

Guida al  
**CORPO DI CONOSCENZE DI SCRUM**  
**(SBOK™ GUIDE)**

Edizione 2016



Guida al

**CORPO DI CONOSCENZE DI  
SCRUM**

***(SBOK™ Guide)***

---

**Edizione 2016**

Una Guida Completa per la Consegna dei Progetti con Scrum\

---

© 2016 SCRUMstudy™, un marchio di VMEdU, Inc. Tutti i diritti riservati.

Library of Congress Cataloging-in-Publication Data

*Guida al Corpo di Conoscenze di Scrum (Guida SBOK™) – edizione 2016*

Titolo originale: *A Guide to the Scrum Body of Knowledge (SBOK™ Guide) – 2016 edition*

Includere riferimenti bibliografici e indice.

ISBN: 978-0-9899252-0-4

1. Il Framework Scrum. I. SCRUMstudy™. II. Guida SBOK™

2013950625

ISBN: 978-0-9899252-0-4

Pubblicato da:

SCRUMstudy™, un marchio di VMEdU, Inc.

410 N 44th Street, Suite 240

Phoenix, Arizona 85008 USA

Telefono: +1-480-882-0706

Fax: +1-240-238-2987

Email: [sbok@scrumstudy.com](mailto:sbok@scrumstudy.com)

Sito web: [www.scrumstudy.com](http://www.scrumstudy.com)

“SBOK”, il logo SCRUMstudy logo, “SDC”, “SMC”, “SAMC”, “SPOC”, e “ESM” sono marchi registrati di SCRUMstudy™ (un marchio di VMEdU, Inc.). Per l'elenco completo dei marchi di SCRUMstudy™, contattare la Divisione Legale di SCRUMstudy™.

La *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)* è distribuita per scopi educativi. SCRUMstudy™ o VMEdU, Inc. non garantiscono che la stessa sia adatta per qualsiasi altro scopo e non forniscono alcuna garanzia espressa o implicita di alcun tipo e non assumono responsabilità per errori o omissioni. Nessuna responsabilità viene assunta per danni accidentali o indiretti connessi o nascenti dall'utilizzo delle informazioni qui contenute.

SCRUMstudy™ accetta volentieri correzioni e commenti a questo libro. Non esitate quindi a inviare commenti su errori tipografici, di formattazione, o di altro genere. Potete fare una copia delle pagine pertinenti, segnare l'errore, e spedire il tutto all'indirizzo sopra indicato o inviare una email a [sbok@scrumstudy.com](mailto:sbok@scrumstudy.com).

Nessuna parte di questa opera può essere riprodotta o trasmessa in qualsiasi forma o con qualunque mezzo, elettronico, manuale, mediante fotocopiatura o registrazione, o attraverso qualsiasi sistema di conservazione e recupero delle informazioni, senza il preventivo permesso scritto dell'editore.

10 9 8 7 6 5 4 3 2

---

---

## PREFAZIONE

La *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)* fornisce le linee guida per il buon esito della implementazione di Scrum – la più popolare metodologia Agile di sviluppo del prodotto e di consegna del progetto. Scrum, così come definito nella *Guida SBOK™*, è un framework applicabile a portfolio, programmi o progetti di qualunque dimensione e complessità; e può essere applicato in maniera efficace in *qualsiasi* settore industriale per creare un prodotto, un servizio, o un altro risultato.

La *Guida SBOK™* è stata sviluppata come una guida standard per le organizzazioni e i professionisti che vogliono implementare Scrum, oltre che per coloro che già lo fanno e che vogliono attuare i miglioramenti necessari ai propri processi. Questo manuale è destinato ad essere utilizzato come riferimento e guida alla conoscenza sia da parte degli esperti di Scrum sia da altri professionisti dello sviluppo di prodotti o servizi, oltre che da persone che non hanno una pregressa esperienza o conoscenza di Scrum né di qualunque altra metodologia per la consegna dei progetti.

La *Guida SBOK™* attinge dalla combinazione di conoscenza e opinioni ottenute da migliaia di progetti di una molteplicità di organizzazioni e settori industriali. Sono stati inoltre forniti contributi da parte di esperti che hanno insegnato Scrum e hanno tenuto corsi sulla consegna dei progetti a più di 400.000 professionisti in 150 paesi diversi. Il suo sviluppo è stato realmente un impegno collaborativo di un gran numero di esperti di una molteplicità di discipline. In particolare, vorrei ringraziare i diciassette co-autori ed esperti della materia ed i ventotto revisori che hanno grandemente contribuito alla creazione della *SBOK™ Guide*.

L'adozione diffusa del framework *Guida SBOK™* dovrebbe aiutare a standardizzare il modo in cui Scrum viene applicato ai progetti dalla globalità delle organizzazioni, oltre ad aiutare a migliorare in modo significativo il Ritorno sull'Investimento. In aggiunta, dovrebbe promuovere una maggiore riflessione e discussione riguardo all'applicazione di Scrum a molti tipi di progetti, e questo a sua volta contribuirà ad una espansione ed arricchimento del corpo di conoscenze e, di conseguenza, a rendere possibili futuri aggiornamenti di questa guida.

Anche se la *Guida SBOK™* rappresenta un riferimento ed un framework omnicomprensivo per la consegna dei progetti con il metodo Scrum, i suoi contenuti sono organizzati, per un facile riferimento, senza tenere conto della precedente conoscenza dell'argomento da parte del lettore. Mi auguro che ciascun lettore possa trarre da questa guida insegnamento e piacere tanto quanto ne hanno tratto i molti autori e revisori dal processo di collazione della conoscenza e della saggezza collettive in essa contenute.



Tridibesh Satpathy,

Autore principale della *Guida SBOK™*

---



# TAVOLA DEI CONTENUTI

1.	INTRODUZIONE.....	1
1.1	Panoramica di Scrum.....	2
1.1.1	Breve Storia di Scrum .....	3
1.2	Perché utilizzare Scrum .....	4
1.2.1	Scalabilità di Scrum.....	5
1.3	Scopo della <i>Guida SBOK™</i> .....	6
1.4	Struttura della <i>Guida SBOK™</i> .....	7
1.4.1	Come Utilizzare la <i>Guida SBOK™</i> .....	8
1.4.2	I Principi di Scrum .....	9
1.4.3	Gli Aspetti di Scrum .....	12
1.4.4	I Processi di Scrum.....	17
1.5	Confronto fra Scrum e il Project Management Tradizionale.....	21
2.	PRINCIPI.....	22
2.1	Introduzione .....	22
2.2	Guida per i Ruoli .....	23
2.3	Controllo Empirico di Processo .....	23
2.3.1	Trasparenza .....	23
2.3.2	Ispezione.....	25
2.3.3	Adattamento .....	25
2.4	Auto-organizzazione .....	28
2.4.1	Benefici dell'Auto-organizzazione .....	28
2.5	Collaborazione.....	30
2.5.1	Benefici della Collaborazione nei Progetti Scrum .....	30
2.5.2	Importanza della Co-ubicazione per la Collaborazione.....	32
2.6	Prioritizzazione basata sul Valore .....	32
2.7	Time-boxing .....	34
2.7.1	I Time-box di Scrum .....	34

2.8	Sviluppo Iterativo .....	37
2.9	Confronto fra Scrum e il Project Management Tradizionale.....	39
3.	ORGANIZZAZIONE .....	40
3.1	Introduzione .....	40
3.2	Guida per i Ruoli .....	41
3.3	I Ruoli del Progetto Scrum .....	41
3.3.1	Ruoli Core .....	42
3.3.2	Ruoli Non-core.....	43
3.4	Product Owner.....	44
3.4.1	Voce del Cliente (Voice of the Customer - VOC) .....	46
3.4.2	Chief Product Owner .....	46
3.5	Scrum Master.....	46
3.5.1	Chief Scrum Master .....	48
3.6	Scrum Team .....	49
3.6.1	Selezione del Personale.....	50
3.6.2	Dimensione dello Scrum Team.....	51
3.7	Scrum nei Progetti, Programmi e Portfolio.....	51
3.7.1	Definizione di Progetto, Programma e Portfolio .....	51
3.7.2	Scrum nei Progetti.....	52
3.7.3	Scrum nei Portfolio e nei Programmi .....	55
3.7.4	Mantenere il Coinvolgimento degli Stakeholder .....	57
3.8	Riepilogo delle Responsabilità .....	58
3.9	Confronto fra Scrum e il Project Management Tradizionale.....	58
3.10	Teorie Diffuse sulle Risorse Umane e loro Rilevanza per Scrum .....	60
3.10.1	Modello delle Dinamiche di Gruppo di Tuckman.....	60
3.10.2	Gestione dei Conflitti.....	61
3.10.3	Tecniche di Gestione dei Conflitti.....	61
3.10.4	Stili di Leadership.....	63
3.10.5	Teoria della Gerarchia dei Bisogni di Maslow .....	65
3.10.6	Teoria X e Teoria Y .....	66

4.	GIUSTIFICAZIONE COMMERCIALE.....	67
4.1	Introduzione .....	67
4.2	Guida per i Ruoli .....	68
4.3	Consegna basata sul Valore .....	68
4.3.1	Responsabilità del Product Owner in tema di Giustificazione Commerciale.....	70
4.3.2	Responsabilità degli Altri Ruoli di Scrum in tema di Giustificazione Commerciale.	70
4.4	Importanza della Giustificazione Commerciale .....	71
4.4.1	Fattori Utilizzati per Determinare la Giustificazione Commerciale.....	71
4.4.2	Giustificazione Commerciale e Ciclo di Vita del Progetto .....	72
4.5	Tecniche relative alla Giustificazione Commerciale .....	75
4.5.1	Stima del Valore del Progetto .....	75
4.5.2	Pianificazione Incentrata sul Valore .....	78
4.5.3	Relative Prioritization Ranking .....	80
4.5.4	Story Mapping .....	80
4.6	Giustificazione Continua del Valore .....	80
4.6.1	Analisi dell'Earned Value .....	81
4.6.2	Diagramma di Flusso Cumulativo (CFD) .....	84
4.7	Confermare la Realizzazione dei Benefici .....	85
4.7.1	Prototipi, Simulazioni e Dimostrazioni .....	85
4.8	Riepilogo delle Responsabilità.....	86
4.9	Confronto fra Scrum e il Project Management Tradizionale.....	87
5.	QUALITÀ.....	88
5.1	Introduzione .....	88
5.2	Guida per i Ruoli .....	89
5.3	Definizione di Qualità .....	89
5.3.1	Qualità e Ambito .....	90
5.3.2	Qualità e Valore di Business .....	90
5.4	Criteri di Accettazione e Prioritized Product Backlog .....	92
5.4.1	Scrivere i Criteri di Accettazione .....	94
5.4.2	Criteri di Accettazione Minimi.....	94
5.4.3	Definizione di 'Done' .....	95

5.4.4	Accettazione o Rifiuto degli Elementi del Prioritized Product Backlog .....	96
5.5	La Gestione della Qualità in Scrum.....	96
5.5.1	Pianificazione della Qualità .....	97
5.5.2	Controllo di Qualità e Garanzia di Qualità.....	98
5.5.3	Ciclo Plan-Do-Check-Act (PDCA).....	99
5.6	Riepilogo delle Responsabilità .....	100
5.7	Confronto fra Scrum e il Project Management Tradizionale.....	101
6.	CAMBIAMENTO.....	103
6.1	Introduzione .....	103
6.2	Guida per i Ruoli .....	104
6.3	Panoramica.....	104
6.3.1	Richieste di Modifica non Approvate e Approvate .....	105
6.4	Il Cambiamento in Scrum.....	107
6.4.1	Bilanciare Flessibilità e Stabilità .....	107
6.4.2	Realizzare la Flessibilità .....	108
6.5	Integrare il Cambiamento.....	113
6.5.1	Modifiche ad uno Sprint .....	114
6.6	Il Cambiamento nei Portfolio e nei Programmi.....	119
6.6.1	Nel Portfolio .....	119
6.6.2	Nel Programma.....	119
6.7	Riepilogo delle Responsabilità .....	121
6.8	Confronto fra Scrum e il Project Management Tradizionale.....	122
7.	RISCHIO .....	123
7.1	Introduzione .....	123
7.2	Guida per i Ruoli .....	124
7.3	Che cos'è il Rischio?.....	124
7.3.1	Differenza fra Rischi e Questioni .....	124
7.3.2	Attitudine al Rischio .....	125
7.4	Procedura di Gestione del Rischio.....	126
7.4.1	Identificazione del Rischio .....	126
7.4.2	Valutazione del Rischio.....	127

7.4.3	Priorizzazione del Rischio .....	131
7.4.4	Mitigazione del Rischio.....	132
7.4.5	Comunicazione del Rischio .....	133
7.5	Minimizzare il Rischio con Scrum .....	134
7.6	I rischi nei Portfolio e nei Programmi .....	135
7.6.1	Nel Portfolio .....	135
7.6.2	Nel Programma.....	136
7.7	Riepilogo delle Responsabilità .....	138
7.8	Confronto fra Scrum e il Project Management Tradizionale.....	139
8.	INIZIO.....	140
8.1	Creare la Vision del Progetto .....	144
8.1.1	Input .....	146
8.1.2	Strumenti.....	149
8.1.3	Output .....	150
8.2	Identificare lo Scrum Master e gli Stakeholder.....	152
8.2.1	Input .....	154
8.2.2	Strumenti.....	156
8.2.3	Output .....	157
8.3	Formare lo Scrum Team.....	158
8.3.1	Input .....	160
8.3.2	Strumenti.....	161
8.3.3	Output .....	162
8.4	Sviluppare le Epic .....	164
8.4.1	Input .....	165
8.4.2	Strumenti.....	168
8.4.3	Output .....	170
8.5	Creare il Prioritized Product Backlog .....	172
8.5.1	Input .....	173
8.5.2	Strumenti.....	175
8.5.3	Output .....	176
8.6	Eeguire la Pianificazione dei Rilasci.....	178

8.6.1	Input .....	179
8.6.2	Strumenti .....	181
8.6.3	Output .....	181
8.7	Diagramma di Flusso dei Dati della Fase .....	183
9.	PIANIFICAZIONE E STIMA .....	184
9.1	Creare le User Story .....	188
9.1.1	Input .....	189
9.1.2	Strumenti .....	190
9.1.3	Output .....	192
9.2	Approvare, Stimare e Prendere in Carico le User Story .....	194
9.2.1	Input .....	195
9.2.2	Strumenti .....	195
9.2.3	Output .....	198
9.3	Creare le Attività .....	199
9.3.1	Input .....	200
9.3.2	Strumenti .....	200
9.3.3	Output .....	202
9.4	Stimare le Attività .....	203
9.4.1	Input .....	204
9.4.2	Strumenti .....	205
9.4.3	Output .....	206
9.5	Creare lo Sprint Backlog .....	207
9.5.1	Input .....	208
9.5.2	Strumenti .....	209
9.5.3	Output .....	209
9.6	Diagramma di Flusso dei Dati della Fase .....	211
10.	IMPLEMENTAZIONE .....	212
10.1	Creare i Deliverable .....	216
10.1.1	Input .....	217
10.1.2	Strumenti .....	219

10.1.3	Output .....	220
10.2	Condurre il Daily Standup .....	222
10.2.1	Input .....	223
10.2.2	Strumenti .....	224
10.2.3	Output .....	225
10.3	Mettere a Punto il Prioritized Product Backlog .....	227
10.3.1	Input .....	228
10.3.2	Strumenti .....	230
10.3.3	Output .....	231
10.4	Diagramma di Flusso dei Dati della Fase .....	231
11.	REVISIONE E RETROSPETTIVA .....	232
11.1	Convocare lo Scrum degli Scrum .....	236
11.1.1	Input .....	237
11.1.2	Strumenti .....	238
11.1.3	Output .....	239
11.2	Illustrare e Convalidare lo Sprint .....	240
11.2.1	Input .....	241
11.2.2	Strumenti .....	242
11.2.3	Output .....	243
11.3	Retrospektiva dello Sprint .....	244
11.3.1	Input .....	245
11.3.2	Strumenti .....	246
11.3.3	Output .....	248
11.4	Diagramma di Flusso dei Dati della Fase .....	249
12.	RILASCIO .....	250
12.1	Consegnare i Deliverable .....	252
12.1.1	Input .....	253
12.1.2	Strumenti .....	254
12.1.3	Output .....	255
12.2	Retrospektiva del Progetto .....	256
12.2.1	Input .....	257

12.2.2	Strumenti.....	257
12.2.3	Output .....	258
12.3	Diagramma di Flusso dei Dati della Fase .....	259
APPENDICE A. PANORAMICA DI AGILE .....		260
APPENDICE B. AUTORI E REVISORI DELLA <i>GUIDA SBOK™</i> .....		270
BIBLIOGRAFIA.....		272
GLOSSARIO .....		273
INDICE .....		307

## ELENCO DELLE FIGURE

Figura 1-1: Flusso di Scrum per un singolo Sprint.....	2
Figura 1-2: Struttura della <i>Guida SBOK™</i> .....	7
Figura 1-3: Principi di Scrum .....	9
Figura 1-4: L'Organizzazione in Scrum .....	14
Figura 2-1: Trasparenza in Scrum .....	24
Figura 2-2: Ispezione in Scrum.....	25
Figura 2-3: Adattamento in Scrum.....	26
Figura 2-4: Sfide del Project Management Tradizionale .....	27
Figura 2-5: Obiettivi di un Team Auto-Organizzato.....	29
Figura 2-6: Benefici della Collaborazione nei Progetti Scrum.....	31
Figura 2-7: Prioritizzazione basata sul valore .....	34
Figura 2-8: Durate prefissate delle riunioni di Scrum .....	36
Figura 2-9: Confronto fra Scrum ed il Metodo Tradizionale a Cascata (Waterfall).....	38
Figura 3-1: Ruoli di Scrum—Visione d'insieme.....	43
Figura 3-2: Domande alle quali si dà risposta durante uno Scrum of Scrums Meeting .....	48
Figura 3-3: Caratteristiche desiderabili per i Ruoli Core di Scrum .....	50
Figura 3-4: Scrum of Scrums (SoS) Meeting.....	53
Figura 3-5: Scrum applicato ai Progetti, Programmi e Portfolio dell'Organizzazione .....	56
Figura 3-6: Fasi di Sviluppo del Gruppo secondo Tuckman .....	60
Figura 3-7: Teoria della Gerarchia dei Bisogni di Maslow .....	65
Figura 4-1: Confronto fra la Consegna di Valore in Scrum e nei Progetti Tradizionali.....	69
Figura 4-2: Gerarchia delle Responsabilità in tema di Giustificazione Commerciale .....	70
Figura 4-3: Giustificazione Commerciale e Ciclo di Vita del Progetto .....	74
Figura 4-4: Analisi di Kano.....	79
Figura 4-5: Esempio di Diagramma di Flusso Cumulativo (CFD) .....	84
Figura 5-1: Diagramma di Flusso dell'Incremento di Prodotto .....	93
Figura 5-2: Criteri di Accettazione a Cascata .....	95
Figura 5-3: Ciclo PDCA in Scrum .....	100
Figura 6-1: Esempio di Processo di Approvazione dei Cambiamenti .....	106
Figura 6-2: Aggiornamento del Prioritized Product Backlog con i Cambiamenti Approvati.....	106
Figura 6-3: Le Caratteristiche di Scrum per Realizzare la Flessibilità .....	108
Figura 6-4: Motivazioni degli Stakeholder per presentare Richieste di Modifica.....	109
Figura 6-5: Le motivazioni dello Scrum Core Team per presentare Richieste di Modifiche.....	110
Figura 6-6: L'integrazione del Cambiamento in Scrum.....	114
Figura 6-7: Impatto del Cambiamento Atteso sulla Lunghezza dello Sprint .....	116
Figura 6-8: Incorporare i Cambiamenti nel Portfolio e nel Programma.....	120
Figura 7-1: Esempio di Albero delle Probabilità.....	128
Figura 7-2: Esempio di Diagramma di Pareto.....	129

Figura 7-3: Esempio di Matrice di Probabilità e Impatto .....	130
Figura 7-4: Processo di Priorizzazione del Rischio .....	132
Figura 7-5: Esempio di Risk Burndown Chart.....	134
Figura 7-6: Gestire i Rischi nei Portfolio e nei Programmi .....	137
Figura 8-1: Visione d'insieme della Fase di Inizio .....	142
Figura 8-2: Visione d'insieme della Fase di Inizio (Elementi essenziali) .....	143
Figura 8-3: Creare la Vision del Progetto—Input, Strumenti e Output .....	144
Figura 8-4: Creare la Vision del Progetto—Diagramma di Flusso dei Dati .....	145
Figura 8-5: Il processo di Gap Analysis .....	150
Figura 8-6: Identificare lo Scrum Master e gli Stakeholder—Input, Strumenti e Output.....	152
Figura 8-7: Identificare lo Scrum Master e gli Stakeholder—Diagramma di Flusso dei Dati.....	153
Figura 8-8: Formare lo Scrum Team—Input, Strumenti e Output .....	158
Figura 8-9: Formare lo Scrum Team—Diagramma di Flusso dei Dati .....	159
Figura 8-10: Sviluppare le Epic—Input, Strumenti e Output .....	164
Figura 8-11: Sviluppare le Epic—Diagramma di Flusso dei Dati .....	165
Figura 8-12: Creare il Prioritized Product Backlog—Input, Strumenti e Output .....	172
Figura 8-13: Creare il Prioritized Product Backlog—Diagramma di Flusso dei Dati .....	173
Figura 8-14: Eseguire la Pianificazione dei Rilasci—Input, Strumenti e Output .....	178
Figura 8-15: Eseguire la Pianificazione dei Rilasci—Diagramma di Flusso dei Dati.....	179
Figura 8-16: Fase di Inizio—Diagramma di Flusso dei Dati.....	183
Figura 9-1: Visione d'insieme della Fase di Pianificazione e Stima .....	186
Figura 9-2: Visione d'insieme della Fase di Pianificazione e Stima (Elementi essenziali) .....	187
Figura 9-3: Creare le User Story—Input, Strumenti e Output .....	188
Figura 9-4: Creare le User Story—Diagramma di Flusso dei Dati .....	189
Figura 9-5: Approvare, Stimare e Prendere in Carico le User Story—Input, Strumenti e Output.....	194
Figura 9-6: Approvare, Sttimare e Prendere in Carico le User Story—Diagramma di Flusso dei Dati.....	194
Figura 9-7: Creare le Attività—Input, Strumenti e Output .....	199
Figura 9-8: Creare le Attività—Diagramma di Flusso dei Dati .....	199
Figura 9-9: Task Planning Meeting .....	200
Figura 9-10: Stimare le Attività—Input, Strumenti e Output.....	203
Figura 9-11: Stimare le Attività—Diagramma di Flusso dei Dati .....	203
Figura 9-12: Creare lo Sprint Backlog—Input, Strumenti e Output .....	207
Figura 9-13: Creare lo Sprint Backlog—Diagramma di Flusso dei Dati .....	207
Figura 9-14: Fase di Pianificazione e Stima—Diagramma di Flusso dei Dati .....	211
Figura 10-1: Visione d'insieme della Fase di Implementazione .....	214
Figura 10-2: Visione d'insieme della Fase di Implementazione (Elementi essenziali) .....	215
Figura 10-3: Creare i Deliverable—Input, Strumenti e Output .....	216
Figura 10-4: Creare i Deliverable—Diagramma di Flusso dei Dati .....	216
Figura 10-5: Scrumboard.....	217
Figura 10-6: Condurre il Daily Standup—Input, Strumenti e Output .....	222
Figura 10-7: Conduct Daily Standup—Data Flow Diagram.....	222

Figura 10-8: Mettere a Punto il Prioritized Product Backlog—Input, Strumenti e Output.....	227
Figura 10-9: Mettere a Punto il Prioritized Product Backlog—Diagramma di Flusso dei Dati.....	228
Figura 10-10: Fase di Implementazione—Diagramma di Flusso dei Dati.....	231
Figura 11-1: Visione d’Insieme della Fase di Revisione e Retrospettiva .....	234
Figura 11-2: Visione d’Insieme della Fase di Revisione e Retrospettiva (Elementi essenziali) .....	235
Figura 11-3: Convocare lo Scrum degli Scrum—Input, Strumenti e Output .....	236
Figura 11-4: Convocare lo Scrum degli Scrum—Diagramma di Flusso dei Dati.....	236
Figura 11-5: Illustrare e Convalidare lo Sprint—Input, Strumenti e Output.....	240
Figura 11-6: Illustrare e Convalidare lo Sprint—Diagramma di Flusso dei Dati.....	241
Figura 11-7: Retrospettiva dello Sprint—Input, Strumenti e Output.....	244
Figura 11-8: Retrospettiva dello Sprint—Diagramma di Flusso dei Dati.....	245
Figura 11-9: Fase di Revisione e Retrospettiva—Diagramma di Flusso dei Dati .....	249
Figura 12-1: Visione d’insieme della Fase di Rilascio.....	251
Figura 12-2: Visione d’insieme della fase di Rilascio (Elementi essenziali) .....	251
Figura 12-3: Consegnare i Deliverable—Input, Strumenti e Output.....	252
Figura 12-4: Consegnare i Deliverable—Diagramma di Flusso dei Prodotti.....	252
Figura 12-5: Retrospettiva del Progetto—Input, Strumenti e Output .....	256
Figura 12-6: Retrospettiva del Progetto—Diagramma di Flusso dei Dati .....	256
Figura 12-7: Fase di Rilascio—Diagramma di Flusso dei Dati.....	259

---

## ELENCO DELLE TAVOLE

Tavola 1-1: Riepilogo dei Processi di Scrum .....	17
Tavola 1-2: Confronto fra Scrum e il Project Management Tradizionale .....	21
Tavola 3-1: Responsabilità del Product Owner nei Processi di Scrum .....	46
Tavola 3-2: Responsabilità dello Scrum Master nei Processi di Scrum .....	48
Tavola 3-3: Responsabilità dello Scrum Team nei Processi di Scrum.....	50
Tavola 3-4: Riepilogo delle Responsabilità Attinenti all'Organizzazione.....	58
Tavola 4-1: Formule dell'Earned Value .....	82
Tavola 4-2: Riepilogo delle Responsabilità Collegate alla Giustificazione Commerciale .....	86
Tavola 5-1: Riepilogo delle Responsabilità attinenti alla Qualità .....	101
Tavola 6-1: Riepilogo delle Responsabilità Collegate al Cambiamento.....	121
Tavola 7-1: Riepilogo delle Responsabilità Riferibili al Rischio.....	138

# 1. INTRODUZIONE

La *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)* fornisce le linee guida per il buon esito della implementazione di Scrum – la più popolare metodologia Agile di sviluppo del prodotto e di consegna del progetto. Questo libro mette a disposizione un framework completo che include i principi, gli aspetti ed i processi di Scrum.

Scrum, così come definito nella *Guida SBOK™*, è applicabile nei seguenti casi:

- Portfolio, programmi, e/o progetti di *qualsiasi* settore industriale
- Prodotti, servizi, o qualsiasi altro risultato che deve essere consegnato agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* può fare riferimento ad un prodotto, ad un servizio o ad un altro risultato atteso. Scrum può essere applicato in maniera efficace a qualsiasi progetto di un qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti complessi e di grandi dimensioni che arrivano anche a diverse centinaia di membri del team.

Questo primo capitolo descrive lo scopo e la struttura della *Guida SBOK™* e fornisce un'introduzione dei concetti chiave di Scrum. Contiene quindi un riepilogo dei principi, degli aspetti e dei processi di Scrum. Il capitolo 2 approfondisce i sei principi di Scrum che rappresentano i fondamenti sui quali si basa il framework Scrum. I capitoli da 3 a 7 sviluppano i cinque aspetti di Scrum che devono essere presi in considerazione nel corso di qualsiasi progetto: organizzazione, giustificazione commerciale, qualità, cambiamento e rischio. I capitoli da 8 a 12 trattano i 19 processi di Scrum che contribuiscono alla realizzazione di un progetto Scrum. Questi processi fanno parte delle cinque fasi di Scrum: Inizio, Pianificazione e Stima, Implementazione, Revisione e Retrospettiva, Rilascio. Queste fasi descrivono nel dettaglio gli input e gli output di ciascun processo, oltre ai vari strumenti che possono essere utilizzati in ciascun processo. Alcuni input, strumenti e output sono obbligatori e sono indicati come tali; altri sono opzionali, dipendendo dal progetto specifico, dai requisiti dell'organizzazione, e/o dalle linee guida espresse dallo Scrum Guidance Body (SGB) dell'organizzazione. Da ultimo, l'Appendice A contiene una panoramica del Manifesto Agile (Fowler and Highsmith, 2001) e un orientamento generale sui vari metodi Agile per coloro che desiderano acquisire maggiori informazioni riguardo ad Agile.

Questo capitolo è suddiviso nelle seguenti sezioni:

## 1.1 Panoramica di Scrum

## 1.2 Perché utilizzare Scrum

## 1.3 Scopo della *Guida SBOK™*

## 1.4 Struttura della *Guida SBOK™*

## 1.5 Confronto fra Scrum ed il Project Management tradizionale

## 1.1 Panoramica di Scrum

Un progetto Scrum implica un impegno collaborativo per creare un nuovo prodotto, servizio o altro risultato così come definito nella Dichiarazione della Vision del Progetto. I progetti sono condizionati da vincoli di tempo, costi, ambito, qualità, risorse, capacità organizzative, ed altri limiti che possono rendere difficile la loro pianificazione, esecuzione, gestione e in definitiva il loro successo. Tuttavia, un'organizzazione trae importanti benefici economici dalla corretta implementazione dei risultati di un progetto portato a termine. Per questo è importante per le organizzazioni selezionare ed attuare una metodologia appropriata per la gestione dei progetti.

Scrum è una delle più popolari metodologie Agile. Si tratta di una metodologia adattiva, iterativa, veloce, flessibile ed efficace, volta alla realizzazione di un valore significativo in tempi rapidi e per tutta la durata del progetto.

Scrum assicura la trasparenza delle comunicazioni e crea un ambiente di responsabilità collettiva e di progressione continua. Il framework Scrum, così come definito nella *Guida SBOK™*, è strutturato in maniera tale da supportare lo sviluppo di prodotti e servizi in tutti i tipi di settori industriali ed in qualsiasi tipo di progetto, a prescindere dalla sua complessità.

Un punto di forza chiave di Scrum risiede nell'uso di team interfunzionali, auto-organizzati e investiti di potere, che dividono il proprio lavoro in cicli brevi e concentrati, chiamati Sprint. La figura 1-1 fornisce una visione di insieme del flusso di un progetto Scrum.

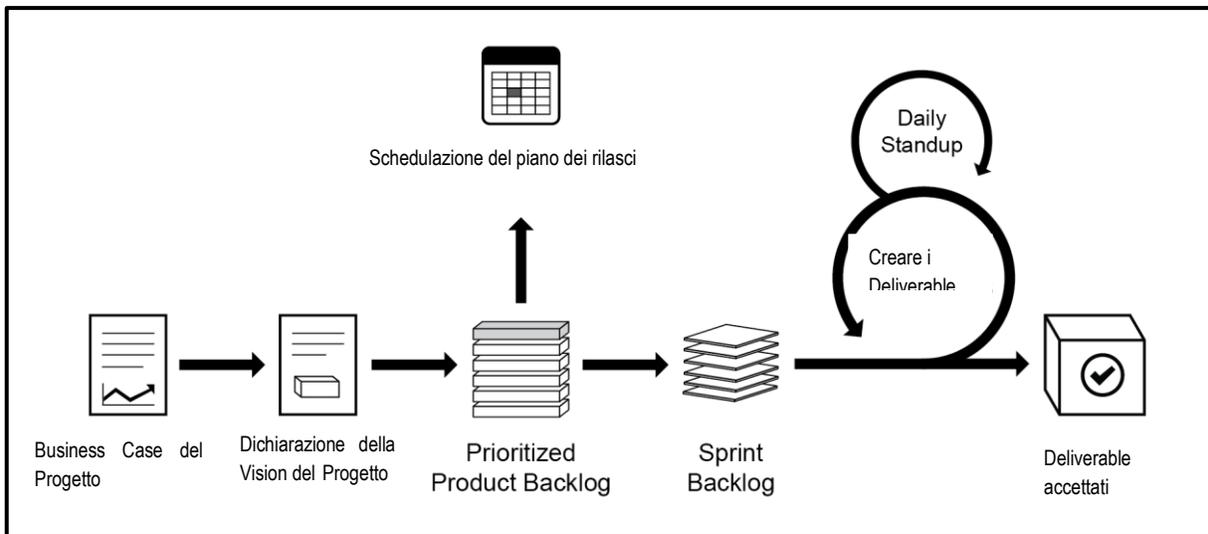


Figura 1-1: Flusso di Scrum per un singolo Sprint

Il ciclo Scrum inizia con una Riunione degli Stakeholder, durante la quale viene creata la Vision del Progetto. Il Product Owner sviluppa quindi un Prioritized Product Backlog che contiene un elenco dei requisiti di business e di progetto ordinati per priorità scritti in forma di User Story. Ogni Sprint inizia con uno Sprint

Planning Meeting, durante il quale si prendono in considerazione le User Story ad alta priorità ai fini dell'inclusione nello Sprint. Uno Sprint dura in genere da una a sei settimane, durante le quali lo Scrum Team lavora alla creazione di deliverable potenzialmente consegnabili o di incrementi di prodotto. Durante lo Sprint sono condotti i Daily Standup Meeting, nei quali i membri del team discutono lo stato di avanzamento giornaliero. Verso la fine dello Sprint, si tiene uno Sprint Review Meeting nel corso del quale viene fornita una dimostrazione dei Deliverable al Product Owner e agli stakeholder appropriati. Il Product Owner accetta i Deliverable solo se soddisfano i Criteri di Accettazione predefiniti. Il ciclo dello Sprint termina con un Retrospect Sprint Meeting, nella quale il team discute i modi per migliorare i processi e le prestazioni mentre si appresta a dare il via allo Sprint successivo.

### 1.1.1 Breve Storia di Scrum

A metà degli anni '80, Hirotaka Takeuchi e Ikujiro Nonaka definirono una strategia di sviluppo del prodotto flessibile ed omnicomprensiva nella quale il team di sviluppo lavora come un'unità per raggiungere un obiettivo comune. Essi descrissero un approccio innovativo allo sviluppo del prodotto, che chiamarono approccio olistico o "del rugby", "nel quale un team prova a coprire la distanza come un'unità, passando la palla avanti e indietro" ("where a team tries to go the distance as a unit, passing the ball back and forth"). I due studiosi basarono il loro approccio su casi studio di varie industrie di produzione. Secondo Takeuchi e Nonaka lo sviluppo del prodotto non doveva essere come una staffetta in sequenza, ma doveva essere piuttosto analogo al gioco del rugby nel quale il team lavora insieme, passando la palla avanti e indietro mentre avanza sul campo come un'unità. Il concetto di una "mischia" proprio del rugby (in cui si forma un gruppo di giocatori per far ripartire il gioco) fu presentato in questo articolo per descrivere la proposta degli autori secondo la quale lo sviluppo del prodotto dovrebbe comportare "l'avanzamento della mischia fino in fondo" ("moving the Scrum downfield").

Ken Schwaber e Jeff Sutherland elaborarono il concetto di Scrum e la sua applicabilità allo sviluppo del software in una presentazione alla conferenza *Object-Oriented Programming, Systems, Languages & Applications* (OOPSLA) tenutasi nel 1995 ad Austin, Texas. Da allora, moltissimi professionisti, esperti ed autori di Scrum hanno continuato a perfezionare la concettualizzazione e la metodologia di Scrum. Negli ultimi anni, Scrum è cresciuto in popolarità ed è ora la metodologia di sviluppo del progetto preferita per molte organizzazioni in tutto il mondo.

## 1.2 Perché utilizzare Scrum

Alcuni dei benefici derivanti dall'utilizzo di Scrum in qualsiasi progetto sono:

1. **Adattabilità**— Il controllo empirico di processo e la consegna iterativa rendono i progetti adattabili e aperti all'incorporazione del cambiamento.
2. **Trasparenza**— Tutti i radiatori delle informazioni come la Scrumboard e la Sprint Burdown Chart sono condivisi, favorendo così un ambiente di lavoro aperto.
3. **Feedback Continuo**—Viene fornito un feedback continuo attraverso i processi *Condurre il Daily Standup* e *Illustrare e Convalidare lo Sprint*.
4. **Miglioramento Continuo**—I deliverable sono migliorati progressivamente Sprint dopo Sprint, grazie al processo *Mettere a Punto il Prioritized Product Backlog*.
5. **Consegna Continua di Valore**— I processi iterativi consentono la consegna continua di valore attraverso il processo *Consegnare i Deliverable* con la frequenza richiesta dal cliente.
6. **Ritmo Sostenibile**— I processi Scrum sono progettati per consentire alle persone coinvolte di poter lavorare ad un ritmo sostenibile che possono, in teoria, continuare a tenere indefinitamente.
7. **Consegna Anticipata del Valore più Elevato**— Il processo *Creare il Prioritized Product Backlog* assicura che i requisiti di più alto valore per il cliente siano soddisfatti per primi.
8. **Processo di Sviluppo Efficiente**— Il Time-boxing (durata predeterminata) e la riduzione al minimo del lavoro non essenziale conducono a livelli maggiori di efficienza.
9. **Motivazione**—I processi *Condurre il Daily Standup* e *Retrospettiva dello Sprint* contribuiscono ad aumentare i livelli di motivazione dei dipendenti.
10. **Risoluzione dei Problemi più Veloce**— La collaborazione e la co-ubicazione di team interfunzionali conduce ad una risoluzione più veloce dei problemi.
11. **Deliverable Efficaci**—Il processo *Creare il Prioritized Product Backlog* e le revisioni regolari dopo la creazione dei deliverable assicurano deliverable efficaci a vantaggio del cliente.
12. **Centralità del Cliente**— L'attenzione al valore di business e il mantenimento di un approccio collaborativo con gli stakeholder lo rende un framework orientato al cliente.
13. **Ambiente di Alta Fiducia**—I processi *Condurre il Daily Standup* e *Retrospettiva dello Sprint* promuovono la trasparenza e la collaborazione, dando luogo ad un ambiente di lavoro di elevata fiducia con conseguente basso attrito fra i partecipanti al progetto.
14. **Responsabilità Collettiva**— Il processo *Approvare, Stimare e Prendere in carico le User Story* permette ai membri del team di assumersi la responsabilità del progetto, da ciò scaturendo una migliore qualità del loro lavoro.

15. **Velocità Elevata**—Un framework collaborativo consente ai team interfunzionali e molto competenti di realizzare il loro pieno potenziale e una velocità elevata.
16. **Ambiente Innovativo**—I processi *Retrospektiva dello Sprint* e *Retrospektiva del Progetto* creano un clima di introspezione, apprendimento e adattabilità che porta ad un ambiente di lavoro innovativo e creativo.

### 1.2.1 Scalabilità di Scrum

Per essere efficaci, gli Scrum Team dovrebbero avere idealmente dai sei ai dieci membri. Da questa pratica discende forse l'erroneo convincimento che il framework Scrum possa essere utilizzato solo per i progetti piccoli. Tuttavia, lo stesso può essere facilmente scalato per un uso efficace in progetti di grandi dimensioni. Nelle situazioni in cui la dimensione dello Scrum Team eccede le dieci persone, per lavorare al progetto possono essere costituiti vari Scrum Team. Il processo *Convocare lo Scrum degli Scrum* facilita il coordinamento fra gli Scrum Team, permettendo l'efficace implementazione del metodo nei progetti di più grandi dimensioni.

I progetti grandi o complessi sono spesso implementati come parte di un programma o portfolio. Il framework Scrum può essere applicato anche per gestire i programmi e i portfolio. L'approccio logico delle linee guida e dei principi di questo framework può essere usato per gestire progetti di qualsiasi dimensione, che abbracciano diverse zone geografiche e organizzazioni. I progetti di grandi dimensioni possono avere numerosi Scrum Team che lavorano in parallelo, per cui si rende necessario sincronizzare e facilitare il flusso di informazioni e migliorare la comunicazione. Il processo *Convocare lo Scrum degli Scrum* è quello che assicura questa sincronizzazione. In questa riunione sono rappresentati i vari Scrum Team, con l'obiettivo di fornire aggiornamenti sullo stato di avanzamento, discutere le sfide affrontate durante il progetto e coordinare le attività. Non ci sono regole prestabilite riguardo alla frequenza di queste riunioni. I fattori che ne determinano la frequenza sono la quantità di interdipendenze fra i vari team, la dimensione del progetto, il livello di complessità e le raccomandazioni provenienti dallo Scrum Guidance Body.

### 1.3 Scopo della *Guida SBOK™*

Negli ultimi anni, è divenuto evidente che le organizzazioni che utilizzano Scrum come framework preferito per la consegna dei progetti realizzano stabilmente elevati Ritorni sull'Investimento. La focalizzazione sulla consegna basata sul valore propria di Scrum aiuta gli Scrum Team a consegnare risultati quanto prima possibile nel corso del progetto.

La *Guida SBOK™* è stata sviluppata come un mezzo per creare un punto di riferimento indispensabile per le organizzazioni e i professionisti della gestione del progetto che vogliono implementare Scrum, oltre che per coloro che già lo fanno e che intendono apportare i cambiamenti desiderati ai propri processi. Tale guida si basa sull'esperienza attinta da migliaia di progetti di una grande varietà di organizzazioni e settori industriali. Nel suo sviluppo sono stati presi in considerazione i contributi di molti esperti di Scrum e professionisti della gestione dei progetti.

La *Guida SBOK™* è particolarmente preziosa:

- Per i membri dello Scrum Core Team, che include:
  - I Product Owner che vogliono comprendere appieno il framework Scrum ed soprattutto le preoccupazioni legate al cliente o agli stakeholder che coinvolgono aspetti della giustificazione commerciale, della qualità, del cambiamento e del rischio associati ai progetti Scrum.
  - Gli Scrum Master che vogliono apprendere il loro specifico ruolo di supervisori dell'applicazione del framework Scrum ai progetti Scrum.
  - I membri dello Scrum Team che desiderano comprendere meglio i processi di Scrum e gli strumenti ad essi associati da poter utilizzare per creare il prodotto o servizio del progetto.
- Come guida completa per tutti i professionisti di Scrum che lavorano a progetti Scrum in qualsiasi tipo di organizzazione o settore industriale.
- Come fonte di riferimento per chiunque interagisca con lo Scrum Core Team, ivi compresi - ma non in via esaustiva - il Portfolio Product Owner, il Portfolio Scrum Master, il Program Product Owner, il Program Scrum Master, lo Scrum Guidance Body e gli Stakeholder (vale a dire lo sponsor, il cliente e gli utenti).
- Come un manuale utile per chiunque non abbia una precedente esperienza o conoscenza del framework Scrum ma che desideri apprendere di più sull'argomento.

Il contenuto della *Guida SBOK™* è inoltre di aiuto per coloro che si preparano a sostenere i seguenti esami di certificazione SCRUMstudy™:

- Scrum Developer Certified (SDC™)
- Scrum Master Certified (SMC™)
- SCRUMstudy Agile Master Certified (SAMC™)
- Scrum Product Owner Certified (SPOC™)
- Expert Scrum Master (ESM™)

## 1.4 Struttura della Guida SBOK™

La Guida SBOK™ è suddivisa in linea generale nelle seguenti tre aree:

1. **Principi** trattati nel capitolo 2, che spiega nel dettaglio i sei principi che costituiscono le fondamenta di Scrum.
2. **Aspetti** trattati nei capitoli da 3 a 7, che descrivono i cinque aspetti che sono considerati importanti per tutti i progetti Scrum.
3. **Processi** trattati nei capitoli da 8 a 12, che trattano i diciannove processi e gli input, gli strumenti e gli output ad essi associati.

La Figura 1-2 illustra la struttura della Guida SBOK™, la quale mette in evidenza che i principi, gli aspetti e di processi interagiscono gli uni con gli altri e sono ugualmente importanti per raggiungere una migliore comprensione del framework Scrum.

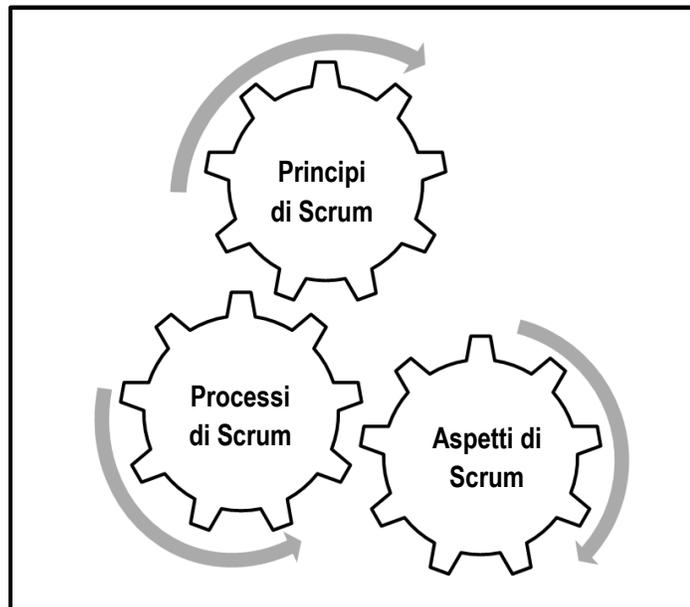


Figura 1-2: Struttura della Guida SBOK™

### 1.4.1 Come Utilizzare la *Guida SBOK™*

La *Guida SBOK™* può essere utilizzata come punto di riferimento e conoscenza sia da coloro che hanno esperienza di Scrum sia dagli altri professionisti dello sviluppo del prodotto e del servizio, oltre che dalle persone che non hanno alcuna precedente esperienza o conoscenza di Scrum o di una metodologia di project management. I contenuti sono organizzati in modo da facilitarne la consultazione da parte dei tre ruoli dello Scrum Core Team: Scrum Master, Product Owner, e Scrum Team.

I capitoli riguardanti i sei principi di Scrum (capitolo 2) e i cinque aspetti di Scrum (capitoli da 3 a 7) includono una Guida per i Ruoli. Questa guida fornisce indicazioni sulla rilevanza di ciascuna sezione del capitolo per i vari ruoli dello Scrum Core Team.

Al fine di rendere possibile la migliore applicazione del framework Scrum, la *Guida SBOK™* ha differenziato in maniera chiara gli input, gli strumenti e gli output obbligatori da quelli non obbligatori o opzionali. Gli input, gli strumenti e gli output contrassegnati dagli asterischi (\*) sono obbligatori, mentre quelli senza asterischi sono opzionali. Si raccomanda a coloro che si accostano per la prima volta a Scrum di concentrarsi prima di tutto sugli input, gli strumenti e gli output obbligatori, mentre i professionisti con maggiore esperienza dovrebbero leggere integralmente i capitoli relativi ai processi.

## 1.4.2 I Principi di Scrum

I principi di Scrum rappresentano le linee guida centrali per l'applicazione del framework Scrum e dovrebbero essere usati obbligatoriamente in tutti i progetti Scrum. I sei principi di Scrum esposti nel capitolo 2 sono:

1. Controllo Empirico di Processo
2. Auto-organizzazione
3. Collaborazione
4. Prioritizzazione basata sul valore
5. Time-boxing
6. Sviluppo Iterativo

La Figura 1-3 illustra i sei principi di Scrum.

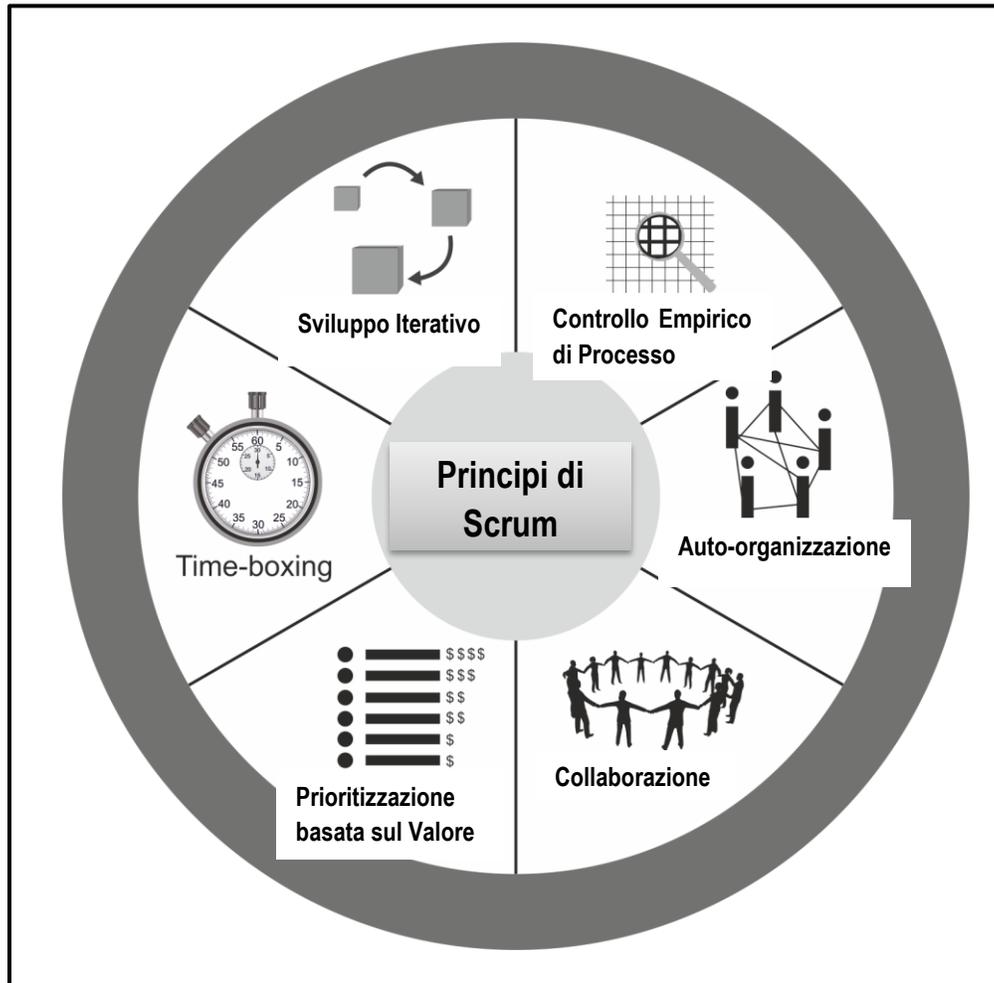


Figura 1-3: Principi di Scrum



I principi di Scrum possono essere applicati a qualsiasi tipo di progetto di qualunque organizzazione ed è necessario conformarsi ad essi per assicurare l'efficace implementazione del framework Scrum. I principi di Scrum non sono negoziabili e devono essere applicati così come specificato nella *Guida SBOK™*. Mantenere l'integrità dei principi ed utilizzarli in modo appropriato infonde fiducia riguardo al fatto che il framework Scrum possa far raggiungere gli obiettivi del progetto. Gli aspetti e i processi di Scrum, tuttavia, possono essere modificati per soddisfare i requisiti del progetto o dell'organizzazione.

1. **Controllo Empirico di Processo**—Questo principio mette in evidenza il nucleo centrale della filosofia di Scrum basata sulle tre idee principali di trasparenza, ispezione e adattamento.
2. **Auto-organizzazione**— Questo principio si focalizza sui lavoratori di oggi, che realizzano un valore significativamente più grande quando sono auto-organizzati e questo si traduce in un'adesione totale da parte del team ed in una assunzione di responsabilità condivisa; e, non ultimo, in un ambiente innovativo e creativo che contribuisce maggiormente alla crescita.
3. **Collaborazione**— Questo principio si focalizza sulle tre dimensioni fondamentali connesse al lavoro collaborativo: consapevolezza, articolazione e appropriazione. Inoltre promuove la gestione del progetto come processo condiviso di creazione del valore nel quale i team lavorano e interagiscono insieme per realizzare il massimo valore.
4. **Prioritizzazione basata sul Valore**—Questo principio evidenzia l'attenzione di Scrum alla consegna del massimo valore di business, dall'inizio del progetto e per tutta la sua durata.
5. **Time-boxing (Durata Predeterminata)**—Questo principio descrive come in Scrum il tempo sia considerato un vincolo limitante, e sia utilizzato per aiutare a gestire efficacemente la pianificazione e l'esecuzione del progetto. In Scrum gli elementi per i quali il tempo è predeterminato comprendono gli Sprint, i Daily Standup Meeting, gli Sprint Planning Meeting e gli Sprint Review Meeting.
6. **Sviluppo Iterativo**—Questo principio definisce lo sviluppo iterativo e mette in evidenza come gestire meglio i cambiamenti e costruire prodotti che soddisfino i bisogni del cliente. Esso delinea inoltre le responsabilità del Product Owner e dell'organizzazione connesse allo sviluppo iterativo.

## 1.4.3 Gli Aspetti di Scrum

Gli aspetti di Scrum devono essere affrontati e gestiti lungo tutto il corso di un progetto Scrum. I cinque aspetti di Scrum esposti nei capitoli da 3 a 7 sono:

### 1.4.3.1 Organizzazione

Comprendere la definizione dei ruoli e delle responsabilità di un progetto Scrum è molto importante per assicurare il buon esito dell'implementazione di Scrum.

I ruoli di Scrum rientrano in due categorie generali:

1. **Ruoli Core**— I Ruoli Core sono quelli che sono richiesti obbligatoriamente per realizzare il prodotto o servizio del progetto. Le persone a cui sono assegnati i ruoli core sono completamente dedicate al progetto e sono in definitiva responsabili del successo di ogni singola iterazione del progetto e del progetto nel suo insieme.

Questi ruoli comprendono:

- Il **Product Owner** è la persona responsabile della realizzazione del massimo valore di business per il progetto. Lui o lei è altresì responsabile dell'enunciazione dei requisiti del cliente e del mantenimento della giustificazione commerciale del progetto. Il Product Owner rappresenta la Voce del Cliente (Voice of the Customer).
- Lo **Scrum Master** è un facilitatore che ha il compito di fare in modo che lo Scrum Team abbia un ambiente che agevoli la riuscita del progetto. Lo Scrum Master guida, facilita ed insegna le pratiche di Scrum a tutti coloro che sono coinvolti nel progetto; elimina gli impedimenti del team; e fa in modo che i processi di Scrum siano seguiti correttamente.
- Lo **Scrum Team** è il gruppo o team di persone responsabili di comprendere i requisiti specificati dal Product Owner e di creare i Deliverable del progetto.

2. **Ruoli Non-core**—I Ruoli Non-core sono quelli che non sono obbligatoriamente richiesti per il progetto Scrum e possono includere i membri del team che hanno un interesse nel progetto. Questi non hanno un ruolo formale all'interno del team di progetto e possono interfacciarsi con il team, ma non essere responsabili per il successo del progetto. I ruoli non-core devono essere tenuti in considerazione in qualsiasi progetto Scrum.

I ruoli non-core comprendono:

- Gli **Stakeholder**, che è un termine collettivo che ricomprende i clienti, gli utenti, e gli sponsor, si interfacciano di frequente con lo Scrum Core Team ed influenzano il progetto durante tutto il suo sviluppo. Ciò che più conta è che il progetto produce i benefici della collaborazione a vantaggio degli stakeholder.
- Lo **Scrum Guidance Body** (SGB) è un ruolo opzionale, che in genere consiste in un insieme di documenti e/o in un gruppo di esperti che sono di norma coinvolti nella definizione degli obiettivi connessi alla qualità, ai regolamenti governativi, alla sicurezza ed ad altri parametri chiave dell'organizzazione. Questo SGB guida il lavoro svolto dal Product Owner, dallo Scrum Master e dallo Scrum Team.
- I **Venditori**, che includono le persone o le organizzazioni esterne, forniscono prodotti e/o servizi che non rientrano nelle competenze principali dell'organizzazione di progetto.
- Il **Chief Product Owner** è un ruolo proprio dei progetti più grandi con numerosi Scrum Team. Questo ruolo ha il compito di facilitare il lavoro di molteplici Product Owner e di mantenere la giustificazione commerciale dei progetti di dimensioni più ampie.
- Il **Chief Scrum Master** ha la responsabilità di coordinare le attività connesse a Scrum nei grandi progetti, che possono richiedere il lavoro in parallelo di molteplici Scrum Team.

La Figura 1-4 illustra la struttura dell'Organizzazione di Scrum.

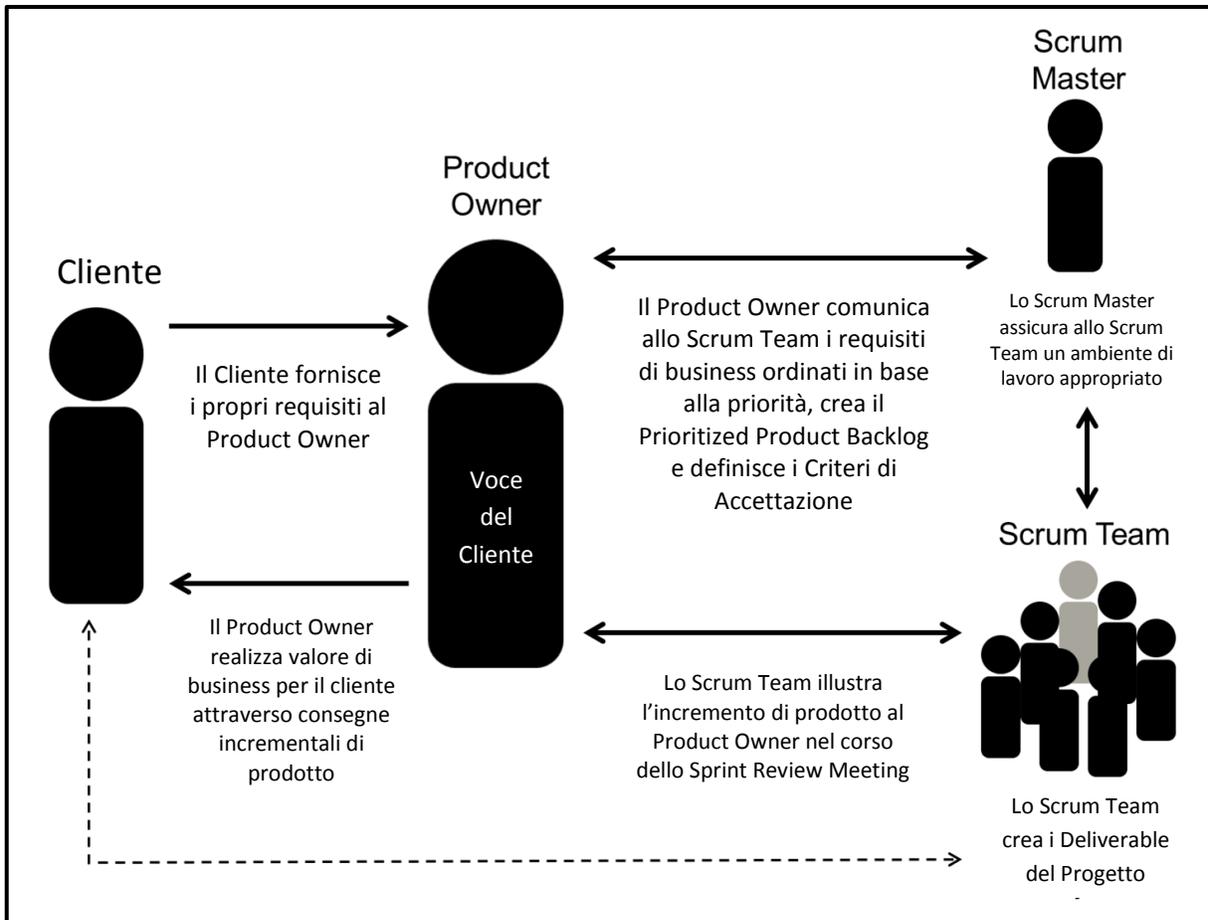


Figura 1-4: L'Organizzazione in Scrum

L'aspetto dell'Organizzazione di Scrum si occupa anche dei requisiti di struttura del team necessari per implementare Scrum nei programmi e nei portfolio.

### 1.4.3.2 Giustificazione Commerciale

Per un'organizzazione è importante eseguire una valutazione di business appropriata prima di far partire qualsiasi progetto. Ciò aiuta gli organi decisionali chiave a comprendere l'esigenza di business di un cambiamento o di un nuovo prodotto o servizio, la giustificazione per andare avanti con un progetto e la sua fattibilità.

La giustificazione commerciale in Scrum si basa sul concetto della Consegna basata sul Valore. Una delle caratteristiche chiave di tutti i progetti è l'incertezza degli esiti o dei risultati. È impossibile garantire il successo di un progetto, a prescindere dalla sua dimensione o complessità. Proprio partendo dalla considerazione dell'incertezza del buon esito, Scrum tenta di iniziare a consegnare i risultati del progetto

quanto prima sia possibile. Questa consegna anticipata di risultati, e perciò di valore, offre un'opportunità di reinvestimento e dimostra la pregevolezza del progetto agli stakeholder interessati.

L'adattabilità di Scrum permette il cambiamento degli obiettivi e dei processi del progetto qualora dovesse cambiare la giustificazione commerciale. È importante notare che anche se il Product Owner è il principale responsabile della giustificazione commerciale, gli altri membri del team vi contribuiscono in maniera significativa.

### 1.4.3.3 Qualità

In Scrum la qualità è definita come l'abilità del prodotto o dei deliverable completati di soddisfare i Criteri di Accettazione e realizzare il valore di business atteso dal cliente.

Per assicurare che un progetto soddisfi i requisiti di qualità, Scrum adotta un approccio di miglioramento continuo attraverso il quale il team apprende dall'esperienza e dal coinvolgimento degli stakeholder, in modo da tenere costantemente aggiornato il Prioritized Product Backlog con le eventuali modifiche dei requisiti. Il Prioritized Product Backlog semplicemente non è mai completo fino alla chiusura del progetto o alla sua interruzione. Eventuali cambiamenti dei requisiti rispecchiano i cambiamenti nell'ambiente di business interno o esterno e permettono al team di lavorare senza interruzioni e adattarsi per realizzare quei requisiti.

Dal momento che Scrum richiede che il lavoro sia completato in incrementi successivi durante gli Sprint, ciò significa che gli errori o i difetti vengono notati prima grazie a test di qualità ripetitivi, piuttosto che quando il prodotto o il servizio finale è prossimo al completamento. Inoltre, importanti attività connesse alla qualità (ad esempio sviluppo, collaudo e documentazione) sono completate come parte del medesimo Sprint dal medesimo team; questo assicura che la qualità sia inerente a qualsiasi deliverable creato come parte di uno Sprint. Tali deliverable scaturenti dai progetti Scrum, e potenzialmente consegnabili, sono detti 'Completati' ('Done').

Pertanto, il miglioramento continuo con test ripetitivi accresce la probabilità di raggiungere i livelli di qualità attesi in un progetto Scrum. Discussioni costanti fra lo Scrum Core Team e gli stakeholder (fra cui clienti e utenti), con incrementi effettivi del prodotto che vengono realizzati alla fine di ciascuno Sprint, assicurano che il divario fra le aspettative del cliente riguardo al progetto ed i deliverable effettivamente prodotti si riduca costantemente.

Lo Scrum Guidance Body può fornire inoltre linee guida sulla qualità che possono essere applicabili a tutti i progetti Scrum dell'organizzazione.

### 1.4.3.4 Cambiamento

Tutti i progetti, indipendentemente dal metodo o framework utilizzato, sono esposti al cambiamento. È assolutamente indispensabile che i membri del team di progetto comprendano che i processi di sviluppo di Scrum sono progettati per abbracciare il cambiamento.

Le organizzazioni devono provare a massimizzare i benefici nascenti dal cambiamento e a minimizzarne gli impatti negativi attraverso processi accurati di gestione del cambiamento, secondo i principi di Scrum.

Un principio fondamentale di Scrum è il suo riconoscimento che a) gli stakeholder (ad esempio i clienti, gli utenti e gli sponsor) nel corso del progetto cambiano idea su ciò che vogliono e di cui hanno bisogno (circostanza a volte chiamata “requirements churn”, letteralmente “abbandono dei requisiti”) e b) è molto difficile, se non impossibile, che gli stakeholder definiscano tutti i requisiti durante l’inizio del progetto.

I progetti Scrum accolgono volentieri il cambiamento attraverso l’uso di brevi Sprint iterativi che incorporano il feedback del cliente nei deliverable di ciascuno Sprint. Ciò consente al cliente di interagire regolarmente con i membri dello Scrum Team, di visionare i deliverable non appena sono pronti e di cambiare i requisiti all’inizio dello Sprint se necessario.

Inoltre, i team di gestione del portfolio o del programma possono rispondere a Richieste di Modifica attinenti ai progetti Scrum inerenti al loro livello.

#### **1.4.3.5 Rischio**

Il rischio è definito come un evento incerto o un insieme di eventi incerti che possono impattare gli obiettivi di un progetto e potrebbero contribuire al suo successo o fallimento. I rischi che hanno un probabile impatto positivo sul progetto vengono definiti come opportunità, mentre le minacce sono i rischi che potrebbero impattare il progetto in modo negativo. La gestione del rischio deve essere eseguita in maniera proattiva ed è un processo iterativo che deve iniziare nella fase di inizio del progetto e continuare lungo tutto il suo ciclo di vita. Il processo di gestione dei rischi dovrebbe seguire alcuni passi standardizzati per assicurare l’identificazione del rischio, la sua valutazione, la determinazione delle azioni appropriate e la loro conseguente messa in atto.

I rischi dovrebbero essere identificati, valutati e trattati in base a due fattori: la probabilità di accadimento di ciascun rischio e il possibile impatto nel caso di tale evenienza. I rischi con un valore di probabilità ed impatto elevato (che si determina moltiplicando i due fattori) dovrebbero essere affrontati prima di quelli con un valore relativamente più basso. In generale, una volta che si identifica un rischio, è importante comprenderne le probabili cause ed i potenziali effetti in caso di suo accadimento.

## 1.4.4 I Processi di Scrum

I processi di Scrum riguardano le specifiche attività ed il flusso di un progetto Scrum. In totale ci sono diciannove processi che sono raggruppati in cinque fasi. Queste fasi sono presentate nei capitoli da 8 a 12 della *Guida SBOK™*, così come mostrato nella Tavola 1-1.

Chapter	Phase	Processes
8	Inizio	<ol style="list-style-type: none"> <li>1. Creare la Vision del Progetto</li> <li>2. Identificare lo Scrum Master e gli Stakeholder</li> <li>3. Formare lo Scrum Team</li> <li>4. Sviluppare le Epic</li> <li>5. Creare il Prioritized Product Backlog</li> <li>6. Eseguire la Pianificazione dei Rilasci</li> </ol>
9	Pianificazione e Stima	<ol style="list-style-type: none"> <li>7. Creare le User Story</li> <li>8. Approvare, Stimare e Prendere in Carico le User Story</li> <li>9. Creare le Attività</li> <li>10. Stimare le Attività</li> <li>11. Creare lo Sprint Backlog</li> </ol>
10	Implementazione	<ol style="list-style-type: none"> <li>12. Creare i Deliverable</li> <li>13. Condurre il Daily Standup</li> <li>14. Sistemare il Prioritized Product Backlog</li> </ol>
11	Revisione e Retrospettiva	<ol style="list-style-type: none"> <li>15. Convocare lo Scrum degli Scrum</li> <li>16. Illustrare e Convalidare lo Sprint</li> <li>17. Retrospettiva dello Sprint</li> </ol>
12	Rilascio	<ol style="list-style-type: none"> <li>18. Consegnare i Deliverable</li> <li>19. Retrospettiva del Progetto</li> </ol>

Tavola 1-1: Riepilogo dei Processi di Scrum

Queste fasi descrivono ciascun processo nel dettaglio con gli input, gli strumenti e gli output associati. In ciascun processo, alcuni input, strumenti ed output sono obbligatori (quelli con un asterisco [\*] dopo il nome), mentre altri sono opzionali. L'inclusione degli input, strumenti ed output opzionali dipenderà dal particolare progetto, organizzazione o settore industriale. Gli input, gli strumenti e gli output denotati come obbligatori sono importanti per il buon esito dell'implementazione di Scrum in qualsiasi organizzazione.

#### 1.4.4.1 Inizio

1. *Creare la Vision del Progetto*—In questo processo, si esamina il Business Case del Progetto per creare una dichiarazione della Vision del Progetto che servirà da stimolo e costituirà il fulcro dell'intero progetto. In questo processo viene identificato il Product Owner.
2. *Identificare lo Scrum Master e gli Stakeholder*—In questo processo vengono identificati lo Scrum Master e gli Stakeholder utilizzando Criteri di Selezione specifici.
3. *Formare lo Scrum Team*—In questo processo vengono identificati i membri dello Scrum Team. Di solito il Product Owner è il principale responsabile della selezione dei membri del team, ma spesso svolge questo compito in collaborazione con lo Scrum Master.
4. *Sviluppare le Epic*—In questo processo, la Dichiarazione della Vision del Progetto costituisce la base per lo sviluppo delle Epic. Possono essere tenute delle Riunioni di Gruppi di Utenti per discutere le Epic appropriate.
5. *Creare il Prioritized Product Backlog*—In questo processo le Epic sono perfezionate, elaborate e poi ordinate per priorità per creare un Prioritized Product Backlog del progetto. In questo momento sono stabiliti anche i Done Criteria.
6. *Eeguire la Pianificazione dei Rilasci*—In questo processo, lo Scrum Core Team riesamina le User Story del Prioritized Product Backlog per sviluppare una Schedulazione del Piano dei Rilasci, che è essenzialmente un programma di distribuzione per fasi che può essere condiviso con gli Stakeholder del progetto. In questo processo viene inoltre stabilita la lunghezza dello Sprint

#### 1.4.4.2 Pianificazione e Stima

7. *Creare le User Story*—In questo processo vengono create le User Story ed i connessi Criteri di Accettazione. Le User Story sono di solito scritte dal Product Owner e sono volte ad assicurare che i requisiti del cliente siano descritti in maniera chiara e possano essere pienamente compresi da tutti gli Stakeholder. Possono essere tenute Esercitazioni di Scrittura delle User Story che coinvolgono i membri dello Scrum Team nella creazione delle User Story. Le User Story sono incorporate nel Prioritized Product Backlog.
8. *Approvare, Stimare e Prendere in Carico le User Story*—In questo processo, il Product Owner approva le User Story di uno Sprint. Successivamente, lo Scrum Master e lo Scrum Team stimano l'impegno richiesto per sviluppare la funzionalità descritta in ciascuna User Story e lo Scrum Team assume l'impegno di consegnare i requisiti del cliente sotto forma di User Story Approvate, Stimate e Prese in Carico.

9. *Creare le Attività*—In questo processo, le User Story Approvate, Stimate e Prese in Carico sono scomposte in specifiche attività e viene redatta una Lista delle Attività. A questo scopo viene spesso tenuto un Task Planning Meeting.
10. *Stimare le Attività*—In questo processo, durante iTask Estimation Meeting, lo Scrum Team stima l'impegno necessario per il compimento di ciascuna attività presente nella Lista delle Attività. Il risultato di questo processo è una Lista delle Attività con Impegno Stimato.
11. *Creare lo Sprint Backlog*—In questo processo, lo Scrum Core Team tiene degli Sprint Planning Meeting durante i quali il gruppo crea uno Sprint Backlog che contiene tutte le attività da completare nello Sprint.

#### 1.4.4.3 Implementazione

12. *Creare i Deliverable*—In questo processo, lo Scrum Team lavora alle attività dello Sprint Backlog per creare i Deliverable dello Sprint. Spesso si utilizza una Scrumboard per tenere traccia del lavoro e delle attività che si stanno eseguendo. Le questioni o i problemi che lo Scrum Team si trova ad affrontare possono essere annotate in un Registro degli Impedimenti.
13. *Condurre il Daily Standup*—In questo processo viene condotta giornalmente una riunione estremamente mirata, di durata prestabilita, chiamata Daily Standup Meeting. Questo rappresenta per lo Scrum Team il luogo di discussione per un aggiornamento reciproco sullo stato di avanzamento e sugli eventuali impedimenti che ognuno sta riscontrando.
14. *Sistemare il Prioritized Product Backlog*—In questo processo si provvede al continuo aggiornamento e mantenimento del Prioritized Product Backlog. Può essere tenuto un Prioritized Product Backlog Review Meeting nel quale eventuali cambiamenti o aggiornamenti del backlog vengono discussi ed inseriti se del caso.

#### 1.4.4.4 Revisione e Retrospettiva

15. *Convocare lo Scrum degli Scrum*—In questo processo, i rappresentanti dei vari Scrum Team si riuniscono per lo Scrum of Scrums (SoS) Meeting a intervalli prestabiliti o ogni qualvolta sia necessario per collaborare e monitorare il rispettivo stato di avanzamento, gli impedimenti e le dipendenze fra i vari team. Questo processo riguarda solo i progetti di grandi dimensioni nei quali sono coinvolti molti Scrum Team.
16. *Illustrare e Convalidare lo Sprint*—In questo processo lo Scrum Team, nel corso di uno Sprint Review Meeting, illustra i Deliverable dello Sprint al Product Owner e agli opportuni Stakeholder. Lo

scopo di questa riunione è di ottenere l'approvazione e l'accettazione dei Deliverable creati nello Sprint da parte del Product Owner.

17. *Retrospettiva dello Sprint*—In questo processo, lo Scrum Master e lo Scrum Team si riuniscono per discutere le lezioni apprese nel corso dello Sprint. Queste informazioni sono registrate come lezioni apprese che possono essere applicate ai futuri Sprint. Spesso, come risultato di questa discussione, possono scaturire Miglioramenti Fattibili Concordati o un Aggiornamento delle Raccomandazioni dello Scrum Guidance Body.

#### **1.4.4.5 Rilascio**

18. *Consegnare i Deliverable*—In questo processo i Deliverable Accettati sono consegnati o trasferiti agli Stakeholder pertinenti. Un Accordo Formale sui Deliverable Funzionanti documenta il positivo completamento dello Sprint.
19. *Retrospettiva del Progetto*—In questo processo, che conclude il progetto, gli stakeholder dell'organizzazione e i membri dello Scrum Core Team si riuniscono per esaminare retrospettivamente il progetto e identificare, documentare ed assimilare le lezioni apprese. Queste lezioni portano spesso alla documentazione di Miglioramenti Fattibili Concordati, da implementare in progetti futuri.

## 1.5 Confronto fra Scrum e il Project Management Tradizionale

La Tabella 1-2 riassume molte delle differenze fra Scrum ed i modelli tradizionali di project management.

1

	Scrum	Project Management Tradizionale
L'attenzione è concentrata su	Persone	Processi
Documentazione	Minima—solo quando richiesto	Omnicomprendiva
Tipo di Processo	Iterativo	Lineare
Pianificazione Anticipata	Bassa	Alta
Prioritizzazione dei Requisiti	Basata sul valore di business e aggiornata regolarmente	Fissata nel piano di Progetto
Garanzia di Qualità	Incentrata sul Cliente	Incentrata sul Processo
Organizzazione	Auto-organizzata	Gestita
Stile di Gestione	Decentrato	Centralizzato
Cambiamento	Aggiornamenti del Prioritized Product Backlog	Sistema formale di Gestione del Cambiamento
Leadership	Collaborativa, Leadership del Servitore	Comando e controllo
Misurazione della prestazione	Valore di Business	Conformità al Piano
Ritorno sull'Investimento (ROI)	Anticipato/durante tutta la vita del progetto	Alla fine della vita del progetto
Coinvolgimento del Cliente	Elevato per tutta la durata del progetto	Varia in base al ciclo di vita del progetto

Tavola 1-2: Confronto fra Scrum e il Project Management Tradizionale

## 2. PRINCIPI

### 2.1 Introduzione

I Principi di Scrum sono le fondamenta su cui si basa il framework Scrum. I principi di Scrum possono essere applicati a qualsiasi tipo di progetto o di organizzazione, ed è obbligatorio conformarvisi per assicurare una applicazione appropriata di Scrum. Gli aspetti e i processi di Scrum possono essere modificati per soddisfare i requisiti del progetto o dell'organizzazione che ne fa uso, ma i principi di Scrum non sono negoziabili e devono essere applicati così come descritto nel framework presentato nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*.

Mantenere intatti i principi ed utilizzarli in modo appropriato infonde fiducia in chi utilizza il framework Scrum riguardo al raggiungimento degli obiettivi del progetto. I principi sono considerati le linee guida fondamentali per l'applicazione del framework Scrum.

I principi, così come definiti nella *Guida SBOK™*, sono applicabili a:

- Portfolio, programmi, e/o progetti di *qualsiasi* industria
- Prodotti, servizi, o qualunque altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Nella *Guida SBOK™* il termine “prodotto” può riferirsi ad un prodotto, servizio o altro deliverable. Scrum può essere applicato efficacemente a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi con fino a diverse centinaia di membri del team.

Questo capitolo è suddiviso nelle seguenti sezioni:

**2.2 Guida per i Ruoli**—Questa sezione identifica quali parti o paragrafi del capitolo sono più attinenti per ciascuno dei ruoli core di Scrum - Product Owner, Scrum Master e Scrum Team.

**2.3 Controllo Empirico di Processo**—Questa sezione descrive il primo principio di Scrum, e le tre idee principali di trasparenza, ispezione e adattamento.

**2.4 Auto-organizzazione**—Questa sezione evidenzia il secondo principio di Scrum, che si focalizza sui lavoratori di oggi, i quali realizzano un valore significativamente maggiore quando sono auto-organizzati e questo si traduce in un'adesione totale da parte del team e nella assunzione di responsabilità condivisa; ed anche in un ambiente innovativo e creativo che contribuisce maggiormente alla crescita.

**2.5 Collaborazione**—Questa sezione si concentra sul terzo principio di Scrum per il quale lo sviluppo del prodotto è un processo condiviso di creazione del valore che ha bisogno che tutti gli stakeholder lavorino e interagiscano insieme per consegnare il massimo valore. Tale sezione si focalizza inoltre sulle tre dimensioni fondamentali del lavoro collaborativo: consapevolezza, articolazione e appropriazione.

**2.6 Prioritizzazione basata sul Valore**—Questa sezione illustra il quarto principio di Scrum, che mette in evidenza l'impulso del framework Scrum a consegnare il massimo valore di business in uno spazio minimo di tempo.

**2.7 Time-boxing**—Questa sezione spiega il quinto principio di Scrum, che tratta il tempo come un vincolo limitante. Qui si parla inoltre dello Sprint, del Daily Standup Meeting, e delle varie altre riunioni connesse allo Sprint, come lo Sprint Planning Meeting, le quali tutte sono di durata temporale predeterminata (Time-boxed).

**2.8 Sviluppo Iterativo**—Questa sezione affronta il sesto principio di Scrum, il quale pone l'accento sul fatto che lo sviluppo iterativo aiuta a gestire meglio i cambiamenti e a costruire prodotti che soddisfino i bisogni del cliente.

**2.9 Confronto fra Scrum e il Project Management Tradizionale**—Questa sezione mette in evidenza le differenze chiave fra i principi di Scrum e i principi del project management tradizionale (modello a cascata) e spiega come Scrum funzioni meglio in un mondo in rapida evoluzione come quello odierno.

## 2.2 Guida per i Ruoli

Tutti le sezioni di questo capitolo sono importanti per tutti i ruoli dello Scrum Core Team—Product Owner, Scrum Master e Scrum Team. Una chiara comprensione dei principi di Scrum da parte di tutti gli stakeholder è essenziale per fare del framework Scrum un successo in qualsiasi organizzazione.

## 2.3 Controllo Empirico di Processo

In Scrum, le decisioni vengono prese sulla base dell'osservazione e della sperimentazione piuttosto che su una dettagliata pianificazione anticipata. Il controllo empirico di processo fa affidamento sulle tre idee principali di trasparenza, ispezione e adattamento.

### 2.3.1 Trasparenza

La trasparenza permette a chiunque di osservare tutti i risvolti di un qualunque processo Scrum. Questo promuove un flusso facile e trasparente di informazioni in tutta l'organizzazione e crea una cultura di lavoro aperto. In Scrum, la trasparenza è rappresentata attraverso:

- Una Dichiarazione della Vision del Progetto che può essere visionata da tutti gli stakeholder e dallo Scrum Team
- Un Prioritized Product Backlog aperto contenente le User Story messe in ordine di priorità che possono essere esaminate da qualunque persona, sia interna che esterna allo Scrum Team

- Una Schedulazione del Piano dei Rilasci che può essere coordinata per vari Scrum Team
- Una chiara visibilità dello stato di avanzamento del team grazie all'uso di una Scrumboard, di un Burndown Chart e di altri radiatori delle informazioni
- I Daily Standup Meeting condotti durante il processo *Condurre il Daily Standup*, nei quali tutti i membri del team riferiscono cosa hanno fatto il giorno precedente, cosa pianificano di fare oggi, ed eventuali problemi che gli impediscono di completare le attività dello Sprint in corso
- Gli Sprint Review Meeting condotti durante il processo *Illustrare e Convalidare lo Sprint*, nei quali lo Scrum Team illustra al Product Owner e agli Stakeholder i Deliverable dello Sprint potenzialmente consegnabili

La Figura 2-1 sintetizza il concetto di trasparenza in Scrum.

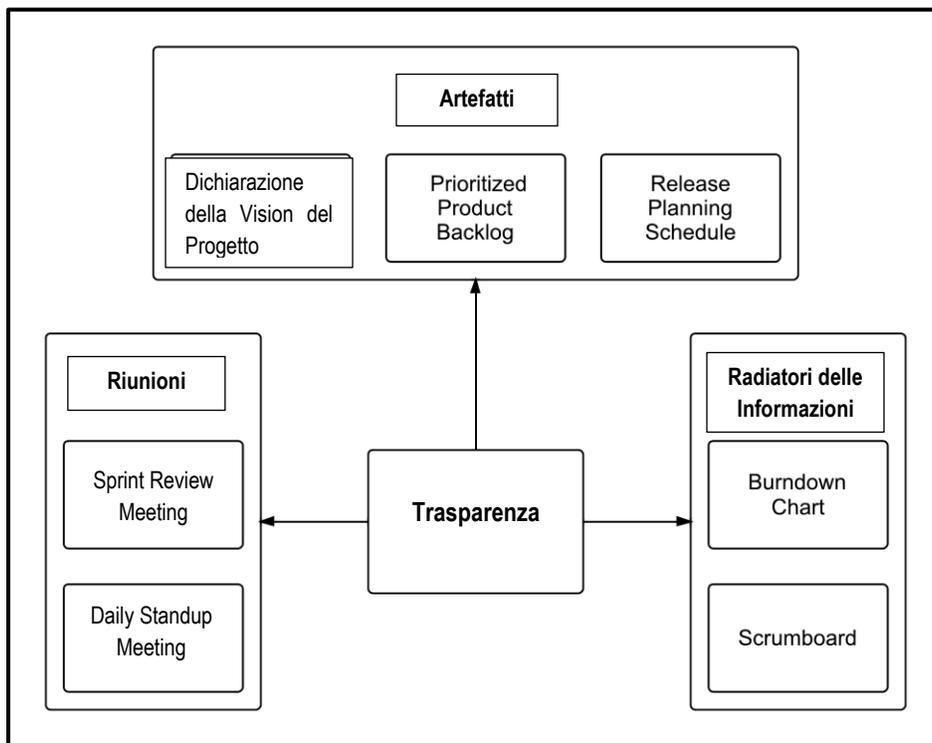


Figura 2-1: Trasparenza in Scrum

### 2.3.2 Ispezione

In Scrum l'ispezione è rappresentata attraverso:

- L'utilizzo di una comune Scrumboard e di altri radiatori delle informazioni che mostrano lo stato di avanzamento dello Scrum Team nel completamento delle attività dello Sprint in corso.
- La raccolta di feedback dal cliente e da altri stakeholder durante i processi *Sviluppare le Epic*, *Creare il Prioritized Product Backlog* e *Eeguire la Pianificazione dei Rilasci*.
- L'ispezione e l'approvazione dei Deliverable da parte del Product Owner e del Cliente nel processo *Illustrare e Convalidare lo Sprint*.

La Figura 2-2 sintetizza il concetto di Ispezione secondo Scrum.

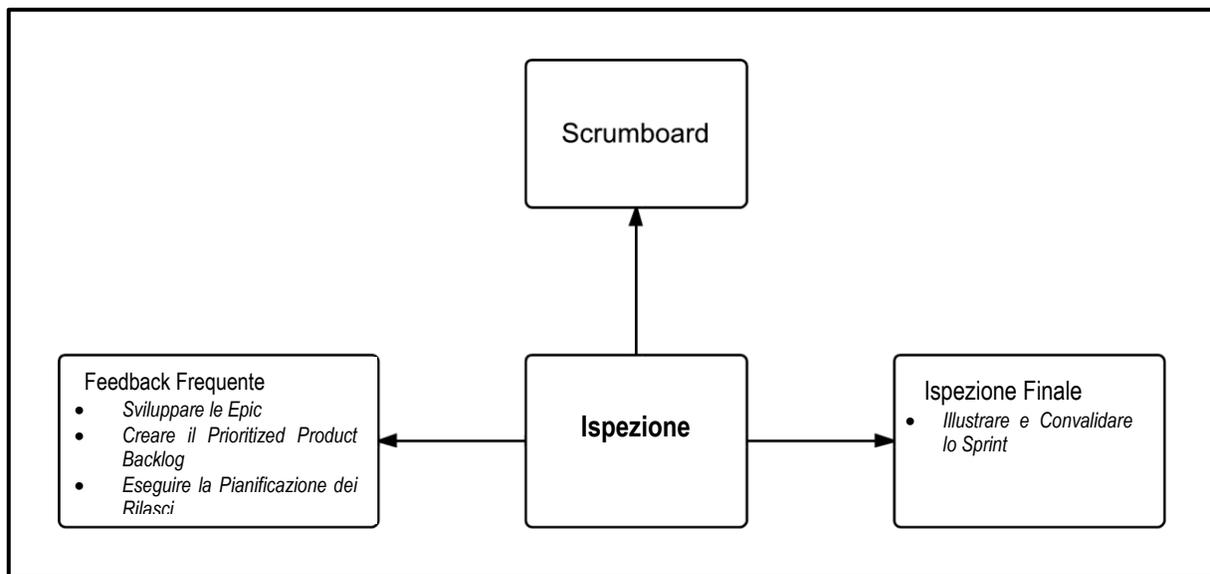


Figura 2-2: Ispezione in Scrum

### 2.3.3 Adattamento

L'adattamento avviene quando lo Scrum Core Team e gli Stakeholder apprendono per mezzo della trasparenza e dell'ispezione e poi si adattano apportando miglioramenti al lavoro che stanno facendo. Alcuni esempi di adattamento sono:

- Nei Daily Standup Meeting i membri dello Scrum Team discutono apertamente gli impedimenti al completamento delle proprie attività e cercano l'aiuto degli altri membri del team. I membri con maggiore esperienza all'interno dello Scrum Team fanno inoltre da mentori a quelli che hanno meno esperienza in termini di conoscenza del progetto o della tecnologia.

- L'identificazione dei rischi viene eseguita e ripetuta per tutta la durata del progetto. I rischi identificati diventano input per molti processi di Scrum, fra cui *Creare il Prioritized Product Backlog*, *Mettere a Punto il Prioritized Product Backlog* e *Illustrare e Convalidare lo Sprint*.
- I miglioramenti possono anche dare luogo a Richieste di Modifica, che sono discusse e approvate durante i processi *Sviluppare le Epic*, *Creare il Prioritized Product Backlog* e *Mettere a Punto il Prioritized Product Backlog*.
- Lo Scrum Guidance Body interagisce con i membri dello Scrum Team durante i processi *Creare le User Story*, *Stimare le Attività*, *Creare i Deliverable* e *Mettere a Punto il Prioritized Product Backlog* per offrire un orientamento e fornire competenze esperte ove necessario.
- Nel processo *Retrospectiva dello Sprint* si stabiliscono i Miglioramenti Fattibili Concordati, sulla base degli output provenienti dal processo *Illustrare e Convalidare lo Sprint*.
- Nel Retrospect Project Meeting, i partecipanti documentano le Lezioni Apprese ed eseguono revisioni alla ricerca di opportunità per migliorare i processi e affrontare le inefficienze.

La Figura 2-3 sintetizza il concetto di adattamento secondo Scrum.

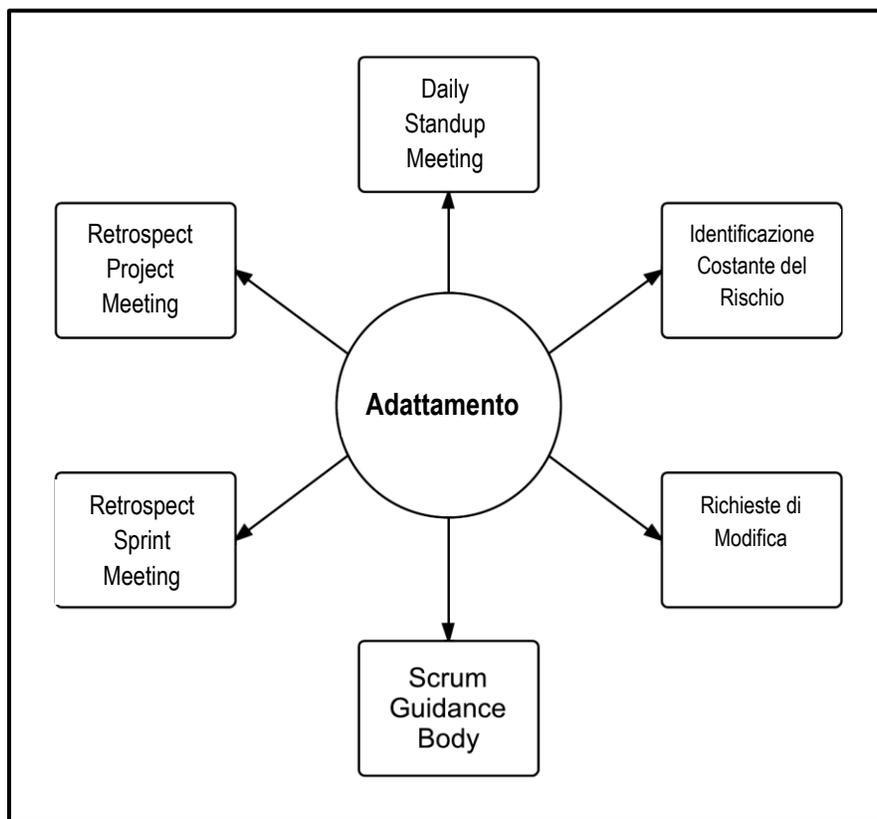


Figura 2-3: Adattamento in Scrum

Con altri metodi, come il tradizionale modello a cascata (Waterfall), è necessario predisporre in anticipo gran parte della pianificazione e il cliente in genere non esamina i componenti del prodotto fino quasi alla fine

della fase, o alla fine dell'intero progetto. Questo metodo presenta spesso rischi enormi per il successo del progetto perché può avere maggiori possibilità di incidere significativamente sulla consegna del progetto e sull'accettazione del cliente. L'interpretazione e la comprensione del prodotto finito da parte del cliente può essere molto diversa da ciò che è stato effettivamente compreso e prodotto dal team e questo potrebbe non emergere se non in uno stadio molto avanzato dello sviluppo del progetto.

La Figura 2-4 illustra un esempio di queste sfide.

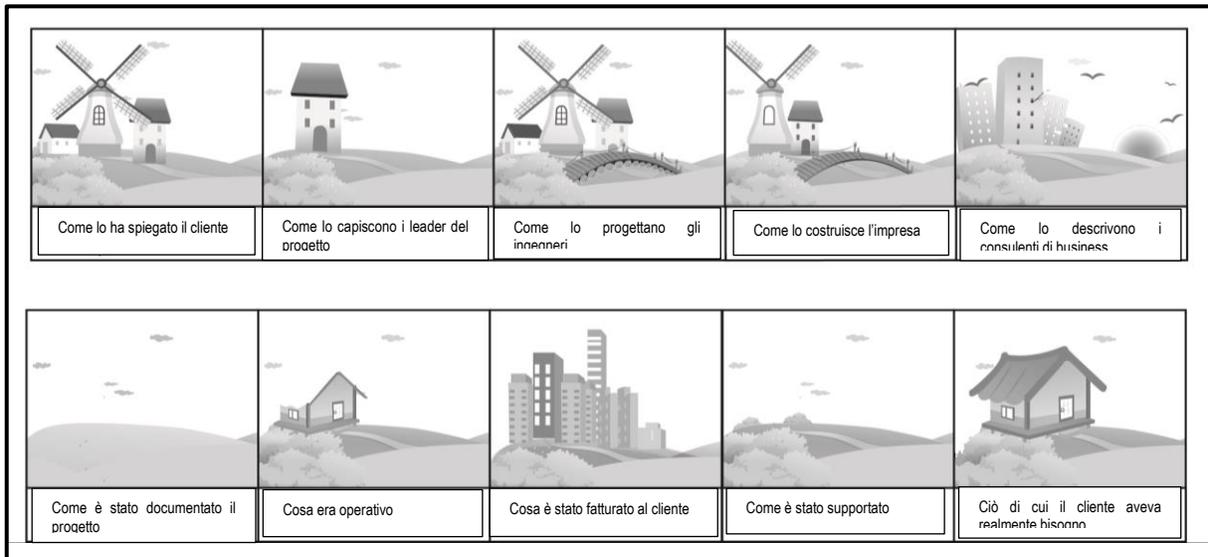


Figura 2-4: Sfide del Project Management Tradizionale

## 2.4 Auto-organizzazione

Scrum crede che i lavoratori siano auto-motivati e cerchino di accettare responsabilità più grandi. Pertanto, essi realizzano un valore molto maggiore quando sono auto-organizzati.

Lo stile di leadership preferito da Scrum è la “leadership del servitore” (“servant leadership”), che pone l’accento sul raggiungimento dei risultati attraverso la focalizzazione sui bisogni dello Scrum Team. Per una discussione sui vari stili di leadership e di gestione consultare la sezione 3.10.3.

### 2.4.1 Benefici dell’Auto-organizzazione

L’auto-organizzazione come principio essenziale di Scrum porta a:

- Adesione totale (c.d. buy-in) del team e responsabilità condivisa
- Motivazione, che porta ad un miglioramento del livello della prestazione del team
- Ambiente innovativo e creativo che contribuisce alla crescita

Auto-organizzazione non significa che i membri del team possono agire in qualunque modo vogliano. Significa solo che una volta che la Vision del Progetto è definita nel processo *Creare la Vision del Progetto*, il Product Owner, lo Scrum Master e lo Scrum sono identificati. Inoltre lo stesso Scrum Core Team lavora a strettissimo contatto con gli Stakeholder per perfezionare al meglio i requisiti, man mano che porta avanti i processi *Sviluppare le Epic* e *Creare le User Story*. La competenza del team è usata per valutare gli input necessari all’esecuzione del lavoro pianificato per il progetto. Questo giudizio e competenza sono applicati a tutti gli aspetti tecnici e di gestione del progetto durante il processo *Creare i Deliverable*.

Anche se la prioritizzazione è compito soprattutto del Product Owner che rappresenta la Voce del Cliente (Voice of Customer), lo Scrum Team auto-organizzato è coinvolto nella scomposizione e nella stima delle attività durante i processi *Creare le Attività* e *Stimare le Attività*. Durante questi processi, ciascun membro del team è responsabile di stabilire quale lavoro lui o lei eseguirà. Se durante l’esecuzione di uno Sprint i membri del team hanno bisogno di un aiuto per completare le proprie attività, Scrum affronta tale problematica attraverso l’interazione obbligatoria ad intervalli regolari che si realizza con i Daily Standup Meeting. Lo stesso Scrum Team interagisce con gli altri team grazie agli Scrum of Scrums (SoS) Meeting e se necessario può cercare ulteriori indicazioni dallo Scrum Guidance Body.

Infine, lo Scrum Team e lo Scrum Master lavorano a stretto contatto per illustrare l’incremento di prodotto creato durante lo Sprint nel processo *Illustrare e Convalidare lo Sprint*, nel quale i deliverable completati in modo corretto sono accettati. Dal momento che i Deliverable sono potenzialmente consegnabili (ed il Prioritized Product Backlog è prioritizzato per User Story ordinate in virtù del valore da esse creato), il Product Owner ed il cliente possono visualizzare ed esprimere in maniera chiara il valore che viene creato dopo ogni Sprint; e in cambio gli Scrum Team hanno la soddisfazione di vedere il loro duro lavoro accettato dal cliente e dagli altri stakeholder.

Gli obiettivi principali dei team auto-organizzati sono i seguenti:

- Comprendere la Vision del Progetto e perché il progetto realizza valore per l'organizzazione
- Stimare le User Story durante il processo *Approvare, Stimare e Prendere in Carico le User Story* e assegnarsi le attività durante il processo *Creare lo Sprint Backlog*
- Creare le attività in modo indipendente durante il processo *Creare le Attività*
- Applicare e sfruttare la propria competenza di team interfunzionale per lavorare sulle attività durante il processo *Creare i Deliverable*
- Consegnare risultati tangibili che sono accettati dal cliente e dagli altri stakeholder durante il processo *Illustrare e Convalidare lo Sprint*
- Risolvere insieme i problemi individuali affrontandoli durante i Daily Standup Meeting
- Chiarire eventuali discrepanze o dubbi ed essere aperti ad apprendere nuove cose
- Aumentare continuamente la conoscenza e la capacità con regolari interazioni all'interno del team
- Mantenere la stabilità dei membri del team per tutta la durata del progetto, non cambiandone i membri a meno che non sia inevitabile.

La Figura 2-5 illustra gli obiettivi di un team auto-organizzato.

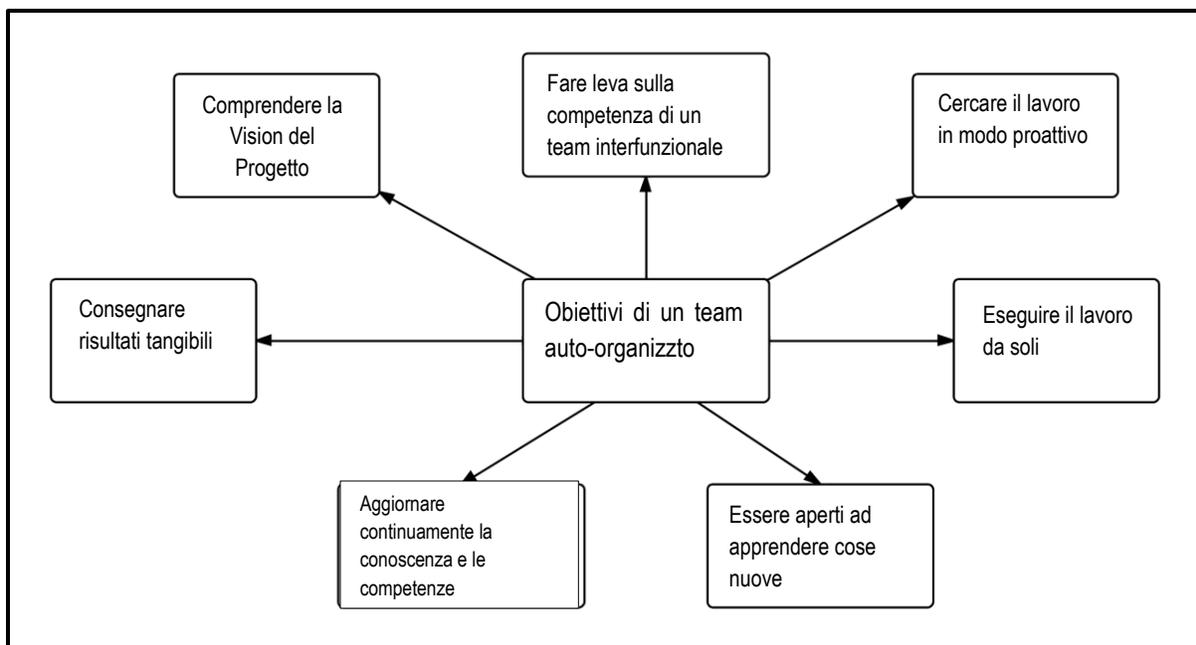


Figura 2-5: Obiettivi di un Team Auto-Organizzato

## 2.5 Collaborazione

In Scrum la collaborazione si riferisce allo Scrum Core Team che lavora insieme e si interfaccia con gli Stakeholder per creare e convalidare i deliverable del progetto allo scopo di soddisfare gli obiettivi delineati nella Vision del Progetto. È importante notare qui la differenza fra cooperazione e collaborazione. La cooperazione si ha quando il prodotto del lavoro consiste nella somma degli impegni di lavoro delle varie persone di un team. La collaborazione si ha quando un team lavora insieme per trarre vantaggio dai reciproci contributi allo scopo di produrre qualcosa di più grande.

Le dimensioni fondamentali del lavoro collaborativo sono:

- **Consapevolezza**—Le persone che lavorano insieme hanno bisogno di essere consapevoli le une del lavoro delle altre.
- **Articolazione**— Le persone che collaborano devono suddividere il lavoro in unità, ripartire le unità fra i membri del team e successivamente riaggregare il lavoro quando è completato.
- **Appropriazione**—Adattare la tecnologia alla propria particolare situazione; la tecnologia può essere usata in modo completamente diverso da quello atteso da chi l'ha progettata.

### 2.5.1 Benefici della Collaborazione nei Progetti Scrum

Il Manifesto Agile (Fowler e Highsmith, 2001) pone l'accento sulla "collaborazione con il cliente prima della negoziazione del contratto". Perciò, il framework Scrum adotta un approccio nel quale i membri dello Scrum Core Team (Product Owner, Scrum Master e Scrum Team) collaborano fra di loro e con gli stakeholder per creare i deliverable che offrono il maggior valore possibile al cliente. Questa collaborazione si realizza per tutta la durata del progetto.

La Collaborazione consente di ottenere i seguenti benefici:

1. Viene ridotta al minimo la necessità di cambiamenti dovuti a requisiti mal chiariti. Per esempio, durante i processi *Creare la Vision del Progetto*, *Sviluppare le Epic* e *Creare il Prioritized Product Backlog*, il Product Owner collabora con gli stakeholder per creare rispettivamente la Vision del Progetto, le Epic ed il Prioritized Product Backlog. Questo farà sì che ci sia chiarezza fra i membri dello Scrum Core Team riguardo al lavoro necessario per portare a compimento il progetto. Lo Scrum Team collabora continuamente con il Product Owner e con gli stakeholder per mezzo di un Prioritized Product Backlog trasparente per creare i deliverable del progetto. I processi *Condurre il Daily Standup*, *Mettere a Punto il Prioritized Product Backlog* e *Retrospectiva dello Sprint* danno modo allo Scrum Core Team di discutere di ciò che è stato fatto e di collaborare a ciò che deve essere fatto. In questo modo il numero di Richieste di Modifica da parte del Cliente e di rilavorazioni è ridotto al minimo.
2. I rischi sono identificati e trattati in modo efficiente. Per esempio, i rischi di progetto sono identificati e valutati dai membri dello Scrum Core Team nell'ambito dei processi *Sviluppare le Epic*, *Creare i*

*Deliverable e Condurre il Daily Standup.* Gli strumenti costituiti dalle riunioni di Scrum, come il Daily Standup Meeting, lo Sprint Planning Meeting, il Prioritized Product Backlog Review Meeting, e così via, offrono al team la possibilità non solo di identificare e valutare i rischi, ma anche di implementare le risposte per i rischi ad alta priorità.

3. Si realizza il pieno potenziale del team. Ad esempio, il processo *Condurre il Daily Standup* dà modo allo Scrum Team di collaborare e comprendere i punti di forza e di debolezza dei propri membri. Se un membro del team non ha rispettato la scadenza di un'attività, i membri dello Scrum Team collaborano tutti insieme per finire quella attività e raggiungere gli obiettivi concordati per il completamento dello Sprint.
4. Si garantisce il miglioramento continuo attraverso le lezioni apprese. Ad esempio, lo Scrum Team utilizza il processo *Retrospettiva dello Sprint* per identificare ciò che è andato bene e ciò che non è andato bene nel precedente Sprint. Questo offre allo Scrum Master l'opportunità di lavorare con il team ad una rielaborazione e ad un miglioramento in vista del successivo Sprint programmato. Ciò consentirà inoltre una collaborazione ancora più efficace nello Sprint successivo.

La Figura 2-6 illustra i benefici della collaborazione nei progetti Scrum.

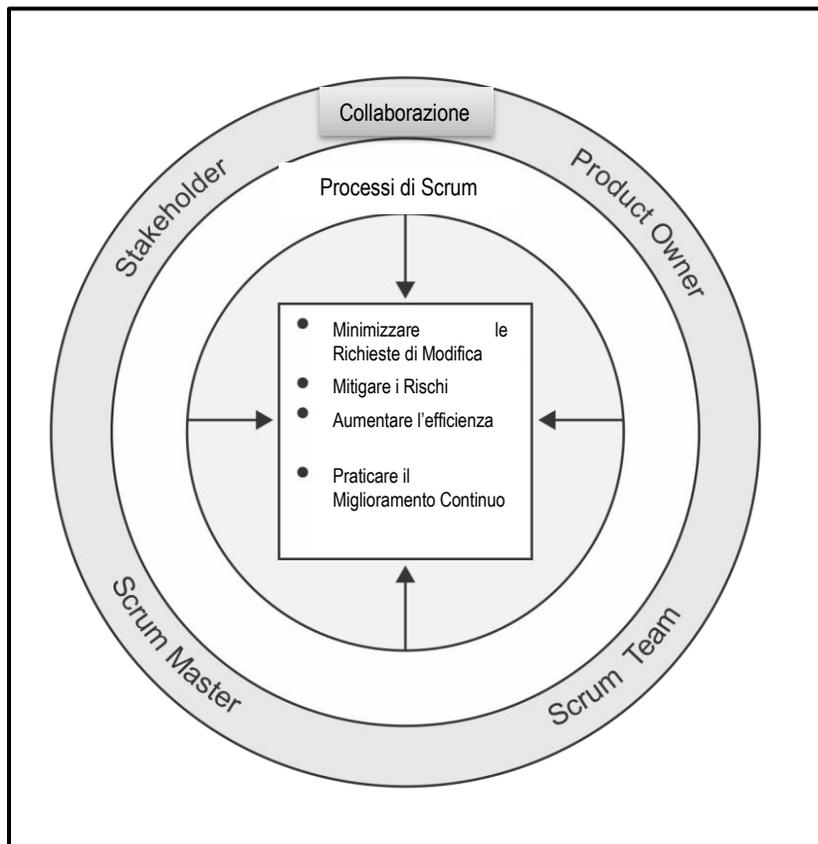


Figura 2-6: Benefici della Collaborazione nei Progetti Scrum

## 2.5.2 Importanza della Co-ubicazione per la Collaborazione

Per molte delle pratiche di Scrum è necessaria una comunicazione a banda larga (vale a dire orale e diretta). Per permettere ciò, è preferibile la co-ubicazione dei membri del team. La co-ubicazione permette l'interazione sia formale che informale fra i membri del team. La stessa offre il vantaggio di avere i membri del team sempre a portata di mano per il coordinamento, la risoluzione dei problemi e l'apprendimento. Di seguito sono elencati alcuni dei benefici della co-ubicazione:

- Le domande ricevono risposte in tempi rapidi.
- I problemi vengono risolti immediatamente.
- Le interazioni provocano minore attrito.
- La fiducia si guadagna e viene premiata molto più rapidamente.

Gli strumenti di Collaborazione che si possono utilizzare per i team co-ubicati o distribuiti sono:

1. **Team Co-ubicati** (cioè team che lavorano nello stesso ufficio)–Per Scrum è preferibile avere team co-ubicati. In tal caso, le modalità di comunicazione preferite sono le interazioni faccia a faccia, le Decision Room o War Room, le Scrumboard, gli espositori da parete, i tavoli condivisivi, e così via.
2. **Team Distribuiti** (cioè team che lavorano in luoghi fisici diversi)– Anche se sono preferibili i team co-ubicati, a volte è possibile che lo Scrum Team sia distribuito a causa dell'outsourcing, della delocalizzazione, di sedi fisiche diverse, della scelta di lavorare da casa, ecc. Alcuni strumenti da utilizzare per un'efficace collaborazione nei team distribuiti sono le videoconferenze, la messaggistica istantanea, le chat, i social media, la condivisione di schermi e strumenti software che simulano le funzionalità delle Scrumboard, degli espositori da parete, e così via.

## 2.6 Prioritizzazione basata sul Valore

Il framework Scrum è guidato dall'obiettivo di realizzare il massimo valore di business nel minimo lasso di tempo. Uno degli strumenti più efficaci per consegnare il massimo valore nel più breve tempo possibile è rappresentato dalla prioritizzazione.

La prioritizzazione può essere definita come la determinazione dell'ordine e la separazione di ciò che deve essere fatto ora da quello che occorre fare in un secondo momento. Il concetto di prioritizzazione non è nuovo al project management. Il tradizionale modello di project management a cascata (Waterfall) propone l'utilizzo di molteplici strumenti per la prioritizzazione delle attività. Dal punto di vista del Project Manager, la prioritizzazione è essenziale in quanto alcune attività devono essere eseguite prima per accelerare il processo di sviluppo e realizzare gli obiettivi del progetto. Alcune delle tecniche tradizionali di

priorizzazione delle attività sono rappresentate dalla fissazione di scadenze per le attività delegate e dall'utilizzo di matrici di prioritizzazione.

Scrum, però, utilizza la Priorizzazione basata sul Valore come uno dei principi fondamentali che guidano la struttura e la funzione dell'intero framework Scrum – quello che aiuta i progetti a trarre beneficio dall'adattabilità e dallo sviluppo iterativo del prodotto o servizio. Più significativamente, Scrum punta a consegnare al cliente un prodotto o servizio di valore fin da subito e continuamente.

Il Product Owner esegue la prioritizzazione nel momento in cui ordina per priorità le User Story all'interno del Prioritized Product Backlog. Il Prioritized Product Backlog contiene un elenco di tutti i requisiti richiesti per portare a compimento il progetto.

Una volta che ha ricevuto i requisiti di business dal cliente e li ha messi giù in forma di User Story fattibili, il Product Owner lavora con il cliente e lo sponsor per capire quali requisiti di business offrano il massimo valore di business. Il Product Owner deve capire cosa vuole il cliente e ciò a cui dà valore, in modo da organizzare gli elementi del Prioritized Product Backlog (le User Story) in base all'importanza relativa. A volte, un cliente può esigere che sia assegnata una priorità alta a tutte le User Story. Sebbene ciò possa essere vero, anche un elenco di User Story ad alta priorità deve essere organizzato al suo interno per ordine di priorità. La prioritizzazione di un backlog implica la decisione circa la criticità di ciascuna User Story. I requisiti di valore elevato vengono identificati e collocati in cima al Prioritized Product Backlog. I processi nei quali viene messo in pratica il principio della Priorizzazione basata sul Valore sono *Creare il Prioritized Product Backlog* e *Mettere a Punto il Prioritized Product Backlog*.

Contemporaneamente, il Product Owner deve lavorare con lo Scrum Team per capire quali possano essere i rischi e le incertezze del progetto, in quanto a questi potrebbero essere associate conseguenze negative. Di ciò si dovrebbe tenere conto quando si prioritizzano le User Story in base al loro valore (per maggiori informazioni consultare il capitolo sul Rischio). Lo Scrum Team avvisa inoltre il Product Owner di qualsiasi dipendenza che dovesse emergere dall'implementazione. Queste dipendenze devono essere tenute presenti durante la prioritizzazione. La prioritizzazione può essere basata su una stima soggettiva del valore di business o della redditività previsti, oppure può basarsi su risultati e analisi di mercato, utilizzando strumenti fra cui, ma non solo, le interviste al cliente, i sondaggi, i modelli finanziari e le tecniche analitiche.

Il Product Owner deve tradurre gli input e i bisogni degli stakeholder del progetto per creare il Prioritized Product Backlog. Pertanto, nel mettere in ordine di priorità le User Story all'interno del Prioritized Product Backlog, vengono considerati i seguenti tre fattori (si veda la Figura 2-7):

1. Valore
2. Rischio o incertezza
3. Dipendenze

La prioritizzazione si traduce quindi in deliverable che soddisfano i requisiti del cliente con l'obiettivo di consegnare il massimo valore di business nel minor lasso di tempo possibile.

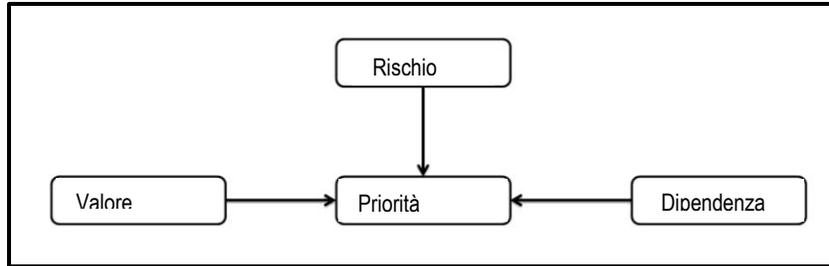


Figura 2-7: Prioritizzazione basata sul valore

## 2.7 Time-boxing

Scrum tratta il tempo come uno dei vincoli più importanti nella gestione di un progetto. Per far fronte al vincolo di tempo, Scrum introduce un concetto chiamato 'Time-boxing' il quale propone di fissare un certo quantitativo di tempo per ciascun processo e attività di un progetto Scrum. Questo fa sì che i membri dello Scrum Team non accettino né troppo né troppo poco lavoro per un determinato periodo di tempo e non spendano il loro tempo e le loro energie su un lavoro poco chiaro.

Alcuni dei vantaggi del Time-boxing sono:

- Processi efficienti di sviluppo
- Minori spese generali
- Alta velocità dei team

Il Time-boxing può essere utilizzato in molti processi di Scrum, per esempio nel processo *Condurre il Daily Standup* la durata del Daily Standup Meeting è predeterminata. A volte il Time-boxing può essere utilizzato per evitare il miglioramento eccessivo di un elemento (vale a dire il gold-plating).

Il Time-boxing è una pratica fondamentale in Scrum e dovrebbe essere applicata con scrupolo. Il Time-boxing arbitrario può portare alla demotivazione del team e può avere come conseguenza la instaurazione di un clima apprensivo, per cui dovrebbe essere usato in modo appropriato.

### 2.7.1 I Time-box di Scrum

- **Sprint**—Uno Sprint è un'iterazione di durata predeterminata da una a sei settimane, durante le quali lo Scrum Master guida, facilita, e protegge lo Scrum Team dagli impedimenti sia interni che esterni nel corso del processo *Creare i Deliverable*. Questo aiuta ad evitare la 'deriva' della vision che potrebbe ripercuotersi sull'obiettivo dello Sprint. Durante questo periodo di tempo, il team lavora per convertire i requisiti del Prioritized Product Backlog in funzionalità del prodotto consegnabili. Per ricavare il massimo dei benefici da un progetto Scrum, si raccomanda sempre di

fissare la durata dello Sprint in 4 settimane, salvo che nei progetti con requisiti molto stabili, nei quali gli Sprint possono durare fino a 6 settimane.

- **Daily Standup Meeting**— Il Daily Standup Meeting è una breve riunione quotidiana, che ha una durata predeterminata di 15 minuti. I membri del team si riuniscono per fare un resoconto sullo stato di avanzamento del progetto rispondendo alle seguenti tre domande:
  1. Cosa ho completato ieri?
  2. Cosa completerò oggi?
  3. Quali eventuali impedimenti o ostacoli sto riscontrando attualmente?

Questa riunione viene tenuta dal team come parte del processo *Condurre il Daily Standup*.

- **Sprint Planning Meeting**—Questa riunione viene condotta prima dello Sprint come parte del processo *Creare lo Sprint Backlog*. Ha una durata predeterminata di otto ore per uno Sprint di un mese. Lo Sprint Planning Meeting si divide in due parti:
  1. Definizione dell'Obiettivo—Durante la prima metà della riunione, il Product Owner spiega allo Scrum Team le User Story o i requisiti del Prioritized Product Backlog che hanno la priorità più alta. Lo Scrum Team definisce quindi, in collaborazione con il Product Owner, l'obiettivo dello Sprint.
  2. Stima delle Attività—Durante la seconda metà della riunione, lo Scrum Team decide “come” completare gli elementi selezionati del Prioritized Product Backlog per realizzare l'obiettivo dello Sprint.

A volte, i Task Planning Meeting (condotti durante il processo *Creare le Attività*) e i Task Estimation Meeting (condotti durante il processo *Stimare le Attività*) vengono chiamati anche Sprint Planning Meeting.

- **Sprint Review Meeting**—Lo Sprint Review Meeting ha una durata predeterminata di quattro ore per uno Sprint di un mese. Durante lo Sprint Review Meeting, che si tiene nel processo *Illustrare e Convalidare lo Sprint*, lo Scrum Team presenta al Product Owner i deliverable dello Sprint in corso. Il Product Owner rivede il prodotto (o l'incremento di prodotto) in rapporto ai Criteri di Accettazione concordati e procede ad accettare o rifiutare le User Story completate.
- **Retrospect Sprint Meeting**— Il Retrospect Sprint Meeting ha una durata predeterminata di 4 ore per uno Sprint di un mese e viene eseguito come parte del processo *Retrospectiva dello Sprint*. Durante questo incontro, lo Scrum Team si riunisce per rivedere lo Sprint precedente e riflettere su di esso in termini di processi seguiti, strumenti impiegati, meccanismi di collaborazione e comunicazione ed altri aspetti di rilievo del progetto.  
Il team discute di ciò che è andato bene e di ciò che è andato male nello Sprint precedente, con l'obiettivo di imparare e apportare miglioramenti negli Sprint a seguire. Alcune opportunità di

miglioramento o best practice provenienti da questa riunione possono anche entrare a fare parte degli aggiornamenti dei documenti dello Scrum Guidance Body.

La Figura 2-8 illustra le durate predeterminate delle riunioni relative a Scrum.

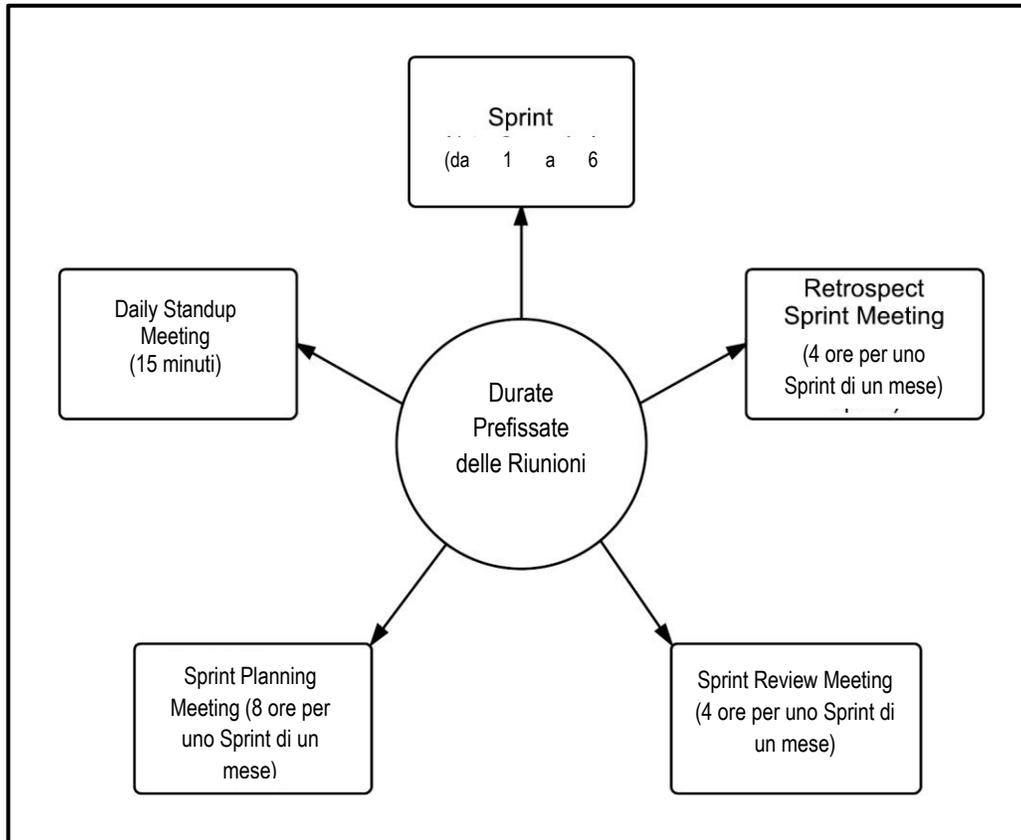


Figura 2-8: Durate prefissate delle riunioni di Scrum

## 2.8 Sviluppo Iterativo

Il framework Scrum è guidato dall'obiettivo di consegnare il massimo valore di business in un minimo lasso di tempo. Per raggiungere in pratica questo risultato, Scrum punta sullo Sviluppo Iterativo dei Deliverable.

Nella maggior parte dei progetti complessi, il cliente può non essere in grado di definire requisiti molto concreti oppure non è sicuro di come possa presentarsi il prodotto finale. Il modello iterativo è più flessibile nel garantire la possibilità di includere nel progetto eventuali modifiche richieste dal cliente. È possibile dover scrivere le User Story per tutta la durata del progetto. Nelle fasi iniziali di scrittura, la maggior parte delle User Story sono funzionalità di alto livello. Queste User Story sono conosciute come Epic. Le Epic sono di norma troppo ampie per essere completate dai team in un singolo Sprint. Per questo motivo, vengono scomposte in User Story più piccole.

Ogni aspetto complesso del progetto viene scomposto attraverso un'elaborazione progressiva durante il processo *Mettere a Punto il Prioritized Product Backlog*. I processi *Creare le User Story* e *Stimare, Approvare e Prendere in Carico le User Story* vengono utilizzati per aggiungere nuovi requisiti al Prioritized Product Backlog. Il compito del Product Owner è di assicurare l'aumento del ROI concentrandosi sul valore e sulla sua continua consegna in ogni Sprint. Il Product Owner deve comprendere molto bene la giustificazione commerciale del progetto e il valore che si suppone venga consegnato dal progetto stesso quando delinea il contenuto del Prioritized Product Backlog, e in tal modo decidere quali deliverable e quindi quali valori devono essere consegnati in ogni Sprint. Dopo di che, i processi *Creare le Attività*, *Stimare le Attività* e *Creare lo Sprint Backlog* producono lo Sprint Backlog che il team utilizza per creare i deliverable.

Il processo *Creare i Deliverable* viene eseguito in ogni Sprint per creare gli output dello Sprint stesso. Lo Scrum Master deve fare in modo che i processi di Scrum siano seguiti e facilitare il team affinché possa lavorare nel modo più produttivo possibile. Lo Scrum Team si auto-organizza al fine di creare i Deliverable dello Sprint partendo dalle User Story inserite nello Sprint Backlog. Nei progetti di grandi dimensioni, vari team interfunzionali lavorano in parallelo attraverso gli Sprint, fornendo soluzioni potenzialmente consegnabili alla fine di ogni Sprint. Dopo il completamento dello Sprint, il Product Owner accetta o rifiuta i deliverable sulla base dei Criteri di Accettazione nell'ambito del processo *Illustrare e Convalidare lo Sprint*.

Come illustrato nella Figura 2-9, i progetti Scrum sono completati in modo iterativo e consegnano valore durante tutto il ciclo di vita del progetto.

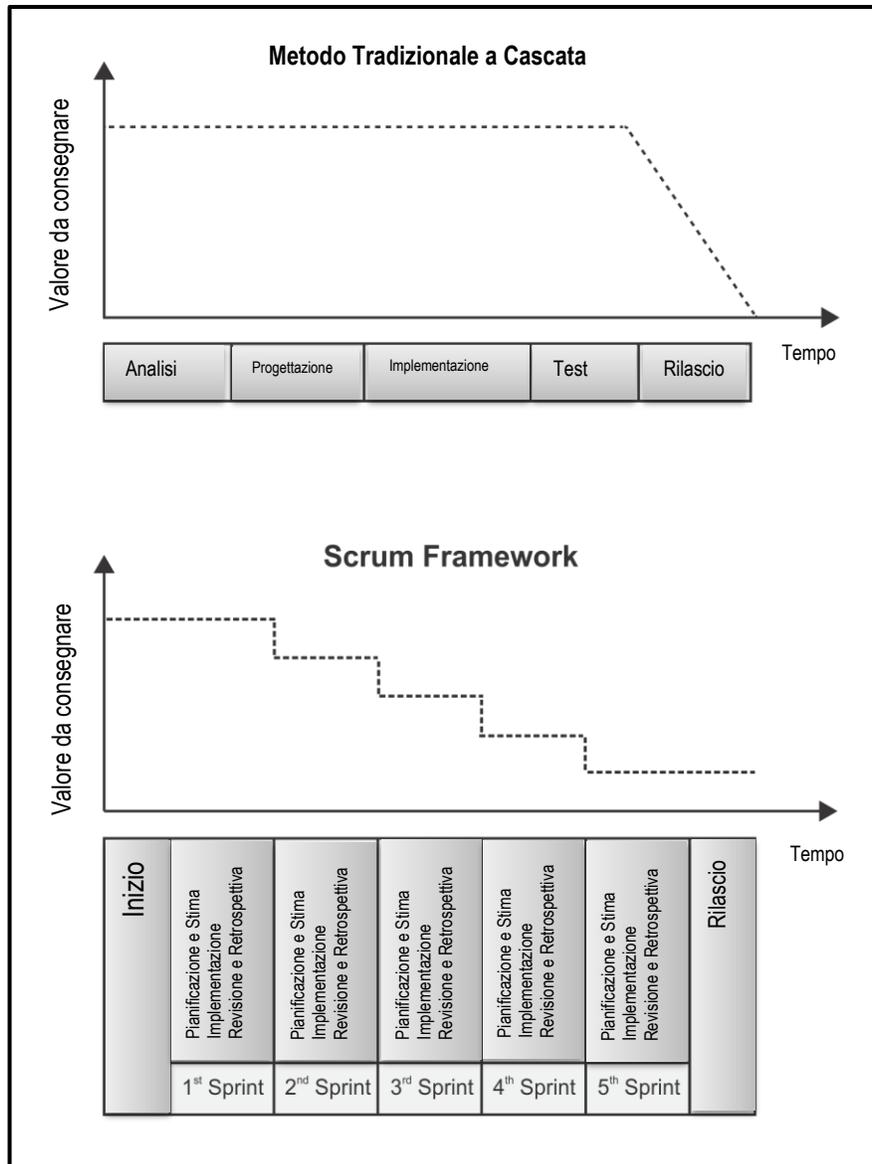


Figura 2-9: Confronto fra Scrum ed il Metodo Tradizionale a Cascata (Waterfall)

Il vantaggio dello sviluppo iterativo è che permette la correzione dell'andamento, in quanto tutte le persone coinvolte hanno una migliore comprensione di ciò che deve essere consegnato dal progetto e incorporano quanto appreso in modo iterativo. Per questo il tempo e l'impegno richiesti per raggiungere il punto finale si riducono enormemente e il team produce deliverable che sono più adatti al contesto finale di business.

## 2.9 Confronto fra Scrum e il Project Management Tradizionale

L'attenzione del Project Management tradizionale è rivolta alla esecuzione di una dettagliata pianificazione anticipata del progetto con particolare attenzione alla fissazione dell'ambito, dei costi e delle tempistiche e alla gestione di tali parametri. Il project management tradizionale può a volte portare a situazioni nelle quali il piano ha avuto successo ma il cliente non è soddisfatto.

Il Framework Scrum si fonda sulla convinzione che i lavoratori di oggi possono offrire molto più che la loro sola competenza tecnica, e che cercare di definire e pianificare tutto in un ambiente in continuo cambiamento non è efficiente. Per questo, Scrum promuove il processo decisionale iterativo e basato sui dati. In Scrum, l'attenzione è rivolta principalmente alla consegna di prodotti che soddisfino i requisiti del cliente in piccoli ma utilizzabili incrementi iterativi.

Per consegnare la massima quantità di valore nel minor tempo possibile, Scrum promuove la prioritizzazione e il Time-boxing prima della fissazione dell'ambito, dei costi e delle tempistiche di un progetto. Una caratteristica importante di Scrum è l'auto-organizzazione, che permette a coloro che eseguono effettivamente il lavoro di stimare le attività e assumersene la responsabilità.

## 3. ORGANIZZAZIONE

### 3.1 Introduzione

In questa sezione discuteremo dei vari aspetti dell'organizzazione di un progetto Scrum, oltre che dei ruoli core e non-core e di come costituire Scrum Team ad alte prestazioni.

L'Organizzazione, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Questo capitolo si divide nelle seguenti sezioni:

**3.2 Guida per i Ruoli** — Questa sezione identifica quali parti o paragrafi del capitolo sono importanti per il Product Owner, per lo Scrum Master e per lo Scrum Team.

**3.3 Ruoli del Progetto Scrum**— Questa sezione tratta tutti i ruoli chiave core e non-core associati ad un progetto Scrum.

**3.4 Product Owner**—Questa sezione evidenzia le responsabilità fondamentali del Product Owner rispetto ad un progetto Scrum.

**3.5 Scrum Master**— Questa sezione si concentra sulle responsabilità fondamentali dello Scrum Master nel contesto di un progetto Scrum.

**3.6 Scrum Team**—Questa sezione pone l'accento sulle responsabilità fondamentali dello Scrum Team nel contesto di un progetto Scrum.

**3.7 Scrum nei Progetti, Programmi e Portfolio** —Questa sezione illustra come adattare ed utilizzare il framework Scrum nei differenti contesti dei programmi e dei portfolio. Inoltre evidenzia le specifiche responsabilità dei membri dello Scrum Team riguardo alla comunicazione, alla integrazione e al lavoro con i team aziendali e di gestione del programma.

**3.8 Responsabilità** —Questa sezione descrive le responsabilità attinenti alla tematica Organizzazione facenti capo, in base ai rispettivi ruoli, a tutti coloro che lavorano ad un progetto.

**3.9 Confronto fra Scrum e il Project Management Tradizionale**—Questa sezione spiega le differenze fondamentali e i vantaggi del modello Scrum rispetto al modello tradizionale di Project Management a cascata (Waterfall).

**3.10 Teorie diffuse sulle Risorse Umane e loro rilevanza rispetto a Scrum**—Questa sezione contiene alcune delle più popolari teorie sulle risorse umane utili per tutti i membri dello Scrum Core Team.

## 3.2 Guida per i Ruoli

1. **Product Owner**— I Product Owner devono necessariamente leggere l'intero capitolo.
2. **Scrum Master**— Anche lo Scrum Master dovrebbe conoscere l'intero capitolo, con particolare attenzione ai paragrafi 3.3, 3.5, 3.6, 3.8 e 3.10.4.
3. **Scrum Team**— Lo Scrum Team dovrebbe concentrarsi soprattutto sui paragrafi 3.3, 3.6 e 3.8.

## 3.3 I Ruoli del Progetto Scrum

Per assicurare il successo nella implementazione di progetti Scrum è molto importante comprendere la definizione dei ruoli e delle responsabilità.

I ruoli di Scrum ricadono in due categorie generali:

1. **Ruoli Core**— I ruoli core sono quelli obbligatoriamente richiesti per la produzione del prodotto del progetto, sono dedicati al progetto e sono in sostanza responsabili del successo di ciascuno Sprint del progetto e del progetto nel suo insieme.
2. **Ruoli Non-core**— I ruoli non-core sono i ruoli non richiesti obbligatoriamente per il progetto Scrum, e possono includere membri del team che sono interessati al progetto, non hanno un ruolo formale all'interno del team di progetto, possono interfacciarsi con il team, ma non sono necessariamente responsabili del successo del progetto. Anche i ruoli non-core dovrebbero essere tenuti in considerazione in qualunque progetto Scrum.

### 3.3.1 Ruoli Core

In Scrum ci sono tre ruoli core, che sono in sostanza responsabili della realizzazione degli obiettivi del progetto. I ruoli core sono il Product Owner, lo Scrum Master e lo Scrum Team. Nel loro insieme vengono chiamati lo Scrum Core Team. È importante notare che nessuno di questi tre ruoli esercita un'autorità sugli altri.

#### 1. Product Owner

Il Product Owner è il soggetto responsabile di trarre il massimo valore di business dal progetto. Lui o lei ha la responsabilità di esprimere al meglio i requisiti del cliente e di mantenere salda la giustificazione commerciale del progetto. Il Product Owner rappresenta la Voce del Cliente.

Parallelamente al ruolo di Product Owner di un progetto, può essere presente un Program Product Owner per il programma o un Portfolio Product Owner per il portfolio.

#### 2. Scrum Master

Lo Scrum Master è un facilitatore che fa in modo che lo Scrum Team abbia a disposizione un ambiente che favorisca la possibilità di completare con successo lo sviluppo del prodotto. Lo Scrum Master guida, facilita, ed insegna le pratiche di Scrum a tutti coloro che sono coinvolti nel progetto; risolve gli impedimenti riscontrati dal team; e infine si assicura che vengano seguiti i processi di Scrum.

Si fa notare che il ruolo dello Scrum Master è molto diverso dal ruolo ricoperto dal Project Manager in un modello tradizionale di project management a cascata (Waterfall), nel quale il Project Manager lavora come gestore o leader del progetto. Lo Scrum Master lavora solo come facilitatore e si trova allo stesso livello gerarchico di qualunque altro componente dello Scrum Team: qualsiasi persona appartenente allo Scrum Team che impara come facilitare i progetti Scrum può diventare lo Scrum Master di un progetto o di un singolo Sprint.

Parallelamente al ruolo di Scrum Master di un progetto, può essere presente un Program Scrum Master per il programma o un Portfolio Scrum Master per il portfolio.

#### 3. Scrum Team

Lo Scrum Team è un gruppo o team di persone che hanno la responsabilità di capire i requisiti di business specificati dal Product Owner, di stimare le User Story, e infine di creare i Deliverable del progetto.

La Figura 3-1 presenta una visione d'insieme dei ruoli Core dei Team Scrum

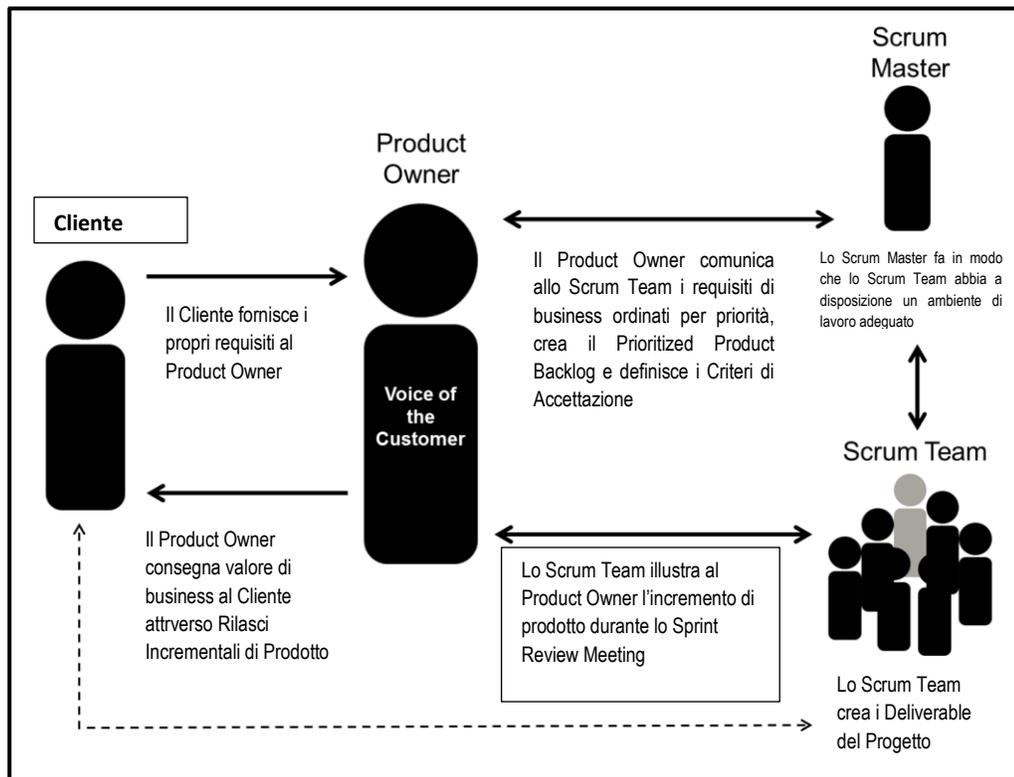


Figura 3-1: Ruoli di Scrum—Visione d'insieme

### 3.3.2 Ruoli Non-core

I ruoli non-core sono quelli non obbligatoriamente richiesti per il progetto Scrum e possono non essere continuamente o direttamente coinvolti nei processi di Scrum. Tuttavia, è importante conoscere i ruoli non-core in quanto questi possono svolgere una parte importante in alcuni progetti Scrum.

Fra i ruoli non-core sono compresi:

#### 1. Stakeholder

Stakeholder è un termine collettivo che ricomprende i clienti, gli utenti e gli sponsor, i quali tutti si interfacciano spesso con il Product Owner, lo Scrum Master e lo Scrum Team per fornire loro input e facilitare la creazione del prodotto, servizio o altro risultato del progetto. Gli Stakeholder influenzano il progetto durante tutto il suo sviluppo. Gli Stakeholder possono inoltre avere un ruolo durante i processi *Sviluppare le Epic*, *Creare il Prioritized Product Backlog*, *Eseguire la Pianificazione dei Rilasci*, *Retrospektiva dello Sprint*, ed in altri importanti processi di Scrum.

- **Cliente**

Il cliente è l'individuo o l'organizzazione che acquista il prodotto, il servizio o altro risultato del progetto. Per ciascuna organizzazione, a seconda del progetto, ci possono essere sia clienti interni (cioè all'interno della stessa organizzazione), che clienti esterni (cioè, al di fuori dell'organizzazione).

- **Utenti**

Gli utenti sono gli individui o le organizzazioni che usano direttamente il prodotto, servizio o altro risultato del progetto. Così come per i clienti, ciascuna organizzazione può avere utenti sia interni che esterni. Inoltre, in alcuni settori industriali clienti e utenti possono coincidere.

- **Sponsor**

Lo sponsor è la persona o l'organizzazione che mette a disposizione le risorse e il supporto per il progetto. Tale figura rappresenta anche lo stakeholder al quale alla fine tutti devono rispondere.

A volte, la stessa persona o organizzazione può rivestire più ruoli da stakeholder; per esempio, lo sponsor ed il cliente possono coincidere.

## 2. Venditori

I venditori sono le persone o le organizzazioni esterne che forniscono prodotti o servizi che non rientrano nelle competenze principali dell'organizzazione di progetto.

## 4. Scrum Guidance Body

Lo Scrum Guidance Body (SGB) è un ruolo opzionale. Generalmente consiste in un gruppo di documenti e/o in un gruppo di esperti che si occupano di solito della definizione degli obiettivi relativi alla qualità, ai regolamenti governativi, alla sicurezza, e ad altri parametri chiave dell'organizzazione. Questi obiettivi guidano il lavoro svolto dal Product Owner, dallo Scrum Master e dallo Scrum Team. Lo Scrum Guidance Body aiuta inoltre a individuare le best practice da utilizzare in tutti i progetti Scrum dell'organizzazione.

Lo Scrum Guidance Body non prende decisioni riguardo al progetto. Agisce invece come una struttura di consulenza o orientamento per tutti i livelli gerarchici dell'organizzazione di progetto – quello di portfolio, quello di programma e quello di progetto. Gli Scrum Team hanno la possibilità, se necessario, di richiedere un parere allo Scrum Guidance Body.

## 3.4 Product Owner

Il Product Owner rappresenta gli interessi della comunità degli stakeholder di fronte allo Scrum Team. Il Product Owner ha la responsabilità di fare in modo che i requisiti di funzionalità del prodotto o servizio siano comunicati in modo chiaro allo Scrum Team, di definire i Criteri di Accettazione e di assicurarne la soddisfazione. In altre parole, il Product Owner deve assicurare la consegna di valore da parte dello Scrum Team. Il Product Owner deve sempre mantenere un duplice punto di vista. Deve capire e supportare i bisogni e gli interessi di tutti gli stakeholder e nel contempo comprendere anche le esigenze ed il funzionamento dello Scrum Team. Dal momento che il Product Owner deve capire le esigenze e le priorità degli stakeholder, questo ruolo viene di solito chiamato la Voce del Cliente.

La Tavola 3-1 riassume le responsabilità del Product Owner nei vari processi di Scrum.

Processo	Responsabilità del Product Owner
8.1 Creare la Vision del Progetto	<ul style="list-style-type: none"> <li>Definisce la Vision del Progetto</li> <li>Aiuta a creare il Project Charter e il Budget di Progetto</li> </ul>
8.2 Identificare lo Scrum Master e gli Stakeholder	<ul style="list-style-type: none"> <li>Aiuta a finalizzare la nomina dello Scrum Master del progetto</li> <li>Identifica gli Stakeholder</li> </ul>
8.3 Formare lo Scrum Team	<ul style="list-style-type: none"> <li>Aiuta a determinare i membri dello Scrum Team</li> <li>Aiuta a sviluppare un Piano di Collaborazione</li> <li>Contribuisce insieme allo/agli Scrum Master alla creazione del Piano di Team Building</li> </ul>
8.4 Sviluppare le Epic	<ul style="list-style-type: none"> <li>Crea le Epic e le <i>Persona</i></li> </ul>
8.5 Creare il Prioritized Product Backlog	<ul style="list-style-type: none"> <li>Prioritizza gli elementi del Prioritized Product Backlog</li> <li>Definisce i Done Criteria</li> </ul>
8.6 Eseguire la Pianificazione dei Rilasci	<ul style="list-style-type: none"> <li>Crea la Schedulazione del Piano dei Rilasci</li> <li>Aiuta a stabilire la Lunghezza dello Sprint</li> </ul>
9.1 Creare le User Story	<ul style="list-style-type: none"> <li>Aiuta a creare le User Story</li> <li>Definisce i Criteri di Accettazione per ciascuna User Story</li> </ul>
9.2 Approvare, Stimare e Prendere in Carico le User Story	<ul style="list-style-type: none"> <li>Approva le User Story</li> <li>Facilita lo Scrum Team e gli affida le User Story</li> </ul>
9.3 Creare le Attività	<ul style="list-style-type: none"> <li>Spiega le User Story allo Scrum Team mentre questo crea la Lista delle Attività</li> </ul>
9.4 Stimare le Attività	<ul style="list-style-type: none"> <li>Fornisce orientamento e chiarimenti allo Scrum Team sulla stima dell'impegno necessario all'esecuzione delle attività</li> </ul>
9.5 Creare lo Sprint Backlog	<ul style="list-style-type: none"> <li>Spiega i requisiti allo Scrum Team mentre questo predispone lo Sprint Backlog</li> </ul>
10.1 Creare i Deliverable	<ul style="list-style-type: none"> <li>Spiega i requisiti di business allo Scrum Team</li> </ul>
10.3 Mettere a Punto il Prioritized Product Backlog	<ul style="list-style-type: none"> <li>Mette a Punto il Prioritized Product Backlog</li> </ul>
11.2 Illustrare e Convalidare gli Sprint	<ul style="list-style-type: none"> <li>Accetta/Rifiuta i Deliverable</li> <li>Fornisce allo Scrum Master e agli Scrum Team il feedback</li> </ul>

	necessario <ul style="list-style-type: none"> <li>• Aggiorna il Piano dei Rilasci e il Prioritized Product Backlog</li> </ul>
12.1 Consegnare i Deliverable	<ul style="list-style-type: none"> <li>• Aiuta a distribuire i Rilasci di Prodotto coordinandosi con il cliente</li> </ul>
12.2 Retrospettiva del Progetto	<ul style="list-style-type: none"> <li>• Partecipa ai Retrospective Sprint Meeting</li> </ul>

Tavola 3-1: Responsabilità del Product Owner nei Processi di Scrum

Le altre responsabilità di un Product Owner riguardano:

- Stabilire i requisiti generali iniziali del progetto e dare il via alle attività di progetto; questo può comportare un'interazione con il Program Product Owner e il Portfolio Product Owner per assicurare l'allineamento del progetto alle direttive fornite dal senior management.
- Rappresentare gli utenti del prodotto o servizio con una comprensione accurata della comunità degli utenti.
- Ottenere la certezza della disponibilità delle risorse finanziarie iniziali e per tutta la durata del progetto.
- Focalizzarsi sulla creazione di valore e sul complessivo Ritorno sull'Investimento (ROI).
- Valutare la fattibilità ed assicurare la consegna del prodotto o servizio.

### 3.4.1 Voce del Cliente (Voice of the Customer - VOC)

Come rappresentante del cliente, si dice che il Product Owner è la Voce del Cliente, perché fa in modo che i bisogni dichiarati e non dichiarati del cliente siano tradotti in User Story inserite nel Prioritized Product Backlog e poi utilizzate per creare i Deliverable di progetto per il cliente.

### 3.4.2 Chief Product Owner

Nel caso di progetti di grandi dimensioni con molti Scrum Team, può essere necessario avere un Chief Product Owner. Questo ruolo ha la responsabilità di coordinare il lavoro di molteplici Product Owner. Il Chief Product Owner prepara e mantiene il Prioritized Product Backlog complessivo del progetto di grandi dimensioni, utilizzandolo per coordinare il lavoro dei Product Owner dei vari Scrum Team. I Product Owner, a loro volta, gestiscono le proprie rispettive parti del Prioritized Product Backlog.

Il Chief Product Owner si interfaccia anche con il Program Product Owner per assicurare l'allineamento del progetto di grandi dimensioni con gli obiettivi specifici e generali del programma.

## 3.5 Scrum Master

Lo Scrum Master è il “servant leader”, cioè il leader servitore, dello Scrum Team, modera e facilita le relazioni del team fungendone da maestro e motivatore. Lo Scrum Master ha la responsabilità di fare in modo che il team abbia a disposizione un ambiente di lavoro produttivo proteggendolo da influenze esterne, rimuovendo qualsiasi ostacolo e facendo rispettare i principi, gli aspetti e i processi di Scrum.

La Tavola 3-2 riassume le responsabilità dello Scrum Master nei vari processi di Scrum.

Processi	Responsabilità dello Scrum Master
8.2 Identificare lo Scrum Master e gli Stakeholder	<ul style="list-style-type: none"> <li>• Aiuta a identificare gli Stakeholder del progetto</li> </ul>
8.3 Formare lo Scrum Team	<ul style="list-style-type: none"> <li>• Facilita la selezione dello Scrum Team</li> <li>• Facilita la creazione del Piano di Collaborazione e del Piano di Team Building</li> <li>• Assicura che siano disponibili risorse di riserva per l'agevole funzionamento del progetto</li> </ul>
8.4 Sviluppare le Epic	<ul style="list-style-type: none"> <li>• Facilita la creazione delle Epic e delle <i>Persona</i></li> </ul>
8.5 Creare il Prioritized Product Backlog	<ul style="list-style-type: none"> <li>• Aiuta il Product Owner a creare il Prioritized Product Backlog e a definire i Done Criteria</li> </ul>
8.6 Eseguire la Pianificazione dei Rilasci	<ul style="list-style-type: none"> <li>• Coordina la creazione della Schedulazione del Piano dei Rilasci</li> <li>• Stabilisce la Lunghezza dello Sprint</li> </ul>
9.1 Creare le User Story	<ul style="list-style-type: none"> <li>• Assiste lo Scrum Team nella creazione delle User Story e dei loro Criteri di Accettazione</li> </ul>
9.2 Approvare, Stimare e Prendere in Carico le User Story	<ul style="list-style-type: none"> <li>• Facilita le riunioni dello Scrum Team volte alla stima e alla creazione delle User Story</li> </ul>
9.3 Creare le Attività	<ul style="list-style-type: none"> <li>• Facilita lo Scrum Team nella creazione della Lista delle Attività per lo Sprint successivo</li> </ul>
9.4 Stimare le Attività	<ul style="list-style-type: none"> <li>• Assiste lo Scrum Team nella stima dell'impegno necessario per completare le attività concordate per lo Sprint</li> </ul>
9.5 Creare lo Sprint Backlog	<ul style="list-style-type: none"> <li>• Assiste lo Scrum Team nello sviluppo dello Sprint Backlog e dello Sprint Burndown Chart</li> </ul>
10.1 Creare i Deliverable	<ul style="list-style-type: none"> <li>• Supporta lo Scrum Team nella creazione dei Deliverable concordati per lo Sprint</li> <li>• Aiuta ad aggiornare la Scrumboard e il Registro degli Impedimenti</li> </ul>
10.2 Condurre il Daily Standup	<ul style="list-style-type: none"> <li>• Fa in modo che la Scrumboard e il Registro degli Impedimenti rimangano sempre aggiornati</li> </ul>
10.3 Mettere a Punto il Prioritized Product Backlog	<ul style="list-style-type: none"> <li>• Facilita i Prioritized Product Backlog Review Meeting</li> </ul>
11.1 Convocare lo Scrum degli Scrum	<ul style="list-style-type: none"> <li>• Si assicura che le questioni che riguardano lo Scrum Team siano discusse e risolte</li> </ul>
11.2 Illustrare e Convalidare gli	<ul style="list-style-type: none"> <li>• Facilita la presentazione dei Deliverable completati dallo Scrum</li> </ul>

Sprint	Team ai fini dell'approvazione del Product Owner
11.3 Retrospettiva dello Sprint	<ul style="list-style-type: none"> <li>• Si assicura che negli Sprint successivi possa esserci un ambiente ideale per lo Scrum Team</li> </ul>
12.2 Retrospettiva del Progetto	<ul style="list-style-type: none"> <li>• Rappresenta lo Scrum Core Team per fornire all'occorrenza lezioni ricavate dal progetto attuale.</li> </ul>

Tavola 3-2: Responsabilità dello Scrum Master nei Processi di Scrum

### 3.5.1 Chief Scrum Master

I progetti di grandi dimensioni richiedono il lavoro parallelo di molti Scrum Team. È possibile che le informazioni raccolte da un team debbano essere comunicate adeguatamente agli altri team. Il Chief Scrum Master è responsabile di questa attività.

Il coordinamento dei vari Scrum Team che lavorano ad un progetto si realizza di solito per mezzo dello Scrum of Scrums (SoS) Meeting (vedi paragrafo 3.7.2.1). Questa riunione è analoga al Daily Standup Meeting ed è facilitata dal Chief Scrum Master. Il Chief Scrum Master è solitamente la persona che ha la responsabilità di affrontare gli impedimenti che impattano più di uno Scrum Team.

La Figura 3-2 elenca le domande alle quali si dà risposta durante uno Scrum of Scrums (SoS) Meeting

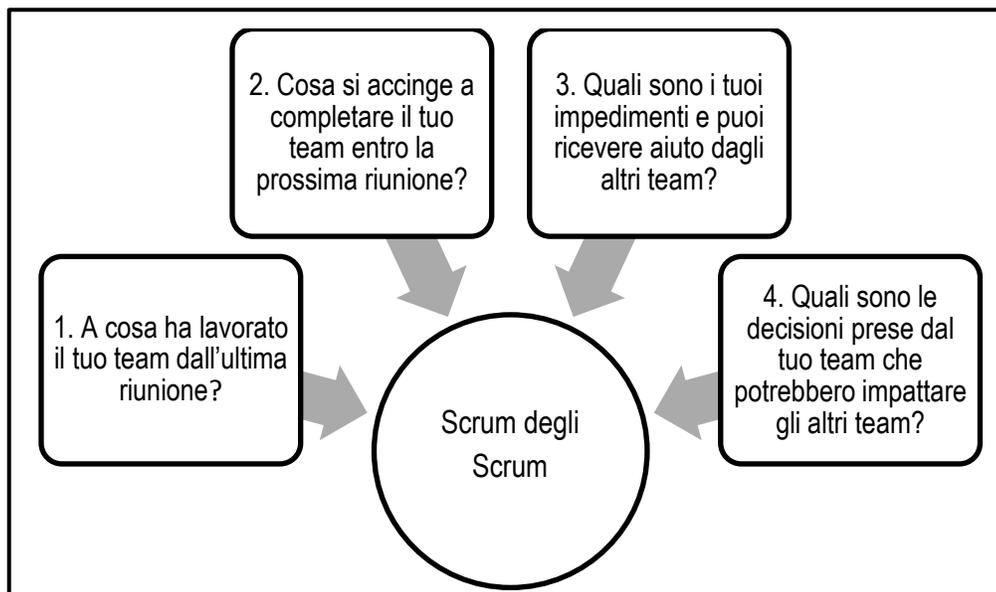


Figura 3-2: Domande alle quali si dà risposta durante uno Scrum of Scrums Meeting

Di solito, eventuali questioni fra i team sono affrontate dalle parti interessate in una sessione immediatamente successiva allo Scrum of Scrums Meeting. Questa sessione viene facilitata dal Chief Scrum Master.

## 3.6 Scrum Team

Lo Scrum Team viene a volte chiamato Team di Sviluppo poiché ha la responsabilità di sviluppare il prodotto, servizio o altro risultato. Esso consiste in un gruppo di persone che lavorano alle User Story contenute nello Sprint Backlog per creare i Deliverable del progetto.

La Tavola 3-3 riassume le responsabilità dello Scrum Team nei vari processi di Scrum.

Processi	Responsabilità dello Scrum Team
8.3 Formare lo Scrum Team	<ul style="list-style-type: none"> <li>Fornisce input per la creazione del Piano di Collaborazione e del Piano di Team Building</li> </ul>
8.4 Sviluppare le Epic	<ul style="list-style-type: none"> <li>Si assicura una piena comprensione delle Epic e delle <i>Persona</i></li> </ul>
8.5 Creare il Prioritized Product Backlog	<ul style="list-style-type: none"> <li>Comprende le User Story del Prioritized Product Backlog</li> </ul>
8.6 Eseguire la Pianificazione dei Rilasci	<ul style="list-style-type: none"> <li>Concorda la Lunghezza dello Sprint con gli altri membri dello Scrum Core Team</li> <li>Cerca spiegazioni sui prodotti nuovi o sulle eventuali modifiche ai prodotti esistenti all'interno del Prioritized Product Backlog rifinito</li> </ul>
9.1 Creare le User Story	<ul style="list-style-type: none"> <li>Fornisce input al Product Owner riguardo alla creazione delle User Story</li> </ul>
9.2 Approvare, Stimare e Prendere in Carico le User Story	<ul style="list-style-type: none"> <li>Stima le User Story approvate dal Product Owner</li> <li>Prende in Carico le User Story da eseguire in uno Sprint</li> </ul>
9.3 Creare le Attività	<ul style="list-style-type: none"> <li>Sviluppa la Lista delle Attività sulla base delle User Story concordate e delle dipendenze</li> </ul>
9.4 Stimare le Attività	<ul style="list-style-type: none"> <li>Stima l'impegno necessario per le attività identificate e, all'occorrenza, aggiorna la Lista delle Attività</li> </ul>
9.5 Creare lo Sprint Backlog	<ul style="list-style-type: none"> <li>Sviluppa lo Sprint Backlog e lo Sprint Burndown Chart</li> </ul>
10.1 Creare i Deliverable	<ul style="list-style-type: none"> <li>Crea i Deliverable</li> <li>Identifica i rischi e implementa le eventuali azioni di mitigazione del rischio</li> <li>Aggiorna il Registro degli Impedimenti e le dipendenze</li> </ul>
10.2 Condurre il Daily Standup	<ul style="list-style-type: none"> <li>Aggiorna il Burndown Chart, la Scrumboard, e il Registro degli Impedimenti</li> <li>Discute le questioni riscontrate dai singoli membri e cerca soluzioni per motivare il team</li> </ul>

	<ul style="list-style-type: none"> <li>• Identifica gli eventuali rischi</li> <li>• Presenta le Richieste di Modifica, se necessario</li> </ul>
10.3 Mettere a Punto il Prioritized Product Backlog	<ul style="list-style-type: none"> <li>• Partecipa ai Prioritized Product Backlog Review Meeting</li> </ul>
11.1 Convocare lo Scrum degli Scrum	<ul style="list-style-type: none"> <li>• Fornisce input allo Scrum Master per gli Scrum of Scrums (SoS) Meeting</li> </ul>
11.2 Illustrare e Convalidare gli Sprint	<ul style="list-style-type: none"> <li>• Illustra i deliverable completati al Product Owner per ottenerne l'approvazione</li> </ul>
11.3 Retrospettiva dello Sprint	<ul style="list-style-type: none"> <li>• Identifica le eventuali opportunità di miglioramento provenienti dallo Sprint in corso e concorda sugli eventuali miglioramenti fattibili per lo Sprint successivo</li> </ul>
12.2 Retrospettiva del Progetto	<ul style="list-style-type: none"> <li>• Partecipa al Retrospect Project Meeting</li> </ul>

Tavola 3-3: Responsabilità dello Scrum Team nei Processi di Scrum

### 3.6.1 Selezione del Personale

La Figura 3-3 elenca le caratteristiche desiderabili per i ruoli core di Scrum.

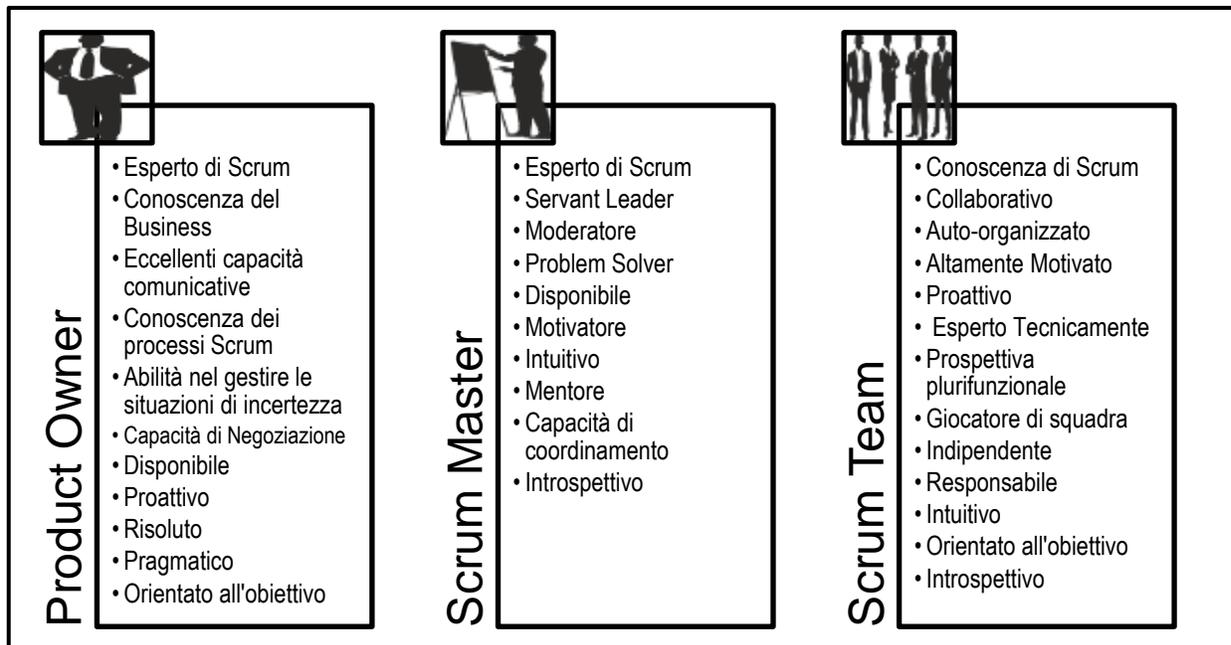


Figura 3-3: Caratteristiche desiderabili per i Ruoli Core di Scrum

### 3.6.2 Dimensione dello Scrum Team

È importante che lo Scrum Team possieda tutte le competenze essenziali richieste per portare a compimento il lavoro del progetto. È inoltre necessario un elevato livello di collaborazione per massimizzare la produttività e ridurre quindi al minimo la necessità di coordinamento per fare il lavoro.

La dimensione ottimale di uno Scrum Team è dai sei ai dieci membri – grande abbastanza da assicurare la presenza di un adeguato set di competenze, ma sufficientemente piccolo da poter collaborare facilmente. Un beneficio fondamentale di un team dai sei ai dieci membri è dato dal fatto che la comunicazione e la gestione sono generalmente semplici e richiedono uno sforzo minimo. Tuttavia, ci possono essere anche dei lati negativi. Uno dei più importanti è che i team più piccoli subiscono un impatto più significativo dalla perdita di un membro del team rispetto ai team più grandi, anche se è per un breve periodo di tempo. A questo problema si può ovviare se i membri del team hanno conoscenze esperte e competenze al di fuori del loro specifico ruolo. Tuttavia, ciò può essere difficile e dipende dal tipo di progetto, settore industriale, e dimensione dell'organizzazione. Si raccomanda inoltre di prevedere delle riserve per sostituire chiunque possa essere costretto a lasciare lo Scrum Team.

## 3.7 Scrum nei Progetti, Programmi e Portfolio

### 3.7.1 Definizione di Progetto, Programma e Portfolio

- **Progetto**— Un progetto è un'impresa collaborativa volta a creare nuovi prodotti o servizi o a consegnare risultati così come definiti nella Dichiarazione della Vision del Progetto. I progetti sono di norma condizionati da vincoli di tempo, di costi, di ambito, di qualità, di persone e di capacità organizzative. L'obiettivo del team di progetto è quello di creare i Deliverable come definiti nel Prioritized Product Backlog.
- **Programma**— Un programma è un gruppo di progetti collegati, che ha l'obiettivo di realizzare i risultati di business come definiti nella Dichiarazione della Vision del Programma. Il Prioritized Program Backlog incorpora i Prioritized Product Backlog di tutti i progetti compresi nel programma.
- **Portfolio**—Un portfolio è un gruppo di programmi collegati, che ha l'obiettivo di realizzare i risultati di business come definiti nella Dichiarazione della Vision del Portfolio. Il Prioritized Portfolio Backlog incorpora i Prioritized Program Backlog di tutti i programmi del portfolio.

Quelli che seguono sono esempi di progetti, programmi e portfolio di differenti industrie e settori:

*Esempio 1: Società di Costruzioni Edilizie*

- Progetto- Costruzione di una casa
- Programma—Costruzione di un complesso residenziale
- Portfolio—Tutti i progetti di edilizia abitativa della società

*Esempio 2: Organizzazione aerospaziale*

- Progetto—Costruzione del veicolo di lancio
- Programma—Lancio riuscito di un satellite
- Portfolio—Tutti i programmi satellitari attivi

*Esempio 3: Compagnia di InformationTechnology (IT)*

- Progetto—Sviluppo del modulo carrello della spesa
- Programma—Sviluppo di un sito internet di e-commerce dotato di tutte le funzioni
- Portfolio—Tutti i siti internet sviluppati dalla compagnia fino ad oggi

### 3.7.2 Scrum nei Progetti

Poiché Scrum preferisce i team piccoli, si potrebbe essere indotti a ritenere che questo metodo può essere utilizzato solo per i progetti di piccole dimensioni, ma non è così. Scrum può essere utilizzato in maniera efficace anche nei progetti di grandi dimensioni. Quando c'è bisogno di più di dieci persone per portare a compimento il lavoro, si possono costituire tanti Scrum Team. Il team di progetto è costituito da vari Scrum Team che lavorano insieme per creare i Deliverable e i Rilasci del Prodotto, così come per realizzare i risultati desiderati per l'intero progetto.

Siccome un progetto può avere molti Scrum Team che lavorano in parallelo, diventa importante il coordinamento fra i diversi team. Lo Scrum Team normalmente comunica e si coordina con gli altri in molti modi, ma l'approccio più comune è conosciuto con il nome di Scrum of Scrums (SoS) Meeting. I membri che rappresentano ciascun team si riuniscono per discutere lo stato di avanzamento, le questioni e per coordinare le attività dei vari team. Queste riunioni hanno una struttura simile ai Daily Standup Meeting; tuttavia, la frequenza dello Scrum of Scrum può essere fissata ad intervalli predeterminati oppure coordinata in base alle richieste dei diversi Scrum Team.

#### 3.7.2.1 Scrum of Scrums (SoS) Meeting

Uno Scrum of Scrums Meeting è un elemento importante per l'applicazione di Scrum ai progetti di grandi dimensioni. Normalmente, alla riunione presenza un rappresentante per ciascuno Scrum Team, di solito lo Scrum Master, ma se necessario può partecipare alla riunione anche un qualsiasi membro dello Scrum Team. Questa riunione è facilitata dal Chief Scrum Master e ha lo scopo di focalizzarsi sulle aree di coordinamento e di integrazione fra i diversi Scrum Team. Viene tenuta ad intervalli regolari o quando richiesto dagli Scrum Team.

Nelle organizzazioni che hanno tanti Scrum Team che lavorano simultaneamente su porzioni di un progetto, il SoS Meeting può essere scalato ad un ulteriore livello che viene chiamato Scrum of Scrum of Scrums Meeting. In questa situazione, si tiene un SoS Meeting per coordinare ciascun gruppo di Scrum Team che lavorano su porzioni di progetti fra loro collegati e poi può essere tenuto uno Scrum of Scrum of Scrums Meeting per coordinare ed integrare i progetti ad un livello più alto. I team devono valutare attentamente i benefici che possono trarre dagli Scrum of Scrum of Scrums Meetings, poiché questo terzo livello aggiunge una notevole dose di complessità dal punto di vista logistico.

La Figura 3-4 illustra il concetto dello Scrum of Scrums (SoS) Meeting e quello dello Scrum of Scrum of Scrums Meeting.

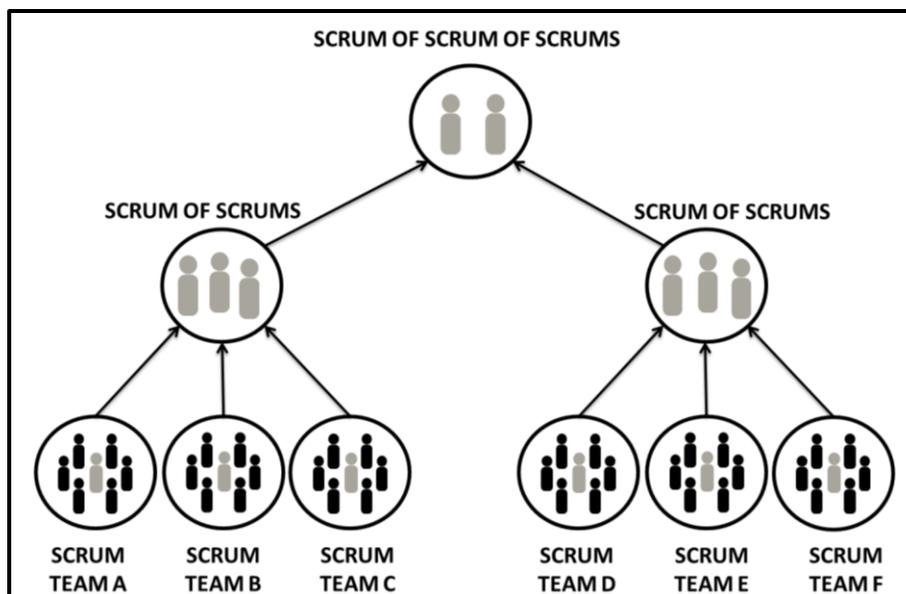


Figura 3-4: Scrum of Scrums (SoS) Meeting

In questo esempio, ci sono sei Scrum Team che lavorano simultaneamente. Gli Scrum Team A, B, e C lavorano su porzioni di un progetto collegato mentre gli Scrum Team D, E, e F lavorano su porzioni di un altro progetto collegato. Si tiene uno Scrum of Scrums Meeting per coordinare le interdipendenze fra i progetti collegati. Può essere condotto uno Scrum of Scrums of Scrums Meeting per coordinare e gestire le dipendenze fra tutti i progetti.



### 3.7.3 Scrum nei Portfolio e nei Programmi

#### 3.7.3.1 Portfolio

A livello di portfolio, i ruoli importanti per gestire i portfolio di Scrum sono:

1. **Portfolio Product Owner**—Definisce gli obiettivi strategici e le priorità del portfolio.
2. **Portfolio Scrum Master**—Risolve i problemi, rimuove gli impedimenti, facilita e conduce le riunioni di portfolio.

Questi ruoli sono simili a quelli del Product Owner e dello Scrum Master, salvo che per il fatto che essi soddisfano i bisogni del proprio portfolio o dell'intera azienda piuttosto che quelli di un singolo Scrum Team.

#### 3.7.3.2 Programmi

Nell'ambito dei programmi, i ruoli importanti per gestire di programmi Scrum sono:

1. **Program Product Owner**—Definisce gli obiettivi strategici e le priorità del programma.
2. **Program Scrum Master**—Risolve i problemi, rimuove gli impedimenti, facilita, e conduce le riunioni di programma.

Questi ruoli sono simili a quelli del Product Owner e dello Scrum Master, salvo che per il fatto che essi soddisfano i bisogni del proprio programma o dell'unità di business piuttosto che quelli di un singolo Scrum Team.

La Figura 3-5 illustra in che modo l'organizzazione può utilizzare Scrum per i portfolio, i programmi o i progetti.

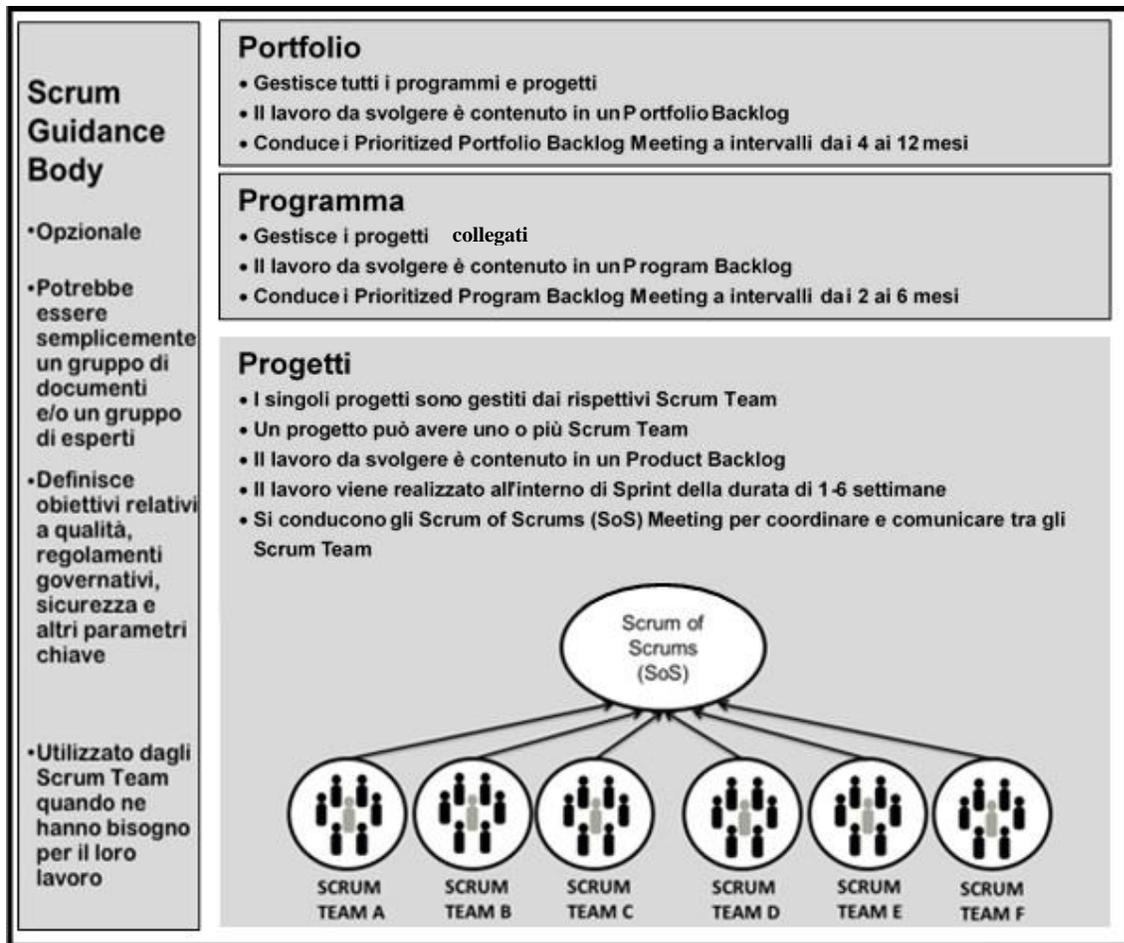


Figura 3-5: Scrum applicato ai Progetti, Programmi e Portfolio dell'Organizzazione

### 3.7.3.3 Lavorare con i Team di Portfolio e di Programma

Quando si applica Scrum per gestire progetti che rientrano nel contesto di un programma o portfolio, è fortemente raccomandata una piena adesione ai principi generali di Scrum esposti in questa pubblicazione. Resta inteso però che, per poter tenere conto delle attività e delle interdipendenze del programma o portfolio complessivi, possono essere necessarie modifiche marginali all'insieme degli strumenti così come alla struttura dell'organizzazione. Se è presente lo Scrum Guidance Body, questo potrebbe avere il compito di analizzare l'organizzazione a differenti livelli per capire e definire un'applicazione appropriata di Scrum e per fungere da organo consultivo per tutti coloro che lavorano ad un progetto, programma o portfolio.

I portfolio e i programmi hanno team separati che perseguono distinti insiemi di obiettivi. I team di gestione del programma puntano a consegnare capacità e a realizzare determinati obiettivi che contribuiscono al raggiungimento degli specifici obiettivi del programma. Viceversa, il team di portfolio deve bilanciare gli obiettivi di vari programmi per realizzare gli obiettivi strategici dell'organizzazione nel suo insieme.

### 3.7.3.4 Gestire le Comunicazioni con i Team di Portfolio e di Programma

I problemi e le questioni che si riscontrano quando si utilizza Scrum all'interno di un programma o portfolio riguardano principalmente il coordinamento di numerosi team. Questa situazione, se non gestita con attenzione, può portare ad un insuccesso. Gli strumenti utilizzati per le comunicazioni hanno bisogno di essere ampliati per soddisfare le esigenze dei molti team coinvolti in un programma o portfolio. Ogni Scrum Team deve considerare non solo le comunicazioni interne, ma anche quelle esterne con gli altri team e con gli stakeholder del programma o del portfolio.

### 3.7.4 Mantenere il Coinvolgimento degli Stakeholder

Scrum richiede un supporto totale da parte degli stakeholder del progetto. La responsabilità di tenere vivo l'impegno degli stakeholder fa capo al Product Owner. Quelle che seguono rappresentano le azioni raccomandate per mantenere il coinvolgimento ed il supporto degli stakeholder:

- Assicurare l'effettiva collaborazione e la partecipazione degli stakeholder al progetto
- Valutare continuamente l'impatto sul business
- Mantenere comunicazioni regolari con gli stakeholder
- Gestire le aspettative degli stakeholder

Uno stakeholder chiave è lo sponsor – il soggetto che mette a disposizione i fondi e le altre risorse che servono al progetto. Gli sponsor vogliono avere chiaro l'utile finanziario collegato ad un prodotto o servizio e di solito si preoccupano più dei risultati finali che delle singole attività.

È importante che gli sponsor che finanziano il progetto abbiano un quadro chiaro sui seguenti punti:

- Benefici derivanti dall'implementazione di Scrum
- Scadenze obiettivo e costi stimati dei progetti Scrum
- Rischi complessivi dei progetti Scrum e passi da compiere per mitigarli
- Date attese dei rilasci e dei Deliverable finali

### 3.8 Riepilogo delle Responsabilità

Ruolo	Responsabilità
Scrum Guidance Body	<ul style="list-style-type: none"> <li>Stabilisce le linee guida e le metriche complessive per lo sviluppo delle descrizioni dei ruoli dei membri dello Scrum Team</li> <li>Funge da consulente per i progetti dell'organizzazione a vari livelli</li> <li>Comprende e definisce gli adeguati livelli di raggruppamento, i ruoli e le riunioni per i progetti Scrum</li> </ul>
Portfolio Product Owner	<ul style="list-style-type: none"> <li>Definisce gli obiettivi strategici e le priorità dei portfolio</li> </ul>
Portfolio Scrum Master	<ul style="list-style-type: none"> <li>Risolve i problemi e coordina le riunioni di portfolio</li> </ul>
Program Product Owner	<ul style="list-style-type: none"> <li>Definisce gli obiettivi strategici e le priorità dei programmi</li> </ul>
Program Scrum Master	<ul style="list-style-type: none"> <li>Risolve i problemi e coordina le riunioni di programma</li> </ul>
Stakeholder	<ul style="list-style-type: none"> <li>È un termine collettivo che include clienti, utenti e sponsor</li> <li>Si interfaccia spesso con il Product Owner, lo Scrum Master e lo Scrum Team per fornire loro input e facilitare la creazione dei Deliverable del progetto.</li> </ul>
Product Owner	<ul style="list-style-type: none"> <li>Crea i requisiti complessivi iniziali del progetto e fa partire il progetto</li> <li>Nomina le persone appropriate per i ruoli di Scrum Master e Scrum Team</li> <li>Procura le risorse finanziarie iniziali e correnti per il progetto</li> <li>Stabilisce la Vision del Prodotto</li> <li>Valuta la fattibilità ed assicura la consegna del prodotto o servizio</li> <li>Assicura la trasparenza e la chiarezza degli elementi del Prioritized Product Backlog</li> <li>Decide il contenuto minimo di un rilascio commerciabile</li> <li>Fornisce i Criteri di Accettazione delle User Story da sviluppare in uno Sprint</li> <li>Esamina i deliverable</li> <li>Decide la durata dello Sprint</li> </ul>
Scrum Master	<ul style="list-style-type: none"> <li>Assicura che tutti i membri del team, incluso il Product Owner, seguano correttamente i processi di Scrum</li> <li>Assicura che lo sviluppo del prodotto o servizio proceda senza problemi e che i membri dello Scrum Team abbiano a disposizione tutti gli strumenti necessari alla esecuzione del lavoro</li> <li>Sovrintende al Release Planning Meeting e organizza altre riunioni</li> </ul>
Scrum Team	<ul style="list-style-type: none"> <li>Si assume la responsabilità collettiva e assicura che i deliverable del progetto siano creati sulla base dei requisiti.</li> <li>Assicura al Product Owner e allo Scrum Master che il lavoro assegnato viene eseguito conformemente al piano</li> </ul>

Tavola 3-4: Riepilogo delle Responsabilità Attinenti all'Organizzazione

### 3.9 Confronto fra Scrum e il Project Management Tradizionale

La struttura dell'Organizzazione e la definizione dei ruoli e delle responsabilità ad essi associate sono tra le aree in cui Scrum si differenzia in modo sostanziale dai metodi tradizionali di project management.

Nei metodi tradizionali di project management, la struttura organizzativa è gerarchica e l'autorità viene delegata per tutti gli aspetti del progetto da un livello superiore ad uno inferiore (ad esempio, lo sponsor del progetto delega l'autorità al project manager e questi la delega ai membri del team). I metodi di project management tradizionale mettono l'accento sulla responsabilità ultima di un singolo individuo riguardo alle varie responsabilità del progetto piuttosto che sulla responsabilità ultima del gruppo come tale. Qualsiasi deviazione dall'autorità delegata è vista come un sintomo di problemi e può essere portata all'attenzione del livello gerarchico superiore. Di solito è il project manager ad essere responsabile del successo del progetto ed è lui o lei che prende le decisioni sui vari aspetti del progetto, tra cui l'inizio, la pianificazione, la stima, l'esecuzione, il monitoraggio e controllo, e la chiusura.

Scrum invece pone l'accento sull'auto-organizzazione e l'auto-motivazione nelle quali il team si assume una responsabilità maggiore nel fare di un progetto un successo. Questo assicura anche la piena adesione (il c.d. buy-in) e la responsabilità condivisa da parte del team, che a loro volta si traducono nella motivazione del team, dalla quale scaturisce come conseguenza un'ottimizzazione del rendimento. Il Product Owner, lo Scrum Master e lo Scrum Team lavorano a stretto contatto con gli Stakeholder pertinenti per rifinire i requisiti man mano che portano avanti i processi *Sviluppare le Epic*, *Creare il Prioritized Product Backlog* e *Creare le User Story*. Questo fa sì che in Scrum non ci sia spazio per la pianificazione isolata. L'esperienza e la competenza nello sviluppo del prodotto vengono utilizzate per valutare gli input di cui si ha bisogno per pianificare, stimare ed eseguire il lavoro di progetto. La collaborazione fra tutti i membri dello Scrum Core Team assicura l'esecuzione del progetto in un ambiente innovativo e creativo, che contribuisce alla crescita e all'armonia del team.

## 3.10 Teorie Diffuse sulle Risorse Umane e loro Rilevanza per Scrum

### 3.10.1 Modello delle Dinamiche di Gruppo di Tuckman

L'approccio e il metodo Scrum possono sembrare all'inizio abbastanza diversi e difficili per un nuovo Scrum Team. Un nuovo Scrum Team, come qualsiasi altro nuovo team, durante il suo primo progetto Scrum passa generalmente attraverso un processo in quattro fasi. Questo processo è conosciuto come il Modello di Tuckman sulle dinamiche di gruppo (Tuckman, 1965). L'idea principale è che queste quattro fasi —Forming, Storming, Norming e Performing—sono obbligatorie perché un team possa svilupparsi riducendo i problemi e le sfide, cercando soluzioni, pianificando il lavoro e consegnando risultati.

Le quattro fasi del modello sono le seguenti:

1. **Forming**—Questa viene spesso vissuta come una fase divertente in quanto è tutto nuovo ed il team non ha ancora incontrato alcuna difficoltà riguardo al progetto.
2. **Storming**—Durante questa fase, il team prova a eseguire il lavoro; tuttavia possono verificarsi lotte di potere e spesso c'è caos e confusione fra i membri del team.
3. **Norming**—Questo è il momento in cui il team inizia a maturare, chiarisce le proprie differenze interne e cerca soluzioni per lavorare insieme. È considerato un periodo di assestamento.
4. **Performing**—Durante questa fase il team raggiunge il suo massimo di coesione ed opera al suo più alto livello in termini di prestazioni. I membri si sono trasformati in un team efficiente di professionisti alla pari che sono uniformemente produttivi.

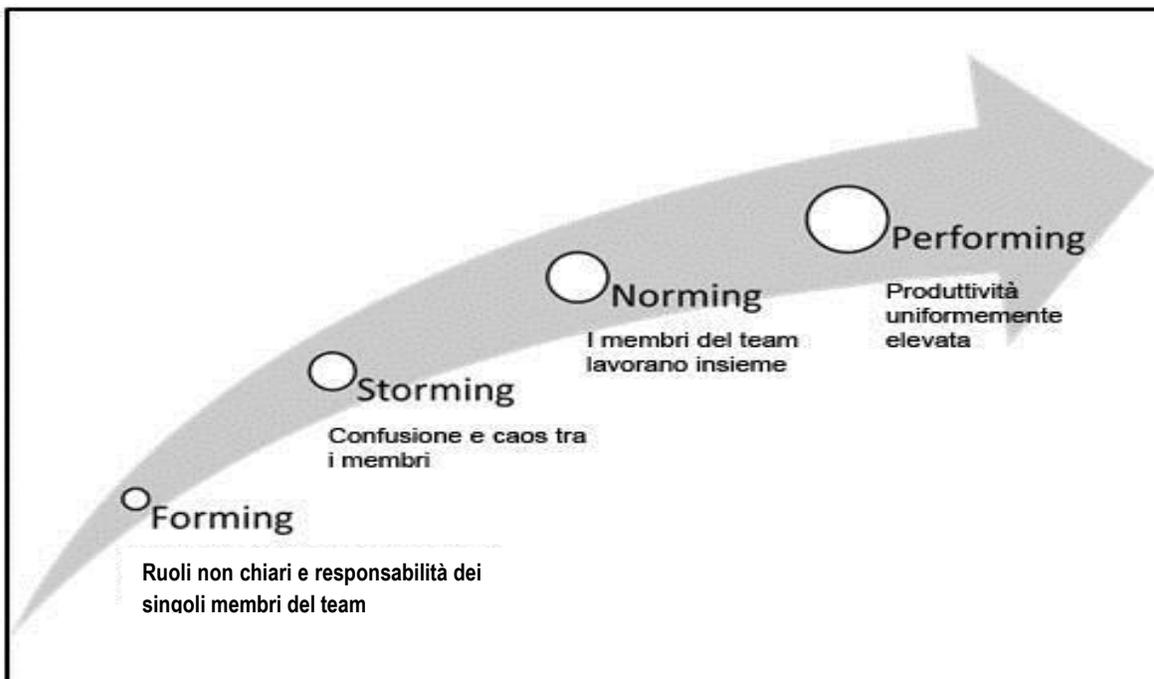


Figura 3-6: Fasi di Sviluppo del Gruppo secondo Tuckman

### 3.10.2 Gestione dei Conflitti

Le organizzazioni che applicano il framework Scrum incoraggiano un ambiente aperto e il dialogo fra i dipendenti. I conflitti fra i membri dello Scrum Team vengono di solito risolti in modo autonomo, con un coinvolgimento minimo o addirittura nullo del management o di altri soggetti esterni allo Scrum Team.

Il conflitto può essere salutare quando favorisce le discussioni del team e incoraggia i dibattiti, in quanto questo porta di solito benefici al progetto e ai rispettivi membri del team. Per questo è importante incoraggiare la soluzione dei conflitti, promuovendo un ambiente aperto in cui i membri del team si sentono invitati a esprimere le proprie opinioni e preoccupazioni sia riguardo ai rapporti reciproci che sul progetto, e alla fine si trovano d'accordo su ciò che deve essere consegnato e su come sarà eseguito il lavoro di ciascuno Sprint.

Le tecniche di gestione dei conflitti sono utilizzate dai membri del team per gestire qualsiasi conflitto insorga durante un progetto Scrum. Le fonti di conflitto nascono principalmente da questioni legate alle schedulazioni, alle priorità, alla catena gerarchica del reporting, a questioni tecniche, a procedure, a problemi caratteriali e per finire ai costi.

### 3.10.3 Tecniche di Gestione dei Conflitti

Di solito ci sono quattro diversi approcci alla gestione del conflitto in un'organizzazione che applica i processi Scrum:

1. Win-Win
2. Lose-Win
3. Lose-Lose
4. Win-Lose

#### 3.10.3.1 Win-Win

Normalmente per i membri del team è meglio affrontare i problemi in modo diretto con un atteggiamento cooperativo e un dialogo aperto per superare eventuali disaccordi al fine di raggiungere il pieno consenso. Questo approccio viene chiamato *Win-Win*. Le organizzazioni che implementano Scrum dovrebbero promuovere un ambiente in cui i dipendenti si sentono a proprio agio a discutere e affrontare apertamente problemi o questioni e a superarli per raggiungere risultati di tipo Win-Win (cioè in cui sono tutti vincitori).

#### 3.10.3.2 Lose-Win

Alcuni membri del team possono a volte avere la sensazione che il loro personale contributo non sia riconosciuto o valorizzato dagli altri, oppure di non venire trattati in modo uguale. Questo potrebbe portarli a rinunciare a dare un contributo effettivo al progetto e a mostrarsi d'accordo con qualsiasi cosa gli si dica di fare, anche se in realtà non concordano affatto. Questo approccio è chiamato *Lose-Win*. Questa situazione si può verificare se all'interno del team (compresi i manager) ci sono membri che usano uno stile autoritario o direttivo di impartire gli ordini e/o non trattano tutti i membri del team allo stesso modo. Questo tipo di approccio non rappresenta una tecnica di gestione dei conflitti auspicabile per i progetti Scrum, in quanto per completare ciascuno Sprint con successo è assolutamente necessario il contributo attivo di tutti i membri del team. Lo Scrum Master dovrebbe incoraggiare il coinvolgimento di tutti quei membri del team che mostrino di ritirarsi dalle situazioni di conflitto. Per esempio, è importante che tutti i membri del team prendano la parola e diano il loro contributo in ciascun Daily Standup Meeting, così da rendere noto qualsiasi problema o impedimento e gestirlo in maniera efficace.

### 3.10.3.3 Lose-Lose

Nelle situazioni di conflitto, i membri del team possono tentare di contrattare o ricercare soluzioni che portino ad un accordo solo parziale o a misure temporanee di soddisfazione per le parti contendenti. Negli Scrum Team questa situazione potrebbe verificarsi quando i membri del team tentano di negoziare soluzioni non ottimali ad un problema. Questo approccio implica di solito un "dare e avere" per soddisfare ogni membro del team —invece di provare a risolvere il vero problema. Ciò determina in genere un risultato complessivo di tipo *Lose-Lose* per tutti i soggetti coinvolti e di conseguenza per il progetto. Lo Scrum Team dovrebbe stare attento a far sì che i membri del team non cedano ad una mentalità di tipo Lose-Lose (cioè in cui tutti perdono). Il Daily Standup e altre riunioni di Scrum vengono condotti proprio per fare in modo che vengano risolti i problemi reali attraverso le discussioni reciproche.

### 3.10.3.4 Win-Lose

A volte, uno Scrum Master o un altro membro influente del team può ritenersi un leader o manager di fatto e tentare di imporre il proprio punto di vista a spese di quelli altrui. Questa tecnica di gestione dei conflitti è spesso caratterizzata dalla competitività e in genere sfocia in un risultato di tipo *Win-Lose*. Questo approccio è assolutamente sconsigliato quando si lavora su progetti Scrum, perché gli Scrum Team sono per loro natura auto-organizzati e investiti del potere, per cui nessuna reale autorità viene esercitata da un membro del team su un altro. Anche se lo Scrum Team può includere persone con diversi livelli di esperienza e di competenza, tutti i membri vengono trattati allo stesso modo e nessuno ha l'autorità per essere considerato il principale organo decisionale.

### 3.10.4 Stili di Leadership

Gli stili di leadership variano a seconda dell'organizzazione, della situazione e anche dei soggetti specifici e degli obiettivi del progetto Scrum. Quelli che seguono sono alcuni fra i comuni stili di leadership:

- **Servant Leadership o Leadership del Servitore**—I leader servitori usano l'ascolto, l'empatia, l'impegno e l'intuizione mentre condividono il potere e l'autorità con i membri del team. I leader servitori sono degli steward che raggiungono risultati mettendo al centro dell'attenzione i bisogni del team. Questo stile rappresenta l'incarnazione del ruolo dello Scrum Master.
- **Delegante**—I leader deleganti sono coinvolti nella maggior parte delle decisioni; tuttavia, essi delegano alcune responsabilità di pianificazione e decisionali ai membri del team, soprattutto se questi hanno competenze specifiche per gestire le attività loro assegnate. Questo stile di leadership è appropriato nelle situazioni in cui il leader si trova in sintonia con i dettagli specifici del progetto e quando il tempo a disposizione è limitato.
- **Autocratico**—I leader autocratici prendono decisioni per conto proprio, consentendo ai membri del team un intervento o una discussione minima, se non nulla, prima dell'emissione di una decisione. Questo stile di leadership dovrebbe essere utilizzato solo in rarissime occasioni.
- **Direttivo**—I leader direttivi impartiscono ai membri del team istruzioni riguardo a quali attività siano necessarie, quando dovranno essere eseguite e in che modo dovranno essere eseguite.
- **Laissez Faire**—Con questo stile di leadership ("lasciar fare"), il team viene lasciato per lo più senza sorveglianza, per cui il leader non interferisce nelle attività lavorative quotidiane del team. Questo stile porta spesso ad una situazione di anarchia.
- **Coaching/Di supporto**—I leader coach e di supporto impartiscono istruzioni e poi supportano e monitorano i membri del team attraverso l'ascolto, l'assistenza, l'incoraggiamento e un atteggiamento positivo nei momenti di incertezza.
- **Orientato alle Attività**—I leader orientati alle attività esigono il completamento delle attività e il rispetto delle scadenze.
- **Assertivo**—I leader assertivi affrontano le questioni e mostrano fiducia per stabilire la propria autorità con il rispetto.

#### 3.10.4.1 Leadership del Servitore

Lo stile di leadership preferito per i progetti Scrum è la Servant Leadership o Leadership del Servitore. Questo termine fu descritto per primo da Robert K. Greenleaf in un saggio intitolato *Il Servitore come Leader*. Qui sotto si riporta un estratto nel quale l'autore spiega il concetto:

Il leader servitore è per prima cosa servitore ... Si inizia con la sensazione naturale di voler servire, servire *in primo luogo*. Poi una scelta cosciente porta la persona ad aspirare ad essere leader. Quella persona è nettamente diversa da colui che è prima di tutto *leader*, forse a causa del bisogno

di soddisfare uno eccessivo desiderio di potere o di acquisire beni materiali .... Colui che è prima di tutto leader e colui che è prima di tutto servitore sono due tipologie estreme. Nel mezzo ci sono sfumature e combinazioni che fanno parte della infinita varietà della natura umana ...

La differenza si manifesta nell'attenzione posta da colui che è prima di tutto servitore a fare in modo di servire i bisogni più importanti delle altre persone. La prova migliore, ma anche difficile da applicare, è: Coloro che sono serviti crescono come persone? *Nel venire serviti*, diventano più sani, più saggi, più liberi, più autonomi, e con maggiori probabilità di diventare essi stessi dei servitori? *E inoltre*, Qual è l'effetto sui meno privilegiati della società? Ne beneficeranno o almeno non subiranno ulteriori privazioni? (Greenleaf 1970, 6).

Approfondendo gli scritti di Greenleaf, Larry Spears identifica dieci tratti che ogni vero servant-leader dovrebbe possedere:

1. **Ascolto**—I servant leader devono ascoltare con attenzione e apertura tutto ciò che viene detto e non detto. Essi hanno la capacità di entrare in contatto con la propria voce interiore per comprendere e riflettere sui propri sentimenti.
2. **Empatia**—I bravi servant leader accettano e riconoscono i singoli individui per le loro capacità ed abilità speciali ed uniche. Essi partono dal presupposto che coloro che lavorano nel team hanno buone intenzioni e li accettano come persone, anche quando si evidenziano problemi di comportamento o di prestazione.
3. **Capacità di cura**—La motivazione e il potenziale per curare se stessi e le proprie relazioni con gli altri rappresenta una caratteristica forte dei servant leader. I servant leader riescono a capire quando i propri colleghi stanno vivendo una sofferenza emotiva e colgono l'opportunità per aiutarli.
4. **Consapevolezza**—La consapevolezza ed in particolare l'auto-consapevolezza è una caratteristica dei servant leader. Ciò permette loro di comprendere meglio e di riuscire ad integrare questioni quali quelle relative all'etica, al potere e ai valori. This allows them to better understand and integrate issues such as those related to ethics, power, and values.
5. **Persuasione**—I servant leader utilizzano la persuasione al posto della autorità che deriva dalla loro posizione per ottenere il consenso del gruppo e prendere decisioni. Invece di costringere all'obbedienza, come è tipico di alcuni stili autoritari di management, i servant leader esercitano la persuasione.
6. **Concettualizzazione**—L'abilità di vedere ed analizzare i problemi (all'interno di un'organizzazione) da una prospettiva più ampia, concettuale e visionaria, invece di focalizzarsi semplicemente sugli obiettivi immediati di breve termine è una capacità distintiva dei bravi servanti leader.

7. **Intuizione**—La loro mente intuitiva permette ai servant leader di utilizzare ed applicare le lezioni del passato e le realtà del presente per prevedere quale sarà il risultato delle situazioni e decisioni attuali.
8. **Stewardship**—La stewardship richiede un impegno a servire gli altri. I servant leader preferiscono la persuasione al controllo, per fare in modo di ottenere la fiducia degli altri all'interno dell'organizzazione.
9. **Impegno a far crescere gli altri**—I servant leader sono profondamente impegnati a far crescere le persone che appartengono all'organizzazione. Essi si assumono la responsabilità della crescita personale, professionale e spirituale degli altri (ad esempio, fornendo l'accesso a risorse per lo sviluppo personale e professionale, incoraggiando chi lavora a partecipare al processo decisionale).
10. **Costruzione della comunità**—I servant leader sono interessati a costruire comunità all'interno dell'ambiente di lavoro, soprattutto in considerazione dello spostamento che si sta verificando nelle società dalle comunità più piccole verso le grandi istituzioni che plasmano e controllano le vite degli uomini.

Scrum ritiene che tutti i leader dei progetti Scrum (compresi lo Scrum Master ed il Product Owner) dovrebbero essere dei servant-leader con le caratteristiche sopra descritte.

### 3.10.5 Teoria della Gerarchia dei Bisogni di Maslow

Nel 1943 Maslow presentò una gerarchia dei bisogni secondo la quale persone differenti si trovano a livelli differenti nella loro scala dei bisogni. Di solito le persone partono dalla ricerca dei bisogni fisiologici, per poi spostarsi progressivamente verso l'alto nella gerarchia dei bisogni.

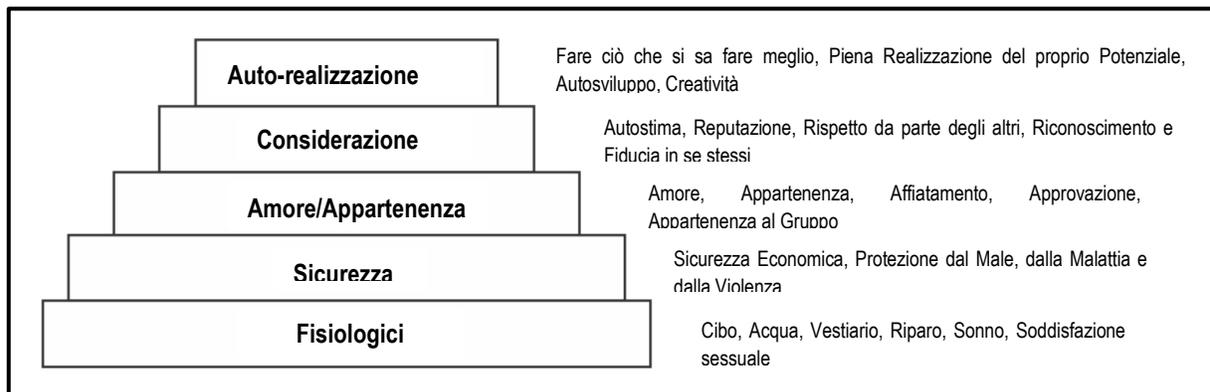


Figura 3-7: Teoria della Gerarchia dei Bisogni di Maslow

Per avere successo, un Team di Scrum ha bisogno che i membri del team, sia core che non-core, abbiano raggiunto i livelli di considerazione o di auto-realizzazione. Il concetto di team auto-organizzati, che rappresenta un principio chiave di Scrum, richiede che i membri del team siano auto-motivati e che partecipino e contribuiscano completamente alla realizzazione degli obiettivi del progetto.

Come leader, lo Scrum Master deve capire in che punto della piramide si trova ciascun membro del team. La comprensione di questo quadro lo aiuterà a stabilire il miglior approccio da seguire per motivare ogni singolo individuo.

Inoltre, tutti gli essere umani, nel corso della vita, passano su e giù da un livello all'altro della gerarchia dei bisogni a causa della motivazione e dell'impegno personale a spostarsi verso l'alto o a volte a causa di fattori fuori dal loro controllo che possono farli scivolare verso il basso. L'obiettivo dello Scrum Master è quello di lavorare con i singoli membri del team per costruire le competenze e la conoscenza di ognuno di loro ed aiutarli a risalire la gerarchia dei bisogni. Tale supporto si traduce in un team formato da individui motivati e che danno un forte contributo al progetto e all'organizzazione nel suo insieme.

### 3.10.6 Teoria X e Teoria Y

Nel 1960 Douglas McGregor propose due teorie manageriali:

- **Teoria X**—I leader della Teoria X ritengono che i dipendenti siano intrinsecamente privi di motivazione e che se possibile eviteranno il lavoro, legittimando quindi l'uso di uno stile di gestione autoritario.
- **Teoria Y**—I leader della Teoria Y, sul versante opposto, ritengono che i dipendenti siano auto-motivati e che cerchino di accettare responsabilità maggiori. La Teoria Y comporta uno stile manageriale più partecipativo.

I progetti Scrum non possono avere successo nelle organizzazioni che hanno leader della teoria X nei ruoli di Scrum Master o di Product Owner. Tutti i leader dei progetti Scrum dovrebbero aderire alla Teoria Y, per cui considerano le persone come risorse importanti e lavorano allo sviluppo delle competenze e all'empowerment dei membri del proprio team, esprimendo nel contempo apprezzamento per il lavoro completato dal team per conseguire gli obiettivi del progetto.

## 4. GIUSTIFICAZIONE COMMERCIALE

### 4.1 Introduzione

Lo scopo di questo capitolo è quello di comprendere il concetto e le finalità della Giustificazione Commerciale in relazione ai progetti Scrum. Per un'organizzazione è importante darsi una giustificazione commerciale adeguata e creare una valida Dichiarazione della Vision del Progetto prima di far partire qualsiasi progetto. Questo aiuta gli organi decisionali chiave a comprendere bene quale sia l'esigenza di business che giustifica un cambiamento o la realizzazione di un nuovo prodotto o servizio e la giustificazione per andare avanti con un progetto. Aiuta inoltre il Product Owner a creare un Prioritized Product Backlog in uno con le aspettative di business del Senior Management e degli Stakeholder.

La *Giustificazione Commerciale*, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Questo capitolo si divide nelle seguenti sezioni:

**4.2 Guida per i Ruoli**— Questa sezione identifica quali parti o paragrafi del capitolo sono importanti per il Product Owner, per lo Scrum Master e per lo Scrum Team.

**4.3 Consegna basata sul Valore**—Questa sezione illustra il concetto di valore di business e la sua importanza per qualsiasi progetto. Fornisce inoltre informazioni riguardo alle responsabilità dei vari soggetti coinvolti nella realizzazione del valore di business, incluso il Product Owner.

**4.4 Importanza della Giustificazione Commerciale**—Questa sezione spiega nel dettaglio l'importanza della giustificazione commerciale, i fattori che la determinano e come viene mantenuta e verificata nel corso del progetto.

**4.5 Tecniche relative alla Giustificazione Commerciale**—Questa sezione descrive nel dettaglio come valutare e verificare la giustificazione commerciale utilizzando vari strumenti.

**4.6 Continua Giustificazione del Valore**—Questa sezione approfondisce l'importanza della giustificazione continua del valore e spiega come si realizza.

**4.7 Confermare la Realizzazione dei Benefici**—Questa sezione descrive come si realizzano i benefici nel corso del progetto.

**4.8 Riepilogo delle Responsabilità**—Questa sezione definisce quali sono le responsabilità dei membri del team di progetto attinenti alla giustificazione commerciale, in base ai rispettivi ruoli.

**4.9 Confronto fra Scrum ed il Project Management Tradizionale**— Questa sezione mette in evidenza i benefici di business del metodo Scrum rispetto ai modelli tradizionali di project management.

## 4.2 Guida per i Ruoli

1. **Product Owner**—La giustificazione commerciale è compito precipuo del Product Owner; per cui la conoscenza dell'intero capitolo è più consigliata a questo ruolo.
2. **Scrum Master**—Lo Scrum Master dovrebbe avere familiarità con il contenuto dell'intero capitolo, con particolare attenzione alle sezioni 4.3, 4.4, 4.6, 4.7 e 4.8.
3. **Scrum Team**—Lo Scrum Team dovrebbe concentrare la propria attenzione soprattutto sulle sezioni 4.3, 4.7 e 4.8.

## 4.3 Consegna basata sul Valore

Un progetto è un'impresa collaborativa volta a creare nuovi Consegna basata sul Valore prodotti o servizi o a consegnare risultati così come definiti nella Dichiarazione della Vision del Progetto. I progetti sono di solito condizionati da vincoli di tempo, costo, ambito, qualità, persone, e infine capacità organizzative. Di solito ci si attende che i risultati generati dai progetti riescano a creare una qualche forma di valore in termini di business o di servizio.

Poiché il valore rappresenta la ragioni principale per la quale qualsiasi organizzazione va avanti con un progetto, la Consegna basata sul Valore deve essere l'obiettivo primario. La consegna di valore è radicata all'interno del framework di Scrum. Scrum facilita la consegna di valore fin dalle primissime fasi del progetto e continua a farlo per tutto il ciclo di vita del progetto.

Una delle caratteristiche chiave di qualsiasi progetto è l'incertezza degli esiti e dei risultati. È impossibile garantire il successo del progetto, a prescindere dalla sua dimensione o complessità. Avendo presente questa incertezza circa le possibilità di raggiungere il successo, è quindi importante iniziare a consegnare risultati concreti quanto prima possibile nel corso del progetto. Questa consegna anticipata di risultati, e perciò di valore, offre l'opportunità di reinvestire e dimostra la validità del progetto agli stakeholder interessati.

Per poter realizzare una Consegna basata sul Valore, è importante:

1. Capire cosa aggiunge valore per i clienti e gli utenti e mettere i requisiti di alto valore in cima alla lista delle priorità del Prioritized Product Backlog.
2. Diminuire l'incertezza e far fronte costantemente ai rischi che potrebbero diminuire il valore qualora si materializzassero. È inoltre importante lavorare a stretto contatto con gli stakeholder del progetto mostrando loro gli incrementi di prodotto alla fine di ogni Sprint, rendendo così possibile un'efficace gestione del cambiamento.
3. *Creare i Deliverable* sulla base delle priorità stabilite, producendo incrementi di prodotto potenzialmente consegnabili durante ogni Sprint, in modo tale che i clienti inizino a realizzare valore fin dalle prime battute del progetto.

Il concetto di Consegna basata sul Valore proprio di Scrum rende questo framework molto allettante per gli stakeholder e il senior management dell'area business. Questo concetto è molto diverso rispetto ai modelli tradizionali di project management nei quali:

1. I requisiti non sono prioritizzati in base al valore di business.
2. Il cambiamento dei requisiti dopo l'inizio del progetto è difficile e si può fare solo attraverso un processo di gestione del cambiamento che richiede tempo.
3. Il valore si realizza solo alla fine del progetto quando viene consegnato il prodotto o servizio finale.

La Figura 4-1 mostra le differenze fra la Consegna basata sul Valore propria di Scrum e i progetti Tradizionali.

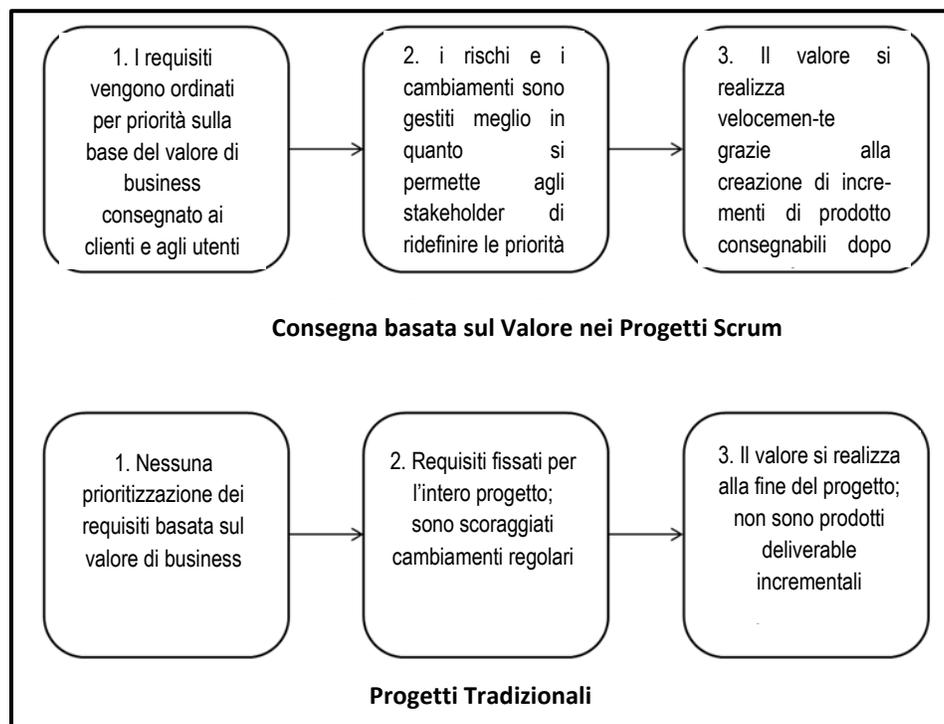


Figura 4-1: Confronto fra la Consegna di Valore in Scrum e nei Progetti Tradizionali

### 4.3.1 Responsabilità del Product Owner in tema di Giustificazione Commerciale

La responsabilità di prioritizzare e di consegnare valore di business con i progetti di un'organizzazione fa capo soprattutto al Product Owner. Per i programmi e i portfolio questa responsabilità spetta rispettivamente al Program Product Owner e al Portfolio Product Owner. Il loro ruolo è quello di agire come efficaci rappresentanti del cliente e/o dello sponsor. Le linee guida per la valutazione e la misurazione del valore di business possono essere stabilite generalmente dallo Scrum Guidance Body.

La Figura 4-2 illustra in ordine gerarchico le responsabilità relative alla giustificazione commerciale.

Portfolio Product Owner	<ul style="list-style-type: none"> <li>• Consegna valore con i portfolio</li> <li>• Crea la giustificazione commerciale dei portfolio</li> <li>• Fornisce un orientamento per il valore dei programmi</li> <li>• Approva la giustificazione commerciale dei programmi</li> </ul>
Program Product Owner	<ul style="list-style-type: none"> <li>• Consegna valore con i programmi</li> <li>• Crea la giustificazione commerciale dei programmi</li> <li>• Fornisce un orientamento per il valore dei progetti</li> <li>• Approva la giustificazione commerciale dei progetti</li> </ul>
Product Owner	<ul style="list-style-type: none"> <li>• Consegna valore con i progetti</li> <li>• Crea la giustificazione commerciale dei progetti</li> <li>• Conferma la realizzazione dei benefici agli stakeholder</li> </ul>

Figura 4-2: Gerarchia delle Responsabilità in tema di Giustificazione Commerciale

### 4.3.2 Responsabilità degli Altri Ruoli di Scrum in tema di Giustificazione Commerciale

È importante notare che, anche se il Product Owner è il principale responsabile della giustificazione commerciale, anche altre persone che lavorano nei progetti Scrum danno un contributo significativo nei termini che seguono:

1. Lo **sponsor** finanzia il progetto e lo monitora costantemente per confermare la realizzazione dei benefici.
2. I **Clienti** e gli **Utenti** sono coinvolti nella definizione della lista prioritizzata dei requisiti e delle User Story del Prioritized Product Backlog, nella revisione dei Deliverable dopo ogni Sprint o rilascio, e nella conferma dell'effettiva realizzazione dei benefici.
3. Lo **Scrum Guidance Body** può fornire linee guida e raccomandazioni riguardo alle tecniche sulla giustificazione commerciale e sulla conferma della realizzazione dei benefici e così via. Gli Scrum Core Team e gli Stakeholder possono quindi fare riferimento a tali linee guida e raccomandazioni.
4. Lo **Scrum Master** facilita la creazione dei deliverable del progetto; gestisce i rischi, i cambiamenti e gli impedimenti durante i processi *Condurre il Daily Standup*, *Retrospettiva dello Sprint*, e altri processi di Scrum. Lo Scrum Master si coordina con lo Scrum Team per creare i deliverable e con il Product Owner e altri stakeholder per assicurare la realizzazione dei benefici del progetto.
5. Lo **Scrum Team** lavora alla creazione dei deliverable del progetto e contribuisce alla realizzazione del valore di business per tutti gli stakeholder e per il progetto. Lo Scrum Team è inoltre coinvolto nei processi *Sviluppare le Epic*, *Creare il Prioritized Product Backlog*, *Creare le User Story*, *Approvare*, *Stimare e Prendere in Carico le Story*, nonché nei processi a questi associati nei quali si procede alla definizione e prioritizzazione dei requisiti di business. Lo Scrum Team contribuisce infine alla identificazione dei rischi e sottopone le Richieste di Modifica per i miglioramenti in occasione degli Sprint Retrospect Meeting e di altre riunioni.

## 4.4 Importanza della Giustificazione Commerciale

La Giustificazione Commerciale spiega le ragioni per intraprendere un progetto. Questo aspetto di Scrum risponde alla domanda “Perché c'è bisogno del progetto?” La Giustificazione Commerciale guida tutto il processo decisionale collegato ad un progetto. Per questo, è importante valutare la fattibilità e realizzabilità di un progetto non solo prima di impegnarsi in spese o investimenti di rilievo nelle fasi iniziali del progetto, ma anche per verificare che la giustificazione commerciale permanga durante tutto il ciclo di vita del progetto. Un progetto dovrebbe essere interrotto se si rivela infattibile; la decisione dovrebbe essere portata all'attenzione degli stakeholder pertinenti e del senior management. La giustificazione commerciale di un progetto deve essere valutata all'inizio del progetto, a intervalli predefiniti nel corso del progetto e ogni qualvolta insorgano questioni o rischi importanti che minacciano la fattibilità del progetto.

### 4.4.1 Fattori Utilizzati per Determinare la Giustificazione Commerciale

Ci sono numerosi fattori che un Product Owner deve considerare per determinare la giustificazione commerciale di un progetto. Quelli che seguono sono alcuni dei più importanti:

### **1. Motivi del Progetto**

I motivi del progetto includono tutti i fattori che rendono necessario il progetto, siano essi positivi o negativi, frutto di una scelta o meno (ad esempio, la scarsa capacità di soddisfare la domanda esistente e prevista, il calo della soddisfazione del cliente, un profitto basso, un requisito legale, ecc.).

### **2. Bisogni di Business**

I bisogni di business sono quei risultati di business che ci si aspetta siano realizzati dal progetto, così come documentato nella Dichiarazione del Vision del Progetto.

### **3. Benefici del Progetto**

I benefici del progetto includono tutti i miglioramenti misurabili di un prodotto, servizio o risultato che possono essere ottenuti attraverso il positivo completamento di un progetto.

### **4. Costo Opportunità**

Il costo opportunità attiene alla migliore opzione di business o miglior progetto che è stato scartato a favore di quello attuale.

### **5. Rischi Principali**

I rischi riguardano qualsiasi incertezza o evento non pianificato che può incidere sulla fattibilità e sul potenziale successo del progetto.

### **6. Tempistiche del Progetto**

Le tempistiche riflettono la lunghezza o durata di un progetto e il tempo in cui saranno realizzati i suoi benefici.

### **7. Costi del Progetto**

I costi del progetto sono rappresentati dall'investimento e dagli altri costi di sviluppo del progetto.

## **4.4.2 Giustificazione Commerciale e Ciclo di Vita del Progetto**

La giustificazione commerciale è in primo luogo valutata prima di iniziare un progetto ed è poi continuamente sottoposta a verifica per tutto il ciclo di vita del progetto. I passi sotto elencati indicano come si determina la giustificazione commerciale:

### 1. Valutare e Presentare un Business Case

La giustificazione commerciale di un progetto di norma è analizzata e confermata dal Product Owner. Questa viene documentata e presentata sotto forma di un Business Case del progetto prima della fase di Inizio e implica la considerazione dei vari fattori specificati nel paragrafo 4.4.1. Dopo aver documentato la giustificazione commerciale, il Product Owner deve creare una Dichiarazione della Vision del Progetto ed ottenerne l'approvazione da parte degli organi decisionali chiave dell'organizzazione. Generalmente questi sono costituiti dagli executive e/o da un comitato di gestione del progetto o del programma.

### 2. Giustificazione Continua del Valore

Una volta che gli organi decisionali approvano la Dichiarazione della Vision del Progetto, questa diviene la baseline e costituisce la giustificazione commerciale. La giustificazione commerciale viene convalidata nel corso dell'esecuzione del progetto, di solito ad intervalli predefiniti o milestone, ad esempio durante le riunioni di portfolio, di programma e durante i Prioritized Product Backlog Review Meeting, nonché quando sono identificate questioni e rischi importanti che minacciano la fattibilità del progetto. Questo si può verificare in occasione di molti dei processi di Scrum, fra cui *Condurre il Daily Standup* e *Mettere a Punto il Prioritized Product Backlog*. Durante tutto il progetto, il Product Owner deve mantenere aggiornata la giustificazione commerciale indicata nella Dichiarazione della Vision del Progetto con tutte le informazioni appropriate relative al progetto, in modo tale da consentire agli organi decisionali chiave di continuare a prendere decisioni informate.

### 3. Confermare la Realizzazione dei Benefici

Il Product Owner conferma il raggiungimento dei benefici per l'organizzazione durante tutto il progetto, oltre che dopo il completamento delle User Story contenute nel Prioritized Product Backlog. I benefici generati dai progetti Scrum vengono realizzati durante i processi *Illustrare e Convalidare lo Sprint*, *Retrospettiva dello Sprint*, *Consegnare i Deliverable* e *Retrospettiva del Progetto*.

La Figura 4-3 riassume i passi per la determinazione della giustificazione commerciale.

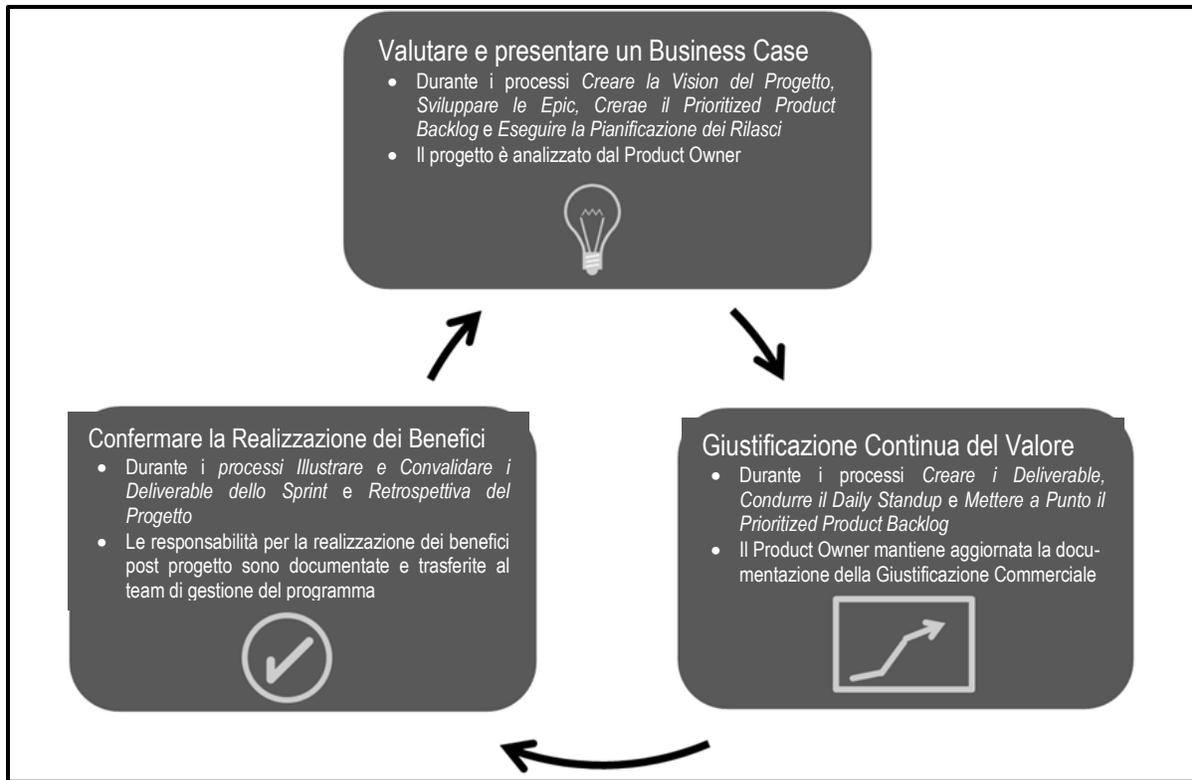


Figura 4-3: Giustificazione Commerciale e Ciclo di Vita del Progetto

## 4.5 Tecniche relative alla Giustificazione Commerciale

I paragrafi che seguono trattano alcuni degli strumenti utilizzati per valutare e misurare la giustificazione commerciale, oltre ad alcuni altri aspetti associati alla giustificazione del progetto e alla selezione del progetto. Non è necessario, o tanto meno raccomandato, usare tutte le tecniche disponibili per ogni progetto. Alcune tecniche non sono appropriate per determinati progetti, inoltre le tecniche possono essere utilizzate per valutare i progetti singolarmente o per confrontare il valore atteso di più progetti.

Lo Scrum Guidance Body (SGB), che può consistere in un gruppo di esperti o in un insieme di documenti sugli standard e le procedure dell'organizzazione, definisce le linee guida e le metriche che saranno utilizzate per valutare il valore di business. Ogni Product Owner, tuttavia, resta responsabile dell'esecuzione delle attività di verifica e monitoraggio del valore di business dei propri rispettivi progetti, programmi o portfolio.

4

### 4.5.1 Stima del Valore del Progetto

Il valore che deve essere generato dai progetti commerciali può essere stimato utilizzando vari metodi, quali il Ritorno sull'Investimento (ROI - Return on Investment), Il Valore Netto Attuale (NPV - Net Present Value), ed il Tasso Interno di Rendimento (IRR - Internal Rate of Return).

#### 1. Ritorno sull'Investimento (ROI)

Il Ritorno sull'Investimento (ROI, quando utilizzato per la giustificazione del progetto, valuta l'utile netto atteso che deve essere ottenuto da un progetto. Si calcola sottraendo i costi o gli investimenti previsti di un progetto dal ricavo atteso e dividendo poi il risultato (profitto netto) per i costi previsti, al fine di ottenere un tasso di rendimento. Nei calcoli del ROI possono poi essere considerati altri fattori come l'inflazione e i tassi di interesse applicati ai prestiti di denaro.

Formula del ROI:

$$\text{ROI} = (\text{Ricavo del Progetto} - \text{Costo del Progetto}) / \text{Costo del Progetto}$$

*Esempio:* Il ROI di un progetto che costerà 125.000\$ per il suo sviluppo, con entrate finanziarie attese stimate in 300.000\$ è calcolato come segue:

$$\text{ROI} = (\$300.000 - \$125.000) / \$125.000 = 1,4$$

Pertanto, il ROI è pari a 1,4 volte l'investimento (o al 140%).

Gli incrementi frequenti del prodotto o servizio rappresentano una base fondamentale di Scrum, in quanto consentono una verifica precoce del ROI. Ciò aiuta nella valutazione della giustificazione del valore continuo.

## 2. Valore Netto Attuale (NPV)

Quello del Valore Netto Attuale (NPV) è un metodo usato per determinare il valore netto presente di un beneficio finanziario futuro, dato come assunto un determinato tasso di inflazione o di interesse. In altre parole, l'NPV è dato dal ricavo totale atteso di un progetto meno il totale del costo previsto del progetto, tenuto conto del valore temporale del denaro.

*Esempio:* Quale fra i due seguenti progetti è meglio scegliere se il criterio di selezione utilizzato è l'NPV?

- Progetto A ha un NPV di 1.500\$ e sarà completato in 5 anni.
- Progetto B ha un NPV di 1.000\$ e sarà completato in 1 anno.

*Soluzione:* Il Progetto A, in quanto il suo NPV è più alto; il fatto che il Progetto B ha una durata minore del Progetto A in questo caso non conta, perché il tempo è già considerato nel calcolo dell'NPV (vale a dire per il fatto che è il valore attuale e non quello futuro che viene preso in considerazione nel calcolo).

## 3. Tasso Interno di Rendimento (IRR)

Il Tasso Interno di Rendimento (IRR) è un tasso di sconto su un investimento che rende il valore attuale dei flussi monetari in entrata pari al valore attuale dei flussi monetari in uscita per valutare il tasso di rendimento di un progetto. Quando si confrontano fra loro più progetti, di norma quello che ha un IRR più alto è migliore.

Anche se l'IRR non viene utilizzato per giustificare i progetti tanto spesso quanto lo sono altre tecniche, come l'NPV, rappresenta comunque un concetto importante da conoscere.

*Esempio:* Sulla base dell'IRR, quale progetto è più auspicabile?

- Progetto A, che ha un IRR del 15% e sarà completato in 5 anni.
- Progetto B, che ha un IRR del 10% e sarà completato in 1 anno.

*Soluzione:* Il Progetto A, dal momento che il suo IRR è più alto; il fatto che il Progetto B ha una durata più breve del Progetto A in questo caso non conta perché il tempo è già considerato nei calcoli dell'IRR (vale a dire, come per l'NPV, che è il valore attuale e non quello futuro che viene usato per determinare l'IRR).



## 4.5.2 Pianificazione Incentrata sul Valore

Dopo aver giustificato e confermato il valore di un progetto, nel pianificarlo il Product Owner dovrà prendere in considerazione le politiche, le procedure, i modelli e gli standard generali dell'organizzazione prescritti dallo Scrum Guidance Body (o da analogo comitato o ufficio di progetto dell'organizzazione), massimizzando nel contempo la Consegna basata sul Valore. L'onere di decidere *come* si crea il valore ricade sugli stakeholder (sponsor, clienti e/o utenti), mentre lo Scrum Team si concentra su cosa deve essere sviluppato. Fra gli strumenti comuni raccomandati da uno Scrum Guidance Body potrebbero rientrare i seguenti:

### 1. Value Stream Mapping

Il Value Stream Mapping utilizza i diagrammi di flusso per illustrare il flusso delle informazioni necessarie a completare un processo. Questa tecnica può essere usata per snellire un processo aiutando ad individuare gli elementi che non aggiungono valore.

### 2. Prioritizzazione basata sul Valore per il Cliente

La Prioritizzazione basata sul Valore per il Cliente dà importanza primaria al cliente e si sforza di implementare per prime le User Story con il valore più elevato. Queste User Story ad alto valore vengono identificate e spostate in cima al Prioritized Product Backlog.

Un team può utilizzare una molteplicità di schemi di prioritizzazione per determinare le funzionalità ad alto valore.

#### a. Schemi Semplici

Gli schemi semplici consistono nel classificare gli elementi come Priorità "1", "2", "3" o "Alta", "Media" e "Bassa" e così via. Anche se si tratta di un approccio semplice e trasparente, può causare dei problemi perché spesso c'è la tendenza a classificare tutto come Priorità "1" o "Alta". Anche gli schemi di prioritizzazione "Alta", "Media" e "Bassa" possono incontrare difficoltà analoghe.

#### b. Prioritizzazione MoSCoW

Lo schema di prioritizzazione MoSCoW prende il nome dalle prime lettere delle frasi "Must have" (cioè "Deve avere"), "Should have" (cioè "Dovrebbe avere"), "Could have" (cioè "Potrebbe avere"), e "Won't have" (cioè "Non avrà"). Questo metodo di prioritizzazione è generalmente più efficace degli schemi semplici. Le categorie sono in ordine decrescente di priorità, dove le funzionalità del tipo "Must have" sono quelle senza le quali il prodotto non avrà valore e quelle del tipo "Won't have" sono quelle che, anche se sarebbe bello avere, non è necessario includere.

#### c. Monopoly Money

Questa tecnica consiste nel consegnare ai clienti una quantità di “soldi del Monopoli” o “soldi falsi” pari al budget del progetto, chiedendo loro di distribuirli fra le User Story prese in esame. In questo modo, il cliente fa una scala di priorità in base a ciò che è disposto a pagare per ciascuna User Story.

#### d. Metodo dei 100 Punti

Il Metodo dei 100 Punti è stato sviluppato da Dean Leffingwell e Don Widrig nel 2003. Consiste nel mettere a disposizione del cliente 100 punti, che lo stesso può utilizzare per votare per le funzionalità che ritiene più importanti.

4

#### e. Analisi di Kano

L'Analisi di Kano è stata sviluppata da Noriaki Kano nel 1984 e consiste nel classificare le funzionalità o i requisiti in quattro categorie basate sulle preferenze del cliente:

1. *Exciters/Delighters* (cioè che provocano eccitazione e meraviglia): Funzionalità che sono nuove o hanno un valore elevato per il cliente
2. *Satisfiers* (cioè che soddisfano): Funzionalità che offrono valore al cliente
3. *Dissatisfiers* (cioè che non soddisfano): Funzionalità che, se non presenti, determinano con ogni probabilità il non gradimento del prodotto da parte del cliente, ma che, se presenti, non incidono sul suo livello di soddisfazione.
4. *Indifferent* (cioè che sono indifferenti): Funzionalità che non influenzeranno in alcun modo il cliente e che dovrebbero essere eliminate.

La Figura 4-4 mostra una rappresentazione grafica della Analisi di Kano.

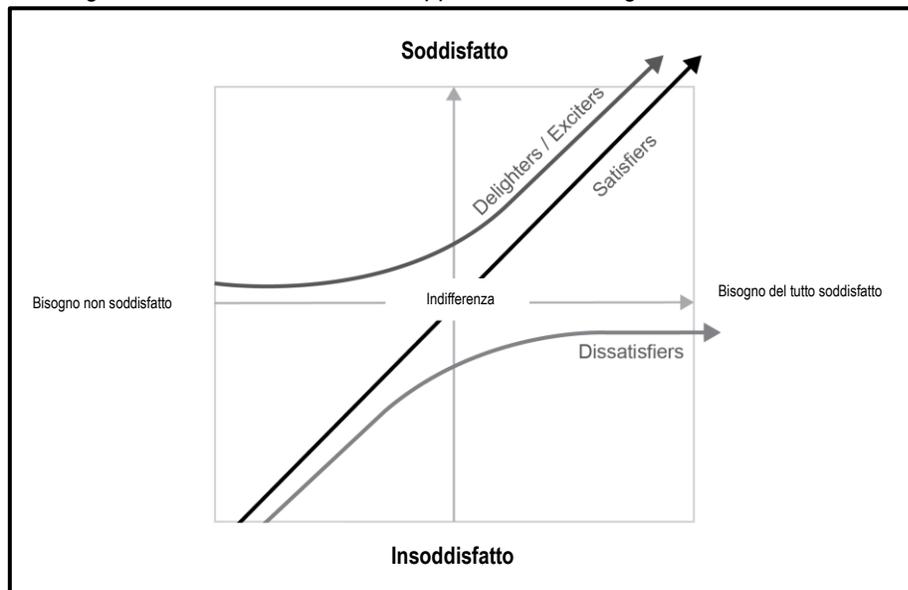


Figura 4-4: Analisi di Kano

È interessante notare che normalmente con il passare del tempo le funzionalità si spostano verso la parte bassa della graduatoria; i clienti inizieranno ad aspettarsi certe funzionalità (ad esempio le fotocamere incorporate nei telefoni) e queste funzionalità si sposteranno dall'iniziale categoria di 'exciters/delighters' a quella di 'satisfiers', e infine a quella di 'dissatisfiers'.

### 4.5.3 Relative Prioritization Ranking

Un semplice elenco di User Story in ordine di priorità rappresenta un metodo efficace per stabilire le User Story desiderate per ciascuna iterazione o rilascio del prodotto o servizio. Lo scopo è quello di creare un'unica semplice lista con l'obiettivo di prioritizzare le funzionalità, piuttosto che farsi distrarre da tanti schemi di prioritizzazione.

Questa lista semplice offre inoltre la base per incorporare i cambiamenti e i rischi identificati ove necessario. Ciascun cambiamento o rischio identificato può essere inserito nella lista sulla base della sua priorità relativa in rapporto alle altre User Story della lista. Di solito i nuovi cambiamenti saranno inclusi a spese delle funzionalità alle quali è stata assegnata una priorità più bassa.

Durante questo processo, è estremamente importante definire le Minimum Marketable Features (MMF), cioè le caratteristiche minime commerciabili, in modo da far avvenire il primo rilascio o la prima iterazione quanto prima possibile e aumentare così il ROI. Normalmente queste User Story si vedranno assegnare una priorità più alta.

### 4.5.4 Story Mapping

Questa è una tecnica che serve a fornire un profilo visivo del prodotto e dei suoi componenti chiave. Lo Story Mapping, formulato per la prima volta da Jeff Patton nel 2005, è comunemente usato per illustrare il percorso (roadmap) dei prodotti.

Le story map raffigurano la sequenza delle iterazioni di sviluppo del prodotto e delineano quali funzionalità saranno incluse nel primo, nel secondo, nel terzo rilascio ed in quelli seguenti.

## 4.6 Giustificazione Continua del Valore

Il valore di business dovrebbe essere valutato regolarmente per stabilire se la giustificazione o la fattibilità dell'esecuzione del progetto continui ad esistere. La valutazione frequente dell'investimento nel progetto rispetto al valore di business creato qualifica la fattibilità di un progetto. I requisiti attesi dal progetto possono cambiare frequentemente e ciò può avere un impatto sia sull'investimento del progetto che sulla creazione del valore. Un aspetto fondamentale di Scrum è la sua capacità di adeguarsi rapidamente al caos creato da

un modello di business in rapida evoluzione. Nei progetti in cui i requisiti dell'utente sono ambigui e c'è una significativa possibilità di cambiamenti frequenti, Scrum assicura notevoli vantaggi rispetto agli altri modelli di sviluppo.

Il monitoraggio del tasso di valore consegnato è un requisito importante dei progetti Scrum. Seguire periodicamente l'andamento della creazione del valore e relazionare su di essa aiuta a valutare lo stato del progetto e fornisce importanti informazioni al cliente e agli altri stakeholder.

#### **4.6.1     Analisi dell'Earned Value**

Anche se sono utilizzati comunemente, strumenti come i diagrammi a barre e i diagrammi di Gantt presentano dei limiti riguardo alla capacità di monitorare e riportare lo stato di avanzamento quando si tratta delle prestazioni del progetto. A questo fine si utilizza l'Analisi dell'Earned Value (Earned Value Analysis - EVA).

L'EVA analizza la prestazione effettiva del progetto rispetto a quella pianificata in un dato momento temporale. Per far sì che le tecniche di monitoraggio siano efficaci, l'iniziale piano baseline del progetto deve essere accurato. L'EVA utilizza spesso grafici e altri elementi visivi (ad esempio curva a S) per rappresentare le informazioni sullo stato del progetto.

L'Analisi dell'Earned Value misura gli scostamenti attuali delle prestazioni di tempo e costo del progetto e fa una previsione del costo finale sulla base della prestazione attuale rilevata. L'EVA viene eseguita di solito alla fine di ciascuno Sprint, dopo che sono state completate le User Story dello Sprint Backlog.

La Tavola 4-1 riepiloga le formule utilizzate nell'Analisi dell'Earned Value.

Definizione dei Termini	Acronimo	Formula
Valore Pianificato (Planned Value)	PV	
Valore Ottenuto (Earned Value)	EV	
Costo Effettivo (Actual Cost)	AC	
Budget al Completamento	BAC	
Scostamento dei Tempi (Schedule Variance)	SV	$EV - PV$
Scostamento dei Costi (Cost Variance)	CV	$EV - AC$
Indice di Efficienza della Schedulazione (Schedule Performance Index)	SPI	$EV / PV$
Indice di Efficienza dei Costi (Cost Performance Index)	CPI	$EV / AC$
Percentuale di Completamento	% Completamento	$(EV / BAC) \times 100$
Stima al Completamento (Estimate at Completion) <ul style="list-style-type: none"> <li>1. Presupposti di Stima non validi</li> <li>2. Le Variazioni Attuali sono atipiche</li> <li>3. Le Variazioni Attuali sono tipiche</li> </ul>	EAC	<ul style="list-style-type: none"> <li>1. <math>AC + ETC</math></li> <li>2. <math>AC + BAC - EV</math></li> <li>3. <math>BAC / CPI</math></li> </ul>
Stima per il Completamento (Estimate to Complete)	ETC	$EAC - AC$
Variazione al Completamento (Variance at Completion)	VAC	$BAC - EAC$

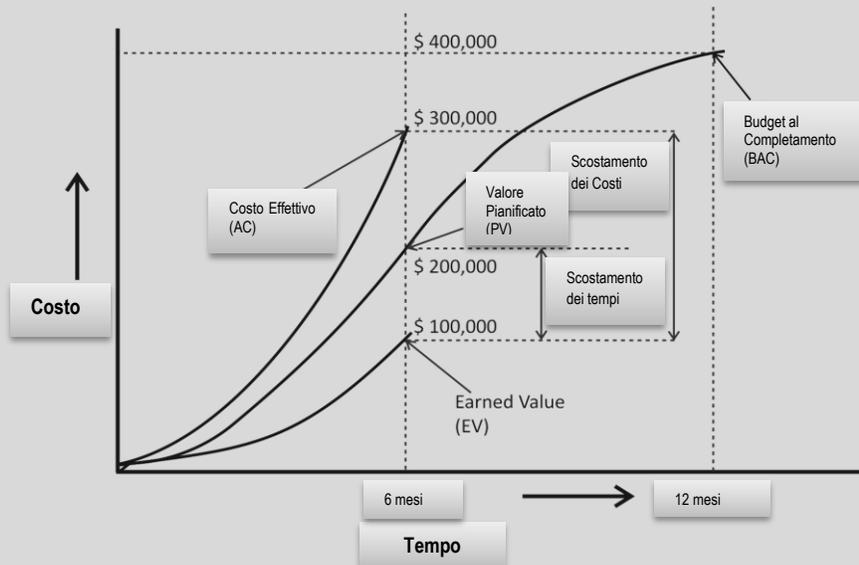
Tavola 4-1: Formule dell'Earned Value

*Esempio:* Deve essere sviluppato un sito web di 4.000 pagine —assumiamo che ciascuna pagina web richieda lo stesso tempo per essere completata, e che ciascuna pagina web rappresenti un'unica User Story di uguale priorità all'interno del Prioritized Product Backlog. Il costo stimato di completamento del progetto è di \$ 400.000 ed il limite temporale è di 12 mesi. Dopo sei mesi, sono stati spesi \$ 300.000 e sono state completate 1.000 pagine web.

Cosa ci è stato dato?

- Budget al Completamento (BAC) = \$ 400.000 (Costo Baseline del progetto)
- Valore Pianificato (PV) = \$ 200.000 (poiché avevamo pianificato di completare 2.000 pagine web)
- Earned Value (EV) = \$100.000 (valore delle 1.000 pagine web completate)
- Costo Effettivo (AC) = \$300.000 (ciò che è stato speso finora)

Curva a S dei dati:



Formule:

- Scostamento dei Tempi (SV) =  $EV - PV = \$100.000 - \$200.000 = - \$100.000$
- Scostamento dei Costi (CV) =  $EV - AC = \$100.000 - \$300.000 = - \$200.000$ 
  - Le variazioni negative nel nostro progetto indicano che siamo in ritardo con i tempi e al di sopra del budget.
- Indice di Efficienza della Schedulazione (SPI) =  $EV / PV = \$100.000 / \$200.000 = 0,5$ 
  - Un SPI < 1 indica che il lavoro completato finora è pari solo al 50% di quello che avevamo pianificato di completare in sei mesi.
- Indice di Efficienza dei Costi (CPI) =  $EV / AC = \$100.000 / \$300.000 = 0,33$ 
  - Un CPI < 1 indica che stiamo ottenendo solo il 33% del lavoro che avrebbe dovuto essere stato fatto per la quantità di denaro speso.
- Percentuale Completata =  $EV / BAC \times 100 = \$100.000 / \$400.000 \times 100 = 25\%$ 
  - Quindi, a questo dato momento risulta completato il 25% del lavoro del progetto.

## 4.6.2 Diagramma di Flusso Cumulativo (CFD)

Un Diagramma di Flusso Cumulativo (CFD - Cumulative Flow Diagram) è un utile strumento per il reporting ed il monitoraggio delle prestazioni del progetto. Tale diagramma fornisce una semplice rappresentazione visiva dello stato di avanzamento del progetto in uno specifico momento temporale. Viene utilizzato di solito per mostrare uno stato di più alto livello del progetto complessivo e non gli aggiornamenti giornalieri dei singoli Sprint.

La Figura 4-5 è un esempio di CFD per un progetto di grandi dimensioni. Mostra quante User Story devono essere ancora create, quante sono in corso di creazione e quante sono state già create. Nel momento in cui cambiano i requisiti del Cliente, c'è un cambiamento nelle User Story Cumulative che devono essere create. I punti di Cambiamento 1 e 2 rappresentano quelli in cui il Product Owner ha rimosso alcune User Story dal Risk Adjusted Prioritized Product Backlog mentre i punti di Cambiamento 3 e 4 sono quelli in cui il Product Owner ha aggiunto nuove User Story al Risk Adjusted Prioritized Product Backlog.

Questo tipo di diagramma può essere un ottimo strumento per identificare i blocchi e i colli di bottiglia all'interno dei processi. Per esempio, se il diagramma mostra che una fascia sta diventando più stretta mentre quella precedente sta diventando più larga con il passare del tempo, questo potrebbe essere il sintomo della presenza di un collo di bottiglia e potrebbero quindi essere necessari dei cambiamenti per aumentare la efficienza e/o migliorare le prestazioni del progetto.

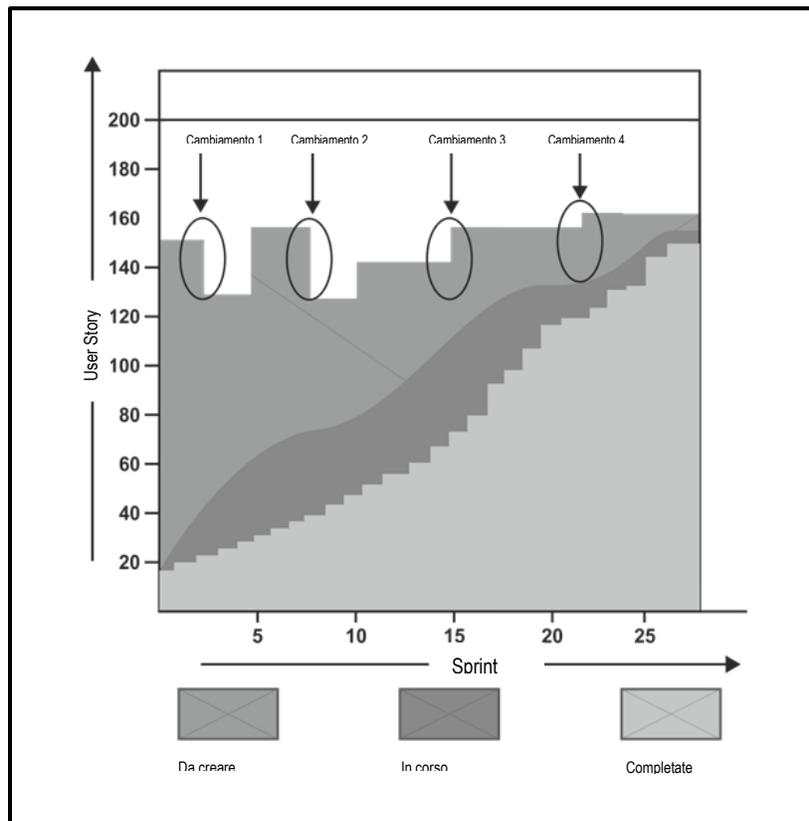


Figura 4-5: Esempio di Diagramma di Flusso Cumulativo (CFD)

## 4.7 Confermare la Realizzazione dei Benefici

Nel corso di un progetto è importante verificare se si stanno realmente realizzando i benefici. Sia che i prodotti di un progetto Scrum siano tangibili sia che siano intangibili, sono necessarie tecniche di verifica adeguate per confermare che il team sta creando i deliverable che otterranno i benefici ed il valore evidenziati all'inizio del progetto.

4

### 4.7.1 Prototipi, Simulazioni e Dimostrazioni

Illustrare i prototipi ai clienti e simulare le loro funzionalità sono tecniche comunemente usate per confermare il valore.

Spesso i clienti possono stabilire più chiaramente se certe funzionalità sono adeguate e calzanti ai propri bisogni dopo averle utilizzate o averne avuto una dimostrazione. Possono ad esempio rendersi conto della necessità di aggiungere alcune funzionalità, o possono decidere di modificare i requisiti di funzionalità stabiliti in precedenza. Nello sviluppo del prodotto, questo tipo di esperienza del cliente è divenuta nota come IKIWISI (I'll Know It When I See It – Lo saprò quando lo vedo).

Attraverso le dimostrazioni o l'accesso a iterazioni anticipate, i clienti possono inoltre valutare in che misura il team è riuscito ad interpretare correttamente i propri requisiti e a soddisfare le proprie aspettative.

## 4.8 Riepilogo delle Responsabilità

Ruolo	Responsabilità
Scrum Guidance Body	<ul style="list-style-type: none"> <li>• Stabilisce le linee guida complessive e le metriche per misurare il valore</li> <li>• Agisce con funzione consultiva e fornisce una guida per i progetti, i programmi e i portfolio secondo necessità.</li> </ul>
Portfolio Product Owner	<ul style="list-style-type: none"> <li>• Assicura la consegna del valore a livello di portfolio</li> <li>• Crea la giustificazione commerciale dei portfolio</li> <li>• Fornisce un orientamento per il valore dei programmi interni ai portfolio</li> <li>• Approva la giustificazione commerciale dei programmi interni al portfolio</li> </ul>
Portfolio Scrum Master	<ul style="list-style-type: none"> <li>• Assicura l'ottenimento dei risultati desiderati del portfolio</li> <li>• Porta avanti la Giustificazione Continua del Valore a livello di portfolio</li> </ul>
Program Product Owner	<ul style="list-style-type: none"> <li>• Assicura la consegna del valore a livello di programma</li> <li>• Crea la giustificazione commerciale dei programmi</li> <li>• Fornisce un orientamento per il valore dei progetti interni ad un programma</li> <li>• Approva la giustificazione commerciale dei progetti interni ad un programma</li> </ul>
Program Scrum Master	<ul style="list-style-type: none"> <li>• Fa in modo che i risultati desiderati del programma siano comunicati e compresi</li> <li>• Porta avanti la Giustificazione Continua del Valore a livello di programma</li> </ul>
Stakeholder	<ul style="list-style-type: none"> <li>• Aiuta a prioritizzare le User Story e i requisiti nel Prioritized Product Backlog</li> <li>• Comunica con lo Scrum Team e conferma la realizzazione del valore alla fine di ogni Sprint, di ogni Rilascio e del progetto.</li> </ul>
Product Owner	<ul style="list-style-type: none"> <li>• Assicura la consegna del valore a livello di progetto</li> <li>• Mantiene la giustificazione commerciale dei progetti</li> <li>• Conferma e comunica i benefici del progetto agli stakeholder</li> </ul>
Scrum Master	<ul style="list-style-type: none"> <li>• Fa in modo che i risultati desiderati del progetto siano comunicati e compresi dallo Scrum Team</li> <li>• Porta avanti la Giustificazione Continua del Valore a livello di progetto</li> </ul>
Scrum Team	<ul style="list-style-type: none"> <li>• Assicura il completamento dei deliverable di progetto conformemente ai Criteri di Accettazione concordati</li> <li>• Porta avanti la Giustificazione Continua del Valore a livello di progetto</li> </ul>

Tavola 4-2: Riepilogo delle Responsabilità Collegate alla Giustificazione Commerciale

## 4.9 Confronto fra Scrum e il Project Management Tradizionale

I progetti tradizionali mettono l'accento su una dettagliata pianificazione anticipata e sulla conformità al piano di progetto creato dal project manager. Di solito i cambiamenti sono gestiti attraverso un sistema formale di gestione dei cambiamenti ed il valore viene generato alla fine del progetto, al momento della consegna del prodotto finale.

Nei progetti Scrum non viene eseguita una pianificazione dettagliata di lungo termine prima dell'esecuzione del progetto. La pianificazione viene realizzata in modo iterativo prima di ciascuno Sprint. Questa modalità permette di rispondere in maniera veloce ed efficace al cambiamento, con conseguente riduzione dei costi e, in definitiva, aumento della redditività e del Ritorno sull'Investimento (ROI). Inoltre, la Consegna basata sul Valore (sezione 4.3) è un beneficio chiave del framework Scrum e consente di eseguire una prioritizzazione sensibilmente migliore e di realizzare più velocemente il valore di business. Grazie alla natura iterativa dello sviluppo propria di Scrum, è sempre disponibile almeno un rilascio del prodotto che possiede le *Minimum Marketable Features* (MMF). Anche se un progetto viene interrotto, normalmente prima della sua fine prematura sono stati comunque generati alcuni benefici o del valore.

## 5. QUALITÀ

### 5.1 Introduzione

Lo scopo di questo capitolo è quello di definire la qualità in rapporto ai progetti e di esporre l'approccio adottato da Scrum per ottenere i livelli di qualità richiesti.

La *Qualità*, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a contare diverse centinaia di membri del team.

Questo capitolo si divide nelle seguenti sezioni:

**5.2 Guida per i Ruoli**— Questa sezione identifica quali parti o paragrafi del capitolo sono importanti per ciascuno dei ruoli di Scrum: Product Owner, Scrum Master e Scrum Team.

**5.3 Definizione di Qualità**—Questa sezione fornisce la definizione di qualità secondo Scrum, distinguendola in modo chiaro dall'ambito, e descrive la relazione intercorrente fra la qualità e il valore di business.

**5.4 Criteri di Accettazione e Prioritized Product Backlog**—Questa sezione evidenzia l'importanza dei Criteri di Accettazione, del Prioritized Product Backlog e delle relazione fra i due. Spiega inoltre la definizione di 'Done' secondo Scrum.

**5.5 La Gestione della Qualità in Scrum**—Questa sezione fornisce dettagli sulla pianificazione della qualità, il controllo della qualità e la garanzia di qualità nel contesto di Scrum.

**5.6 Riepilogo delle Responsabilità**—Questa sezione descrive le responsabilità collegate alla qualità per ciascuna persona o ruolo di un progetto.

**5.7 Confronto fra Scrum e il Project Management Tradizionale**—Questa sezione evidenzia i benefici della gestione della qualità con il metodo Scrum rispetto ai modelli tradizionali di project management.

## 5.2 Guida per i Ruoli

1. Product Owner—Per chi voglia o debba assumere il ruolo di Product Owner in progetti Scrum è importante leggere l'intero capitolo.
2. Scrum Master—Anche lo Scrum Master dovrebbe acquisire familiarità con l'intero capitolo, focalizzandosi in particolare sulle sezioni 5.3, 5.4, 5.5.3 e 5.6.
3. Scrum Team—Lo Scrum Team dovrebbe concentrare principalmente la propria attenzione sulle sezioni 5.3, 5.4 e 5.6.

5

## 5.3 Definizione di Qualità

Ci sono moltissimi modi per definire la qualità.

Per Scrum la qualità si definisce come la capacità del prodotto o dei deliverable completati di soddisfare i Criteri di Accettazione e di realizzare il valore di business atteso dal cliente.

Per fare in modo che un progetto soddisfi i requisiti di qualità, Scrum adotta un approccio di miglioramento continuo, in virtù del quale il team apprende dall'esperienza e dal coinvolgimento degli stakeholder per mantenere il Prioritized Product Backlog costantemente aggiornato con gli eventuali cambiamenti dei requisiti. Il Prioritized Product Backlog non è mai davvero completo fino alla chiusura o alla interruzione del progetto. Gli eventuali cambiamenti dei requisiti rispecchiano cambiamenti dell'ambiente di business interno o esterno e consentono al team di lavorare e adattarsi continuamente per realizzare quei requisiti. Poiché per Scrum il lavoro deve essere completato per incrementi durante gli Sprint, ciò significa che gli errori e i difetti vengono notati prima attraverso test di qualità ripetitivi, invece che quando il prodotto finale o il servizio è prossimo al completamento. In aggiunta, le principali attività collegate alla qualità (ad esempio, sviluppo, test e documentazione) sono completate all'interno del medesimo Sprint e dallo stesso team – il che assicura che la qualità sia inerente a ciascun deliverable finito creato come parte di uno Sprint. Perciò, il miglioramento continuo con test ripetitivi ottimizza la probabilità di realizzare i livelli di qualità attesi in un progetto Scrum. Le costanti discussioni fra lo Scrum Core Team e gli stakeholder (compresi i clienti e gli utenti), unite alla consegna di effettivi incrementi di prodotto alla fine di ogni Sprint, fanno sì che il divario fra le aspettative del cliente nei confronti del progetto e i deliverable realmente prodotti si riduca costantemente.

### 5.3.1 Qualità e Ambito

I requisiti di ambito e di qualità di un progetto sono stabiliti prendendo in considerazione vari fattori, quali:

- L'esigenza di business che il progetto realizzerà
- La capacità e la volontà dell'organizzazione di soddisfare l'esigenza di business identificata
- I bisogni attuali e futuri dei destinatari

L'ambito di un progetto è la somma totale di tutti gli incrementi di prodotto e del lavoro necessario a sviluppare il prodotto finale. La qualità è la capacità dei deliverable di soddisfare i requisiti di qualità del prodotto e di appagare i bisogni del cliente. In Scrum, l'ambito e la qualità del progetto sono registrati nel Prioritized Product Backlog, mentre l'ambito di ciascuno Sprint è stabilito attraverso la rifinitura degli elementi estesi del Prioritized Product Backlog (PBI - Prioritized Product Backlog Items) in un insieme di piccole ma dettagliate User Story che possono essere pianificate, sviluppate e verificate all'interno di uno Sprint.

Il Prioritized Product Backlog è continuamente messo a punto dal Product Owner. Il Product Owner fa in modo che tutte le User Story che ci si aspetta siano eseguite dallo Scrum Team in un determinato Sprint siano rifinite prima dell'inizio di quello Sprint. In generale, i requisiti che hanno maggior valore per la risoluzione dei problemi del cliente o per la soddisfazione dei loro bisogni sono classificati come di alta priorità, mentre agli altri viene assegnata una priorità più bassa. Le User Story meno importanti sono sviluppate negli Sprint successivi o possono essere del tutto escluse in base ai requisiti del cliente. Durante l'esecuzione dello Sprint, il Product Owner, il cliente e lo Scrum Team possono discutere la lista delle funzionalità del prodotto per rispondere ai mutevoli bisogni del cliente.

### 5.3.2 Qualità e Valore di Business

Qualità e valore di business sono strettamente collegati fra loro. Pertanto è fondamentale avere chiari la qualità e l'ambito di un progetto per poter avere uno schema chiaro dei risultati e dei benefici che il progetto ed il suo prodotto devono realizzare per consegnare valore di business. Per stabilire il valore di business di un prodotto, è importante comprendere l'esigenza di business che è alla base dei requisiti del prodotto. Pertanto, l'esigenza di business stabilisce quale sia il prodotto richiesto, e, a sua volta, il prodotto fornisce il valore di business atteso.

La qualità è una variabile complessa. Un'estensione dell'ambito senza un incremento dei tempi o delle risorse tende a ridurre la qualità. Analogamente, anche una riduzione dei tempi o delle risorse senza una restrizione dell'ambito si traduce generalmente in un calo della qualità. Scrum crede nella necessità di mantenere un "ritmo sostenibile" del lavoro, che aiuta con il tempo a migliorare la qualità.

Lo Scrum Guidance Body può definire requisiti minimi di qualità e standard obbligatori per tutti i progetti dell'organizzazione. Gli standard devono essere seguiti da tutti gli Scrum Team all'interno dell'azienda.



## 5.4 Criteri di Accettazione e Prioritized Product Backlog

Il Prioritized Product Backlog è un documento unico sui requisiti che definisce l'ambito del progetto attraverso una lista prioritizzata delle caratteristiche del prodotto o servizio che il progetto deve consegnare. Le caratteristiche richieste sono descritte nella forma di User Story. Le User Story sono specifici requisiti delineati dai vari stakeholder riguardo al prodotto o servizio proposto. Ciascuna User Story avrà dei Criteri di Accettazione della User Story ad essa associati (chiamati anche semplicemente “Criteri di Accettazione”), che rappresentano gli elementi oggettivi attraverso i quali viene giudicata la funzionalità di una User Story. I Criteri di Accettazione sono sviluppati dal Product Owner in base alla propria comprensione esperta dei requisiti del cliente. Il Product Owner comunica poi le User Story inserite nel Prioritized Product Backlog ai membri dello Scrum Team e richiede la loro approvazione. I Criteri di Accettazione devono delineare in maniera esplicita le condizioni che le User Story hanno l'obbligo di soddisfare. Criteri di Accettazione definiti in modo chiaro sono fondamentali per la consegna tempestiva ed efficace della funzionalità individuata nelle User Story e quindi, in definitiva, per il successo dello stesso progetto.

Il Product Owner utilizza questi criteri alla fine di ogni Sprint per verificare i deliverable completati; e può accettare o rigettare singoli deliverable e le User Story ad essi associate. Se i deliverable sono accettati dal Product Owner, allora la User Story si considera ‘Done’, cioè fatta. Una definizione chiara di ciò che può considerarsi ‘Done’ è essenziale in quanto aiuta a chiarire i requisiti e permette al team di conformarsi alle norme di qualità. Inoltre aiuta il team a pensare secondo il punto di vista dell'utente quando lavora alle User Story.

Le User Story corrispondenti a deliverable rifiutati sono reinserite nel Prioritized Product Backlog Aggiornato durante il processo *Mettere a Punto il Prioritized Product Backlog*, per essere poi completate in futuri Sprint. Il rifiuto di pochi, singoli deliverable e delle corrispondenti User Story non comporta il rifiuto del prodotto finale o dell'incremento di prodotto. Il prodotto o incremento di prodotto potrebbe essere potenzialmente disponibile per la consegna anche nel caso in cui sia rifiutata qualche User Story.

La Figura 5-1 illustra il concetto di Criteri di Accettazione coniugato al flusso dell'incremento di prodotto.

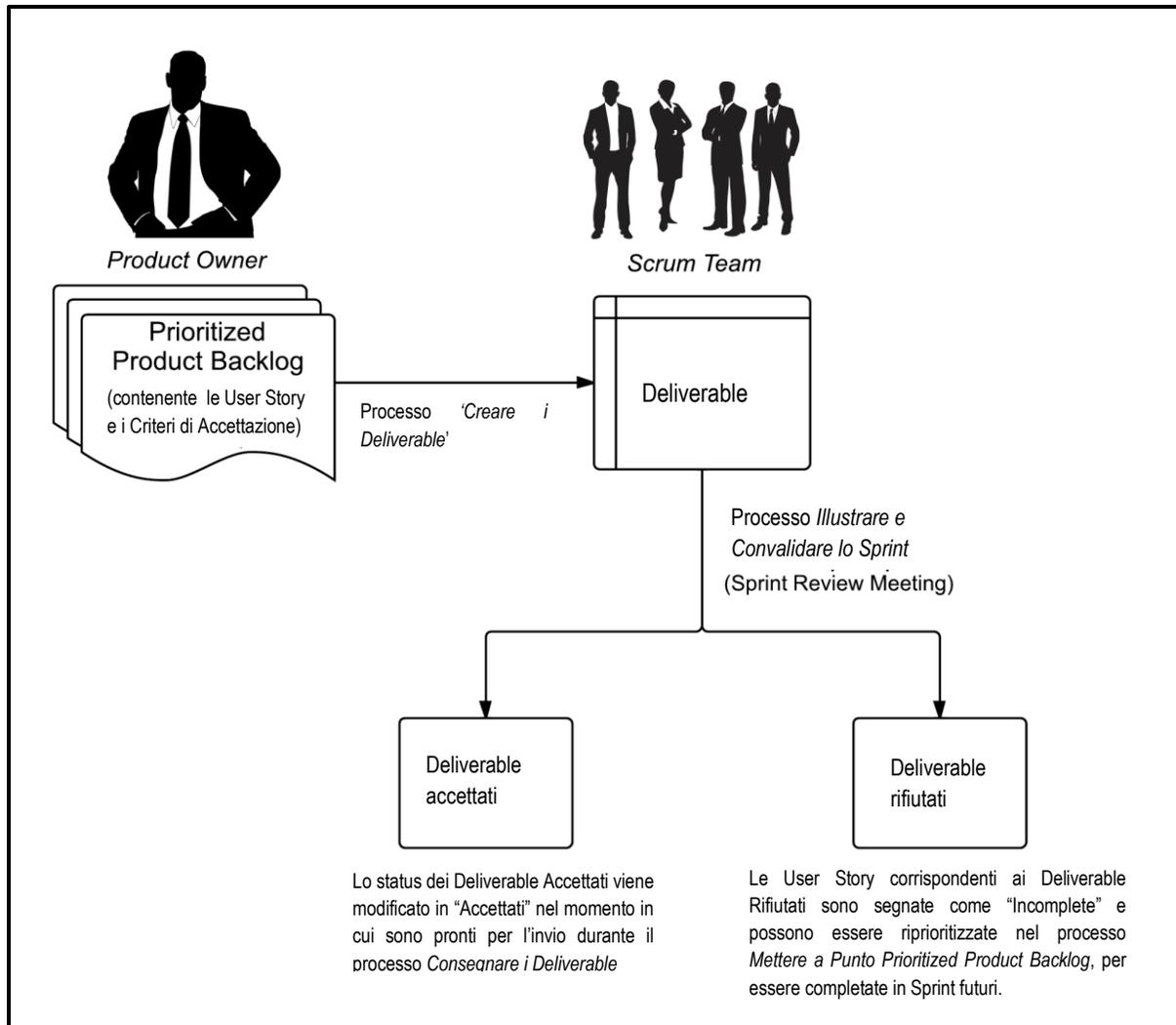


Figura 5-1: Diagramma di Flusso dell'Incremento di Prodotto

### 5.4.1 Scrivere i Criteri di Accettazione

I Criteri di Accettazione sono propri e specifici di ciascuna User Story e non sono il sostituto di una lista dei requisiti.

*Esempio:*

*Persona:* Janine è una libera professionista di 36 anni, sposata con tre figli. È una donna impegnata e di successo, che concilia la sua vita professionale con quella personale. Ha dimestichezza con la tecnologia ed è costantemente fra i primi ad utilizzare servizi e prodotti innovativi. È sempre connessa ad internet con diversi dispositivi e fa regolarmente acquisti dai portali di e-commerce.

User Story: “Io Janine, nella mia qualità di acquirente di prodotti alimentari online, devo poter salvare e vedere la mia bozza di ordine da qualsiasi dispositivo, in modo da poter completare il processo di acquisto quando mi è più comodo.”

Criteri di Accettazione:

- Tutti gli ordini in corso devono essere salvati come bozza di ordine ogni 5 secondi nell’account dell’utente connesso
- Le nuove bozze di ordine devono arrivare come notifiche su qualunque dispositivo al quale l’utente si connetta

Per un Product Owner è importante notare che le User Story che soddisfano la maggior parte dei Criteri di Accettazione, ma comunque non tutti, non possono essere accettate come ‘Done’. I progetti Scrum operano in Sprint di durata predeterminata (Time-boxed), con uno Sprint Backlog dedicato per ciascuno Sprint. Spesso, l’ultimo pezzetto di lavoro può essere la parte più complicata di una User Story e può richiedere più tempo di quanto ci si aspettasse. Se a User Story incomplete si riconosce la qualifica di parzialmente ‘Done’ e quindi le si prosegue nello Sprint successivo, l’avanzamento dello Sprint a seguire si potrebbe interrompere. Per questo motivo, lo status di ‘Done’ è o bianco o nero. Una User Story può essere solo ‘Done’ o ‘not Done’.

### 5.4.2 Criteri di Accettazione Minimi

Un’unità di business di più alto livello può rendere pubblici dei Criteri di Accettazione Minimi obbligatori, che entreranno poi a far parte dei Criteri di Accettazione di tutte le User Story di quella unità di business. Qualsiasi funzionalità definita dall’unità di business deve soddisfare questi Criteri di Accettazione Minimi per essere accettata dal rispettivo Product Owner. L’introduzione di questi Criteri di Accettazione può portare ad una serie di Criteri di Accettazione a cascata per il portfolio, per il programma e per il progetto (si veda la Figura 5-2). Gli standard di qualità generali, le linee guida e i modelli di un intero portfolio sono fissati dal Portfolio Product Owner, mentre i Criteri di Accettazione minimi per i programmi sono fissati dal Program

Product Owner. Pertanto, i Criteri di Accettazione di una User Story che fa parte di un progetto includeranno implicitamente tutti i Criteri di Accettazione minimi dei livelli superiori, ove applicabili.

Portfolio Product Owner	<ul style="list-style-type: none"> <li>• Fissa i Criteri di Accettazione minimi dell'intero portfolio</li> <li>• Rivede i deliverable del portfolio</li> </ul>
Program Product Owner	<p>Fissa i Criteri di Accettazione minimi dell'intero programma, che includono i Criteri di Accettazione del portfolio</p> <ul style="list-style-type: none"> <li>• Rivede i deliverable del programma</li> </ul>
Product Owner	<ul style="list-style-type: none"> <li>• Fissa i Criteri di Accettazione minimi del progetto, che includono i Criteri di Accettazione del programma</li> <li>• Rivede i deliverable del progetto</li> </ul>

5

Figura 5-2: Criteri di Accettazione a Cascata

Una volta definiti, tali Criteri di Accettazione minimi possono poi essere documentati nei documenti dello Scrum Guidance Body e utilizzati come riferimento dagli Scrum Team ove necessario.

### 5.4.3 Definizione di 'Done'

Esiste una differenza fondamentale fra i "Done Criteria" e i "Criteri di Accettazione". Mentre i Criteri di Accettazione sono propri e specifici di una singola User Story, i Done Criteria sono un complesso di regole applicabili a tutte le User Story di un dato Sprint. I Done Criteria generali possono includere una qualunque delle seguenti opzioni:

- Revisione da parte degli altri membri del team
- È stato completato il testing di unità della User Story
- Completamento dei test di garanzia della qualità
- Completamento di tutta la documentazione relativa alla User Story
- Tutte le questioni sono state risolte
- Dimostrazione con esito positivo agli stakeholder e/o ai rappresentanti del business

Come per i Criteri di Accettazione, perché la User Story possa essere considerata 'Done', cioè fatta, devono essere soddisfatte tutte le condizioni dei Done Criteria.

Lo Scrum Team dovrebbe utilizzare una lista di controllo dei Done Criteria generali per accertarsi con sicurezza che un'attività sia finita e che il risultato soddisfi la Definizione di 'Done' (Definition of Done - DoD). Una Definizione di 'Done' chiara è di importanza fondamentale, perché aiuta a rimuovere le ambiguità e permette al team di conformarsi alle norme di qualità prescritte. La definizione di 'Done' è solitamente stabilita e documentata dallo Scrum Guidance Body.

Le registrazioni e i dati richiesti per il rispetto dei requisiti di documentazione del progetto possono essere generati man mano che il team va avanti con gli Sprint e i Rilasci.

L'inserimento di attività come la conduzione di riunioni di revisione e la stesura di documenti di progettazione può contribuire ad assicurare il rispetto degli standard di qualità interni ed esterni. I principi di base di Scrum - come le iterazioni brevi, la costruzione incrementale, il coinvolgimento del cliente, l'adattamento al cambiamento dei requisiti e, per finire, il costante aggiustamento dell'ambito, dei tempi e dei costi all'interno del progetto - saranno applicati in ogni caso.

#### 5.4.4 Accettazione o Rifiuto degli Elementi del Prioritized Product Backlog

Verso la fine di un'iterazione, le rispettive unità di business e gli stakeholder partecipano ad uno Sprint Review Meeting nel quale l'incremento di prodotto viene illustrato al Product Owner, allo sponsor, al cliente e agli utenti. Sebbene venga raccolto il feedback da tutti gli stakeholder, solo il Product Owner ha il potere di accettare o rifiutare una particolare User Story come 'Done', in base ai Criteri di Accettazione concordati. Perciò il ruolo dei Criteri di Accettazione nel mantenimento della qualità è fondamentale e deve essere compreso chiaramente dal team. Spetta allo Scrum Master fare in modo che il Product Owner non modifichi i Criteri di Accettazione di una User Story nel mezzo di uno Sprint. Le User Story parzialmente completate sono rifiutate come 'not Done' e reinserite nel Prioritized Product Backlog.

### 5.5 La Gestione della Qualità in Scrum

Il cliente è lo stakeholder più importante di qualsiasi progetto. Per questo è importante capire i bisogni e di requisiti del cliente. La Voce del Cliente (Voice of the Customer - VOC) può essere definita come i requisiti impliciti ed espliciti del cliente, che devono essere compresi prima di progettare un prodotto o servizio. Generalmente, in un ambiente Scrum, l'attenzione del Product Owner è concentrata sui requisiti di business e sugli obiettivi, che insieme rappresentano la Voce del Cliente. Il Product Owner può trarre un grande beneficio dagli orientamenti messi a disposizione dallo Scrum Guidance Body (con i documenti o gli standard sulla qualità, o tramite gli esperti della qualità). Questi specialisti dovrebbero lavorare con il Product Owner e il cliente per assicurare alle User Story l'adeguato livello di dettaglio e di informazioni, dal momento che le User Story rappresentano la base del successo di qualsiasi progetto Scrum.

Da notare che gli stakeholder esterni non sono direttamente coinvolti a livello di Scrum Team, mentre interagiscono soprattutto con il Product Owner. Per qualsiasi progetto Scrum, il cliente può essere indifferentemente:

- Interno (cioè appartenente alla stessa organizzazione)
- Esterno (cioè al di fuori dell'organizzazione)

In Scrum la gestione della qualità consente ai clienti di prendere coscienza dei problemi del progetto fin dal suo inizio e li aiuta a rendersi conto se un progetto sta funzionando o meno. In Scrum la qualità riguarda la soddisfazione del cliente e un prodotto funzionante, non necessariamente la rispondenza a metriche

arbitrarie. Questa distinzione assume molta importanza dal punto di vista dei clienti, dal momento che sono loro che investono tempo e denaro nel progetto.

La gestione della qualità in Scrum è facilitata da tre attività fra loro correlate:

1. Pianificazione della qualità
2. Controllo della qualità
3. Garanzia della qualità

### 5.5.1 Pianificazione della Qualità

Uno dei principi guida di Scrum è quello di sviluppare per prima la funzionalità che ha la priorità più alta per il cliente. Le funzionalità meno importanti sono sviluppate negli Sprint successivi o possono essere escluse del tutto in base ai requisiti del cliente. Questo approccio dà modo allo Scrum Team di avere tempo per concentrarsi sulla qualità della funzionalità essenziale. Un beneficio fondamentale della pianificazione della qualità è la riduzione del technical debt. Il technical debt (debito tecnico)—chiamato anche debito di progettazione (design debt) o debito del codice (code debt)—si riferisce al lavoro a cui il team dà priorità più bassa, oppure omette o non completa, in quanto lavora alla creazione dei deliverable principali associati al prodotto del progetto. Il technical debt si accumula e si paga nel futuro.

Di seguito sono elencate alcune delle cause del technical debt:

- Soluzioni tampone e costruzione di deliverable che non si conformano agli standard di qualità, sicurezza, obiettivi architetturali di lungo termine, ecc.
- Attività di testing inadeguata o incompleta
- Documentazione inappropriata o incompleta
- Mancanza di coordinamento fra i vari membri del team, o situazioni in cui diversi Scrum Team iniziano a lavorare in modo isolato, con conseguente scarsa concentrazione sulla integrazione finale dei vari componenti, necessaria a garantire il successo del progetto o programma
- Scarsa condivisione della conoscenza di business e di processo fra gli stakeholder e i team di progetto
- Eccessiva focalizzazione sugli obiettivi di breve termine del progetto invece che sugli obiettivi aziendali di lungo termine. Questa disattenzione può dar luogo a Deliverable Funzionanti di bassa qualità, che provocano significativi costi di manutenzione e di aggiornamento.

Nei progetti Scrum, grazie a Criteri di Accettazione e Done Criteria chiaramente definiti, non viene trasferito alcun technical debt da uno Sprint al successivo. La funzionalità deve soddisfare questi criteri per essere considerata 'Done'. Dal momento che il Prioritized Product Backlog viene costantemente messo a punto e le User Story vengono prioritzate, il team crea regolarmente Deliverable Funzionanti, prevenendo così l'accumulazione di un significativo technical debt. Inoltre, lo Scrum Guidance Body può includere documentazione e definizioni dei processi che contribuiscono a ridurre il technical debt.

Per mantenere una quantità minima di technical debt, è importante definire il prodotto che si richiede a uno Sprint e all'intero progetto insieme ai Criteri di Accettazione, agli eventuali metodi di sviluppo da seguire, e alle responsabilità chiave attribuite ai membri dello Scrum Team in materia di qualità. La definizione dei Criteri di Accettazione è una parte importante della pianificazione della qualità e consente di mettere in atto un controllo di qualità efficace durante il progetto.

Il technical debt è una sfida molto impegnativa per alcune tecniche di project management tradizionale, nelle quali lo sviluppo, il testing, la documentazione, ecc. sono eseguite in sequenza e molto spesso da persone diverse, senza che nessuno sia responsabile per un particolare Deliverable Funzionante. Il risultato è che il technical debt si accumula, con conseguenti costi di mantenimento, di integrazione e di rilascio del prodotto significativamente più alti nelle fasi finali della consegna di un progetto. Inoltre, in questi casi il costo delle modifiche è molto alto poiché i problemi emergono solo nelle ultime fasi del progetto. Il framework Scrum previene le questioni collegate al technical debt assicurando che i deliverable classificabili come 'Done' con i relativi Criteri di Accettazione siano definiti come parte dello Sprint Backlog e che anche le attività fondamentali fra cui lo sviluppo, il testing e la documentazione siano eseguite nell'ambito del medesimo Sprint e dallo stesso Scrum Team.

#### **5.5.1.1 Integrazione Continua e Ritmo Sostenibile**

Il mantenimento di un ritmo sostenibile è uno dei principali cardini di Scrum. Il ritmo sostenibile si traduce nell'aumento della soddisfazione del personale, nella stabilità e in una maggiore accuratezza delle stime, fattori che portano tutti in definitiva ad una maggiore soddisfazione del cliente. Per sviluppare un prodotto veramente di alta qualità e mantenere un ambiente di lavoro sano, è importante eseguire regolarmente attività di carattere integrativo piuttosto che rinviare fino alla fine il lavoro di integrazione. Per fornire valore a intervalli frequenti, il team deve sviluppare, testare e integrare continuamente le funzionalità di ciascun elemento del Prioritized Product Backlog (PBI) in ogni Sprint, facendo uso di tecniche come l'integrazione continua e i test automatizzati del prodotto. È inoltre importante, dal punto di vista del team, fare in modo che l'impegno profuso nello Sprint corrente sia analogo a quello dello Sprint precedente, al fine di mantenere un ritmo costante per tutti gli Sprint del progetto. Questo aiuta il team ad evitare fasi di intensi periodi di lavoro, mettendolo in condizione di riuscire sempre a approfondire il livello di sforzo necessario a compiere il lavoro che deve essere svolto.

#### **5.5.2 Controllo di Qualità e Garanzia di Qualità**

Il controllo di qualità si riferisce alla esecuzione delle attività pianificate relative alla qualità da parte dello Scrum Team nel processo di creazione di deliverable che sono potenzialmente consegnabili. Tale controllo include anche l'apprendimento da ciascun insieme di attività completate al fine di realizzare il miglioramento continuo. È importante possedere, all'interno di un team interfunzionale, le capacità necessarie ad eseguire le attività di controllo di qualità. Durante lo Sprint Retrospect Meeting, i membri del team discutono le lezioni

apprese. Queste lezioni fungono da input per il miglioramento continuo e contribuiscono al miglioramento del controllo di qualità sempre in corso.

La qualità è richiesta non solo nei prodotti ma anche nei processi. La garanzia di qualità si riferisce alla valutazione dei processi e degli standard che governano la gestione della qualità in un progetto per confermare che continuano ad essere pertinenti. Le attività di garanzia della qualità sono eseguite come parte del lavoro di progetto. In effetti, la garanzia di qualità è un fattore importante della definizione di 'Done'. Il deliverable non è completato se non è stata eseguita l'opportuna garanzia di qualità. Spesso, la garanzia di qualità è dimostrata durante lo Sprint Review Meeting.

I Product Owner dei rispettivi progetti, programmi e portfolio possono monitorare e valutare le attività di garanzia di qualità per assicurarsi che ogni team continui ad essere in accordo e a conformarsi agli standard di qualità fissati. Una garanzia di qualità completa può riguardare il test finale di un prodotto, un Rilascio o uno Sprint. Si può eseguire un confronto fra il numero di questioni emerse e il numero di User Story completate. I componenti del prodotto che presentano difetti possono essere incorporati come elementi del Prioritized Product Backlog (PBI), ai quali possono lavorare l'intero team o una singola persona in determinati momenti dello Sprint, a seconda del numero dei difetti.

A volte, lo Scrum Guidance Body può definire i processi e i documenti a cui gli Scrum Team devono fare riferimento nell'eseguire i loro progetti, per garantire che tutti i progetti dell'azienda seguano le medesime norme di qualità.

### 5.5.3 Ciclo Plan-Do-Check-Act (PDCA)

Il Ciclo Plan-Do-Check-Act—noto anche come Ciclo di Deming o di Shewhart—fu sviluppato dal Dott. W. Edwards Deming, considerato il padre del moderno controllo di qualità e dal Dott. Walter A. Shewhart. Quelli che seguono sono alcuni punti importanti della filosofia di Deming:

- Le linee guida del management definiscono la qualità. Quando il management è in grado di mettere a disposizione un ambiente favorevole ed è capace di motivare i propri dipendenti per migliorare la qualità su base continua, ciascun dipendente sarà in grado di dare il proprio fattivo contributo alla realizzazione di un prodotto di qualità superiore. La "Teoria della Conoscenza Approfondita" di Deming raccomanda ciò che il management dovrebbe fare per creare un ambiente nel quale ogni dipendente può contribuire in modo significativo al miglioramento della qualità.

Deming modificò successivamente il ciclo Plan-Do-Check-Act in Plan-Do-Study-Act (PDSA, perché ritenne che il termine 'Study' – Esamina – mettesse in luce il concetto di analisi piuttosto che semplicemente quello di ispezione, come suggerito dal termine 'Check' – Controlla).

Sia Scrum che il Ciclo PDCA di Deming/Shewhart sono metodi iterativi che si focalizzano sul miglioramento continuo. La Figura 5-3 illustra le fasi del Ciclo e la loro correlazione con i vari processi di Scrum.

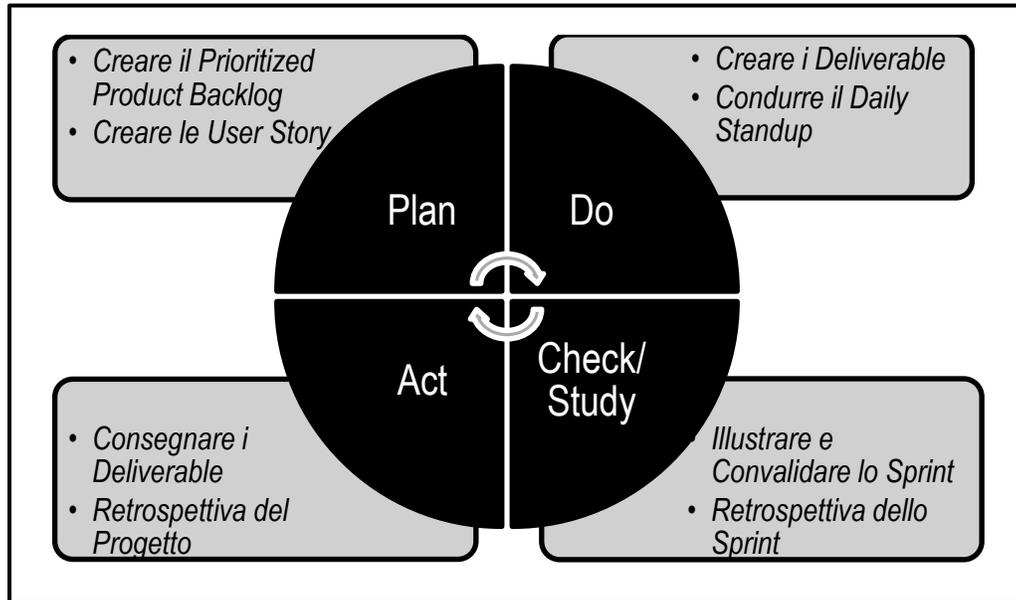


Figura 5-3: Ciclo PDCA in Scrum

## 5.6 Riepilogo delle Responsabilità

Ruolo	Responsabilità
Scrum Guidance Body	<ul style="list-style-type: none"> <li>• Fornisce la definizione di 'Done'</li> <li>• Fornisce lo schema e i riferimenti per lo sviluppo dei Criteri di Accettazione</li> <li>• Definisce la gamma di strumenti che possono essere utilizzati dallo Scrum Team per sviluppare e verificare il prodotto</li> </ul>
Portfolio Product Owner	<ul style="list-style-type: none"> <li>• Fissa i Criteri di Accettazione minimi dell'intero portfolio</li> <li>• Rivede i Deliverable del portfolio</li> </ul>
Portfolio Scrum Master	<ul style="list-style-type: none"> <li>• Fa in modo che sia mantenuto un ritmo sostenibile e che ci si concentri sulla qualità delle funzionalità piuttosto che strettamente sulla velocità</li> </ul>
Program Product Owner	<ul style="list-style-type: none"> <li>• Fissa i Criteri di Accettazione minimi dell'intero programma</li> <li>• Rivede i Deliverable del programma</li> </ul>
Program Scrum Master	<ul style="list-style-type: none"> <li>• Fa in modo che sia mantenuto un ritmo sostenibile e che ci si concentri sulla qualità delle funzionalità piuttosto che strettamente sulla velocità</li> </ul>
Stakeholder	<ul style="list-style-type: none"> <li>• Rivedono e accettano i Deliverable e il prodotto finale</li> </ul>
Product Owner	<ul style="list-style-type: none"> <li>• Stabilisce i requisiti di business del prodotto e definisce chiaramente i requisiti del Prioritized Product Backlog</li> <li>• Valuta la fattibilità e assicura che i Deliverable soddisfino i requisiti di qualità</li> <li>• Fissa i Criteri di Accettazione minimi dell'intero progetto, inclusi i Criteri di Accettazione del relativo programma</li> </ul>

	<ul style="list-style-type: none"> <li>• Facilita la creazione dei Criteri di Accettazione delle User Story</li> <li>• Rivede e convalida i Deliverable durante il processo <i>Illustrare e Convalidare lo Sprint</i></li> </ul>
Scrum Master	<ul style="list-style-type: none"> <li>• Facilita una mentalità “di squadra” quando si tratta di qualità</li> <li>• Elimina gli ostacoli ambientali che possono incidere sulla qualità dei deliverable e dei processi</li> <li>• Fa in modo che sia mantenuto un ritmo sostenibile e che ci si concentri sulla qualità delle funzionalità piuttosto che strettamente sulla velocità</li> <li>• Assicura che i processi Scrum siano seguiti correttamente da tutti i membri del team, incluso il Product Owner</li> </ul>
Scrum Team	<ul style="list-style-type: none"> <li>• Sviluppa e mantiene tutti i Deliverable durante gli Sprint fino a quando non sono trasferiti agli utenti finali</li> <li>• Mette in pratica e incoraggia una buona comunicazione che garantisca chiarezza e piena comprensione dei requisiti</li> <li>• Condivide la conoscenza per permettere ai membri del team di familiarizzare con l'intero complesso delle funzionalità e quindi trarre beneficio dall'altrui esperienza</li> <li>• Effettua prontamente le opportune modifiche ai Deliverable</li> </ul>

Tavola 5-1: Riepilogo delle Responsabilità attinenti alla Qualità

## 5.7 Confronto fra Scrum e il Project Management Tradizionale

Anche se ci sono delle analogie fra Scrum e i metodi tradizionali di project management riguardo alla definizione di ‘qualità’ (vale a dire, la capacità del prodotto di soddisfare i Criteri di Accettazione concordati e di realizzare il valore di business atteso dal cliente), ci sono delle differenze riguardo a come i due approcci affrontano l’implementazione e la realizzazione dei livelli di qualità richiesti.

Nei metodi tradizionali di project management, gli utenti chiariscono le proprie aspettative; il project manager definisce quelle aspettative in termini misurabili e ottiene l’approvazione da parte degli utenti. Dopo una pianificazione dettagliata, il team di progetto sviluppa il prodotto in un periodo di tempo concordato. Se deve essere modificato qualcuno dei criteri concordati, le modifiche possono avvenire solo attraverso un sistema formale di gestione dei cambiamenti nel quale viene stimato l’impatto delle modifiche e il Project Manager ottiene l’approvazione da parte di tutti gli stakeholder pertinenti.

In Scrum, tuttavia, il Product Owner collabora con lo Scrum Team e definisce i Criteri di Accettazione delle User Story relative al prodotto da consegnare. Lo Scrum Team poi sviluppa il prodotto in una serie di brevi iterazioni chiamate Sprint. Il Product Owner può operare cambiamenti ai requisiti per stare al passo con i bisogni dell’utente e questi cambiamenti possono essere presi in esame dallo Scrum Team terminando anticipatamente lo Sprint corrente oppure includendo i requisiti modificati nello Sprint successivo, dal momento che ogni Sprint è di durata molto breve (vale a dire, da una a sei settimane).

Uno dei maggiori vantaggi di Scrum è il risalto dato alla creazione di deliverable potenzialmente consegnabili alla fine del ciclo di ogni Sprint, invece che alla fine dell'intero progetto. Quindi, il Product Owner e i clienti ispezionano, approvano e accettano costantemente i Deliverable dopo ogni Sprint. Inoltre, anche se un progetto Scrum viene terminato anticipatamente, prima dell'interruzione si è comunque generato del valore, che è costituito appunto dai Deliverable creati nei singoli Sprint.

## 6. CAMBIAMENTO

### 6.1 Introduzione

Tutti i progetti, indipendentemente dal metodo o framework adottato, sono esposti al cambiamento. È assolutamente indispensabile che i membri del team di progetto comprendano che i processi di sviluppo di Scrum sono concepiti proprio per abbracciare il cambiamento. Le organizzazioni devono cercare di massimizzare i benefici che scaturiscono dal cambiamento e minimizzarne gli eventuali impatti negativi attraverso processi accurati di gestione del cambiamento, nel rispetto dei principi di Scrum.

Il *Cambiamento*, così come definito nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Questo capitolo si divide nelle seguenti sezioni:

**6.2 Guida per i Ruoli** — Questa sezione fornisce un’indicazione su quali paragrafi sono importanti per ciascuno dei principali ruoli di Scrum: Product Owner, Scrum Master e Scrum Team.

**6.3 Panoramica**—Questa sezione definisce il concetto di cambiamento, specificamente all’interno del contesto dei processi di Scrum. Inoltre tratta del modo in cui sono gestite le Richieste di Modifica nei processi Scrum.

**6.4 Cambiamento in Scrum**—Questa sezione approfondisce l’importanza di un’efficace gestione del cambiamento in un progetto Scrum. Tratta inoltre del modo in cui si possono realizzare flessibilità e stabilità attraverso una gestione appropriata delle Richieste di Modifica che si presentano nel corso di un progetto.

**6.5 Integrazione del Cambiamento**—Questa sezione chiarisce nel dettaglio come vengono valutate e approvate (o rifiutate) le Richieste di Modifica in applicazione del framework Scrum.

**6.6 Cambiamento nei Programmi e nei Portfolio**—Questa sezione descrive l’impatto dei cambiamenti sui programmi e sui portfolio.

**6.7 Riepilogo delle Responsabilità**—Questa sezione definisce le responsabilità dei membri del team di progetto in ordine alla gestione del cambiamento.

**6.8 Confronto fra Scrum e il Project Management Tradizionale**—Questa sezione parla dei benefici della gestione del cambiamento con i metodi Scrum rispetto ai metodi utilizzati nei modelli tradizionali di project management.

## 6.2 Guida per i Ruoli

1. **Product Owner**—La responsabilità di avviare il cambiamento fa capo prima di tutto al Product Owner; pertanto, per questo ruolo è consigliato l'intero capitolo.
2. **Scrum Master**—Anche lo Scrum Master dovrebbe avere familiarità con l'intero capitolo, con particolare attenzione alle sezioni 6.3, 6.4, 6.5 e 6.7.
3. **Scrum Team**—Lo Scrum Team dovrebbe concentrarsi principalmente sulle sezioni 6.3, 6.4.2, 6.5 e 6.7.

## 6.3 Panoramica

Il cambiamento è inevitabile in tutti i progetti. Nell'ipercompetitivo mondo di oggi, in cui la tecnologia, le condizioni di mercato e i modelli di business sono in continuo movimento, il cambiamento rappresenta l'unica costante.

Per Scrum è fondamentale riconoscere che a) gli stakeholder (ad esempio clienti, utenti e sponsor) nel corso del progetto cambiano idea su quello che vogliono e su ciò di cui hanno bisogno (il cosiddetto 'requirements churn', cioè l'abbandono dei requisiti) e b) che è molto difficile, se non impossibile, per gli stakeholder definire tutti i requisiti durante l'inizio del progetto.

I progetti di sviluppo Scrum accolgono il cambiamento tramite piccoli cicli di sviluppo, che incorporano il feedback del cliente sui deliverable del progetto dopo ogni singolo Sprint. Questo permette al cliente di interagire regolarmente con i membri dello Scrum Team, di vedere gli incrementi di prodotto appena sono pronti e di modificare i requisiti fin dall'inizio del ciclo di sviluppo. Inoltre, i team di gestione del portfolio o del programma possono rispondere alle Richieste di Modifica attinenti ai progetti Scrum e riferibili al loro livello.

Scrum incorpora un principio fondamentale del Manifesto Agile (Fowler e Highsmith, 2001): "Rispondere al cambiamento prima di seguire un piano". L'applicazione di Scrum si basa sul concetto che il cambiamento va abbracciato e trasformato in vantaggio competitivo. Per tale motivo, è più importante essere flessibili che seguire strettamente un piano predefinito. Questo significa che è essenziale avere un approccio adattivo al project management che consenta il cambiamento durante l'intero ciclo di sviluppo del prodotto o servizio.

L'adattabilità al cambiamento è uno dei vantaggi chiave del framework Scrum. Anche se Scrum funziona bene in tutti i progetti di qualsiasi settore industriale, può essere particolarmente efficace quando il prodotto o altri requisiti del progetto non sono del tutto chiari o non possono essere definiti bene in anticipo, quando il

mercato dello specifico prodotto è volatile, e/o quando si punta a rendere il team sufficientemente flessibile da riuscire ad incorporare il cambiamento dei requisiti. Scrum è utile specialmente per i progetti complessi che presentano un grado elevato di incertezza. La pianificazione e le previsioni di lungo termine sono di solito inefficaci per questo tipo di progetti, che implicano alti livelli di rischio. Scrum guida il team al raggiungimento dei risultati di business di maggior valore attraverso i principi della *trasparenza*, dell'*ispezione* e dell'*adattamento*.

### 6.3.1 Richieste di Modifica non Approvate e Approvate

Le richieste di procedere a dei cambiamenti sono di solito presentate come Richieste di Modifica. Le Richieste di Modifica rimangono nello stato di 'non approvate' fino a quando non ottengono l'approvazione formale. Lo Scrum Guidance Body definisce di solito un processo di approvazione e gestione dei cambiamenti valido per tutta l'organizzazione. In mancanza di un processo formale, è consigliabile far approvare direttamente dal Product Owner le piccole modifiche che non hanno un impatto significativo sul progetto. La tolleranza per queste piccole modifiche può essere definita a livello di organizzazione oppure dallo sponsor per uno specifico progetto. Nel maggior parte dei progetti, il 90% delle Richieste di Modifica può essere classificato come piccole modifiche che devono essere approvate dal Product Owner. Per questo motivo il Product Owner ricopre un ruolo molto importante nella gestione dei cambiamenti in un Progetto Scrum.

Le modifiche che superano il livello di tolleranza assegnato al Product Owner devono essere approvate dagli stakeholder appropriati, che lavorano insieme al Product Owner.

A volte, se da un cambiamento richiesto potrebbe derivare un impatto considerevole sul progetto o sull'organizzazione, può essere necessaria l'approvazione da parte del senior management (ad esempio, Executive Sponsor, Portfolio Product Owner, Program Product Owner, o Chief Product Owner).

Le Richieste di Modifica relative al progetto sono discusse e approvate durante i processi *Sviluppare le Epic*, *Creare il Prioritized Product Backlog* e *Mettere a Punto il Prioritized Product Backlog*. Le *Richieste di Modifica Approvate* sono quindi prioritizzate insieme agli altri requisiti del prodotto e alle loro rispettive User Story e poi incorporate nel Prioritized Product Backlog.

La Figura 6-1 sintetizza il processo di approvazione dei cambiamenti, mentre la Figura 6-2 spiega come si aggiorna il Prioritized Product Backlog con i cambiamenti approvati.

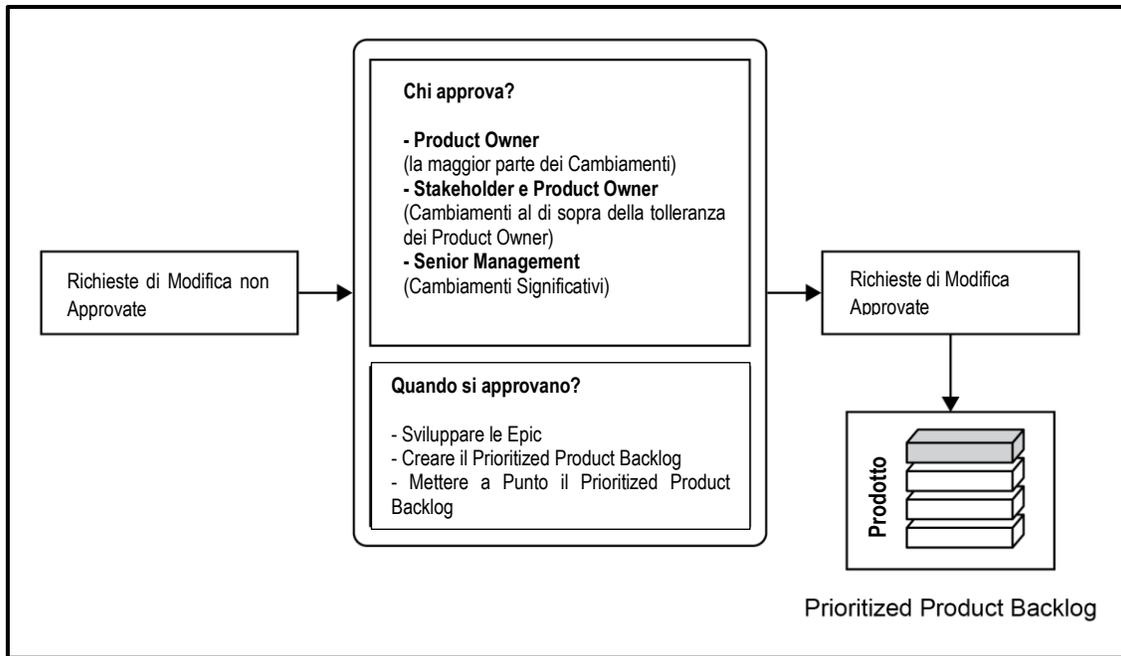


Figura 6-1: Esempio di Processo di Approvazione dei Cambiamenti

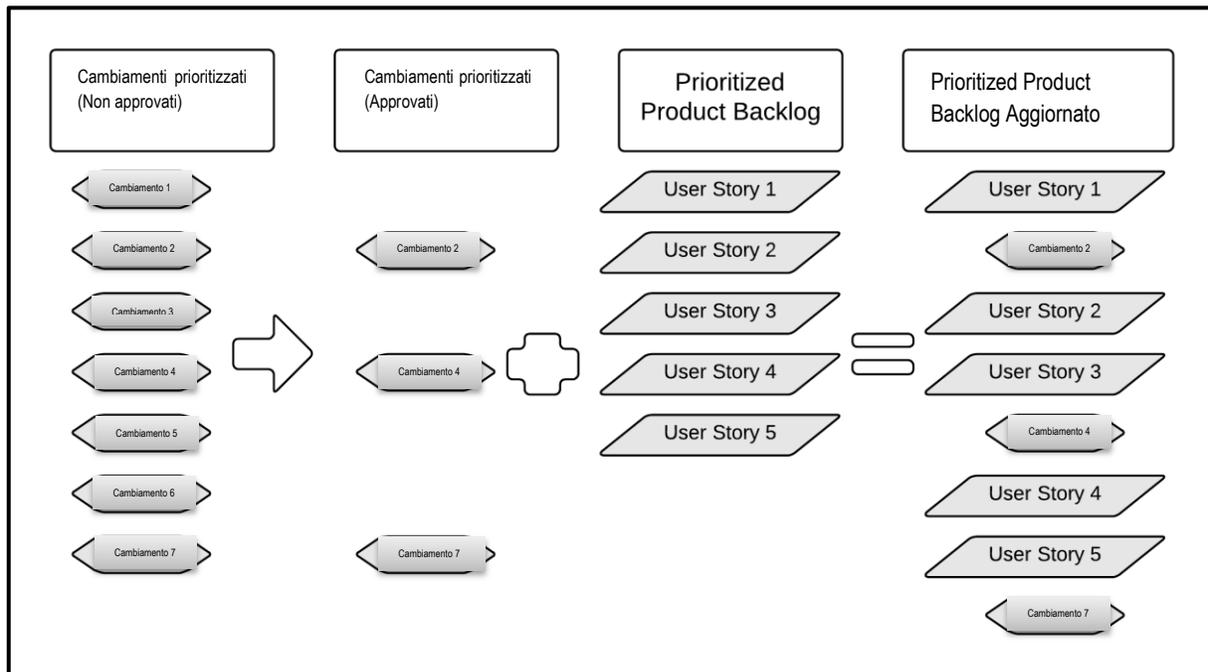


Figura 6-2: Aggiornamento del Prioritized Product Backlog con i Cambiamenti Approvati

## 6.4 Il Cambiamento in Scrum

### 6.4.1 Bilanciare Flessibilità e Stabilità

Scrum contribuisce a far diventare le organizzazioni più flessibili e aperte al cambiamento. Tuttavia, è importante avere chiaro che, sebbene Scrum enfatizzi il concetto di flessibilità, è altrettanto indispensabile mantenere la stabilità durante il processo di cambiamento. Così come l'estrema rigidità è controproducente, lo è allo stesso modo l'estrema flessibilità. La chiave giusta è trovare il corretto equilibrio fra flessibilità e stabilità, perché quest'ultima è necessaria per portare a compimento il lavoro. Scrum utilizza perciò la consegna iterativa e le sue altre caratteristiche e principi per realizzare questo equilibrio. Scrum realizza la flessibilità consentendo la creazione e approvazione delle Richieste di Modifica in qualunque momento del progetto; tuttavia, tali richieste vengono prioritizzate al momento della creazione del Prioritized Product Backlog o al momento del suo aggiornamento. Nello stesso tempo, Scrum assicura la stabilità mantenendo fermo lo Sprint Backlog e non consentendo alcuna interferenza con lo Scrum Team durante uno Sprint.

In Scrum, tutti i requisiti collegati ad uno Sprint in corso sono congelati durante la sua esecuzione. Fino a quando lo Sprint non finisce, non viene introdotto alcun cambiamento, a meno che un particolare cambiamento sia ritenuto così significativo da determinare l'interruzione dello Sprint in questione. Nel caso emerga la necessità di un cambiamento urgente, lo Sprint viene terminato anticipatamente e il Team si riunisce per pianificare un nuovo Sprint. Questo è il modo in cui Scrum accetta i cambiamenti senza creare il problema della modifica delle date di rilascio.

## 6.4.2 Realizzare la Flessibilità

Scrum favorisce la flessibilità attraverso la *trasparenza*, l'*ispezione* e l'*adattamento* per realizzare in definitiva i risultati di business di maggior valore. Scrum fornisce un meccanismo adattivo di project management, nel quale si può dare spazio ad un cambio di requisiti senza impattare significativamente sul complessivo avanzamento del progetto. L'adattamento alle emergenti realtà di business è una parte necessaria del ciclo di sviluppo. In Scrum lo si realizza attraverso cinque caratteristiche fondamentali (vedere la Figura 6-3): sviluppo iterativo del prodotto, Time-boxing, team interfunzionali, prioritizzazione basata sul valore e integrazione continua.

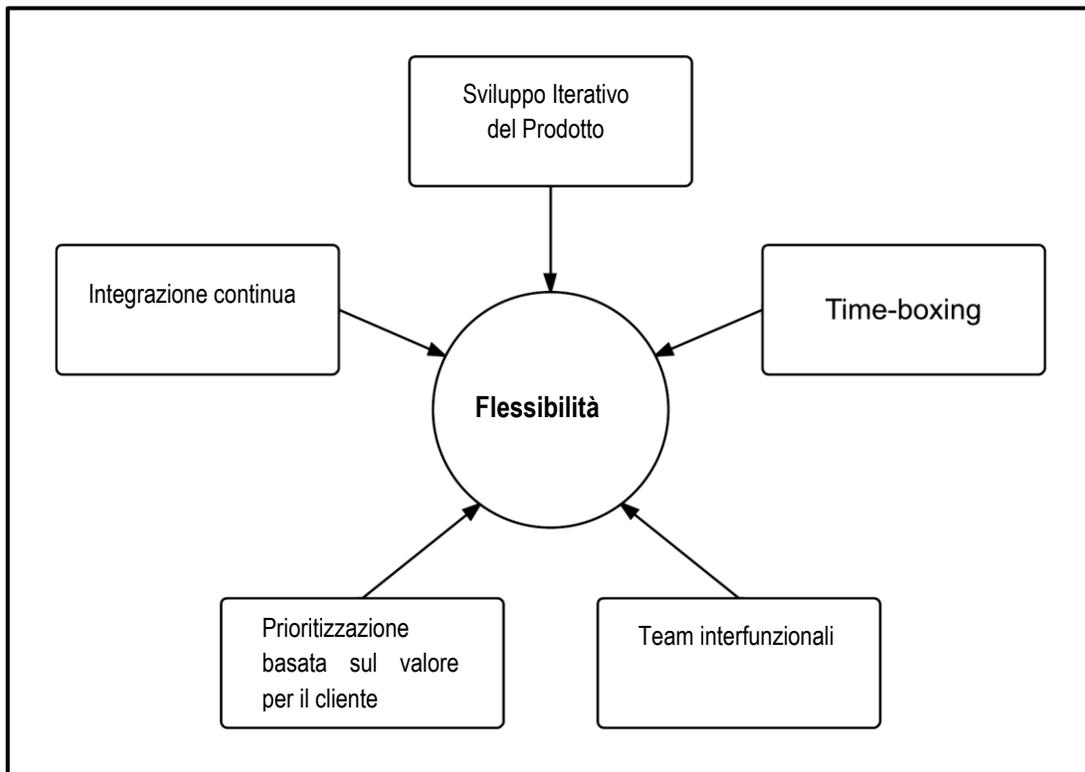


Figura 6-3: Le Caratteristiche di Scrum per Realizzare la Flessibilità

### 6.4.2.1 Flessibilità attraverso lo Sviluppo Iterativo del Prodotto

Scrum segue un approccio iterativo e incrementale allo sviluppo del prodotto e servizio, rendendo possibile l'incorporazione del cambiamento in qualunque fase del processo di sviluppo. Nel momento in cui si sviluppa il prodotto, una Richiesta di Modifica del progetto può provenire da molte fonti, che si elencano qui di seguito:

#### 1. Stakeholder

Gli stakeholder del progetto—in particolare gli sponsor, i clienti e gli utenti—possono presentare Richieste di Modifica in qualunque momento nel corso del progetto. Le Richieste di Modifica possono essere dovute a cambiamenti delle condizioni di mercato, alle direttive dell'organizzazione, a questioni legali o regolamentari, o a varie altre ragioni. Inoltre, gli stakeholder possono sottoporre Richieste di Modifica in occasione della revisione dei deliverable durante i processi *Illustrare e Convalidare lo Sprint*, *Retrospectiva dello Sprint*, *Retrospectiva del Progetto*. Tutte le Richieste di Modifica, una volta approvate, vengono aggiunte al Project Prioritized Product Backlog (chiamato anche Prioritized Product Backlog o Product Backlog). La Figura 6-4 illustra alcune delle ragioni per le quali gli stakeholder iniziano il processo di Richiesta di Modifica.

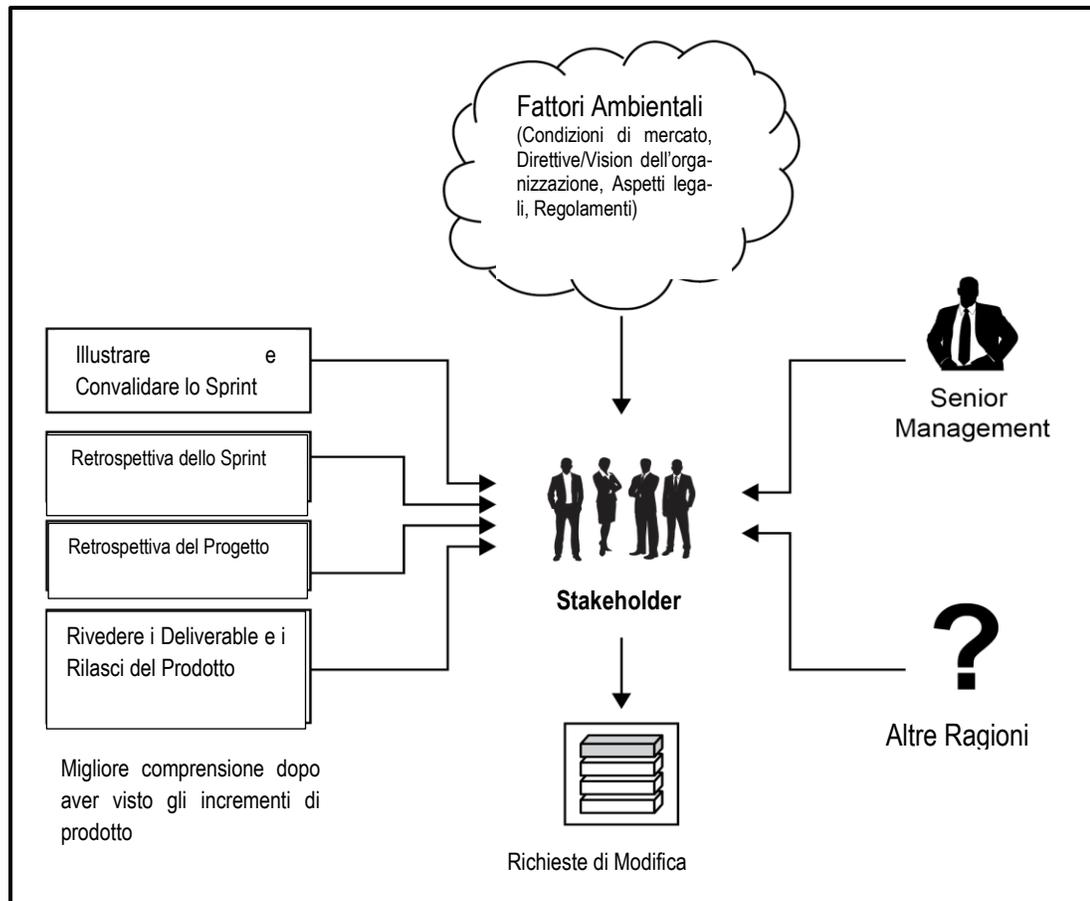


Figura 6-4: Motivazioni degli Stakeholder per presentare Richieste di Modifica

## 2. Scrum Core Team

Lo Scrum Core Team (vale a dire, il Product Owner, lo Scrum Master e lo Scrum Team) è nel suo insieme coinvolto nella creazione dei deliverable del prodotto. L'interazione continua fra i membri dello Scrum Core Team, sia al loro interno che con altri Scrum Team del progetto e con gli stakeholder interni ed esterni del progetto, può spingerli a suggerire cambiamenti o miglioramenti del prodotto, servizio, o altre parti del progetto. Di solito tali cambiamenti - come tutti gli altri - sono tradotti in Richieste di Modifica, ed è compito del Product Owner prendere la decisione finale riguardo a quali modifiche suggerite dallo Scrum Team e dallo Scrum Master si dovranno considerare come formali Richieste di Modifica.

A volte la creazione di alcuni Deliverable potrà presentare delle sfide, che potranno tradursi in Richieste di Modifica. Per esempio, il team può decidere di aggiungere o modificare una funzionalità per migliorare le prestazioni del prodotto. Nella maggioranza dei progetti Scrum, le raccomandazioni di cambiamenti da parte dello Scrum Core Team emergono quando gli Scrum Team lavorano al processo *Creare i Deliverable* o quando partecipano al *Daily Standup Meeting* o ai *Retrospect Sprint Meeting*. La Figura 6-5 illustra alcune delle ragioni per le quali lo Scrum Core Team può avviare delle Richieste di Modifica.

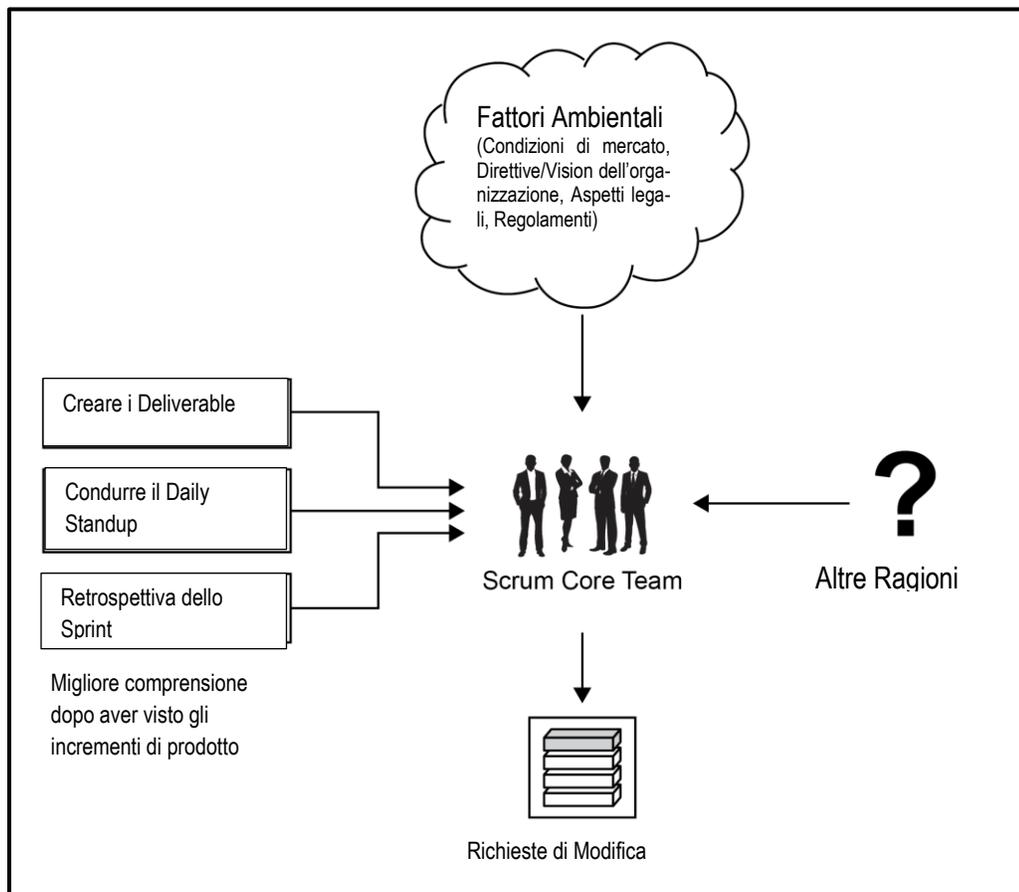


Figura 6-5: Le motivazioni dello Scrum Core Team per presentare Richieste di Modifiche

### 3. Senior Management

Il senior management—fra cui sono compresi il management del portfolio e del programma—può raccomandare cambiamenti che impattano sul progetto. Questo può avvenire a causa di cambiamenti delle direttive strategiche dell'azienda, del panorama concorrenziale, di problematiche di carattere finanziario, e così via. Da notare che questo tipo di cambiamenti viene aggiunto al Prioritized Product Backlog e deve passare per il normale processo di gestione dei cambiamenti. Se qualcuno di questi cambiamenti è urgente, lo Sprint che ne subisce il relativo impatto può dover essere interrotto (consultare la sezione 6.6 per ulteriori dettagli).

### 4. Scrum Guidance Body

Lo Scrum Guidance Body può presentare Richieste di Modifica che incidono su tutti i progetti a causa, per esempio, di una delle circostanze che seguono:

- Cambiamento dei regolamenti governativi (come privacy, standard di sicurezza, o nuove leggi)
- Direttive aziendali sulla qualità, la sicurezza, o altre iniziative dell'organizzazione che devono essere implementate in tutta l'azienda
- Parametri di riferimento o best practice che servono a soddisfare un determinato standard
- Lezioni apprese da precedenti progetti, che possono essere implementate da altri Scrum Team

Il tratto caratteristico di Scrum è la sua tolleranza e adattabilità al cambiamento. Scrum non promuove la strada della determinazione e della ferma definizione anticipata dei piani, in quanto opera sul presupposto che lo sviluppo del progetto è estremamente soggetto al cambiamento e al rischio. Il risultato è un elevato grado di flessibilità e tolleranza al cambiamento. Il progetto è pianificato, eseguito e gestito in modo incrementale, per cui di norma è facile incorporare il cambiamento strada facendo.

#### 6.4.2.2 Flessibilità attraverso il Time-boxing

Il Time-boxing si riferisce alla fissazione di brevi periodi di tempo per l'esecuzione del lavoro. Se il lavoro intrapreso non viene completato entro la fine del Time-box, viene trasferito in un altro Time-box. Esempi di Time-boxing sono la limitazione della durata dei Daily Standup Meeting a 15 minuti e la fissazione della durata degli Sprint entro un arco di tempo che va da una a sei settimane. I time-box rappresentano la struttura dei progetti Scrum, i quali hanno una componente di incertezza, sono dinamici per natura e sono soggetti a cambiamenti frequenti. I Time-box aiutano a valutare lo stato di avanzamento del progetto e permettono al team di individuare facilmente se c'è la necessità di modificare un processo o un approccio.

Gli Sprint di durata predeterminata (cioè time-boxed) contribuiscono notevolmente al rispetto delle scadenze e al raggiungimento di alti livelli di produttività. Gli Sprint favoriscono l'ordine e la coerenza in un ambiente di lavoro instabile e forniscono una piattaforma per valutare i risultati e ottenere il feedback in un breve arco di

tempo. Gli Sprint consentono inoltre una valutazione frequente dello stato di avanzamento e dei metodi utilizzati per gestire il progetto, ivi compresa un'efficace gestione dei cambiamenti. Gli errori o i problemi possono essere identificati precocemente e velocemente corretti.

Attraverso l'uso del Time-boxing per gli Sprint, il team rivede frequentemente il processo di stima del lavoro da compiere, per cui la proiezione del tempo e dell'impegno necessari diviene più accurata ad ogni successivo Sprint, man mano che il progetto avanza. Questi cicli iterativi motivano inoltre i membri del team al raggiungimento dei traguardi previsti e degli obiettivi incrementali, in vista della realizzazione dell'obiettivo più grande.

### 6.4.2.3 Flessibilità attraverso Team Interfunzionali e Auto-organizzati

La struttura interfunzionale e auto-organizzata dello Scrum Team permette ai membri del team di essere fortemente concentrati sui risultati desiderati per lo Sprint. Il team ha un insieme di obiettivi definiti per ogni Sprint e, nel contempo, la flessibilità per poter considerare un cambiamento degli obiettivi prima di iniziare lo Sprint successivo.

Inoltre l'uso di team interfunzionali assicura la presenza di tutte le competenze e le conoscenze necessarie per eseguire il lavoro del progetto all'interno dello stesso team. Questo offre un modello di lavoro efficiente, che si traduce nella creazione di deliverable potenzialmente consegnabili e pronti per la dimostrazione al Product Owner e/o agli altri stakeholder.

L'auto-organizzazione fa sì che i membri dello Scrum Team stabiliscano da soli *come* fare il lavoro del progetto, senza bisogno di un senior manager che controlla dettagliatamente le loro attività.

Essere un team interfunzionale e auto-organizzato permette al gruppo di adattarsi e gestire efficacemente il lavoro in corso e qualsiasi questione o cambiamento secondario senza dover ricorrere al supporto o all'esperienza di membri esterni al team e di creare, nell'apposito processo, deliverable che sono pronti per essere consegnati, se necessario.

### 6.4.2.4 Flessibilità attraverso la Priorizzazione basata sul Valore per il Cliente

La prioritizzazione dei requisiti e del lavoro in un progetto Scrum è sempre stabilita in base al valore fornito al cliente. In prima battuta, al momento della partenza di un progetto, i requisiti iniziali sono prioritizzati sulla base del valore che ciascun requisito fornirà – e questo viene documentato nel Prioritized Product Backlog. Successivamente, quando viene formulata una richiesta di includere un nuovo requisito o di modificarne uno esistente la si valuta durante il processo *Mettere a punto il Prioritized Product Backlog*. Se si ritiene che il cambiamento fornisca un valore maggiore di altri requisiti già esistenti, lo stesso verrà aggiunto al Prioritized Product Backlog aggiornato e prioritizzato di conseguenza. Pertanto, il Prioritized Product Backlog offre l'occasione di incorporare i cambiamenti e di aggiungere nuovi requisiti, se necessario.

È importante notare che i nuovi requisiti e i cambiamenti aggiunti al Prioritized Product Backlog possono abbassare la priorità di altre User Story già presenti nel Backlog: quindi, queste User Story con bassa priorità possono essere lavorate più tardi in base alla loro nuova prioritizzazione. Poiché i clienti sono coinvolti molto strettamente nella prioritizzazione dei requisiti e delle corrispondenti User Story del Prioritized Product Backlog, questa pratica assicura che i requisiti che i clienti ritengono “di alto valore” vengano completati per primi e che il progetto inizi a consegnare un valore significativo molto presto.

#### 6.4.2.5 Flessibilità attraverso l’Integrazione Continua

Utilizzando le tecniche di integrazione continua, i membri dello Scrum Team possono incorporare in qualsiasi momento funzionalità nuove e modificate all’interno dei deliverable. Questo riduce il rischio che molti membri del team si trovino ad apportare modifiche a componenti ridondanti (ad esempio, ad un codice obsoleto nei prodotti software, o a vecchi disegni di parti di produzione). In questo modo si garantisce che solo l’ultima funzionalità o versione sia oggetto di lavorazione e si evitano problemi di compatibilità.

## 6.5 Integrare il Cambiamento

A seconda del settore industriale e del tipo di progetto, la priorità delle funzionalità e dei requisiti di un progetto può rimanere invariata per un lasso di tempo significativo oppure può cambiare frequentemente. Se i requisiti del progetto sono per lo più stabili, nel corso dello sviluppo ci saranno di norma solo cambiamenti secondari al Prioritized Product Backlog e gli Scrum Team possono lavorare al completamento in sequenza dei requisiti che forniranno il massimo valore al cliente, così come prioritizzati nel Prioritized Product Backlog. In questo tipo di ambienti stabili, la lunghezza dello Sprint è di solito maggiore, dalle 4 alle 6 settimane.

Se i requisiti del progetto cambiano nel corso del progetto stesso, per esempio a causa del cambiamento dei requisiti di business, lo stesso metodo è ugualmente efficace. Prima di iniziare uno Sprint - durante i processi *Creare il Prioritized Product Backlog* o *Mettere a punto il Prioritized Product Backlog* - i requisiti che nel Prioritized Product Backlog hanno una priorità più alta sono di solito scelti per essere completati in quello Sprint. Siccome nel Prioritized Product Backlog si è tenuto conto dei cambiamenti, il team deve solo stabilire quante attività può compiere nello Sprint sulla base del tempo e delle risorse a disposizione. La gestione dei cambiamenti viene eseguita attraverso i processi continui di prioritizzazione e di addizione delle attività al Prioritized Product Backlog.

## 6.5.1 Modifiche ad uno Sprint

Se c'è una Richiesta di Modifica che può avere un impatto significativo su uno Sprint in corso, il Product Owner, dopo essersi consultato con gli Stakeholder del caso, decide se la modifica può aspettare fino al prossimo Sprint oppure se la stessa rappresenta una situazione urgente che può richiedere la chiusura immediata dello Sprint attuale e l'avvio di uno nuovo.

Il framework Scrum specifica chiaramente che l'ambito di uno Sprint non può essere modificato una volta che lo Sprint ha inizio. Se il cambiamento richiesto è così importante da rendere inutili i risultati dello Sprint in sua assenza, allora lo Sprint va interrotto. In tutti gli altri casi, il cambiamento viene incorporato in uno Sprint successivo (come mostrato nella Figura 6-6).

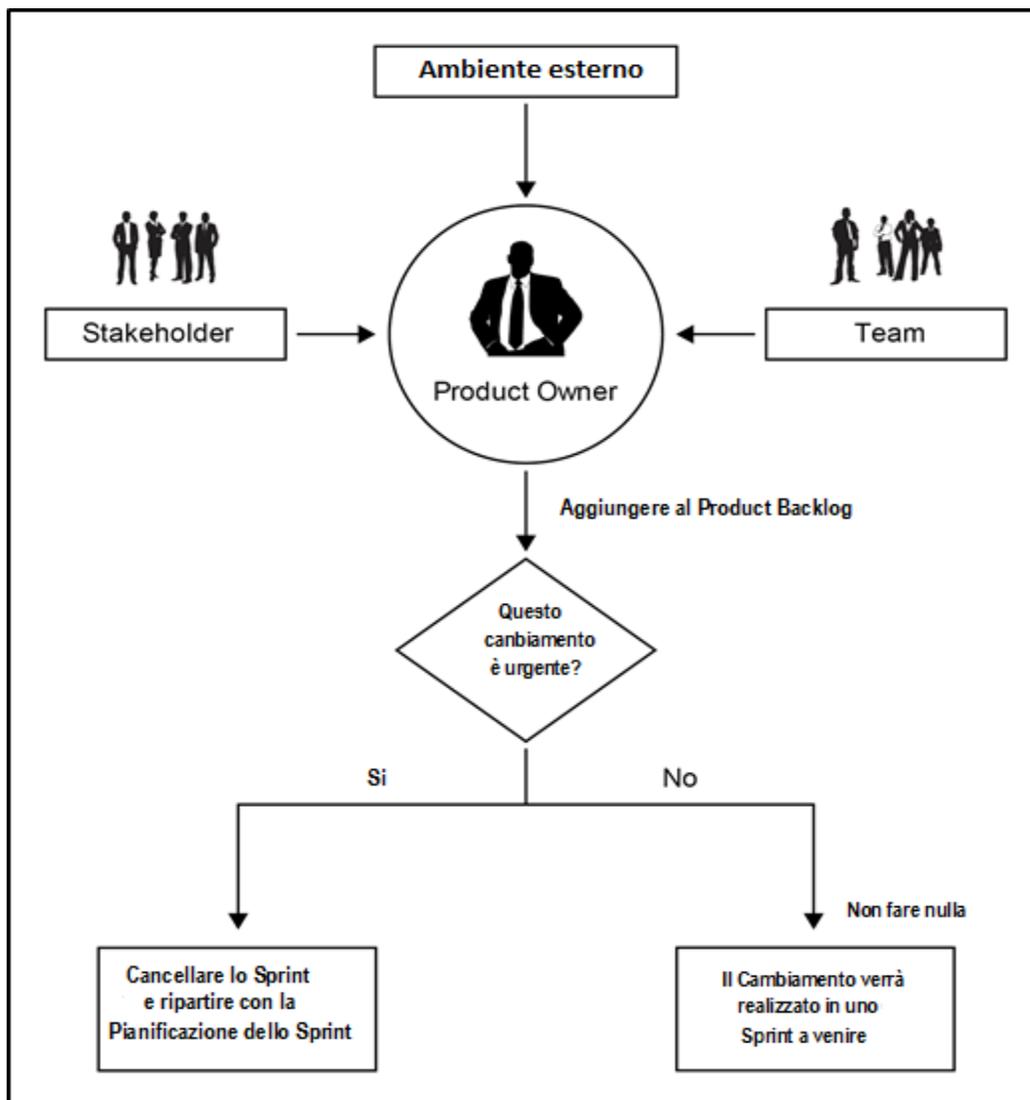


Figura 6-6: L'integrazione del Cambiamento in Scrum

C'è una sola eccezione alla regola di non cambiare l'ambito di uno Sprint una volta che questo è partito. Se lo Scrum Team durante lo Sprint si accorge di aver fortemente sovrastimato l'impegno ed ha una capacità di riserva per poter implementare ulteriori User Story, può chiedere al Product Owner quali User Story dovrebbero essere aggiunte allo Sprint in corso.

Bloccando l'ambito di ogni Sprint, il team è in grado di ottimizzare e di gestire il proprio lavoro e il proprio sforzo in modo efficiente. Un ulteriore beneficio è dato dal fatto che il team non si deve preoccupare di gestire i cambiamenti una volta che ha iniziato a lavorare ad uno Sprint. Questo è un grande vantaggio del framework Scrum rispetto al project management tradizionale.

Nel project management tradizionale, i cambiamenti possono essere richiesti e approvati in qualunque momento durante il ciclo di vita del progetto. Questo fatto crea spesso confusione per i membri del team di progetto, diminuisce la motivazione del team a causa della discontinuità e, da ultimo, si traduce in una mancanza di concentrazione e nella sensazione da parte del team che “non venga mai fatto nulla”. Al contrario, nei progetti Scrum, una volta che parte uno Sprint non sono consentiti cambiamenti. Questo assicura che in ogni Sprint il team completi i deliverable e le attività siano portate a termine. Per di più, il business riconosce l'esistenza di benefici tangibili dai Deliverable potenzialmente consegnabili alla fine di ciascuno Sprint.

Ed ancora, essendo consapevoli del fatto che non sono permesse modifiche una volta che uno Sprint prende il via e che uno Sprint dura da 1 a 6 settimane, il Product Owner e gli Stakeholder definiscono e prioritizzano i requisiti durante gli appositi processi *Creare le Epic*, *Creare il Prioritized Product Backlog* e *Mettere a punto il Prioritized Product Backlog*.

### 6.5.1.1 Impatto dei Cambiamenti Attesi sulla Lunghezza dello Sprint

Poiché durante uno Sprint non sono permessi cambiamenti, l'impatto e la frequenza dei cambiamenti attesi può avere una ripercussione sulla decisione relativa alla lunghezza dello Sprint presa durante il processo *Eseguire la Pianificazione dei Rilasci*.

Se i requisiti del progetto sono per lo più stabili e non ci si aspettano cambiamenti importanti nel futuro prossimo, si può stabilire una Lunghezza dello Sprint maggiore, dalle 4 alle 6 settimane. Questo dà ai membri dello Scrum Team la stabilità di lavorare ai requisiti del Prioritized Product Backlog per lunghi periodi di tempo, senza dover passare per i processi *Creare le User Story*, *Approvare*, *Stimare* e *Prendere in Carico le User Story*, *Creare le Attività*, *Stimare le Attività* e altri processi collegati che si eseguono per ogni Sprint.

Tuttavia, se i requisiti del progetto non sono definiti molto bene o se ci si attendono cambiamenti significativi nell'immediato futuro, la Lunghezza dello Sprint può essere relativamente più corta, da 1 a 3 settimane. Questo fornisce allo Scrum Team sufficiente stabilità per poter lavorare a Sprint più brevi e consegnare risultati che possono essere valutati dal Product Owner e dagli stakeholder alla fine dello Sprint, ma al tempo stesso anche la flessibilità per poter chiarire i requisiti e apportare i cambiamenti al Prioritized Product Backlog alla fine di ogni Sprint.

Per ottenere il massimo dei benefici da un progetto Scrum, si raccomanda sempre di mantenere la durata dello Sprint fissata in 4 settimane, salvo il caso di progetti con requisiti molto stabili, nei quali gli Sprint possono durare fino a 6 settimane.

La Figura 6-7 illustra l'impatto del cambiamento atteso sulla Lunghezza dello Sprint.

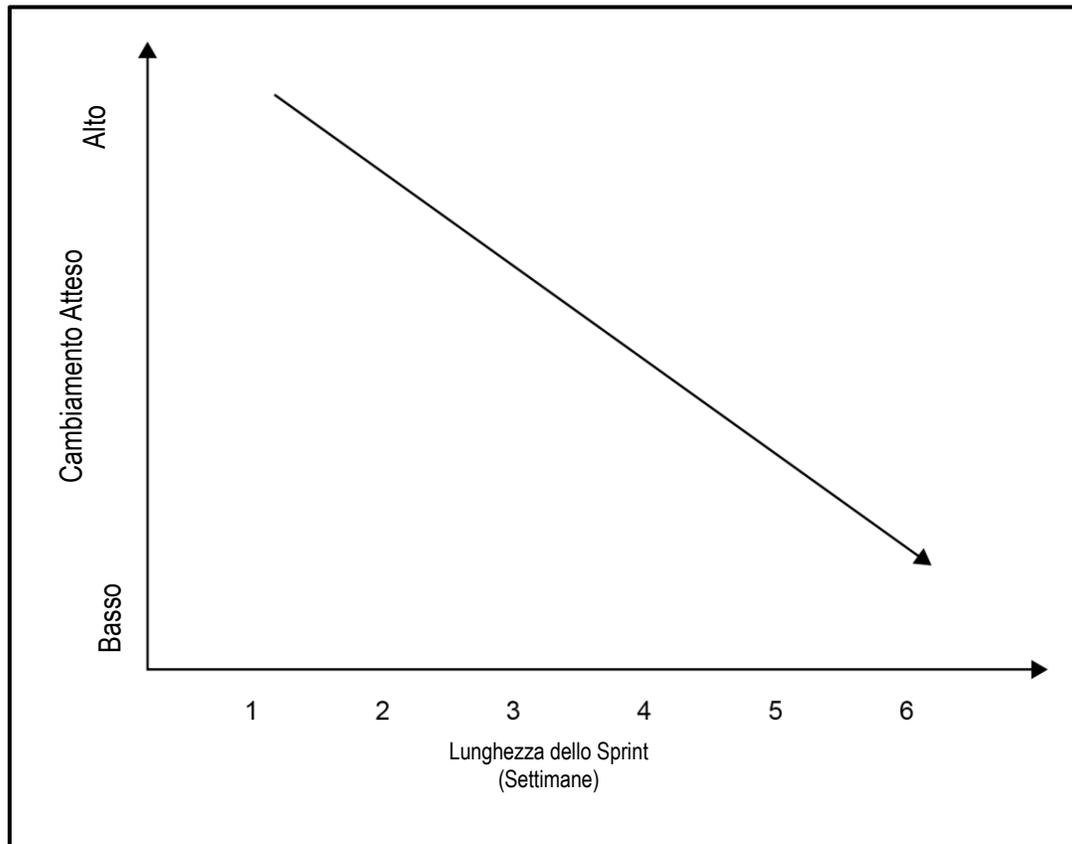


Figura 6-7: Impatto del Cambiamento Atteso sulla Lunghezza dello Sprint

Tuttavia, è importante notare che il cambiamento atteso non rappresenta l'unico fattore considerato per stabilire la Lunghezza dello Sprint. Altri fattori che devono essere anch'essi tenuti presenti sono:

- Il tempo effettivo per portare a compimento il lavoro (se il progetto o il contesto aziendale richiedono uno specifico tempo per compiere le attività, questo tempo potrebbe determinare la Lunghezza dello Sprint)
- La data pianificata per un rilascio (la Lunghezza dello Sprint deve tenere in considerazione le date dei rilasci del complessivo prodotto o servizio)
- Qualunque altro fattore stabilito dal Product Owner o dallo Scrum Master che deve essere considerato nello stabilire la Lunghezza dello Sprint

È importante notare che il cambiamento della Lunghezza dello Sprint non deve essere deciso alla leggera o periodicamente (per esempio non è consigliabile fissare la lunghezza dello sprint in 3 settimane per questo sprint, 2 settimane per il prossimo, 4 per il terzo, ecc.). La Lunghezza dello Sprint dovrebbe essere preferibilmente coerente. Uno degli impatti più grandi del cambiamento della Lunghezza dello Sprint è che causa un reset di tutto il monitoraggio a livello di progetto. Le velocità precedenti possono diventare inutili per le previsioni e la pianificazione degli Sprint futuri. Senza una velocità accurata (che rappresenta un metrica fondamentale per qualsiasi progetto Scrum), non è possibile misurare l'efficacia dello Scrum Team né scegliere adeguatamente il numero di User Story da prendere in carico quando si pianifica lo Sprint successivo. Pertanto, quando si è decisa la Lunghezza dello Sprint, è preferibile mantenerla costante per tutta la durata del progetto o almeno per molti cicli di Sprint.

### 6.5.1.2 Gestire i Cambiamenti attraverso la Messa a Punto del Prioritized Product Backlog

6

Il Prioritized Product Backlog conterrà normalmente tutte le User Story, le loro stime temporali (comprese le eventuali revisioni) e la situazione dei requisiti a priorità più alta. Viene inoltre inserita qualsiasi User Story nuova o rivista scaturente dal cambiamento dei requisiti di business, dalle richieste del cliente, dalle condizioni esterne del mercato, e/o da lezioni apprese dagli Sprint precedenti.

Una delle responsabilità fondamentali del Product Owner è la messa a punto del Prioritized Product Backlog per assicurare che i requisiti del Prioritized Product Backlog prioritizzati per l'inclusione nei successivi due o tre Sprint siano tradotti in User Story adeguate. È raccomandato che il Product Owner dedichi una quantità di tempo significativa alla messa a punto del Prioritized Product Backlog in ciascuno Sprint. Il Product Owner ha la responsabilità di aggiungere e rivedere gli elementi del Prioritized Product Backlog in risposta a eventuali cambiamenti e di arricchire di dettagli le User Story che saranno utilizzate nello Sprint a seguire.

Questo lavoro di messa a punto contribuisce ad assicurare che la rifinitura dei requisiti e delle relative User Story sia eseguita prima dello Sprint Planning Meeting, in modo tale che il team abbia a disposizione un insieme di story ben analizzate e chiaramente definite che possono essere facilmente scomposte in attività e successivamente stimate. Sulla base delle lezioni apprese nello Sprint corrente, ci potrebbero essere dei cambiamenti di requisiti oppure delle riprioritizzazioni che possono essere entrambi facilmente incorporati negli Sprint successivi. Il lavoro di messa a punto supporta e aumenta la flessibilità del modello Scrum grazie alla possibilità di incorporare le più recenti idee tecniche e di business negli Sprint a venire.

Un Product Backlog Review Meeting (chiamato anche Sessione di Messa a Punto del Prioritized Product Backlog) è una riunione formale che si tiene durante il processo *Mettere a Punto il Prioritized Product Backlog* e che aiuta lo Scrum Team a rivedere il Prioritized Product Backlog e a ottenere il consenso sullo stesso. Tuttavia, oltre che nel corso del Prioritized Product Backlog Review Meeting, la messa a punto del Prioritized Product Backlog si deve realizzare nel corso di tutto il progetto e può riguardare situazioni in cui il Product Owner scrive nuove User Story o riprioritizza quelle già esistenti nel Prioritized Product Backlog, oppure i membri dello Scrum Team o gli Stakeholder danno i loro suggerimenti al Product Owner circa nuove User Story, e così via.

È importante notare che qualsiasi elemento del Prioritized Product Backlog è sempre passibile di una nuova stima fino a quando non viene ultimato lo Sprint Backlog nel processo *Creare lo Sprint Backlog*. Dopo tale momento, si possono continuare a fare cambiamenti fino allo Sprint Planning Meeting, se necessario.

#### **6.5.1.2.1 Come rendere efficace il Product Backlog Review Meeting (o Sessione di Messa a Punto del Prioritized Product Backlog)**

Il Product Owner prende il comando del Product Backlog Review Meeting condotto durante il processo *Mettere a Punto il Prioritized Product Backlog*. È importante che il Product Owner fissi gli obiettivi e sviluppi idealmente un'agenda prima dell'inizio del Product Backlog Review Meeting. In mancanza, la sessione sarà priva di struttura e potrebbe dimostrarsi improduttiva. È importante inoltre limitare il numero degli stakeholder che partecipano alla riunione. La presenza di troppi partecipanti tende a far diminuire la complessiva efficienza della riunione. Il Product Owner dovrebbe invitare solo quegli stakeholder il cui feedback è necessario per la sessione di messa a punto. Dovrebbero invece essere inclusi tutti i membri dello Scrum Team, in quanto il loro input riguardo al lavoro che stanno svolgendo e alle eventuali problematiche incontrate è prezioso. Se la sessione di messa a punto porta a qualche riprioritizzazione o cambiamento del Prioritized Product Backlog, è importante che il team sia d'accordo con tali cambiamenti.

Una sessione efficace di messa a punto dovrebbe tradursi in elementi del Prioritized Product Backlog (PBI) chiaramente definiti, in modo che lo Scrum Team abbia ben chiaro quali sono i requisiti del cliente. Questo aiuta inoltre il team a prendere familiarità con tutte le User Story nel caso in cui una o più di queste debba essere inclusa in uno Sprint con poco preavviso. Durante le sessioni di messa a punto possono essere discussi anche i Criteri di Accettazione e i Done Criteria.

Scrum non stabilisce una durata predefinita per gli esercizi di messa a punto. La messa a punto del Prioritized Product Backlog rappresenta per il Product Owner un'attività continua.

#### **6.5.1.3 Gestire i Cambiamenti durante il processo Illustrare e Convalidare lo Sprint**

Anche se il Product Owner ha l'ultima parola sugli elementi del Prioritized Product Backlog e sull'accettazione o il rifiuto delle User Story (corrispondenti a Richieste di Modifica Approvate) presentate durante il processo *Illustrare e Convalidare lo Sprint*, è responsabilità dello Scrum Master assicurarsi che i requisiti e i Criteri di Accettazione delle User Story dello Sprint in corso non vengano cambiati durante lo Sprint Review Meeting. Questo previene il rifiuto di User Story completate dovuto al fatto che non soddisfano requisiti appena modificati. Se è necessario cambiare qualche requisito, i corrispondenti PBI (elementi del Prioritized Product Backlog) devono essere rivisti per tenere conto dei requisiti modificati in uno Sprint a venire.

## 6.6 Il Cambiamento nei Portfolio e nei Programmi

Qualunque cambiamento che emerge nei programmi o nei portfolio può avere un effetto a cascata su tutti i progetti e gli Sprint dipendenti. Pertanto, è consigliabile ridurre al minimo i cambiamenti a questi livelli più alti. Se è necessario un cambiamento e tutti gli stakeholder sono d'accordo nell'apportare il cambiamento a questi livelli, si dovrebbero tenere presenti i seguenti concetti.

### 6.6.1 Nel Portfolio

1. È sconsigliato fare dei cambiamenti fra un Portfolio Backlog Meeting e quello successivo.
2. Se il cambiamento è secondario, il Portfolio Product Owner dovrebbe assicurarsi l'approvazione degli opportuni stakeholder (ad esempio, sponsor, cliente e utenti finali) e poi aggiungere i requisiti al Portfolio Backlog. I Product Owner del programma e del progetto includeranno quei requisiti negli Sprint a venire.
3. Se il cambiamento è importante, gli impegni del portfolio e di tutti i programmi, i progetti e gli Sprint associati devono essere fermati e si dovrà tenere un Portfolio Backlog Meeting per stabilire i passi successivi.
4. I Portfolio Prioritized Product Backlog Meeting (chiamati anche semplicemente Portfolio Backlog Meeting), dovrebbero essere tenuti a intervalli di 4-12 mesi. La frequenza e l'impatto delle modifiche a un portfolio determinano in gran parte lo spazio di tempo che intercorre fra due Portfolio Backlog Meeting. Se nel portfolio ci si attendono molti cambiamenti, è preferibile tenere i Portfolio Backlog Meeting a intervalli più regolari (ad esempio, 4 - 6 mesi); ma se i cambiamenti attesi sono pochi e i requisiti sono stabili, l'intervallo fra due Portfolio Backlog Meeting può essere aumentato (ad esempio, dai 9 ai 12 mesi).

### 6.6.2 Nel Programma

1. È sconsigliato fare dei cambiamenti fra un Program Backlog Meeting e quello successivo.
2. Se il cambiamento è secondario, il Program Product Owner dovrebbe assicurarsi l'approvazione degli opportuni stakeholder (ad esempio, sponsor, cliente e utenti finali) e del Portfolio Product Owner e poi aggiungere i requisiti al Program Backlog. I Product Owner del progetto includeranno quei requisiti negli Sprint a venire.
3. Se il cambiamento è importante, gli impegni del programma e di tutti i progetti e Sprint associati devono essere fermati e si dovrà tenere un Prioritized Product Backlog Meeting per stabilire i passi successivi.
4. I Program Prioritized Product Backlog Meeting (chiamati anche semplicemente Program Backlog Meeting), dovrebbero essere tenuti a intervalli di 2-6 mesi. La frequenza e l'impatto delle modifiche a un programma determinano in gran parte lo spazio di tempo che intercorre fra due Program Backlog Meeting. Se ci si attendono molti cambiamenti nel programma, è preferibile tenere i

Program Backlog Meeting a intervalli più regolari (ad esempio, da 2 a 3 mesi); ma se i cambiamenti attesi sono pochi e i requisiti sono stabili, l'intervallo fra due Program Backlog Meeting può essere aumentato (ad esempio, dai 5 ai 6 mesi).

La Figura 6-8 illustra come gestire i cambiamenti all'interno del flusso Scrum, sia per i portfolio che per i programmi.

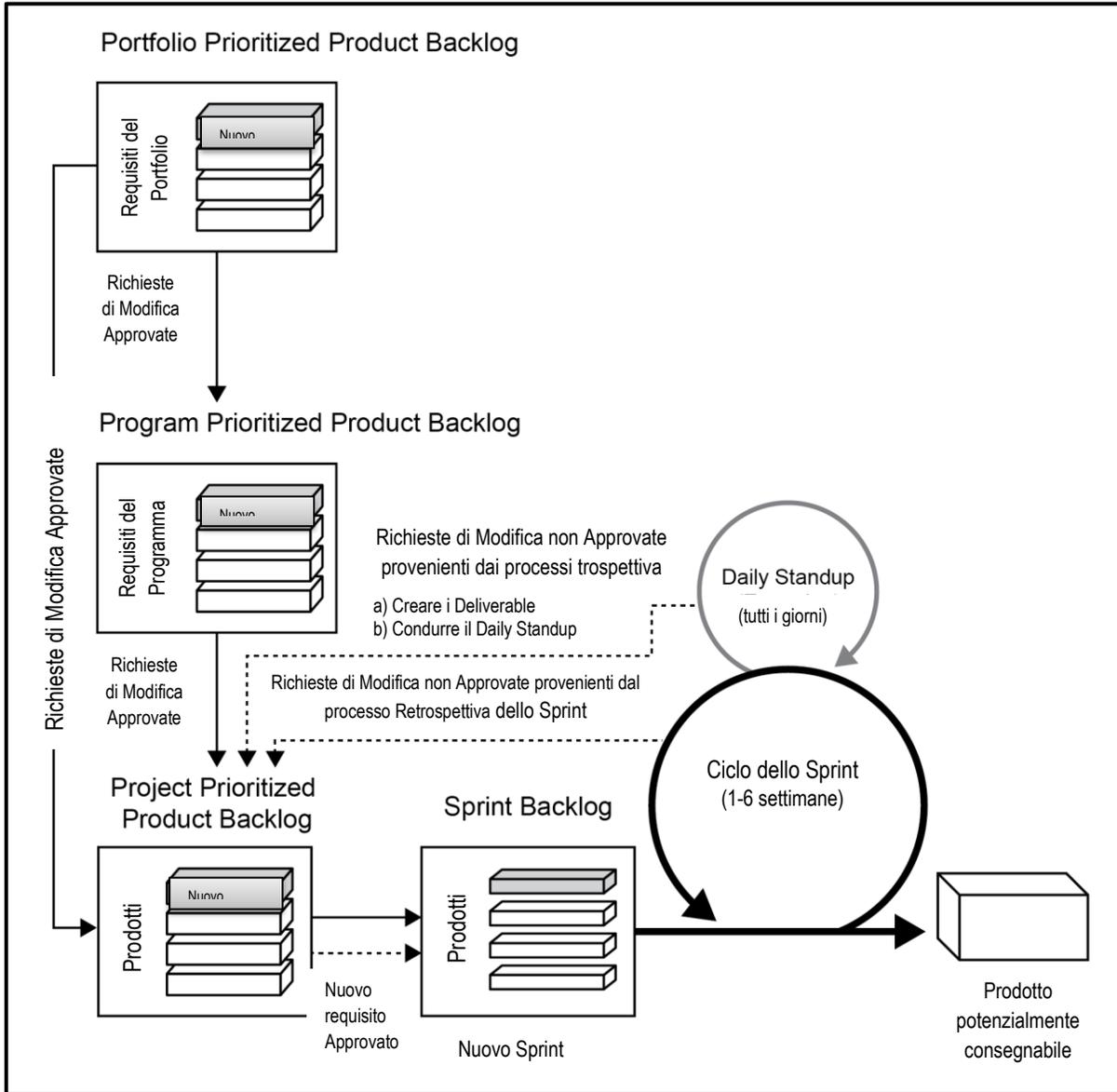


Figura 6-8: Incorporare i Cambiamenti nel Portfolio e nel Programma

## 6.7 Riepilogo delle Responsabilità

Ruolo	Responsabilità
Scrum Guidance Body	<ul style="list-style-type: none"> <li>Fornisce l'indirizzo complessivo sulle procedure di gestione dei cambiamenti da seguire nel corso del progetto</li> </ul>
Portfolio Product Owner	<ul style="list-style-type: none"> <li>Sottopone Richieste di Modifica a livello di portfolio</li> <li>Approva la modifica, rimozione e aggiunta dei prodotti conformemente ai requisiti del portfolio</li> </ul>
Portfolio Scrum Master	<ul style="list-style-type: none"> <li>Facilita l'identificazione, la valutazione e la gestione delle Richieste di Modifica a livello di portfolio</li> </ul>
Program Product Owner	<ul style="list-style-type: none"> <li>Sottopone richieste di modifiche a livello di programma</li> <li>Approva la modifica, rimozione e aggiunta dei prodotti conformemente ai requisiti del programma</li> </ul>
Program Scrum Master	<ul style="list-style-type: none"> <li>Facilita l'identificazione, la valutazione e la gestione delle Richieste di Modifica a livello di programma</li> </ul>
Stakeholder	<ul style="list-style-type: none"> <li>Sottopongono richieste di modifiche</li> <li>Sono coinvolti nell'approvazione e prioritizzazione delle Richieste di Modifica</li> </ul>
Product Owner	<ul style="list-style-type: none"> <li>Sottopone richieste di modifiche relative a un progetto</li> <li>Valuta l'impatto delle richieste di modifiche emerse a livello di portfolio, di programma o di progetto</li> <li>Prioritizza le User Story inserite nel Prioritized Product Backlog del progetto</li> <li>Valuta l'impatto sugli obiettivi del progetto dei problemi identificati dallo Scrum Team</li> <li>Comunica in modo chiaro agli stakeholder la ripriorizzazione degli elementi del Product Backlog</li> </ul>
Scrum Master	<ul style="list-style-type: none"> <li>Facilita l'identificazione, la valutazione e l'escalation dei problemi e delle Richieste di Modifica da parte dello Scrum Team</li> </ul>
Scrum Team	<ul style="list-style-type: none"> <li>Suggerisce miglioramenti o cambiamenti durante i processi <i>Creare i Deliverable</i> e <i>Condurre il Daily Standup</i></li> </ul>

Tavola 6-1: Riepilogo delle Responsabilità Collegate al Cambiamento

## 6.8 Confronto fra Scrum e il Project Management Tradizionale

La gestione del cambiamento nei progetti gestiti tradizionalmente è strettamente collegata alla Gestione della Configurazione. Tutti i cambiamenti sono presi in considerazione in base all'importanza della loro deviazione rispetto a un valore baseline. Al Project Manager vengono assegnate delle tolleranze all'interno delle quali lui o lei può gestire le attività quotidiane e le decisioni del progetto. Quando una Richiesta di Modifica eccede le tolleranze stabilite, il Project Manager ha l'obbligo di portare il cambiamento proposto all'attenzione dei livelli di management più alti e attendere la loro decisione prima di implementare il cambiamento. Il Project Manager per prima cosa inserisce la richiesta di modifica in un Registro delle Questioni o Registro dei Cambiamenti e poi sottopone il cambiamento all'attenzione delle autorità superiori. Tra queste può annoverarsi lo sponsor del progetto, oltre ad altri stakeholder e figure decisionali pertinenti. Ad un certo punto, verrà eseguita una valutazione dell'impatto. Sulla base dell'impatto stimato di quel cambiamento, verrà presa una decisione se implementare o meno il cambiamento. Il Project Manager può anche proporre possibili soluzioni ad eventuali problemi posti dal cambiamento. Se le autorità superiori decidono di procedere all'esecuzione della modifica, il Project Manager ha la responsabilità di assicurarne la corretta implementazione.

In Scrum il cambiamento funziona in modo molto diverso rispetto al Project Management Tradizionale. Il framework Scrum è fortemente sintonizzato sulla gestione efficace ed efficiente dei cambiamenti. Tutte le volte che il Product Owner o lo Scrum Team si accorgono di un problema o difetto o identificano un elemento del Prioritized Product Backlog che deve essere modificato, sostituito o aggiunto, si fa luogo al cambiamento del Prioritized Product Backlog. Analogamente, il senior management, il Product Owner o gli Stakeholder possono aggiungere Richieste di Modifica al Prioritized Product Backlog. Il Product Owner e gli Stakeholder approvano le Richieste di Modifica e riprioritizzano il Backlog di conseguenza. Ogni qual volta vi sia un problema o un nuovo requisito che deve essere trattato immediatamente e che impone un cambiamento che incide sullo Sprint in corso, il Product Owner termina anticipatamente lo Sprint, con l'approvazione degli opportuni stakeholder. Una volta interrotto, lo Sprint sarà ripianificato e fatto ripartire per potervi incorporare i nuovi requisiti.

Tuttavia, se il problema o la modifica non è importante e non giustifica un cambiamento all'interno dello Sprint corrente, il cambiamento verrà aggiunto al Prioritized Product Backlog e incorporato nella pianificazione di uno Sprint successivo. Questo dà agli stakeholder la capacità di rispondere ai cambiamenti dell'ambiente esterno, mantenendo nel contempo un certo grado di controllo sulle attività in corso all'interno del progetto. Inoltre, alla fine di ogni Sprint, i deliverable completati (Done) sono illustrati dallo Scrum Team. Questi deliverable sono potenzialmente consegnabili e possono essere rivisti dal Product Owner e dagli altri stakeholder.

## 7. RISCHIO

### 7.1 Introduzione

Lo scopo di questo capitolo è quello di definire il rischio, discutere della gestione dei rischi in un contesto Scrum e infine prendere in esame gli strumenti che agevolano la gestione dei rischi. Per garantire la fattibilità commerciale, ridurre la probabilità di fallimento del progetto, e prendere decisioni di business più informate, è importante gestire i rischi in maniera efficace attraverso un approccio ben organizzato e metodico.

In un contesto Scrum, i rischi sono generalmente minimizzati, grazie soprattutto al fatto che il lavoro viene realizzato all'interno di Sprint per mezzo dei quali viene prodotta in cicli molto brevi una serie continua di Deliverable, questi vengono poi confrontati con le aspettative e il Product Owner è attivamente coinvolto nel progetto. Tuttavia, anche nel più semplice dei progetti, le cose possono andare male, per cui è importante avere una strategia per identificare e trattare i rischi.

Il *Rischio*, così come definito nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Questo capitolo si divide nelle seguenti sezioni:

**7.2 Guida per i Ruoli** — Questa sezione fornisce un'indicazione su quali sezioni sono importanti per ciascuno dei principali ruoli di Scrum: Product Owner, Scrum Master e Scrum Team.

**7.3 Che cos'è il Rischio?**—Questa sezione definisce il rischio e spiega come lo stesso può incidere sugli obiettivi di un progetto e contribuire al successo o al fallimento di un progetto.

**7.4 Procedura di Gestione del Rischio**—Questa sezione illustra le tecniche fondamentali di gestione del rischio e approfondisce lo sviluppo delle strategie di identificazione, valutazione e gestione dei rischi.

**7.5 Minimizzare il Rischio attraverso Scrum**—Questa sezione spiega gli aspetti chiave di Scrum che ne fanno una struttura di gestione ideale per trattare i rischi in modo efficace a vari livelli - portfolio, programma e progetto.

**7.6 Riepilogo delle Responsabilità**—Questa sezione descrive le responsabilità attribuite in un progetto a ciascuna persona o ruolo relativamente alla gestione del rischio.

**7.7 Confronto fra Scrum e il Project Management Tradizionale**—Questa sezione tratta dei benefici ottenibili dalla gestione del rischio con i metodi Scrum rispetto ai metodi utilizzati nei modelli tradizionali di project management.

## 7.2 Guida per i Ruoli

1. **Product Owner**—Le responsabilità principali della gestione dei rischi in un progetto fanno capo al Product Owner; pertanto lo studio dell'intero capitolo è consigliato soprattutto a questo ruolo.
2. **Scrum Master**—Lo Scrum Master dovrebbe conoscere l'intero capitolo, con particolare attenzione alle sezioni 7.3, 7.4 e 7.7.
3. **Scrum Team**—Lo Scrum Team dovrebbe concentrarsi principalmente sulle sezioni 7.3 e 7.7.

## 7.3 Che cos'è il Rischio?

Il Rischio viene definito come un evento incerto che può incidere sugli obiettivi del progetto e può contribuire al suo successo o al suo fallimento. I rischi con un potenziale impatto positivo sul progetto si definiscono opportunità, mentre le minacce sono rischi che possono avere un impatto negativo sul progetto. La gestione del rischio deve essere eseguita in maniera proattiva ed è un processo iterativo che dovrebbe iniziare alla partenza del progetto e continuare per tutta la vita de progetto. Il processo di gestione del rischio deve seguire alcuni passi standardizzati per assicurare l'identificazione e la valutazione dei rischi, nonché la determinazione e la conseguente messa in pratica di un adeguato piano di risposta.

L'identificazione, la valutazione e la risposta ai rischi si devono basare principalmente su due fattori: la probabilità di un evento e il probabile impatto in caso di suo accadimento. I rischi con una valutazione di alta probabilità e di alto impatto devono essere trattati prima di quelli con una valutazione più bassa. In genere, una volta che un rischio è identificato, è importante capire gli aspetti di base del rischio per quanto riguarda le possibili cause, l'area di incertezza e i potenziali effetti causati dal materializzarsi del rischio.

### 7.3.1 Differenza fra Rischi e Questioni

I rischi sono le incertezze collegate ad un progetto che ne possono alterare significativamente il risultato in senso positivo o negativo. Dal momento che i rischi sono eventi futuri e incerti, non hanno alcun impatto attuale sul progetto, ma possono averne uno potenziale nel futuro. Quelli che seguono sono alcuni esempi di rischio:

- Anche dopo una formazione completa, i rappresentanti del servizio clienti potrebbero non essere pronti a ricevere gli ordini alla data di lancio.
- La squadra dei pittori potrebbe essere ritardata dalla forte pioggia, che potrebbe avere un impatto negativo sulla schedulazione del progetto.

Le questioni sono in genere eventi certi e ben definiti che stanno accadendo attualmente nel progetto, per cui non c'è alcun bisogno di eseguire una valutazione della probabilità come per un rischio. Le questioni devono essere affrontate. Quelli che seguono sono alcuni esempi di questioni:

- Il finanziamento non è approvato.
- I requisiti non sono chiari.

I rischi, se non trattati in tempo, possono trasformarsi in questioni. L'obiettivo della gestione del rischio è quello di essere pronti, con appositi piani, ad affrontare qualsiasi rischio possa verificarsi.

### 7.3.2 Attitudine al Rischio

Gli stakeholder comprendono tutte le persone e le organizzazioni impattate dal progetto oltre a coloro che possono incidere sul progetto. È importante capire l'atteggiamento degli stakeholder nei confronti del rischio. L'attitudine al rischio è influenzata dai seguenti tre fattori:

1. Propensione al rischio: si riferisce alla quantità di incertezza che gli stakeholder o l'organizzazione sono disposti ad assumersi.
2. Tolleranza al rischio: indica il grado, la quantità o il volume di rischio che gli stakeholder saranno disposti a sopportare.
3. Soglia del rischio: si riferisce al livello di rischio accettabile per l'organizzazione degli stakeholder. Un rischio ricadrà sopra o sotto la soglia di rischio. Se è sotto, allora lo stakeholder o l'organizzazione sono più propensi ad accettare il rischio.

In sostanza, l'atteggiamento degli stakeholder verso il rischio determina la quantità di rischio che gli stakeholder considerano accettabile e quindi quando decideranno di intraprendere azioni per mitigare i potenziali impatti sfavorevoli dei rischi. Pertanto, è importante capire il livello di tolleranza degli stakeholder riguardo ai vari fattori, fra cui costi, qualità, ambito e tempi.

L'*Utility Function* è un modello utilizzato per misurare la preferenza di rischio degli stakeholder o il loro atteggiamento nei confronti del rischio. Tale modello definisce il livello o disponibilità degli stakeholder ad accettare il rischio. Le tre categorie di Utility Function sono le seguenti:

1. Avverso al rischio: Lo stakeholder non è disposto ad accettare il rischio, qualunque sia il beneficio previsto o l'opportunità.
2. Neutrale al rischio: Lo stakeholder non è né avverso al rischio né alla ricerca del rischio e qualsiasi decisione data non è influenzata dal livello di incertezza dei risultati.

Quando due possibili scenari portano con sé lo stesso livello di beneficio, lo stakeholder neutrale al rischio non sarà interessato al se uno scenario sia più rischioso dell'altro.

3. Cercatore del rischio: Lo stakeholder è disponibile ad accettare il rischio anche se realizza un aumento del ritorno o del beneficio del progetto del tutto marginale.

## 7.4 Procedura di Gestione del Rischio

La Gestione del Rischio è costituita da cinque passi:

1. Identificazione del rischio: Utilizzare varie tecniche per identificare tutti i potenziali rischi.
2. Valutazione del Rischio: Valutare e stimare i rischi identificati.
3. Prioritizzazione del rischio: Prioritizzare il rischio per l'inclusione nel Prioritized Product Backlog.
4. Mitigazione del rischio: Sviluppare una strategia adeguata per trattare il rischio.
5. Comunicazione del rischio: Comunicare le risultanze dei primi quattro passi agli stakeholder appropriati e determinare la loro percezione riguardo agli eventi incerti.

### 7.4.1 Identificazione del Rischio

I membri dello Scrum Team devono cercare di identificare tutti i rischi che possono potenzialmente incidere sul progetto. Questo lavoro può essere compiuto in maniera accurata solo osservando il progetto da differenti prospettive ed utilizzando una molteplicità di tecniche. L'Identificazione dei Rischi viene eseguita per tutta la durata del progetto e i Rischi Identificati diventano input per molti processi Scrum, fra cui *Creare il Prioritized Product Backlog*, *Mettere a Punto il Prioritized Product Backlog* e *Illustrare e Convalidare lo Sprint*.

Per identificare i rischi sono utilizzate comunemente le tecniche che seguono.

#### 7.4.1.1 Tecniche di Identificazione del Rischio

1. **Rivedere le Lezioni Apprese dai Processi *Retrospettiva dello Sprint* o *Retrospettiva del Progetto***

Imparare dai progetti simili e dai precedenti Sprint dello stesso progetto ed esaminare a fondo gli eventi incerti che hanno inciso su quei progetti e Sprint può essere un modo utile per identificare i rischi.

2. **Liste di Controllo dei Rischi**

Le liste di controllo dei rischi possono contenere i punti chiave da considerare nell'attività di identificazione dei rischi, i rischi in cui ci si imbatte di solito nel progetto Scrum, o anche le categorie di rischio di cui il team si dovrebbe occupare. Le liste di controllo sono un valido strumento di aiuto per una completa identificazione dei rischi.

### 3. Prompt List dei Rischi

Le prompt list sono utilizzate per stimolare le riflessioni riguardo alle possibili fonti di rischio. Sono disponibili pubblicamente prompt list relative a vari settori industriali e tipi di progetto.

### 4. Brainstorming

Si tratta di sessioni nelle quali gli stakeholder pertinenti e i membri dello Scrum Core Team condividono apertamente le idee attraverso discussioni e sessioni di scambio della conoscenza, condotte di solito da un facilitatore.

### 5. Struttura di Scomposizione del Rischio (RBS)

Uno degli strumenti fondamentali per l'attività di identificazione dei rischi è una struttura di scomposizione del rischio (risk breakdown structure). In questa struttura, i rischi sono raggruppati in base alla loro categoria o alle caratteristiche comuni. Ad esempio, i rischi possono essere classificati in finanziari, tecnici, o relativi alla sicurezza. Questo permette al team di pianificare e affrontare meglio ogni singolo rischio.

7

#### 7.4.1.2 Risk-Based Spike

Un concetto che può essere utile adoperare nell'identificazione dei rischi è quello del risk-based spike. Uno spike è un esperimento che comporta la ricerca o la prototipazione per avere una migliore comprensione dei rischi potenziali. In uno spike, viene condotto un esercizio intenso della durata di due o tre giorni (preferibilmente all'inizio di un progetto o prima dei processi *Sviluppare le Epic* o *Creare il Prioritized Product Backlog*) per aiutare il team a stabilire quali eventi incerti potrebbero incidere sul progetto. I risk-based spike sono utili nelle situazioni in cui lo Scrum Team si trova a lavorare e a prendere dimestichezza con nuove tecnologie o strumenti, o quando le User Story sono lunghe. Questo tipo di esercizio aiuta inoltre a stimare in modo più accurato i tempi e l'impegno.

#### 7.4.2 Valutazione del Rischio

La valutazione del rischio aiuta a rendersi conto del potenziale impatto di un rischio, di quanto sia probabile che si verifichi e di quando il rischio potrebbe materializzarsi. Si deve stimare l'effetto complessivo sul valore di business; se quell'impatto è significativo al punto di travolgere la giustificazione commerciale, è obbligatorio prendere una decisione circa l'opportunità di continuare o meno il progetto.

Il rischio viene valutato in termini di probabilità, di prossimità e di impatto. La probabilità di un rischio si riferisce al grado di possibilità che il rischio si verifichi, mentre la prossimità si riferisce al momento temporale in cui il rischio potrebbe verificarsi. L'impatto rappresenta il probabile effetto dei rischi sul progetto o sull'organizzazione.

Per stimare la probabilità di un rischio si possono utilizzare varie tecniche, fra cui gli Alberi di Probabilità, l'Analisi di Pareto e una Matrice di Probabilità e Impatto.

Oltre alla probabilità, la valutazione dei rischi stima anche il potenziale effetto netto dei rischi sul progetto o sull'organizzazione. Questi effetti possono essere stimati per mezzo di tecniche come i Modelli di Rischio e il Valore Monetario Atteso.

### 7.4.2.1 Tecniche di Valutazione del Rischio

#### 1. Riunioni sul Rischio

I rischi possono essere facilmente prioritizzati dal Product Owner convocando una riunione dello Scrum Core Team ed invitandovi eventualmente anche gli Stakeholder pertinenti. Durante la riunione il team può prioritizzare rischi differenti sulla base delle proprie valutazioni soggettive dell'impatto dei rischi sugli obiettivi del progetto.

#### 2. Alberi delle Probabilità

Gli eventi potenziali sono rappresentati in un albero che ha un ramo per ogni possibile risultato di un evento di rischio. La probabilità di ogni possibile risultato viene indicata sul ramo appropriato e poi moltiplicata per il suo impatto stimato, al fine di ottenere un valore atteso per ciascun risultato possibile. I valori dei risultati vengono poi sommati tra di loro per calcolare il complessivo impatto atteso di un determinato rischio sul progetto (vedere la Figura 7-1).

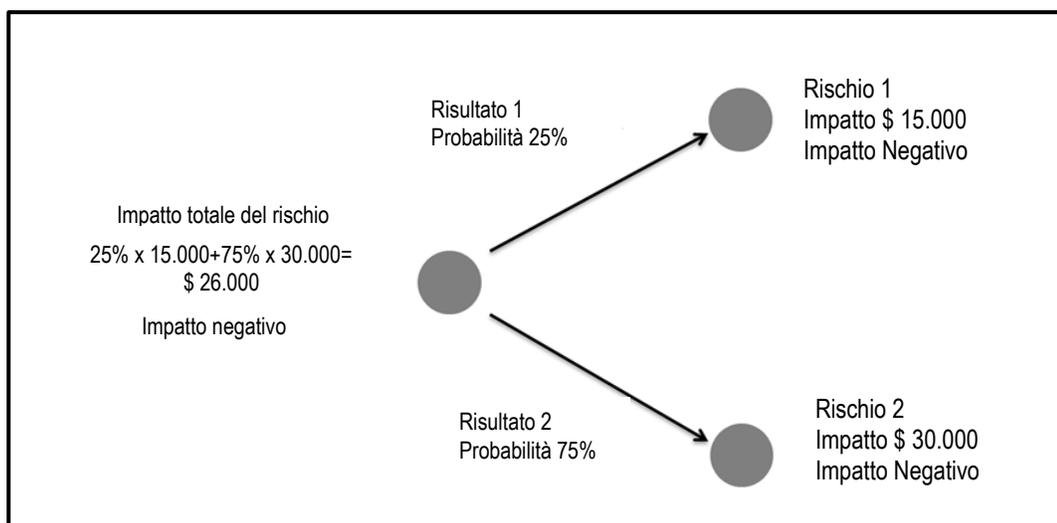


Figura 7-1: Esempio di Albero delle Probabilità

### 3. Analisi di Pareto

Questa tecnica di valutazione del rischio consiste nel classificare i rischi in base alla magnitudine. Ciò aiuta lo Scrum Team a trattare i rischi secondo l'ordine di potenziale impatto che gli stessi hanno sul progetto. Ad esempio, nella Figura 7-2, il Rischio 1 ha l'impatto più alto e deve quindi essere gestito preferibilmente per primo.

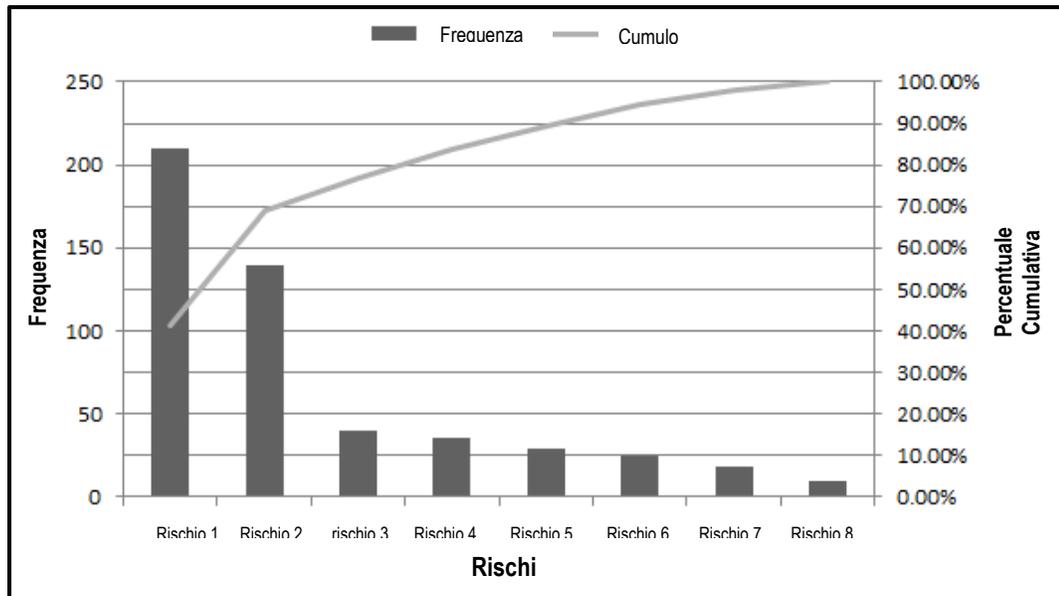


Figura 7-2: Esempio di Diagramma di Pareto

### 4. Griglia di Probabilità e Impatto

Ciascun rischio viene valutato secondo la probabilità che si verifichi e secondo il suo potenziale impatto sugli obiettivi del progetto. In genere, si assegna un punteggio numerico indipendente per la probabilità e un altro per l'impatto. I due valori vengono poi moltiplicati per ricavarne un punteggio di gravità del rischio (o valore di PI), che può essere utilizzato per prioritizzare i rischi.

Ad esempio, il punteggio di gravità del rischio per un rischio con una Probabilità del 50% e un Impatto di 0.6 verrebbe calcolato come segue:

$$0.5(\text{Probabilità}) \times 0.6(\text{Impatto}) = 0.3$$

Gli schemi di punteggio utilizzati sono stabiliti all'interno dell'organizzazione oppure per il progetto specifico. Spesso si usa una scala decimale da zero a uno, dove un punteggio di 0.5 per la probabilità indicherà una probabilità del 50%. Altre opzioni sono una scala da uno a dieci, oppure i valori Alto (3), Medio (2) e Basso (1).

La Figura 7-3 raffigura l'uso della scala decimale. Ad ogni rischio è assegnato un punteggio per la probabilità di accadimento e per l'impatto su una scala oggettiva.

Matrice di Probabilità e Impatto							
Probabilità	Minacce			Opportunità			
	0.90	0.09	0.27	0.72	0.72	0.27	0.09
	0.75	0.075	0.225	0.60	0.60	0.225	0.075
	0.50	0.05	0.15	0.40	0.40	0.15	0.05
	0.30	0.03	0.09	0.24	0.24	0.09	0.03
	0.10	0.01	0.03	0.08	0.08	0.03	0.01
		<b>Basso</b> 0.1	<b>Medio</b> 0.3	<b>Alto</b> 0.8	<b>Alto</b> 0.8	<b>Medio</b> 0.3	<b>Basso</b> 0.1

**Impatto**

Valore di PI basso
  Valore di PI medio
  Valore di PI alto

Figura 7-3: Esempio di Matrice di Probabilità e Impatto

Il metodo di assegnazione dei valori di probabilità e impatto ai rischi varia a seconda del progetto e del numero di rischi da valutare, oltre che dei processi e delle procedure in uso nell'organizzazione. In ogni caso, applicando la semplice formula  $P \times I$ , la gravità del rischio può essere calcolata su una scala numerica o categorica.

## 5. Valore Monetario Atteso (EMV)

Il valore monetario del rischio si basa sul suo Valore Monetario Atteso (Expected Monetary Value - EMV). L'EMV si calcola moltiplicando l'impatto monetario per la probabilità del rischio, secondo quanto indicato dal cliente.

Valore Monetario Atteso = Impatto del Rischio (in dollari) x Probabilità del Rischio (come percentuale)

Per esempio, un rischio con un impatto negativo stimato di \$ 1.000 e una probabilità di accadimento del 50% darà luogo al seguente EMV:

$$EMV = \$1.000 \times 0.50 = \$500$$

### 7.4.3 Prioritizzazione del Rischio

Scrum permette una veloce identificazione e valutazione dei rischi. I Rischi Identificati sono tenuti in considerazione quando si crea un Prioritized Product Backlog durante il processo *Creare il Prioritized Product Backlog*, oppure quando lo si aggiorna durante il processo *Mettere a Punto il Prioritized Product Backlog*—per cui un Prioritized Product Backlog può essere anche chiamato Risk Adjusted Prioritized Product Backlog.

I rischi possono essere identificati e valutati sulla base di una qualsiasi delle tecniche di Identificazione del Rischio e di Valutazione del Rischio menzionate prima.

Nei processi *Creare il Prioritized Product Backlog* o *Mettere a Punto il Prioritized Product Backlog*, le User Story prioritzate del Prioritized Product Backlog e la lista dei rischi prioritzata sono poi combinate fra loro per dare vita ad un Prioritized Product Backlog aggiornato che include i Rischi Identificati.

Passi per aggiornare un Prioritized Product Backlog con i Rischi Identificati:

1. Creare una lista di rischi prioritzati (ad esempio, i rischi possono essere prioritzati in base al valore utilizzando la tecnica del Valore Monetario Atteso).
2. Scegliere i Rischi Identificati che possono essere mitigati e per i quali il team decide di attuare specifiche azioni di mitigazione durante lo Sprint.
3. Creare un elenco di User Story del Prioritized Product Backlog prioritzate secondo il loro valore (ad esempio, il valore di ciascuna User Story può essere stimato sulla base del Ritorno sull'Investimento che ci si attende dalla stessa).
4. Combinare le liste dei passi 2 e 3 e prioritzarle secondo il valore per arrivare al Prioritized Product Backlog Aggiornato.

La Figura 7-4 illustra il processo di prioritizzazione del rischio.

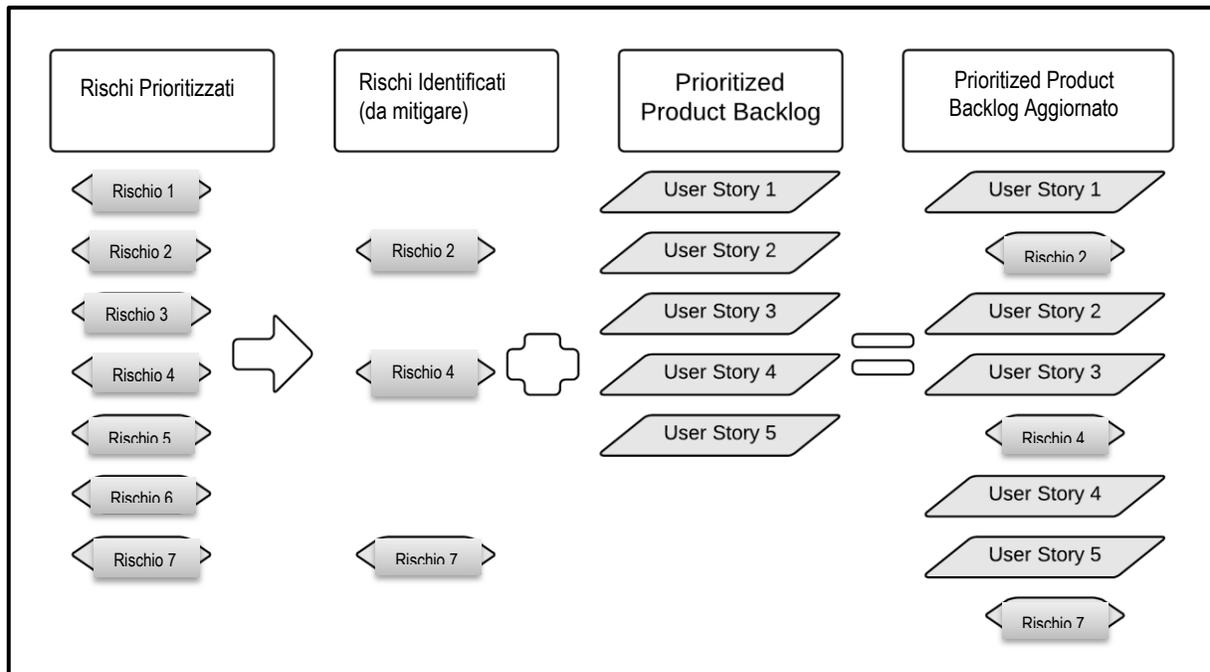


Figura 7-4: Processo di Prioritizzazione del Rischio

#### 7.4.4 Mitigazione del Rischio

La risposta a ciascun rischio dipenderà dalla probabilità e dall'impatto del rischio. In ogni caso, la natura iterativa di Scrum, con i suoi tempi rapidi di consegna e i cicli di feedback, permette la individuazione precoce di possibili insuccessi; pertanto, in pratica, questo metodo incorpora in sé una intrinseca funzione di mitigazione del rischio.

Il rischio può essere mitigato mettendo in atto un certo numero di risposte. Nella maggior parte delle situazioni, le risposte sono proattive o reattive. In caso di rischio, può essere formulato un piano B da utilizzare come alternativa qualora il rischio si materializzi - un piano B è in questo caso una risposta di tipo reattivo. A volte invece i rischi vengono accettati e queste situazioni sono un esempio di risposta al rischio né proattiva né reattiva. I rischi vengono accettati per varie ragioni, ad esempio nel caso in cui la probabilità o l'impatto del rischio sono troppo bassi per giustificare una risposta. Si può optare per l'accettazione del rischio anche nel caso in cui il timore dei rischi secondari può dissuadere il product owner dal prendere qualsiasi iniziativa. L'impegno profuso dal Product Owner per ridurre la probabilità o l'impatto del rischio - o entrambi - è un esempio di risposta proattiva di mitigazione del rischio.

Una volta che i Rishi Identificati sono inclusi nel Prioritized Product Backlog (vedere la Figura 7-4), la mitigazione di molti di questi rischi viene eseguita durante il processo *Creare i Deliverable*, nel quale vengono completate le Attività relative alle User Story definite nel Prioritized Product Backlog.

In Scrum, la proprietà del rischio fa chiaramente capo al Product Owner per la gestione dei rischi relativi agli aspetti di business e allo Scrum Team per l'implementazione delle risposte al rischio nel corso di uno Sprint. Lo Scrum Guidance Body può essere interpellato per un parere su come implementare le risposte al rischio e sulla conformità delle azioni di risposta alle linee guida dell'organizzazione nel suo insieme. Lo Scrum Master tiene strettamente d'occhio i potenziali rischi che potrebbero incidere sul progetto e mantiene informati il Product Owner e lo Scrum Team.

### 7.4.5 Comunicazione del Rischio

Dal momento che gli stakeholder hanno un interesse nel progetto, è importante comunicare con loro riguardo ai rischi. Le informazioni fornite agli stakeholder relativamente ai rischi riguardano l'impatto potenziale e i piani di risposta a ciascun rischio. La comunicazione è un'attività continua che deve attuarsi in parallelo a tutti i quattro passi sequenziali di cui si è parlato finora - identificazione, valutazione, prioritizzazione e mitigazione del rischio. Durante i Daily Standup Meeting lo Scrum Team deve inoltre parlare con lo Scrum Master dei rischi specifici legati alle Attività che sta eseguendo. Il Product Owner ha la responsabilità di prioritizzare i rischi e di comunicare allo Scrum Team la lista delle priorità.

Un importante strumento da utilizzare per comunicare le informazioni relative ai rischi è il Risk Burndown Chart.

#### 7.4.5.1 Risk Burndown Chart

La gestione del rischio è indispensabile per assicurare la creazione di valore; perciò, le attività di gestione del rischio sono eseguite per tutto il ciclo di vita del progetto e non solo durante l'inizio del progetto.

Ciascun rischio può essere valutato con diversi strumenti di Valutazione del Rischio. Tuttavia, lo strumento preferito per la valutazione dei rischi e la creazione di un Risk Burndown Chart è il Valore Monetario Atteso (EMV), già descritto nella sezione 7.4.2.1.

Le informazioni raccolte durante la valutazione del rischio possono essere usate per creare un Risk Burndown Chart. Questo grafico illustra la gravità cumulativa del rischio di progetto nel tempo. Le probabilità dei vari Rischi sono rappresentate graficamente una sopra l'altra per mostrare il rischio cumulativo sull'asse y. Al principio vengono eseguite l'identificazione e la valutazione iniziale dei rischi del progetto e la creazione del Risk Burndown Chart. Successivamente, a intervalli di tempo prestabiliti, possono essere identificati nuovi rischi e i rischi restanti devono essere rivalutati e aggiornati di conseguenza sul grafico. Un momento giusto per compiere questo tipo di attività è durante lo Sprint Planning Meeting. Questo modo di monitorare i rischi permette al team di individuare gli andamenti dell'esposizione al rischio e di adottare le misure appropriate, se necessario.

La Figura 7-5 mostra un esempio di Risk Burndown Chart.

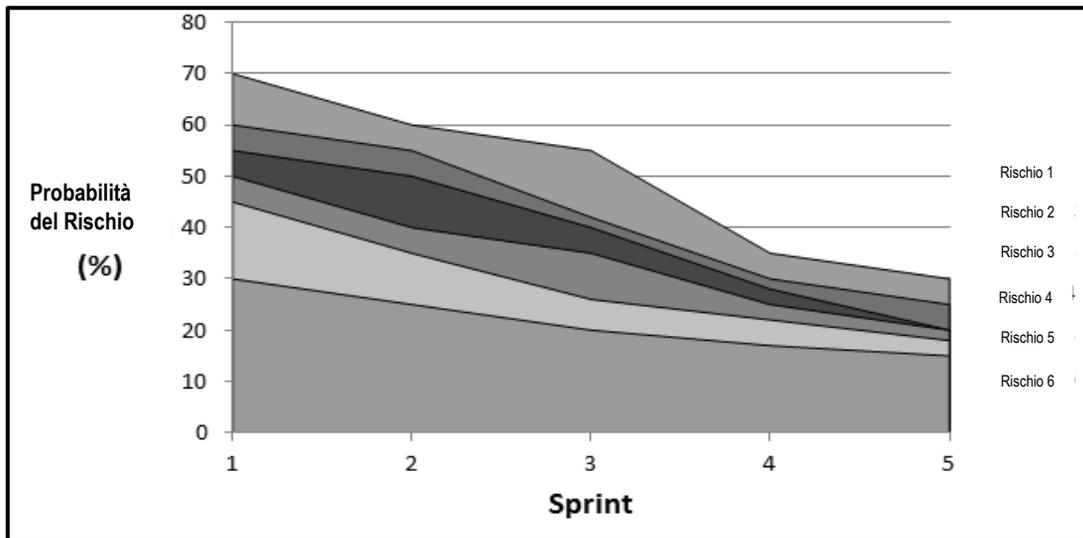


Figura 7-5: Esempio di Risk Burndown Chart

## 7.5 Minimizzare il Rischio con Scrum

Essendo un processo Agile e iterativo, il framework Scrum ha la capacità intrinseca di minimizzare il rischio. Le pratiche Scrum di seguito elencate facilitano l'efficace gestione del rischio:

### 1. La flessibilità riduce i rischi relativi al contesto di business

In Scrum il rischio è minimizzato in larga misura grazie alla flessibilità che lo caratterizza nell'aggiungere o modificare i requisiti in qualunque momento del ciclo di vita del progetto. Questo mette in grado l'organizzazione di rispondere alle minacce e alle opportunità che provengono dal contesto di business e da requisiti imprevisi in qualsiasi momento essi insorgano, di solito con bassi costi di gestione dei rischi stessi.

### 2. Il feedback regolare riduce i rischi relativi alle aspettative

Essendo iterativo, il framework Scrum offre ampie opportunità di ricevere il feedback e stabilire le aspettative lungo tutto il ciclo di vita del progetto. Questo garantisce che gli stakeholder, così come il team, non vengano colti di sorpresa da requisiti mal comunicati.

### 3. La responsabilità collettiva del team riduce i rischi legati alla stima

Lo Scrum Team stima gli elementi dello Sprint Backlog e se ne assume la responsabilità, il che porta ad una stima più accurata e alla consegna puntuale degli incrementi di prodotto.

**4. La trasparenza riduce il rischio di mancata individuazione dei rischi**

Il principio di trasparenza di Scrum intorno al quale è costruito l'intero framework consente di individuare e comunicare precocemente i rischi, con una conseguente migliore gestione e mitigazione del rischio stesso. Inoltre, quando si conducono gli Scrum of Scrums Meeting, gli impedimenti che un team sta riscontrando nel presente possono essere ritenuti un rischio futuro per altri Scrum Team. Di questo si deve dare atto nel Registro degli Impedimenti Aggiornato.

**5. La consegna iterativa riduce il rischio in termini di investimento**

La consegna continua di valore per tutto il ciclo di vita del progetto Scrum, dal momento che dopo ogni Sprint sono creati Deliverable potenzialmente consegnabili, riduce il rischio del cliente in termini di investimento.

**7.6 I rischi nei Portfolio e nei Programmi**

Mentre alcuni rischi sono relativi specificamente ai singoli progetti, altri possono scaturire dai programmi o portfolio e verranno generalmente gestiti in quel contesto. Tuttavia, i rischi relativi ad un portfolio o programma incideranno anche sui progetti che fanno parte rispettivamente del portfolio o programma. Se durante l'attività di valutazione del rischio nei portfolio e nei programmi si stabilisce che un rischio può avere un impatto su un singolo progetto, devono essere comunicate al Product Owner e allo Scrum Team le informazioni riguardanti il rischio che hanno rilevanza per il progetto.

A seconda della gravità o priorità, quando il team del programma o del portfolio comunica un rischio che avrà un impatto sul singolo progetto, lo Scrum Team può essere costretto a interrompere e ripianificare lo Sprint in corso per trattare il rischio in questione. Per i rischi meno urgenti, il team può continuare lo Sprint in corso e trattare il rischio in uno Sprint successivo.

**7.6.1 Nel Portfolio**

1. Quando sono identificati rischi del Portfolio, il Portfolio Product Owner dovrà registrarli e valutare la prossimità, la probabilità e l'impatto di ciascun rischio identificato al fine di prioritarlo e stabilire la risposta appropriata per il portfolio.
2. Il Portfolio Product Owner dovrà inoltre comunicare i rischi agli stakeholder pertinenti, ai team di programma e ai team di progetto. In alcuni casi, il team di portfolio può doversi assumere la responsabilità di specifici rischi.

## 7.6.2 Nel Programma

1. Quando sono identificati rischi del programma, il Program Product Owner deve registrarli nel Risk Adjusted Prioritized Product Backlog del programma, valutare la prossimità, la probabilità e l'impatto di ciascun rischio identificato al fine di priorizzarlo e stabilire le risposte appropriate per i programmi.
2. Il Program Product Owner dovrà inoltre comunicare i rischi agli stakeholder pertinenti e ai team di progetto. In alcuni casi, il team di programma può doversi assumere la responsabilità di specifici rischi.

La Figura 7-6 illustra come gestire i rischi all'interno del flusso di Scrum, sia per i portfolio che per i programmi.

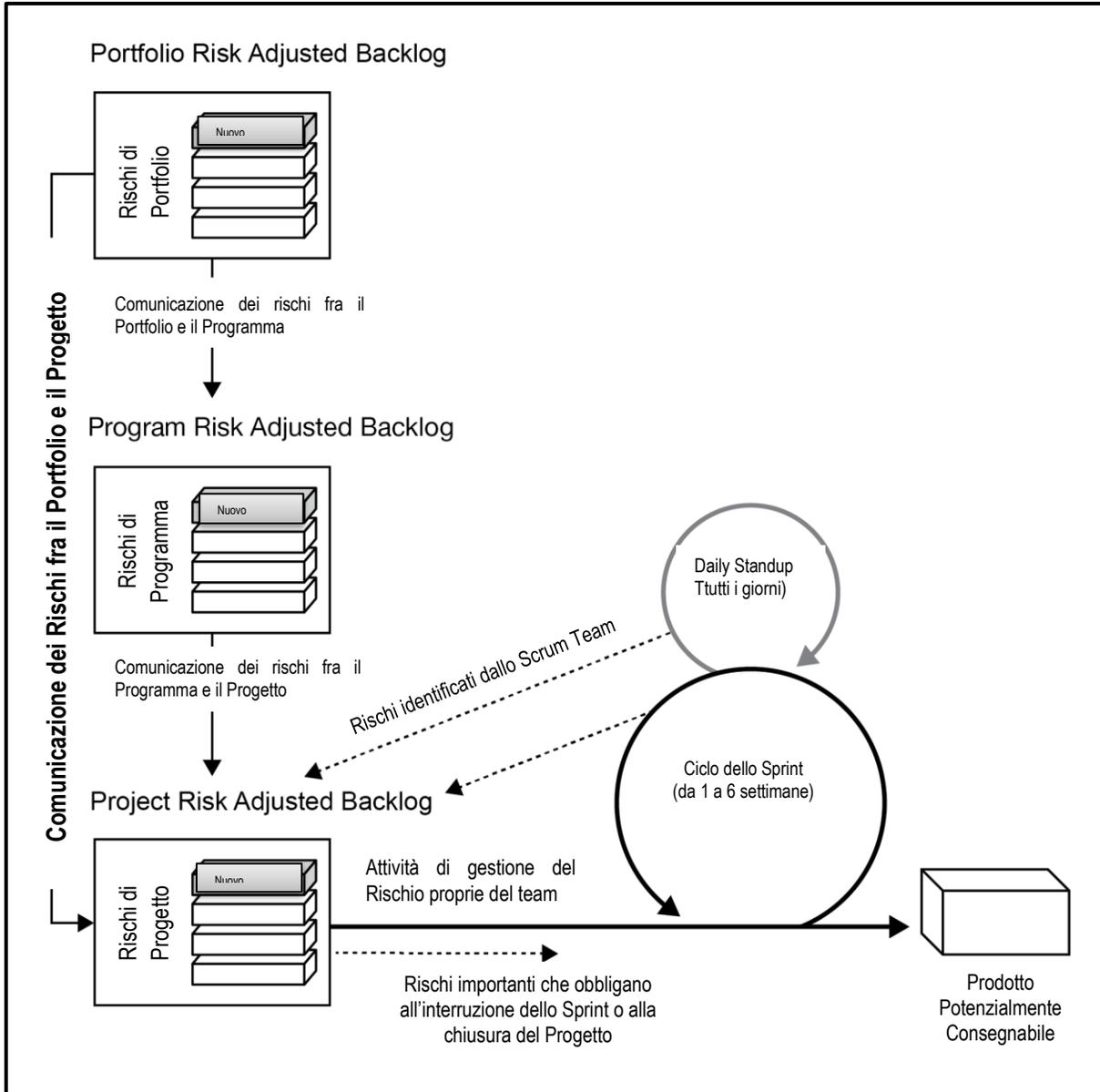


Figura 7-6: Gestire i Rischi nei Portfolio e nei Programmi

## 7.7 Riepilogo delle Responsabilità

In Scrum, le attività di gestione del rischio sono suddivise fra i vari ruoli, pur rimanendo alcune responsabilità in carico a tutto lo Scrum Team indistintamente, mentre lo Scrum Master ha il compito di facilitare il processo.

Ruolo	Responsabilità
Scrum Guidance Body	<ul style="list-style-type: none"> <li>Fornisce l'indirizzo generale riguardo alla procedura di gestione del rischio da seguire nel corso del progetto</li> </ul>
Portfolio Product Owner	<ul style="list-style-type: none"> <li>Registra e valuta i rischi del portfolio</li> <li>Prioritizza e comunica i rischi agli stakeholder pertinenti e ai team di programma e di progetto</li> </ul>
Portfolio Scrum Master	<ul style="list-style-type: none"> <li>Facilita l'identificazione, la valutazione e la comunicazione dei rischi di portfolio</li> </ul>
Program Product Owner	<ul style="list-style-type: none"> <li>Registra e valuta i rischi dei programmi</li> <li>Prioritizza e comunica i rischi agli stakeholder pertinenti e ai team di progetto</li> </ul>
Program Scrum Master	<ul style="list-style-type: none"> <li>Facilita l'identificazione, la valutazione e l'escalation dei rischi di programma</li> </ul>
Stakeholder	<ul style="list-style-type: none"> <li>Si interfacciano con lo Scrum Core Team per fornire input sulla gestione dei rischi che incidono sulla realizzazione dei risultati e dei benefici attesi dal progetto</li> </ul>
Product Owner	<ul style="list-style-type: none"> <li>Registra e valuta i rischi del progetto</li> <li>Prioritizza e comunica i rischi agli stakeholder pertinenti e ai team di programma e di portfolio</li> <li>Si assicura che i livelli di rischio del progetto rimangano entro limiti accettabili</li> </ul>
Scrum Master	<ul style="list-style-type: none"> <li>Facilita l'identificazione e l'escalation dei rischi da parte dello Scrum Team</li> </ul>
Scrum Team	<ul style="list-style-type: none"> <li>Identifica i rischi durante lo sviluppo del prodotto nel corso del processo <i>Creare i Deliverable</i></li> <li>Implementa le attività di gestione del rischio come consigliato dal Product Owner</li> </ul>

Tavola 7-1: Riepilogo delle Responsabilità Riferibili al Rischio

## 7.8 Confronto fra Scrum e il Project Management Tradizionale

Scrum e la maggior parte dei metodi tradizionali di project management definiscono il rischio come ‘un evento o più eventi incerti che possono incidere positivamente o negativamente sulla realizzazione degli obiettivi del progetto’. Inoltre, i rischi sono identificati, valutati, fatti oggetto di un piano di risposta e comunicati continuamente.

Nei modelli tradizionali di project management, si punta su una dettagliata pianificazione anticipata per identificare, valutare e stabilire le risposte al rischio di tutti i rischi di progetto. Durante l'esecuzione del progetto, qualunque membro del team di progetto può identificare dei rischi e il project manager o il project management office o il personale di supporto al progetto può provvedere al loro inserimento nel Promemoria dei Rischi o nel Registro dei Rischi. Il project manager monitora e controlla regolarmente tutti i rischi e di solito identifica specifici soggetti all'interno del team ai quali far assumere la responsabilità riguardo a differenti aspetti dei rischi.

In Scrum, i rischi possono essere identificati da un qualunque membro del team, mentre il Product Owner può provvedere all'aggiornamento dei rischi identificati all'interno Risk Adjusted Prioritized Product Backlog. I principi del Controllo Empirico di Processo e dello Sviluppo Iterativo propri di Scrum consentono allo Scrum Team di poter identificare costantemente i rischi e aggiungerli al Prioritized Product Backlog, nel quale questi rischi sono prioritizzati insieme alle altre User Story già presenti nel backlog, per poter poi essere mitigati negli Sprint successivi. Lo Scrum Team ha su di sé la responsabilità collettiva della gestione di tutti i rischi dello Sprint.

## 8. INIZIO

Questo capitolo parla dei processi relativi alla fase di Inizio di un progetto: *Creare la Vision del Progetto, Identificare lo Scrum Master e gli Stakeholder, Formare lo Scrum Team, Sviluppare le Epic, Creare il Prioritized Product Backlog e Eseguire la Pianificazione dei Rilasci*.

La fase di *Inizio*, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Per facilitare la migliore applicazione del framework Scrum, questo capitolo identifica gli input, gli strumenti e gli output di ciascun processo qualificandoli come “obbligatori” o “opzionali”. Gli input, gli strumenti e gli output contrassegnati da un asterisco (\*) sono obbligatori, mentre quelli senza asterisco sono opzionali.

Si raccomanda allo Scrum Team e a tutti coloro che vengono introdotti al framework e ai processi Scrum di concentrarsi soprattutto sugli input, gli strumenti e gli output obbligatori; i Product Owner, gli Scrum Master e altri professionisti di Scrum più esperti dovrebbero invece sforzarsi di raggiungere una conoscenza più approfondita delle informazioni contenute nell'intero capitolo. È altresì importante avere chiaro che, sebbene tutti i processi siano definiti singolarmente nella *Guida SBOK™*, non è necessario eseguirli in maniera sequenziale o separatamente. A volte, a seconda dei requisiti specifici di ciascun progetto, potrebbe essere più appropriato unire alcuni processi fra loro.

Questo capitolo è scritto dal punto di vista di un singolo Scrum Team che lavora ad un singolo Sprint per produrre Deliverable potenzialmente consegnabili come parte di un progetto più ampio. Tuttavia, le informazioni fornite sono ugualmente applicabili a interi progetti, programmi e portfolio. Ulteriori informazioni sull'uso di Scrum nei progetti, programmi e portfolio sono disponibili nei capitoli da 2 a 7, che trattano i principi e gli aspetti di Scrum.

La Figura 8-1 offre una visione d'insieme dei processi della Fase di Inizio, che sono:

**8.1 Creare La Vision del Progetto**—In questo processo si rivede il business case per dare vita a una Dichiarazione della Vision del Progetto, che servirà da ispirazione e fungerà da punto di riferimento per l'intero progetto. Il Product Owner viene identificato in questo processo.

**8.2 Identificare lo Scrum Master e gli Stakeholder**—In questo processo si procede ad identificare lo Scrum Master e gli Stakeholder utilizzando specifici Criteri di Selezione.

**8.3 Formare lo Scrum Team**—In questo processo si identificano i membri dello Scrum Team. Di solito la responsabilità della scelta dei membri del team fa capo principalmente al Product Owner, il quale però spesso si avvale della collaborazione dello Scrum Master per compiere questa attività.

**8.4 Sviluppare le Epic**—In questo processo la Dichiarazione della Vision del Progetto funge da base per lo sviluppo delle Epic. Per discutere quali Epic siano appropriate si possono tenere delle Riunioni di Gruppi di Utenti.

**8.5 Creare il Prioritized Product Backlog**—In questo processo le Epic vengono rifinite, elaborate e priorizzate per creare un Prioritized Product Backlog del progetto. In questo momento si stabiliscono inoltre i Done Criteria.

**8.6 Eseguire la Pianificazione dei Rilasci**—In questo processo lo Scrum Core Team rivede le User Story del Prioritized Product Backlog per sviluppare una Schedulazione del Piano dei Rilasci, la quale è in sostanza un programma di distribuzione per fasi che può essere condiviso con gli stakeholder del progetto. In questo processo viene stabilita anche la Lunghezza degli Sprint.



Figura 8-1: Visione d'insieme della Fase di Inizio

Nota: Gli asterischi (\*) indicano un input, strumento o output "obbligatorio" del processo corrispondente.

La Figura 8-2 qui sotto mostra gli input, gli strumenti e gli output obbligatori dei processi della fase di Inizio.

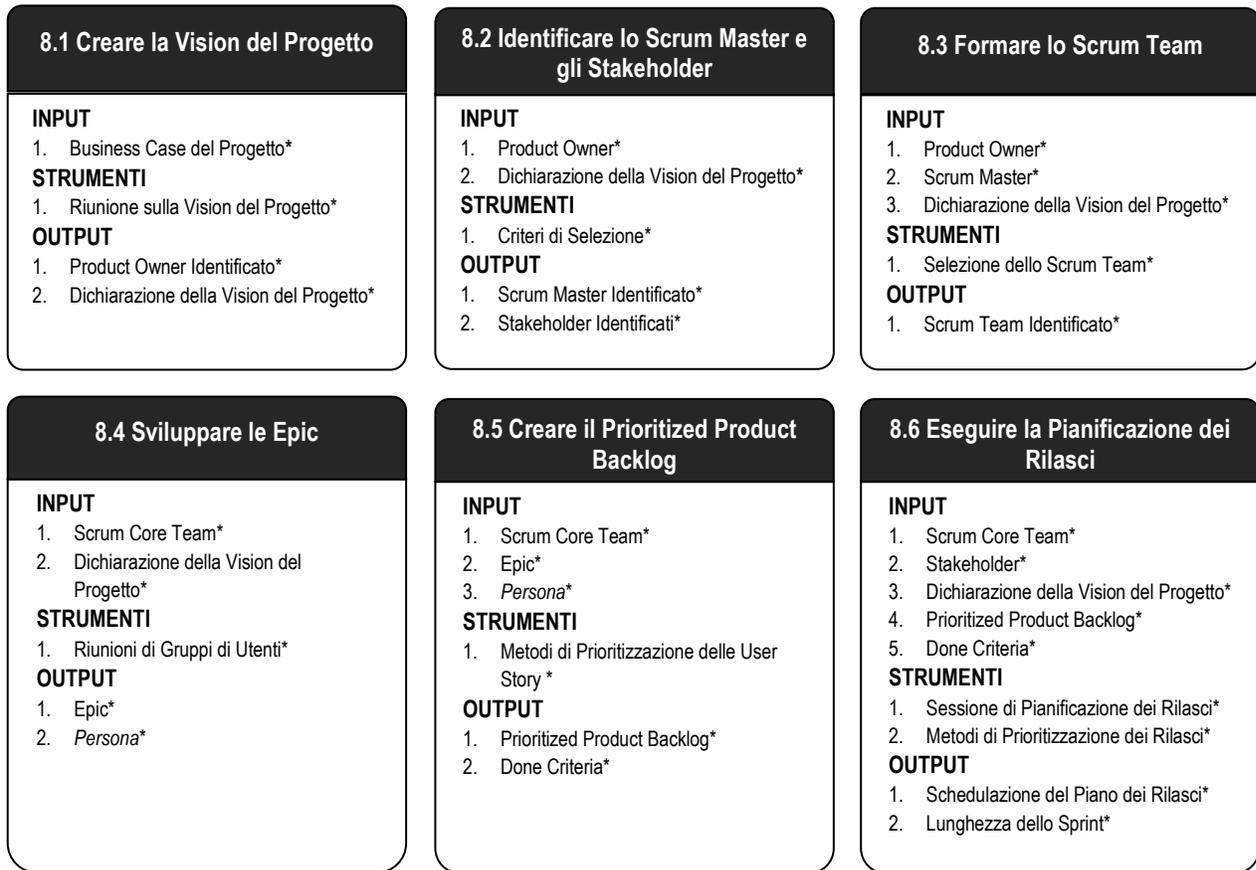


Figura 8-2: Visione d'insieme della Fase di Inizio (Elementi essenziali)

Nota: Gli asterischi (\*) indicano un input, strumento o output "obbligatorio" del processo corrispondente.

## 8.1 Creare la Vision del Progetto

La Figura 8-3 mostra tutti gli input, gli strumenti e gli output del processo *Creare la Vision del Progetto*.

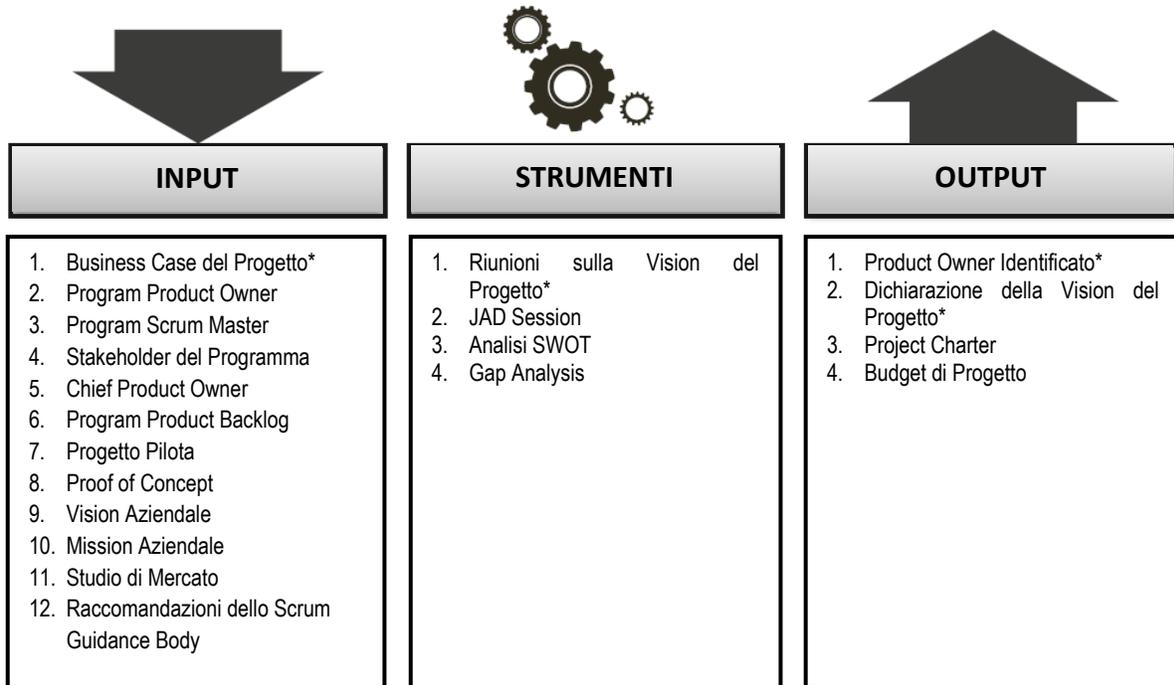


Figura 8-3: Creare la Vision del Progetto—Input, Strumenti e Output

Nota: Gli asterischi (\*) indicano un input, strumento o output "obbligatorio" del processo corrispondente.

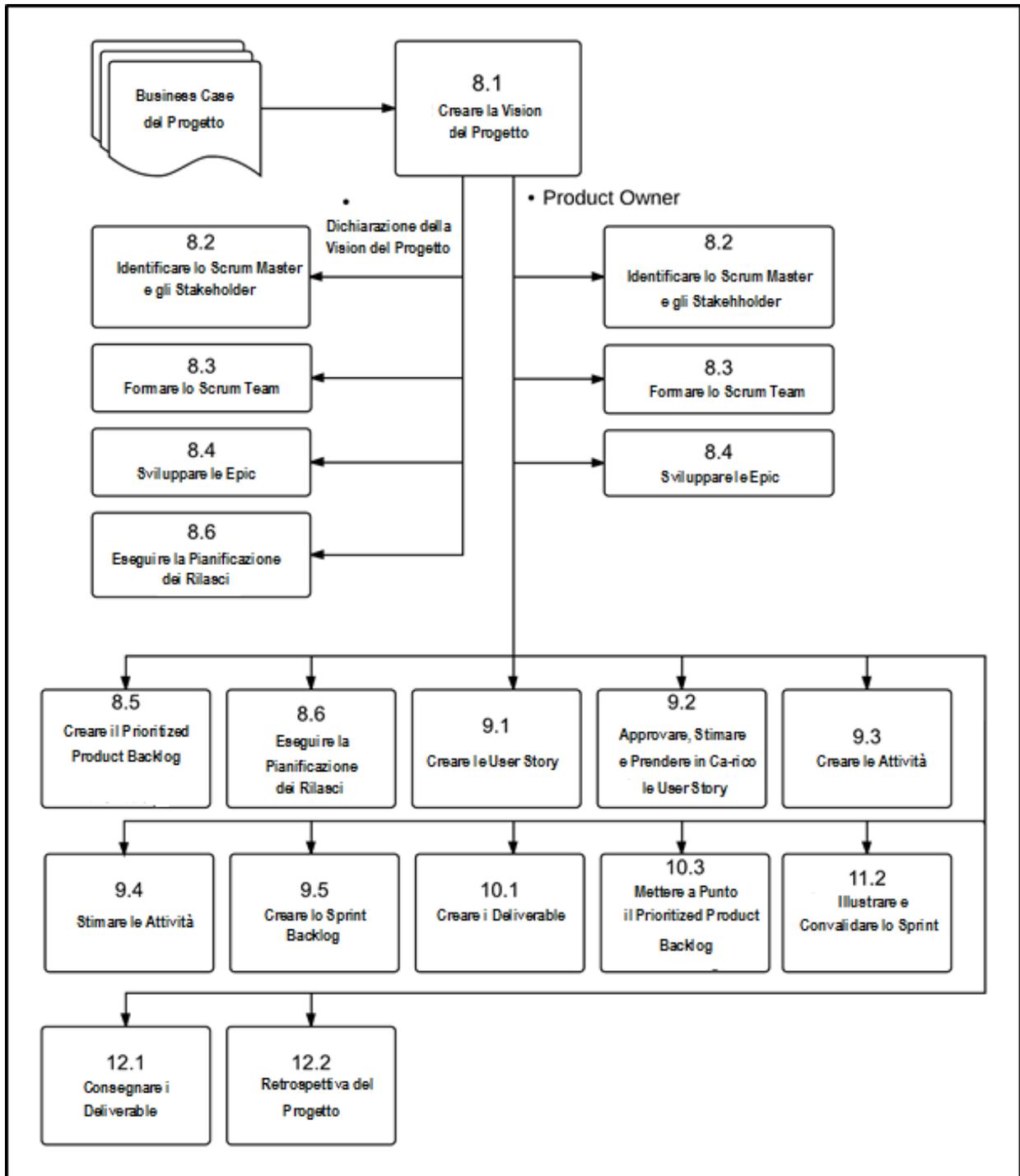


Figura 8-4: Creare la Vision del Progetto—Diagramma di Flusso dei Dati

## 8.1.1 Input

### 8.1.1.1 Business Case del Progetto\*

Un business case può essere un documento molto strutturato o semplicemente una dichiarazione orale che esprime le ragioni che giustificano l'inizio di un progetto. Può essere formale ed esaustivo, oppure breve e informale. Indipendentemente dal formato, include spesso informazioni sostanziali sul retroterra del progetto, l'intento di business prefissato e i risultati desiderati, un rapporto sull'analisi SWOT e sulla Gap analysis, un elenco dei rischi identificati e infine le stime dei tempi, dell'impegno e dei costi.

Il progetto prende il via con la presentazione del Business Case del Progetto. Un business case si presenta agli stakeholder e agli sponsor. Gli stakeholder acquisiscono una piena comprensione dei benefici di business attesi dal progetto e gli sponsor confermano che metteranno a disposizione le risorse finanziarie per il progetto.

### 8.1.1.2 Program Product Owner

Il Program Product Owner è la persona responsabile della massimizzazione del valore di business del programma. Lui o lei ha la responsabilità di esprimere al meglio i requisiti del cliente e di mantenere valida la giustificazione commerciale del programma e può fornire input preziosi riguardo a quali progetti dovrebbero essere inseriti, e in che modo, in un programma. Il Program Product Owner gestisce inoltre il Program Product Backlog.

Il Program Product Owner si interfaccia con il Portfolio Product Owner per garantire l'allineamento del programma con gli obiettivi del portfolio. È inoltre coinvolto nella nomina dei Product Owner dei singoli progetti e contribuisce a garantire che la vision, gli obiettivi, i risultati e i rilasci dei singoli progetti del programma siano in linea con quelli del programma stesso.

### 8.1.1.3 Program Scrum Master

Il Program Scrum Master è un facilitatore che fa in modo che tutti i team di progetto all'interno del programma abbiano a disposizione un ambiente favorevole al completamento con successo dei rispettivi progetti. Il Program Scrum Master indirizza, facilita e insegna le pratiche Scrum a tutti coloro che partecipano al programma; funge da guida per gli Scrum Master dei singoli progetti; risolve gli impedimenti dei vari team di progetto; si coordina con lo Scrum Guidance Body per definire gli obiettivi relativi alla qualità, ai regolamenti governativi, alla sicurezza e altri parametri fondamentali dell'organizzazione; infine, fa in modo che i processi di Scrum siano realmente applicati nell'ambito di tutto il programma.

Il Program Scrum Master si interfaccia con il Portfolio Scrum Master per garantire l'allineamento del programma con gli obiettivi particolari e generali del portfolio. È inoltre coinvolto nella nomina degli Scrum Master dei singoli progetti e contribuisce a garantire che la vision, gli obiettivi, i risultati e i rilasci dei singoli progetti del programma siano in linea con quelli del programma stesso.

#### 8.1.1.4 Stakeholder del Programma

‘Stakeholder del Programma’ è un termine collettivo che ricomprende i clienti, gli utenti e gli sponsor di un programma. Questi esercitano un’influenza su tutti i progetti del programma e lungo tutto lo sviluppo del progetto. Gli Stakeholder del Programma possono contribuire inoltre alla definizione della vision del progetto e fornire delle linee di indirizzo riguardo al valore di business.

Gli Stakeholder del Programma si interfacciano con gli Stakeholder del Portfolio per garantire l’allineamento del programma con gli obiettivi particolari e generali del portfolio. Sono inoltre coinvolti nella nomina degli Stakeholder dei singoli progetti e contribuiscono a garantire che la vision, gli obiettivi, i risultati e i rilasci dei singoli progetti del programma siano in linea con quelli del programma stesso.

#### 8.1.1.5 Chief Product Owner

Nel caso di progetti di grandi dimensioni con numerosi Scrum Team, può essere necessario avere un Chief Product Owner. Questo ruolo ha la responsabilità di coordinare il lavoro di vari Product Owner. Il Chief Product Owner prepara e mantiene il Prioritized Product Backlog complessivo dei progetti di grandi dimensioni, utilizzandolo per coordinare il lavoro dei Product Owner dei vari Scrum Team. I Product Owner, a loro volta, gestiscono le loro rispettive parti del Prioritized Product Backlog.

Il Chief Product Owner si interfaccia inoltre con il Program Product Owner per garantire l’allineamento del progetto di grandi dimensioni con gli obiettivi particolari e generali del programma.

#### 8.1.1.6 Program Product Backlog

Il Program Product Owner sviluppa il Program Product Backlog, che contiene un elenco prioritizzato dei requisiti di business e di progetto di alto livello, scritti preferibilmente sotto forma di voci estese del Program Backlog. Queste sono successivamente rifinite dai Product Owner dei singoli progetti in occasione della creazione e prioritizzazione dei Product Backlog dei propri progetti. Questi Prioritized Product Backlog contengono User Story molto più piccole ma estremamente dettagliate che possono essere approvate, stimate e prese in carico dai singoli Scrum Team.

Il Program Product Backlog è continuamente messo a punto dal Program Product Owner, per garantire che siano aggiunti i requisiti di business nuovi e che siano correttamente documentati e prioritizzati quelli già esistenti. Questo fa sì che vengano prioritizzati come alti i requisiti che contribuiscono maggiormente alla soddisfazione degli obiettivi del programma, mentre agli altri viene data una priorità più bassa.

Il Program Product Backlog creato per il programma presenta un’immagine più ampia di tutti i progetti che fanno parte del programma. Pertanto, può fornire indicazioni rilevanti sugli obiettivi particolari del progetto, il suo ambito, i suoi obiettivi generali e i benefici di business attesi dallo stesso.

### **8.1.1.7 Progetto Pilota**

Se possibile, si può mettere in piedi una dimostrazione o progetto pilota su piccola scala come esperimento per prevedere e stimare la fattibilità, i tempi e costi, i rischi e i possibili effetti del progetto reale. Questo aiuta a valutare il contesto pratico e guida la progettazione del progetto reale prima che questo inizi su scala completa.

### **8.1.1.8 Proof of Concept**

Un Proof of Concept (Prova del Concetto) dimostra e verifica che l'idea sottostante al progetto attuale è potenzialmente fattibile nel contesto del mondo reale. Spesso nella forma di un prototipo, il Proof of Concept è volto a stabilire la fattibilità finanziaria e tecnica, aiuta a capire i requisiti e assiste nella valutazione delle decisioni di progettazione nelle prime fasi del processo. Tuttavia, il Proof of Concept non deve rappresentare necessariamente i Deliverable effettivi del progetto.

### **8.1.1.9 Vision Aziendale**

Avere ben chiara la Vision dell'Azienda contribuisce a mantenere il progetto focalizzato sugli obiettivi dell'organizzazione e sul possibile futuro dell'azienda. Il Product Owner può trarre indicazioni e direttive dalla Vision Aziendale per creare la Dichiarazione della Vision del Progetto.

### **8.1.1.10 Mission Aziendale**

La Mission Aziendale fornisce un quadro per la formulazione delle strategie dell'azienda e guida il processo decisionale complessivo dell'azienda. La Vision del Progetto deve essere inquadrata in modo tale che la sua realizzazione contribuisca alla realizzazione della mission dell'organizzazione.

### **8.1.1.11 Studio di Mercato**

Lo Studio di Mercato si riferisce alla ricerca organizzata, alla raccolta, alla collazione e all'analisi dei dati relativi alle preferenze dei clienti riguardo ai prodotti. Spesso riguarda dati dettagliati sulle tendenze del mercato, sulla segmentazione del mercato e sui processi di marketing. Lo studio di mercato può inoltre includere uno studio analitico dei concorrenti che permette una migliore comprensione dei loro punti di forza e di debolezza e può aiutare gli organi decisionali a formulare prodotti meglio posizionati.

### **8.1.1.12 Raccomandazioni dello Scrum Guidance Body**

Lo Scrum Guidance Body (SGB) è un ruolo opzionale. In genere consiste in un gruppo di documenti e/o in un gruppo di esperti che si occupano di solito di definire gli obiettivi relativi alla qualità, ai regolamenti governativi, alla sicurezza e ad altri parametri fondamentali dell'organizzazione.

Questi obiettivi guidano il lavoro portato avanti dal Product Owner, dallo Scrum Master, e dallo Scrum Team. Lo Scrum Guidance Body aiuta inoltre a individuare e registrare le best practice che dovrebbero essere utilizzate in tutti i progetti Scrum dell'organizzazione.

Lo Scrum Guidance Body non prende decisioni riguardo al progetto. Agisce invece come struttura di consulenza o di indirizzo per tutti i livelli gerarchici dell'organizzazione di progetto —il portfolio, il programma e il progetto. Gli Scrum Team hanno la possibilità di chiedere un parere allo Scrum Guidance Body, se necessario. È importante assicurare l'allineamento della vision del progetto alle raccomandazioni fornite dallo Scrum Guidance Body e la conformità dei processi agli eventuali standard e linee guida stabiliti da tale organismo.

## 8.1.2 Strumenti

### 8.1.2.1 Riunioni sulla Vision del Progetto\*

Ad una Riunione sulla Vision del Progetto partecipano gli Stakeholder del Programma, il Program Product Owner, il Program Scrum Master e il Chief Product Owner. Questa riunione aiuta a identificare il contesto di business, i requisiti di business e le aspettative degli stakeholder, in modo da poter sviluppare una Dichiarazione della Vision del Progetto realmente efficace. Scrum crede nello stretto coinvolgimento e collaborazione con tutti i rappresentanti di business per ottenere la loro piena adesione al progetto e consegnare un valore maggiore.

### 8.1.2.2 JAD Session

Una Joint Application Design (JAD) Session è una tecnica di raccolta dei requisiti. Si tratta di un workshop facilitato molto strutturato che accelera il processo *Creare la Vision del Progetto*, in quanto consente agli Stakeholder e agli altri organi decisionali di pervenire ad un consenso in merito all'ambito, agli obiettivi e ad altre specifiche del progetto.

Consta di metodi per aumentare la partecipazione dell'utente, accelerare lo sviluppo e migliorare le specifiche. Gli Stakeholder del Programma, il Program Product Owner, il Program Scrum Master e il Chief Product Owner possono incontrarsi per delineare e analizzare i risultati di business desiderati e prefigurare la propria vision del progetto Scrum.

### 8.1.2.3 Analisi SWOT

Lo SWOT è un approccio strutturato alla pianificazione del progetto che aiuta a valutare i punti di forza, i punti di debolezza, le opportunità e le minacce relative ad un progetto (in inglese **S**trengths, **W**eaknesses, **O**pportunities e **T**hreats). Questo tipo di analisi aiuta ad identificare i fattori sia interni che esterni che possono incidere sul progetto. I punti di forza e di debolezza sono fattori interni, mentre le opportunità e le minacce sono fattori esterni.

L'identificazione di questi fattori aiuta gli stakeholder e gli altri organi decisionali a definire quali processi, strumenti e tecniche utilizzare per realizzare gli obiettivi del progetto. L'esecuzione di un'analisi SWOT consente di identificare precocemente le priorità, i potenziali cambiamenti e i rischi.

### 8.1.2.4 Gap Analysis

La Gap Analysis è una tecnica utilizzata per confrontare lo stato effettivo attuale con quello desiderato. Nell'ambito di un'organizzazione, tale analisi comporta la determinazione e la documentazione della differenza fra le attuali capacità di business ed il complesso finale di capacità desiderate. Di solito un progetto viene iniziato per portare un'organizzazione ad uno stato desiderato, per cui condurre una Gap Analysis aiuterà gli organi decisionali a stabilire se ci sia o meno bisogno di un progetto.

Nella Figura 8-5 vengono presentati i passi principali della Gap Analysis.

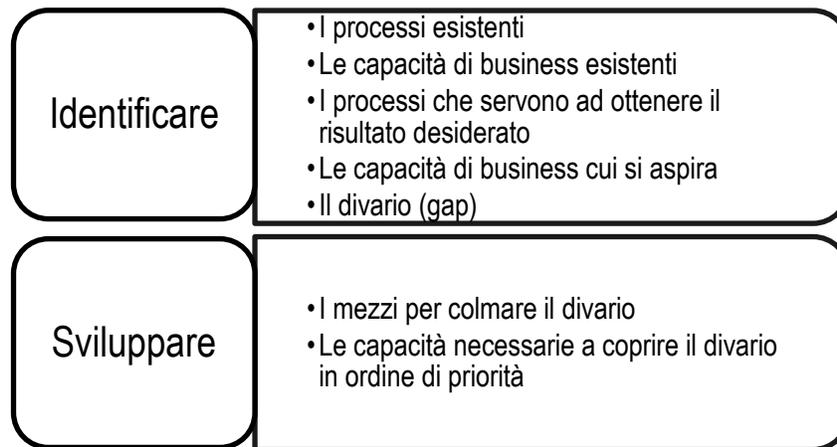


Figura 8-5: Il processo di Gap Analysis

## 8.1.3 Output

### 8.1.3.1 Product Owner Identificato\*

Uno degli output di questo processo è l'identificazione del Product Owner. Il Product Owner è la persona responsabile della realizzazione del massimo valore di business dal progetto. È inoltre responsabile dell'enunciazione dei requisiti del cliente e del mantenimento della giustificazione commerciale del progetto. Il Product Owner rappresenta la Voce del Cliente.

Ogni Scrum Team avrà un Product Owner designato. Un progetto piccolo può avere un solo Product Owner, mentre i progetti di più grandi dimensioni possono averne molti. Questi Product Owner sono responsabili della gestione delle proprie sezioni del Prioritized Product Backlog. I Product Owner scrivono le User Story e gestiscono e mettono a punto il Prioritized Product Backlog.

Il ruolo del Product Owner è descritto in maggiore dettaglio nel Capitolo 3, Sezione 3.4.

### 8.1.3.2 Dichiarazione della Vision del Progetto\*

L'output fondamentale del processo *Creare la Vision del Progetto* è una Dichiarazione della Vision del Progetto ben strutturata. Una buona vision del progetto spiega l'esigenza di business che il progetto è destinato a soddisfare piuttosto che come la soddisferà.

La Dichiarazione della Vision del Progetto non deve essere troppo specifica e deve lasciare spazio ad una certa flessibilità. È possibile che la comprensione attuale del progetto si basi su presupposti che cambieranno con il progredire del progetto, per cui è importante che la vision del progetto sia abbastanza flessibile da accogliere questi cambiamenti. La vision del progetto si deve concentrare sul problema piuttosto che sulla soluzione.

*Esempio:*

VMFoods, una catena tradizionale di prodotti alimentari, vuole espandersi con un portale di commercio elettronico online e ha contattato la tua azienda per creare il prodotto.

Vision del Progetto: Sviluppare per VMFoods un canale di vendite online di facile utilizzo ed esteticamente gradevole.

### 8.1.3.3 Project Charter

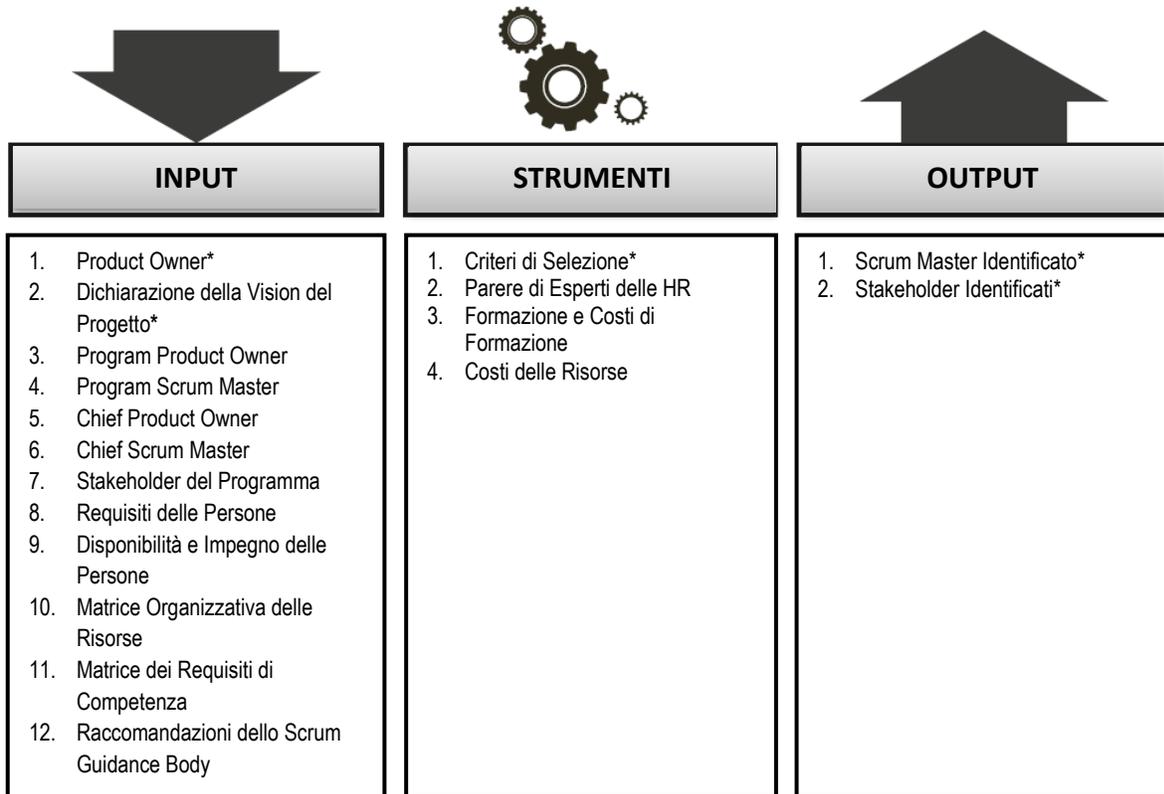
Un Project Charter è una dichiarazione ufficiale degli obiettivi e dei risultati desiderati del progetto. In molte organizzazioni, il Project Charter è il documento che autorizza ufficialmente e formalmente il progetto, conferendo per iscritto al team l'autorità per iniziare il lavoro di progetto.

### 8.1.3.4 Budget di Progetto

Il Budget di Progetto è un documento finanziario che include i costi delle persone, dei materiali, e altre spese relative a un progetto. Il budget di progetto è approvato generalmente dagli sponsor per assicurare la disponibilità di fondi sufficienti. Una volta che il Budget di Progetto è approvato, il Product Owner e lo Scrum Master si dovranno occupare di gestirlo regolarmente e di fare in modo che le persone e le altre risorse necessarie per l'esecuzione delle attività progettuali siano disponibili.

## 8.2 Identificare lo Scrum Master e gli Stakeholder

La Figura 8-6 mostra tutti gli input, gli strumenti e gli output del processo *Identificare lo Scrum Master e gli Stakeholder*.



**Figura 8-6: Identificare lo Scrum Master e gli Stakeholder—Input, Strumenti e Output**

*Nota:* Gli asterischi (\*) indicano un input, strumento o output "obbligatorio" del processo corrispondente

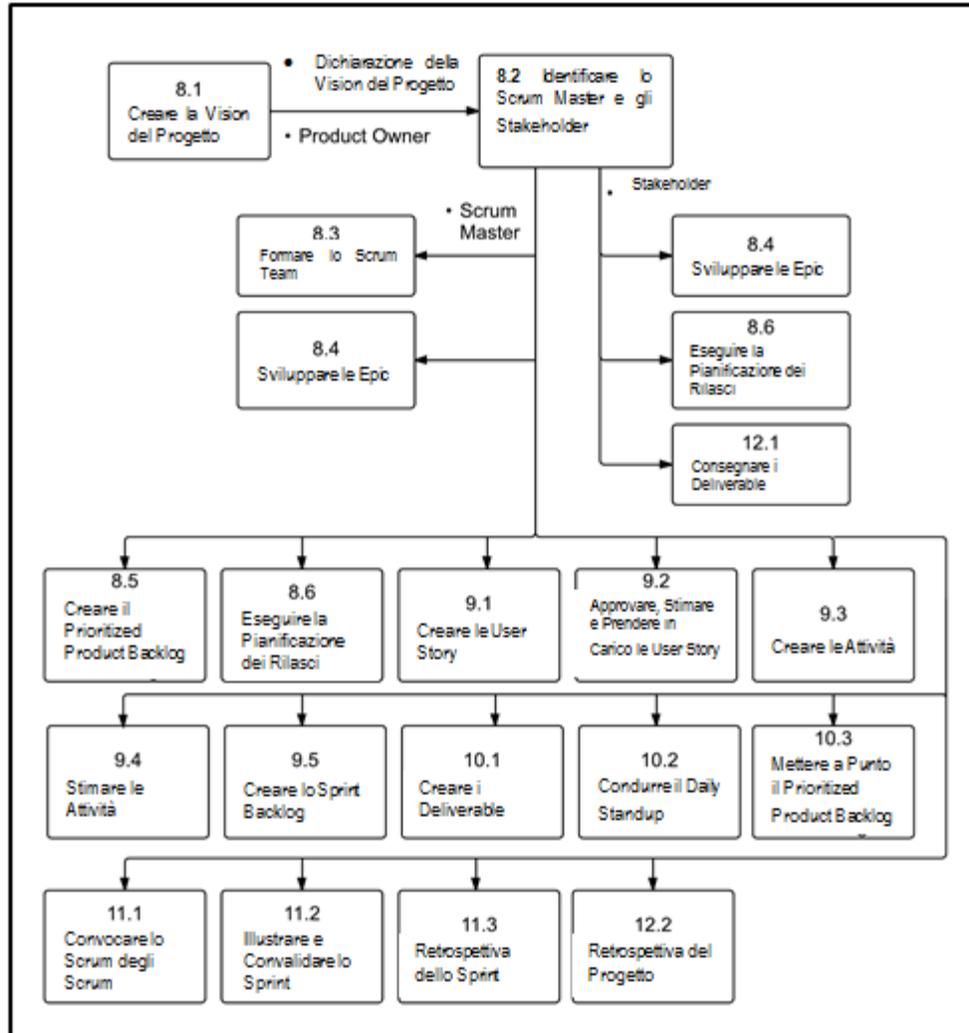


Figura 8-7: Identificare lo Scrum Master e gli Stakeholder—Diagramma di Flusso dei Dati

Note: Asterisks (\*) denote a "mandatory" input, tool, or output for the corresponding process.

## **8.2.1 Input**

### **8.2.1.1 Product Owner\***

Descritto nella sezione 8.1.3.1.

### **8.2.1.2 Dichiarazione della Vision del Progetto\***

Descritta nella sezione 8.1.3.2.

### **8.2.1.3 Program Product Owner**

Descritto nella sezione 8.1.1.2.

### **8.2.1.4 Program Scrum Master**

Descritto nella sezione 8.1.1.3.

### **8.2.1.5 Chief Product Owner**

Descritto nella sezione 8.1.1.5.

### **8.2.1.6 Chief Scrum Master**

I progetti di grandi dimensioni richiedono il lavoro contemporaneo di molteplici Scrum Team. Le informazioni raccolte da un particolare team devono essere correttamente comunicate agli altri team: responsabile di questa attività è appunto il Chief Scrum Master.

Il coordinamento dei vari Scrum Team che lavorano ad uno stesso progetto viene attuato normalmente tramite gli Scrum of Scrums (SoS) Meeting (vedere il capitolo 3, sezione 3.7.2.1). Questo tipo di riunione è simile al Daily Standup Meeting ed è facilitata dal Chief Scrum Master. Il Chief Scrum Master ha la specifica responsabilità di occuparsi degli impedimenti che hanno un impatto su più di uno Scrum Team.

### **8.2.1.7 Stakeholder del Programma**

Descritti nella sezione 8.1.1.4.

### 8.2.1.8 Requisiti delle Persone

L'identificazione dei Requisiti delle Persone è uno dei passi iniziali per la selezione dello Scrum Master e degli Stakeholder. È importante documentare i ruoli e le responsabilità di tutti coloro che saranno coinvolti nel compimento delle attività del progetto. Sono compresi tutti i soggetti che partecipano a qualsiasi titolo al progetto, a prescindere dal fatto che ricoprano un ruolo core o non-core.

Di solito, il Product Owner o lo Scrum Master lavorano insieme al Dipartimento delle Risorse Umane dell'azienda per stabilire e finalizzare i Requisiti delle Persone di un determinato progetto.

### 8.2.1.9 Disponibilità e Impegno delle Persone

Prima di selezionare lo Scrum Master e gli Stakeholder, deve essere confermata la loro disponibilità. Il principio è quello di scegliere come membri del team solo coloro che saranno disponibili e che possono dedicarsi completamente al progetto. La Disponibilità e l'Impegno delle Persone vengono normalmente rappresentati sotto forma di calendari che mostrano quando le varie risorse umane saranno disponibili a lavorare al progetto per tutta la sua durata.

Per essere realmente efficaci, gli Scrum Team dovrebbero avere idealmente dai sei ai dieci membri; inoltre negli Scrum Core Team non è consigliabile procedere alla sostituzione delle persone o al cambiamento dei membri del team. Per questo motivo, è importante che le persone dello Scrum Core Team siano disponibili e completamente dedicate al progetto.

### 8.2.1.10 Matrice Organizzativa delle Risorse

La Matrice Organizzativa delle Risorse è una rappresentazione gerarchica di una combinazione di una struttura organizzativa funzionale e di una struttura organizzativa progettuale. Le organizzazioni a matrice mettono insieme i membri del team di un progetto prendendoli da vari dipartimenti funzionali, come tecnologia dell'informazione, finanze, marketing, vendite, produzione e altri dipartimenti, creando così team interfunzionali.

I membri del team di un'organizzazione a matrice realizzano due obiettivi - quello funzionale e quello di progetto. I membri dei team sono diretti dai Product Owner per quanto riguarda le attività relative al progetto, mentre i manager funzionali eseguono le attività di gestione relative ai propri dipartimenti, come le valutazioni delle prestazioni e le approvazioni dei permessi.

### 8.2.1.11 Matrice dei Requisiti di Competenza

La Matrice dei Requisiti di Competenza, nota anche come struttura delle competenze, è utilizzata per valutare le lacune di competenza e le necessità di formazione dei membri del team. Una matrice delle competenze mappa le competenze, le capacità e il livello di interesse dei membri del team ad utilizzare quelle competenze e capacità in un determinato progetto. Attraverso l'utilizzo di questa matrice,

l'organizzazione può valutare eventuali lacune di competenze da parte dei membri del team e identificare il personale che ha bisogno di ulteriore formazione in un particolare settore o competenza.

### 8.2.1.12 Raccomandazioni dello Scrum Guidance Body

Descritte nella sezione 8.1.1.12.

## 8.2.2 Strumenti

### 8.2.2.1 Criteri di Selezione\*

La selezione degli Scrum Master appropriati e l'identificazione degli Stakeholder pertinenti è cruciale per il successo di qualsiasi progetto. In alcuni progetti, l'indicazione di alcuni membri del team e dei relativi ruoli potrebbe essere frutto della stipula di pre-condizioni.

Nel caso in cui vi sia una certa libertà nella scelta degli Scrum Master, quelli che seguono rappresentano Criteri di Selezione importanti:

1. *Competenze di problem-solving*—Questo è uno dei criteri principali da considerare nella selezione degli Scrum Master. Lo Scrum Master deve possedere le competenze e l'esperienza necessarie a consentire la rimozione di qualsiasi impedimento che ostacoli lo Scrum Team.
2. *Disponibilità*—Lo Scrum Master deve essere disponibile a programmare, sovrintendere e facilitare una molteplicità di riunioni, fra cui il Release Planning Meeting, il Daily Standup Meeting ed altre riunioni relative allo Sprint.
3. *Impegno*—Lo Scrum Master deve essere profondamente impegnato a fare in modo che lo Scrum Team possa lavorare in un ambiente favorevole per assicurare il buon esito dei progetti Scrum.
4. *Stile di Leadership del Servitore (Servant Leadership)*—Per ulteriori dettagli, si prega di fare riferimento al capitolo 3, sezione 3.10.4.1.

Nel procedere all'identificazione degli Stakeholder, è importante ricordare che in questa definizione sono ricompresi tutti i clienti, gli utenti e gli sponsor, che si interfacciano molto spesso con il Product Owner, lo Scrum Master e lo Scrum Team per fornire input e facilitare la creazione dei prodotti del progetto. Gli stakeholder influenzano il progetto per tutto il suo ciclo di vita.

### 8.2.2.2 Parere di Esperti delle Risorse Umane (HR)

Il Parere di Esperti delle Risorse Umane può essere prezioso per l'identificazione dello Scrum Master e degli Stakeholder. Il dipartimento delle Risorse Umane possiede una conoscenza qualificata dei dipendenti di un'organizzazione e dispone di varie tecniche che possono aiutare nell'identificazione dello Scrum Master e degli Stakeholder.

### 8.2.2.3 Formazione e Costi di Formazione

Scrum è un framework radicalmente diverso dai metodi tradizionali di project management. I membri del team non sempre possiedono le conoscenze e le competenze necessarie per lavorare nel contesto Scrum. Il Product Owner deve valutare le necessità di formazione dei potenziali membri del team e facilitare la formazione per superare eventuali lacune di conoscenza che affliggono il team. Il Product Owner è di norma responsabile della valutazione e selezione dei membri del team, ma spesso svolge questo compito consultandosi con lo Scrum Master, che può avere una conoscenza più approfondita delle risorse per averci lavorato insieme in altri progetti.

Ai membri degli Scrum Team deve essere garantita una formazione adeguata non solo prima della partenza del lavoro, ma anche mentre stanno lavorando ai rispettivi progetti. Inoltre i membri dello Scrum Team devono essere pronti ad imparare gli uni dagli altri e dalle persone del team che hanno più esperienza.

### 8.2.2.4 Costi delle Risorse

Una delle considerazioni principali nella scelta delle persone ha a che fare con i compromessi fra il grado di esperienza e il livello di retribuzione. Ci sono anche altri fattori legati alle persone che impattano sul costo e che vanno considerati. In linea ideale, gli Scrum Master, i membri del team e gli Stakeholder dovrebbero essere co-ubicati, in modo da poter comunicare frequentemente e con facilità. Se la co-ubicazione non è possibile e si hanno team dislocati in vari luoghi, si dovranno dedicare risorse aggiuntive al compito di facilitare le comunicazioni, comprendere le differenze culturali, sincronizzare il lavoro e incoraggiare la condivisione della conoscenza.

## 8.2.3 Output

### 8.2.3.1 Scrum Master Identificato\*

Lo Scrum Master è un facilitatore e un 'leader servitore', che fa in modo che lo Scrum Team abbia a disposizione un ambiente favorevole al buon esito del progetto. Lo Scrum Master guida, facilita e insegna le pratiche di Scrum a tutti coloro che sono coinvolti nel progetto, risolve gli impedimenti del team, e infine assicura la corretta implementazione dei processi Scrum. È una responsabilità del Product Owner identificare lo Scrum Master di un progetto Scrum. Il ruolo dello Scrum Master viene descritto in maggiore dettaglio nel capitolo 3, sezione 3.4.

### 8.2.3.2 Stakeholder Identificato\*

Gli stakeholder, termine collettivo che include i clienti, gli utenti e gli sponsor, si interfacciano frequentemente con lo Scrum Core Team e influenzano il progetto lungo tutto il processo di sviluppo del prodotto. È a favore degli stakeholder che il progetto produce i benefici della collaborazione.

Il ruolo degli stakeholder è descritto nel capitolo 3, sezione 3.3.2.

## 8.3 Formare lo Scrum Team

La Figura 8-8 mostra tutti gli input, gli strumenti e gli output del processo *Formare lo Scrum Team*.

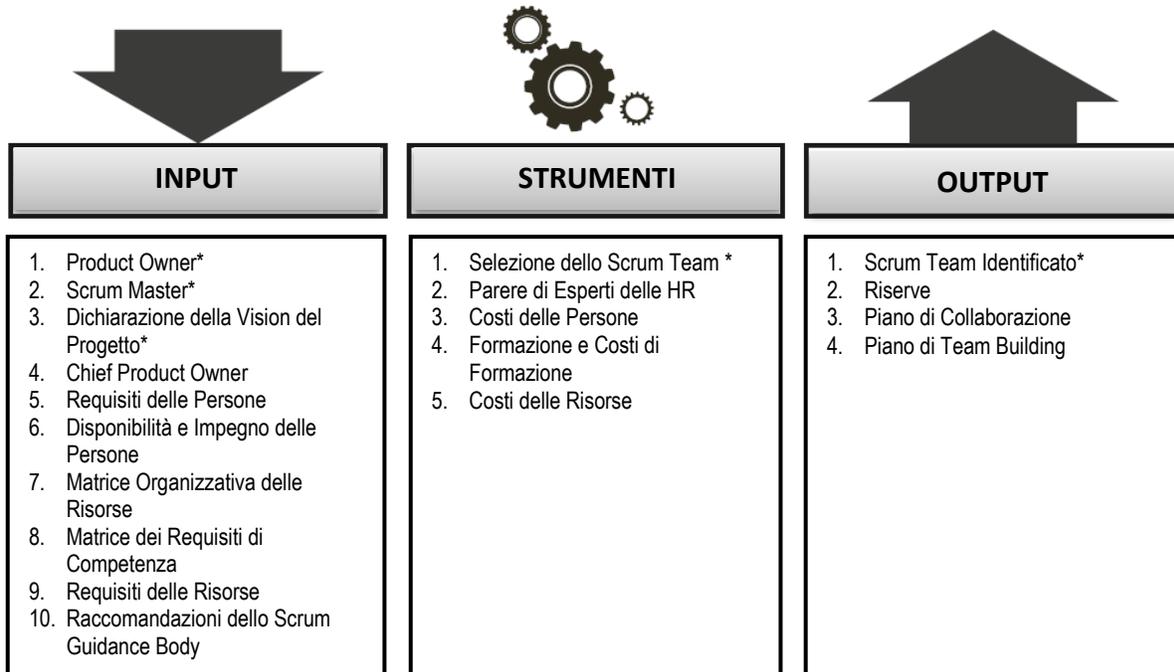


Figura 8-8: Formare lo Scrum Team—Input, Strumenti e Output

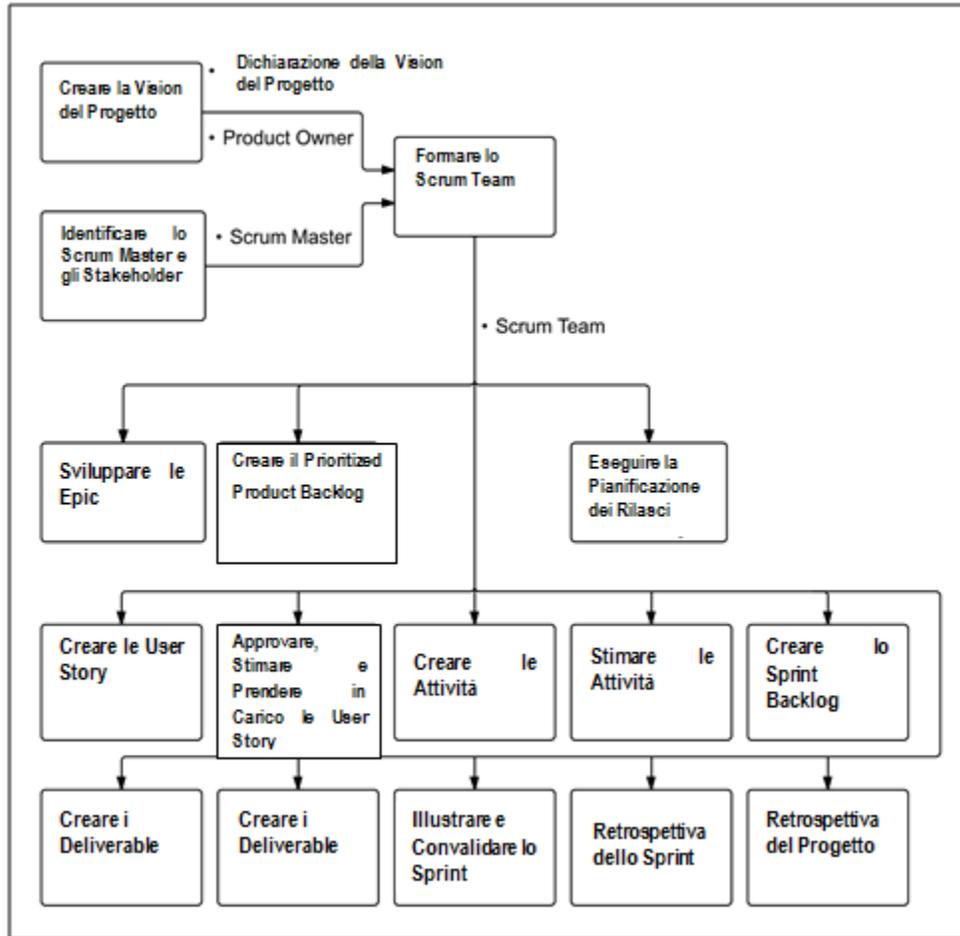


Figura 8-9: Formare lo Scrum Team—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indicano un input, strumento o output "obbligatorio" del processo corrispondente.

## **8.3.1 Input**

### **8.3.1.1 Product Owner\***

Descritto nella sezione 8.1.3.1.

### **8.3.1.2 Scrum Master\***

Descritto nella sezione 8.2.3.1.

### **8.3.1.3 Dichiarazione della Vision del Progetto\***

Descritta nella sezione 8.1.3.2.

### **8.3.1.4 Chief Product Owner**

Descritto nella sezione 8.1.1.5.

### **8.3.1.5 Requisiti delle Persone**

Descritti nella sezione 8.2.1.8.

### **8.3.1.6 Disponibilità e Impegno delle Persone**

Descritti nella sezione 8.2.1.9.

### **8.3.1.7 Matrice Organizzativa delle Risorse**

Descritta nella sezione 8.2.1.10.

### **8.3.1.8 Matrice dei Requisiti di Competenza**

Descritta nella sezione 8.2.1.11.

### **8.3.1.9 Requisiti delle Risorse**

Questi requisiti comprendono tutte le risorse - diverse dalle persone - necessarie per l'efficace funzionamento dello Scrum Team. Queste risorse includono le infrastrutture degli uffici, gli spazi per le riunioni, le attrezzature di lavoro, le Scrumboard, ecc. Nel caso di team virtuali, devono essere considerate

ulteriori risorse, come gli strumenti di collaborazione, sistemi di videoconferenza, spazi per la conservazione dei documenti condivisi, servizi di traduzione, ecc.

### **8.3.1.10 Raccomandazioni dello Scrum Guidance Body**

Descritte nella sezione 8.1.1.12.

## **8.3.2 Strumenti**

### **8.3.2.1 Selezione dello Scrum Team\***

Lo Scrum Team rappresenta il fulcro di ogni progetto Scrum e la scelta dei giusti membri del team è importante per la riuscita dei progetti Scrum. I membri dello Scrum Team sono generalisti/specialisti, nel senso che hanno una certa conoscenza in vari campi e sono esperti almeno in uno. Al di là della specifica competenza in materia, sono le competenze interpersonali che determinano il successo dei team auto-organizzati.

I membri ideali dello Scrum Team sono indipendenti, auto-motivati, incentrati sul cliente, responsabili e collaborativi. Il team deve essere in grado di promuovere un ambiente di pensiero indipendente e di decisioni di gruppo, in modo da tirare fuori il massimo dei benefici dal tipo di struttura.

### **8.3.2.2 Parere di Esperti delle Risorse Umane (HR)**

Il Parere di Esperti delle Risorse Umane (HR) può essere prezioso nella costituzione di uno Scrum Team. Il dipartimento delle Risorse Umane possiede una conoscenza qualificata dei dipendenti di un'organizzazione e dispone di varie tecniche che possono aiutare il Product Owner, lo Scrum Master e gli sponsor nell'identificazione dei membri giusti per il team.

### **8.3.2.3 Costi delle Persone**

Tutti i costi associati ai requisiti delle persone devono essere valutati, analizzati, approvati e messi a budget.

### **8.3.2.4 Formazione e Costi di Formazione**

Può accadere che i membri del team non possiedano le competenze o le conoscenze necessarie per portare a termine le attività specialistiche. Il Product Owner deve valutare le necessità di formazione dei potenziali membri del team e provvedere alla opportuna formazione, quando emergono carenze di competenza o conoscenza.

Per un'implementazione realmente efficace di Scrum, deve sussistere all'interno dell'organizzazione un grado significativo di consapevolezza dei principi e dei valori di Scrum. Questa consapevolezza aiuterà a

mettere in esecuzione Scrum con successo. Lo Scrum Team deve essere sensibilizzato e formato sulle pratiche di Scrum e lo Scrum Master deve svolgere il ruolo di istruttore del team. Dal momento che la pianificazione degli Sprint rappresenta un importante fattore di successo, la formazione aiuterà il team a capire come discutere e identificare obiettivi raggiungibili per lo Sprint. Lo Scrum Master deve tirare fuori il meglio dallo Scrum Team motivandolo e facilitando il processo di sviluppo. Attraverso la formazione e il coaching, lo Scrum Master può aiutare i membri del team ad esporre le problematiche e le sfide che si trovano di fronte. Di norma qualsiasi questione o conflitto di cui il team fa esperienza al suo interno viene risolta dal team stesso con il supporto e l'assistenza dello Scrum Master, se necessario. Lo Scrum Master deve affrontare le questioni come il morale basso o la mancanza di coordinamento nel team. Lui o lei ha la responsabilità di rimuovere gli impedimenti del team. Quando necessario, lo Scrum Master può rimettere al management la soluzione o rimozione delle questioni e degli impedimenti esterni.

La Formazione e i Costi di Formazione sono trattati anche nel processo *Identificare lo Scrum Master e gli Stakeholder*, sezione 8.2.2.3.

### **8.3.2.5 Costi delle Risorse**

I costi associati ai requisiti non legati alle persone devono essere valutati, analizzati, approvati e messi a budget. Una risorsa nel contesto di un progetto è tutto ciò che viene usato per eseguire un compito o un'attività, ivi compresi – ma non solo – le attrezzature, il materiale, i servizi esterni e gli spazi fisici.

## **8.3.3 Output**

### **8.3.3.1 Scrum Team Identificato\***

Lo Scrum Team, a volte chiamato anche Team di Sviluppo, è un gruppo o team di persone che ha la responsabilità di capire i requisiti di business specificati dal Product Owner, stimare le User Story e infine creare i Deliverable del progetto. Gli Scrum Team sono interfunzionali e auto-organizzati. Il team decide la quantità di lavoro da prendere in carico in uno Sprint e stabilisce il modo migliore per eseguirlo. Lo Scrum Team è composto di membri provenienti da più funzioni, che portano a termine tutto il lavoro relativo alla creazione di Deliverable potenzialmente consegnabili, fra cui quello di sviluppo, di testing, di garanzia di qualità, ecc. L'identificazione dello Scrum Team è una responsabilità del Product Owner, spesso svolta di concerto con lo Scrum Master.

Il ruolo dello Scrum Team è descritto in maggior dettaglio nel capitolo 3, sezione 3.6.

### **8.3.3.2 Riserve**

Un altro aspetto importante nella selezione dei team è la creazione di una riserva per ciascun membro dello Scrum Team. Anche se la disponibilità e l'impegno delle persone vengono confermati in anticipo, possono insorgere problemi come una malattia, un'emergenza familiare, oppure un membro del team che lascia

l'organizzazione. Gli Scrum Team lavorano in piccoli gruppi di sei-dieci persone. Disporre di Riserve evita cali importanti di produttività dovuti alla perdita di un membro del team.

### **8.3.3.3 Piano di Collaborazione**

La collaborazione è un elemento molto importante in Scrum. Pianificare in che modo i vari organi decisionali, gli stakeholder e i membri del team si devono impegnare e devono collaborare fra di loro è vitale. Il Piano di Collaborazione è un output opzionale che può assumere una veste formale o informale. A volte può trattarsi semplicemente di un'intesa orale fra i vari stakeholder, dal momento che Scrum evita qualsiasi documentazione che non sia necessaria. Tuttavia, per i progetti più grandi e complessi, soprattutto quelli con team dislocati in vari luoghi, può essere necessario approntare un accordo più formale. Il piano può riguardare le modalità di comunicazione e di collaborazione fra i membri dello Scrum Core Team, gli Stakeholder e gli altri soggetti coinvolti nel progetto Scrum durante il progetto stesso e può inoltre definire specifici strumenti o tecniche da utilizzare a tale scopo. Per esempio, nei team dislocati in vari luoghi, può esserci la necessità di formalizzare un accordo su quando e come saranno condotte le riunioni, quale genere di strumenti di comunicazione sarà utilizzato e chi dovrà partecipare ai vari tipi di riunione.

8

### **8.3.3.4 Piano di Team Building**

Dal momento che uno Scrum Team è interfunzionale, ogni membro deve partecipare attivamente a tutti gli aspetti del progetto. Lo Scrum Master deve identificare le questioni che riguardano i membri del team e affrontarle in modo rigoroso per mantenere salda l'efficacia del team.

Per costruire la coesione del team, lo Scrum Master deve fare in modo che ci siano relazioni positive fra i membri del team e che questi siano compatti nel voler realizzare gli obiettivi del progetto complessivo e dell'organizzazione, generando in questo modo una maggiore efficienza e produttività.

In questo contesto, è utile approfondire lo studio del capitolo 3, sezione 3.10, che tratta delle più diffuse teorie sulle Risorse Umane e della loro rilevanza per Scrum.

## 8.4 Sviluppare le Epic

La Figura 8-10 mostra tutti gli input, gli strumenti e gli output del processo *Sviluppare le Epic*.

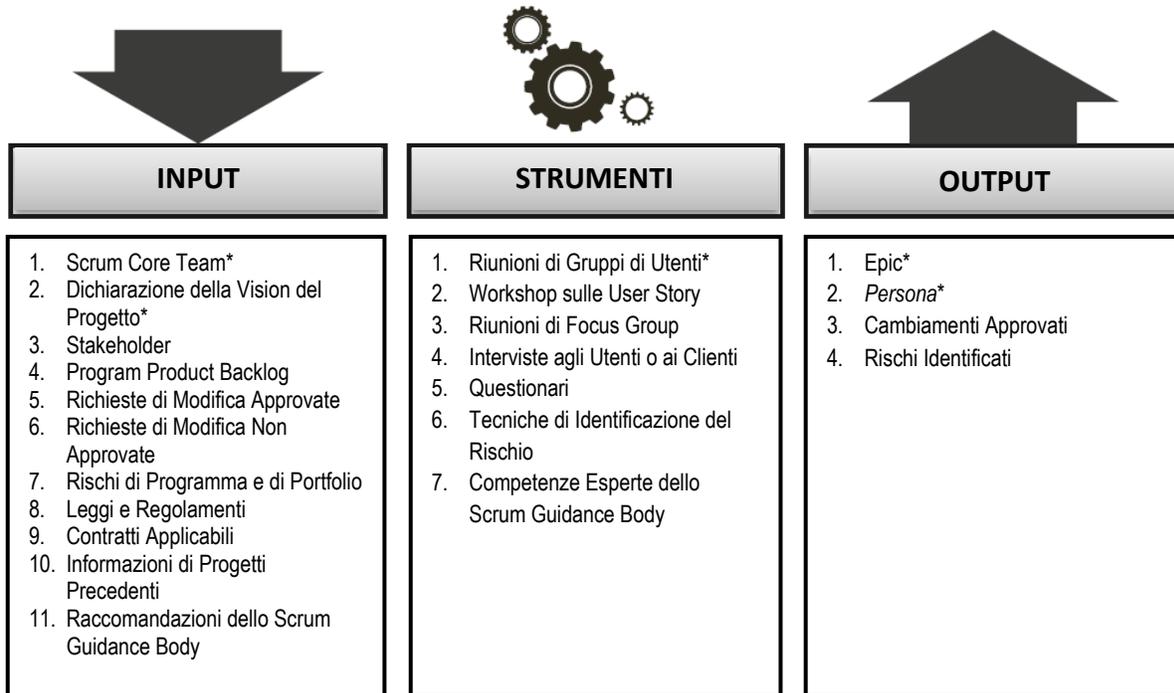


Figura 8-10: Sviluppare le Epic—Input, Strumenti e Output

*Nota:* Gli asterischi (\*) indicano un input, strumento o output "obbligatorio" del processo corrispondente

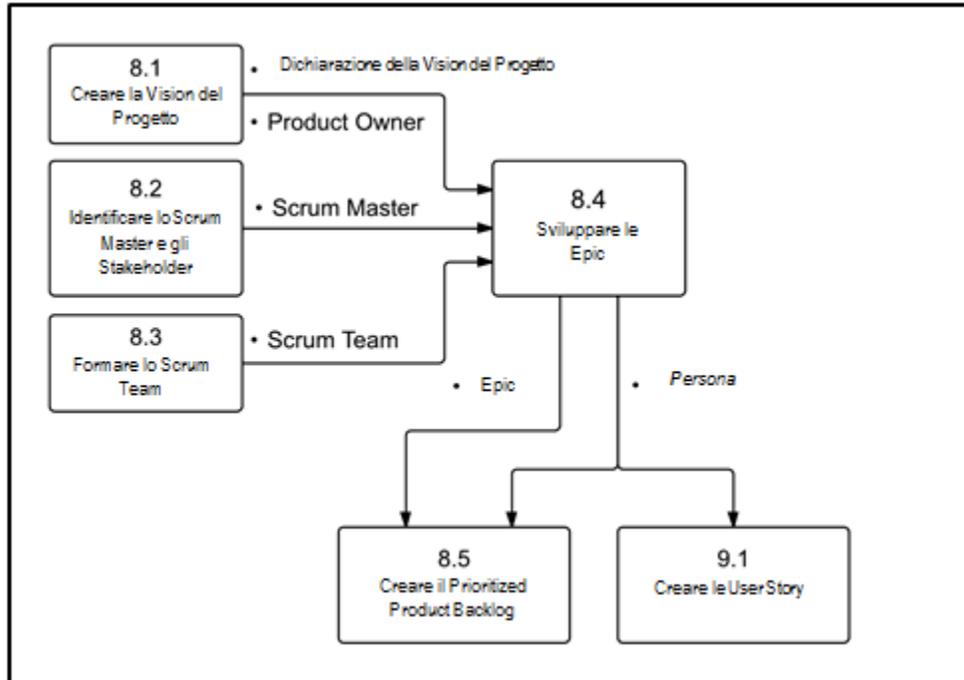


Figura 8-11: Sviluppare le Epic—Diagramma di Flusso dei Dati

## 8.4.1 Input

### 8.4.1.1 Scrum Core Team\*

Lo Scrum Core Team è composto dallo Scrum Team, dallo Scrum Master e dal Product Owner, così come descritto nel capitolo 3, sezione 3.3.1.

### 8.4.1.2 Dichiarazione della Vision del Progetto\*

Descritta nella sezione 8.1.3.2.

### 8.4.1.3 Stakeholder

Descritti nella sezione 8.2.3.2.

### 8.4.1.4 Program Product Backlog

Descritto nella sezione 8.1.1.6.

#### 8.4.1.5 Richieste di Modifica Approvate

Le Richieste di Modifica Approvate provenienti dal programma o dal portfolio costituiscono input da aggiungere alla lista dei cambiamenti approvati del progetto che saranno implementati in Sprint futuri. Ciascun cambiamento può richiedere la sua personale Epic o User Story e potrebbe diventare un input del processo *Sviluppare le Epic*. Le Richieste di Modifica Approvate per questo processo possono provenire anche da altri processi Scrum. Le Richieste di Modifica e le Richieste di Modifica Approvate sono trattate nel capitolo 6, sezioni 6.3.1, 6.4.2.1 e 6.6.

#### 8.4.1.6 Richieste di Modifica Non Approvate

Le richieste di procedere a dei cambiamenti sono di solito presentate come Richieste di Modifica e rimangono nello stato di 'non approvate' fino a quando non sono formalmente approvate. Le Richieste di Modifica Non Approvate per il processo *Sviluppare le Epic* possono provenire dai processi *Creare i Deliverable*, *Condurre il Daily Standup* e da altri processi.

Le Richieste di Modifica e le Richieste di Modifica Non Approvate sono trattate nel capitolo 6, sezioni 6.3.1, 6.4.2.1 e 6.6.

#### 8.4.1.7 Rischi di Programma e di Portfolio

I rischi relativi a un portfolio o a un programma impatteranno anche i progetti che ne fanno rispettivamente parte. Se, durante la valutazione dei rischi nell'ambito dei portfolio e dei programmi, si stabilisce che un rischio può incidere su uno specifico progetto, devono essere comunicate al Product Owner e allo Scrum Team le informazioni pertinenti riguardo a quel rischio. I rischi di Programma e di Portfolio possono costituire input del processo *Sviluppare le Epic* e possono avere un impatto generale sul modo di eseguire il processo.

I Rischi di Programma e di Portfolio sono descritti nel capitolo 7, sezione 7.5.1.

#### 8.4.1.8 Leggi e Regolamenti

È possibile che Leggi e Regolamenti imposti dagli organi di governo incidano sulla pianificazione ed esecuzione di alcuni tipi di progetto. Le leggi non dipendono dall'organizzazione e sono imposte da entità governative. I regolamenti possono essere invece sia interni che esterni. I regolamenti interni sono quelli che trovano applicazione nell'ambito dell'azienda, di solito sulla base di politiche specifiche. Questi regolamenti possono riguardare i sistemi di gestione della qualità, disposizioni finanziarie, disposizioni sul personale, ecc. I regolamenti esterni sono quelli relativi a standard, norme e requisiti stabiliti da organi di governo pubblico.

Nello sviluppare le Epic si deve tenere conto delle Leggi e dei Regolamenti. Le Epic si basano sui requisiti di business. Per soddisfare tali requisiti il team di progetto deve rispettare Leggi e Regolamenti sia interni che esterni.

A volte, alcune delle Leggi e dei Regolamenti che impattano molteplici progetti Scrum possono essere inserite come parte delle Raccomandazioni dello Scrum Guidance Body, trattate nella sezione 8.1.1.12.

#### 8.4.1.9 Contratti Applicabili

Se l'intero progetto o alcune sue parti sono eseguiti in forza di un contratto, è quest'ultimo che definisce l'ambito del lavoro e i termini specifici di esecuzione. Il tipo di contratto utilizzato ha un'influenza sul rischio di progetto.

Alcuni dei tipi più comuni di contratto utilizzati nei progetti Scrum sono i seguenti:

**Contratto a Consegna Incrementale**—Questo contratto prevede punti di controllo a intervalli regolari. Tale forma aiuta il cliente o gli stakeholder a prendere decisioni sullo sviluppo del prodotto periodicamente nel corso del progetto, in corrispondenza di ciascun punto di controllo. Il cliente può accettare lo sviluppo del prodotto, oppure decidere di fermare lo sviluppo del prodotto, o infine richiedere modifiche al prodotto.

**Contratto in Joint Venture**—Questo contratto viene generalmente utilizzato quando due o più parti si associano per eseguire il lavoro di un progetto. Tutte le parti coinvolte nel progetto realizzeranno un Ritorno sull'Investimento, poiché i ricavi o benefici generati saranno condivisi fra i partecipanti.

**Contratto di Sviluppo per Fasi**—Questo contratto rende i fondi disponibili ogni mese o ogni trimestre, dopo che è stato completato con successo un rilascio. Questa tipologia offre un incentivo sia al cliente che al fornitore e assicura che il rischio monetario del cliente sia limitato a quel particolare periodo di tempo, in quanto i rilasci non andati a buon fine non vengono finanziati.

**Contratto a Incentivo e Penale**—Questi contratti si basano sull'accordo che il fornitore sarà ricompensato con un incentivo finanziario se i prodotti del progetto verranno consegnati nei tempi previsti, ma sarà soggetto al pagamento di penali se la consegna avverrà in ritardo.

Altri tipi di contratto diffusi sono il contratto di pagamento in base alle funzionalità consegnate, il contratto 'tempo e materiali', il contratto a prezzo fisso, a ambito fisso e a guadagno fisso.

Le Epiche dovrebbero essere sviluppate tenendo presenti i termini e le condizioni del tipo di contratto che si utilizza.

#### 8.4.1.10 Informazioni di Progetti Precedenti

Le informazioni e le idee acquisite da analoghi precedenti progetti dell'organizzazione sono input preziosi per lo sviluppo delle Epiche e la valutazione del rischio. Le Informazioni di Progetti Precedenti possono includere gli appunti del project manager, i registri di progetto e le considerazioni degli stakeholder.

Alcune informazioni e best practice relative alle informazioni di progetti precedenti possono essere disponibili come Raccomandazioni dello Scrum Guidance Body.

### 8.4.1.11 Raccomandazioni dello Scrum Guidance Body

Trattate nella sezione 8.1.1.12.

Le Raccomandazioni dello Scrum Guidance Body possono contenere informazioni su norme, regolamenti, standard e best practice per sviluppare le Epic.

## 8.4.2 Strumenti

### 8.4.2.1 Riunioni di Gruppi di Utenti\*

Le Riunioni di Gruppi di Utenti coinvolgono gli stakeholder pertinenti (principalmente gli utenti e i clienti del prodotto). Queste riunioni forniscono allo Scrum Core Team informazioni di prima mano riguardo alle aspettative degli utenti. Ciò aiuta a formulare i Criteri di Accettazione del prodotto e offre idee preziose per lo sviluppo delle Epic. Le Riunioni di Gruppi di Utenti sono essenziali per prevenire dispendiose rilavorazioni che possono dipendere da una mancanza di chiarezza riguardo alle aspettative e ai requisiti. Queste riunioni promuovono la piena adesione (il c.d. *buy-in*) al progetto e creano una comprensione comune fra lo Scrum Core Team e gli Stakeholder pertinenti.

### 8.4.2.2 Workshop sulle User Story

Come parte del processo *Sviluppare le Epic* vengono tenuti dei workshop sulle User Story. Queste sessioni, nelle quali è coinvolto l'intero Scrum Core Team e a cui è a volte auspicabile che partecipino anche altri Stakeholder, sono facilitate dallo Scrum Master. Questi workshop aiutano il Product Owner a prioritizzare i requisiti e permettono allo Scrum Core Team di avere una prospettiva condivisa dei Criteri di Accettazione. Assicurano inoltre che le Epic e le User Story descrivano le funzionalità dal punto di vista degli utenti, siano facili da comprendere e possano essere stimate in modo affidabile. I Workshop sulle User Story sono utili per la comprensione delle aspettative dell'utente riguardo ai deliverable e sono eccellenti per il team building. Inoltre facilitano la preparazione della pianificazione dello Sprint successivo. Un Workshop sulle User Story costituisce una buona piattaforma per discutere e spiegare tutti gli elementi di un prodotto e spesso per addentrarsi nei più piccoli dettagli, in modo da fare piena chiarezza.

### 8.4.2.3 Riunioni di Focus Group

I focus group riuniscono varie persone in una sessione guidata per raccogliere le loro opinioni, impressioni, o valutazioni di un prodotto, servizio o risultato desiderato. I membri dei focus group possono liberamente farsi delle domande a vicenda e ottenere chiarimenti su particolari argomenti o concetti. Grazie alle domande, alla critica costruttiva e al feedback, i focus group portano ad un prodotto di migliore qualità e perciò contribuiscono alla soddisfazione delle aspettative degli utenti. A volte in queste riunioni si raggiunge il consenso in alcune aree, mentre in altre può permanere una differenza di opinioni o punti di vista. Nel

caso in cui i membri del gruppo abbiano opinioni o punti di vista differenti, si compie ogni sforzo per superare le differenze e raggiungere il consenso.

Le sessioni di focus group aiutano i team a tirare fuori idee innovative, a risolvere i problemi e a dare suggerimenti per il miglioramento. Queste riunioni agevolano l'accertamento dei fatti e generano idee e feedback da parte dei potenziali utenti e degli sviluppatori del prodotto. Di solito vengono tenute per pianificare, valutare o migliorare un prodotto o servizio. Le opinioni ricavate da queste riunioni possono inoltre essere di aiuto nello sviluppo delle Epic e delle User Story. A volte, si tengono Riunioni di Focus Group anche per risolvere le questioni che possono emergere durante lo sviluppo delle Epic.

#### **8.4.2.4 Interviste agli Utenti o ai Clienti**

Per acquisire il contesto e la visione necessari a sviluppare le Epic, è importante il coinvolgimento degli stakeholder, fra cui lo sponsor, gli utenti e i clienti del prodotto. Il tempo passato a intervistare utenti e clienti si tradurrà nella sicurezza che i requisiti trasfusi nelle Epic siano allineati con la complessiva Vision del Progetto, offrendo in tal modo un valore maggiore.

Queste interviste aiutano a:

- Identificare e comprendere i bisogni e le aspettative degli stakeholder
- Raccogliere opinioni e fatti
- Capire il punto di vista degli stakeholder riguardo al prodotto finale
- Raccogliere il feedback sul prodotto iterato o parzialmente sviluppato

#### **8.4.2.5 Questionari**

Un modo efficace per ottenere opinioni statistiche quantitative e qualitative da un gran numero di utenti o clienti è quello di utilizzare i sondaggi o Questionari. Un Questionario è uno strumento di ricerca che contiene domande da porre a un intervistato per accumulare informazioni riguardo ad una questione o a un argomento specifici. I Questionari possono essere auto-somministrati o somministrati da un intervistatore.

Deve essere posta molta cura nella formulazione dei Questionari, scegliendo i destinatari corretti e stabilendo un metodo appropriato per la distribuzione del sondaggio al fine di evitare errori e distorsioni.

Il Product Owner o lo Scrum Master possono condurre un sondaggio mentre sviluppano le Epic per raccogliere informazioni pertinenti dagli stakeholder o dallo Scrum Team.

#### **8.4.2.6 Tecniche di Identificazione del Rischio**

Descritte nel capitolo 7, sezione 7.4.1.1.

### 8.4.2.7 Competenze Esperte dello Scrum Guidance Body

Descritte nel capitolo 3, sezione 3.3.2.

Con riferimento al processo *Creare le Epic*, le Competenze Esperte dello Scrum Guidance Body possono essere collegate a norme o regolamenti documentati; oppure a standard o best practice per la creazione delle Epic. È anche possibile che ci sia un team di esperti della materia che può assistere il Product Owner nella creazione delle Epic. Questo team può essere formato da Analisti di Business, Capi Progettisti, Sviluppatori Senior, Esperti di Scrum, o altre persone con esperienza. Questo gruppo di esperti di solito non è lo stesso team che resterà fisso a lavorare su un progetto specifico, in quanto tendono a muoversi da uno progetto all'altro durante la 'fase di vendita' o 'fase zero' con i clienti o gli utenti.

## 8.4.3 Output

### 8.4.3.1 Epic\*

Le Epic sono scritte nelle fasi iniziali del progetto quando la maggior parte delle User Story sono solo funzionalità di alto livello o descrizioni del prodotto e i requisiti sono definiti per sommi capi. Si tratta in sostanza di ampie User Story non rifinite inserite nel Prioritized Product Backlog.

Quando queste Epic emergono dal Prioritized Product Backlog per essere completate in uno Sprint imminente, vengono allora scomposte in User Story più piccole e granulari. Queste User Story più piccole sono in genere funzionalità semplici, corte e facili da implementare o blocchi di attività da completare in uno Sprint.

### 8.4.3.2 *Persona*\*

Le *Persona* sono personaggi immaginari molto dettagliati, che rappresentano la maggioranza degli utenti e degli altri stakeholder che non necessariamente utilizzano direttamente il prodotto finale. Le *Persona* sono create per identificare i bisogni della base di utenti target. La creazione di specifiche *Persona* può aiutare il team a comprendere meglio gli utenti e i loro requisiti e obiettivi. Basandosi su una *Persona*, il Product Owner può prioritizzare in modo più efficace le funzionalità per dare vita al Prioritized Product Backlog.

**Creare una *Persona*:** Questa attività implica l'assegnazione di un nome di fantasia e possibilmente di una foto, come un'immagine di repertorio, ad un personaggio. La *Persona* avrà attributi molto specifici come l'età, il genere, il livello di istruzione, l'ambiente in cui agisce, gli interessi e gli obiettivi. Si può inoltre inserire una citazione che illustra i requisiti della *Persona*. Qui sotto viene riportato un esempio di *Persona* per un sito web di viaggi.

*Esempio:*

Vanessa è una donna di 39 anni che risiede a San Francisco. Dopo una carriera di grande successo come avvocato, sta ora coltivando la sua passione per i viaggi. Le piace disporre di varie opzioni quando prende in esame i servizi di trasporto aereo e di alloggio, in modo da poter scegliere il migliore e il più conveniente. I siti web lenti e caotici la scoraggiano.

### 8.4.3.3 Cambiamenti Approvati

Le Richieste di Modifica non Approvate possono essere approvate dal Product Owner durante il processo *Sviluppare le Epic*, a volte con i suggerimenti forniti dagli stakeholder pertinenti. Questi cambiamenti vengono classificati come Cambiamenti Approvati e possono essere prioritizzati e implementati negli Sprint futuri.

Le Richieste di Modifica e le Richieste di Modifica Approvate sono trattate nel capitolo 6, sezioni 6.3.1, 6.4.2.1 e 6.6.

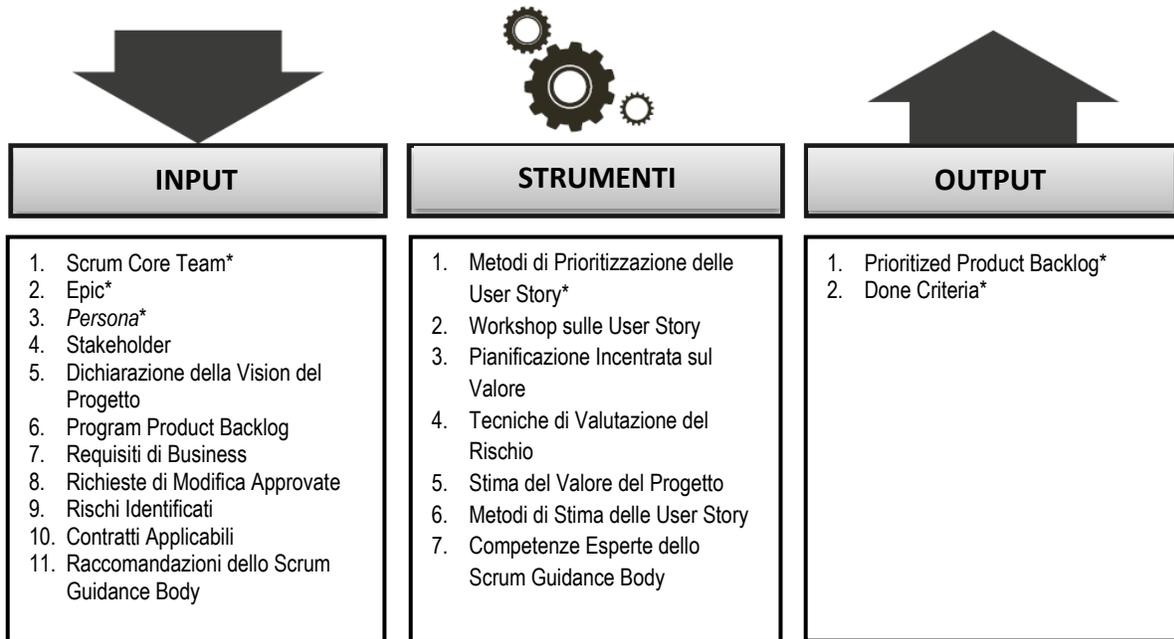
### 8.4.3.4 Rischi Identificati

Durante la creazione delle Epic, possono essere identificati nuovi rischi e questi Rischi Identificati costituiscono un output importante di questa fase. Tali rischi contribuiscono allo sviluppo del Prioritized Product Backlog (detto anche Risk Adjusted Product Backlog).

L'Identificazione dei Rischi viene descritta nel capitolo 7, sezione 7.4.1.

## 8.5 Creare il Prioritized Product Backlog

La Figura 8-12 mostra tutti gli input, gli strumenti e gli output del processo *Creare il Prioritized Product Backlog*.



**Figura 8-12: Creare il Prioritized Product Backlog—Input, Strumenti e Output**

*Nota:* Gli asterischi (\*) indicano un input, strumento o output "obbligatorio" del processo corrispondente.

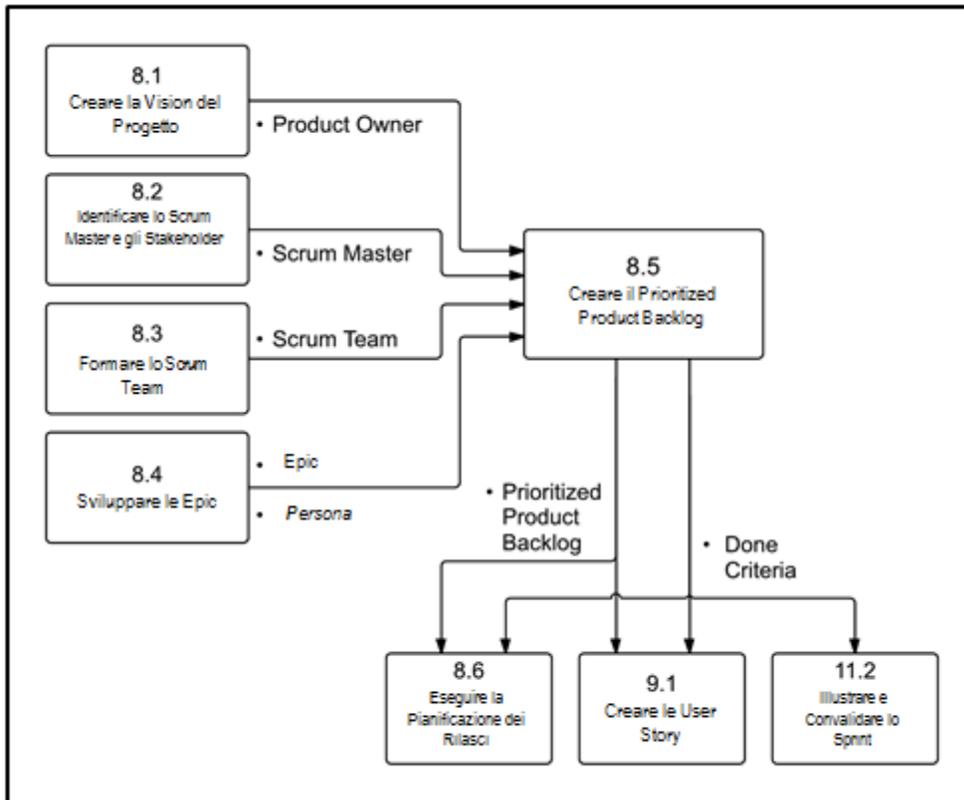


Figura 8-13: Creare il Prioritized Product Backlog—Diagramma di Flusso dei Dati

## 8.5.1 Input

### 8.5.1.1 Scrum Core Team\*

Descritto nella sezione 8.4.1.1.

### 8.5.1.2 Epic\*

Descritte nella sezione 8.4.3.1.

### 8.5.1.3 Persona\*

Descritte nella sezione 8.4.3.2.

### 8.5.1.4 Stakeholder

Descritti nella sezione 8.2.3.2.

#### **8.5.1.5 Dichiarazione della Vision del Progetto**

Descritta nella sezione 8.1.3.2.

#### **8.5.1.6 Program Product Backlog**

Descritto nella sezione 8.1.1.6.

#### **8.5.1.7 Requisiti di Business**

L'insieme di tutti i dati acquisiti grazie ai vari strumenti, come le interviste agli utenti o ai clienti, i Questionari, le JAD Session, la Gap Analysis, l'Analisi SWOT e altre riunioni, aiuta a sviluppare una visione più chiara dei requisiti di business e agevola la creazione del Prioritized Product Backlog.

#### **8.5.1.8 Richieste di Modifica Approvate**

Descritte nella sezione 8.4.3.3.

#### **8.5.1.9 Rischi Identificati**

Descritti nella sezione 8.4.3.4.

#### **8.5.1.10 Contratti Applicabili**

Descritti nella sezione 8.4.1.9.

#### **8.5.1.11 Raccomandazioni dello Scrum Guidance Body**

Trattate nella sezione 8.1.1.12.

Con riferimento al processo *Creare il Prioritized Product Backlog*, le Raccomandazioni dello Scrum Guidance Body possono contenere informazioni su norme, regolamenti, standard e best practice per lo sviluppo del Prioritized Product Backlog.

## 8.5.2 Strumenti

### 8.5.2.1 Metodi di Prioritizzazione delle User Story\*

Di seguito sono presentate alcune delle tecniche utilizzate per prioritizzare le User Story o i requisiti del Prioritized Product Backlog in base al valore di business:

- **Schema di Prioritizzazione MoSCoW** —Lo schema di prioritizzazione MoSCoW prende il suo nome dalle prime lettere delle frasi “Must have” (“Si deve avere”), “Should have” (“Si dovrebbe avere”) “Could have” (“Si potrebbe avere”) e “Won’t have” (“Non si avrà”). Questo metodo di prioritizzazione è in genere più efficace degli schemi semplici. Le etichette sono in ordine di priorità decrescente, ove le User Story di tipo “Must have” sono quelle senza le quali il prodotto non avrà alcun valore e le User Story “Won’t have” sono quelle che, sebbene sarebbe bello avere, non è necessario includere.
- **Comparazione a Coppie**—In questa tecnica, si prepara una lista di tutte le User Story presenti nel Prioritized Product Backlog. Successivamente ciascuna User Story viene presa singolarmente e comparata con le altre User Story della lista, una alla volta. Ogni volta che sono messe a confronto due User Story, viene presa una decisione riguardo a quale delle due sia più importante. Grazie a questo procedimento si può produrre una lista prioritizzata di User Story.
- **Metodo dei 100 Punti**—Il Metodo dei 100 Punti fu sviluppato da Dean Leffingwell e Don Widrig nel 2003. Consiste nel mettere a disposizione del cliente 100 punti che può utilizzare per votare per le User Story più importanti. L’obiettivo è quello di dare un peso maggiore a quelle User Story che hanno una priorità più alta se comparate alle altre User Story disponibili. Ciascun membro del gruppo assegna i punti alle varie User Story, dando più punti a quelle che ritiene più importanti. Alla fine del processo di votazione, si stabilisce l’ordine di priorità calcolando i punti assegnati a ciascuna User Story.
- **Analisi di Kano**

Descritta nel capitolo 4, sezione 4.5.2.

### 8.5.2.2 Workshop sulle User Story

Descritti nella sezione 8.4.2.2.

### 8.5.2.3 Pianificazione Incentrata sul Valore

Descritta nel capitolo 4, sezione 4.5.2.

#### 8.5.2.4 Tecniche di Valutazione del Rischio

Descritte nel capitolo 7, sezione 7.4.2.1.

#### 8.5.2.5 Stima del Valore del Progetto

Descritta nel capitolo 4, sezione 4.5.1.

#### 8.5.2.6 Metodi di Stima delle User Story

Quando si crea il Prioritized Product Backlog, per produrre stime di alto livello delle Epic possono essere usati tutti gli strumenti utilizzati per il processo *Approvare, Stimare e Prendere in Carico le User Story* (così come descritti nel capitolo 9, sezione 9.2.2). Alcuni importanti strumenti sono:

1. Riunioni di Gruppi di Utenti
2. Planning Poker
3. Fist of Five
4. Punti per la Stima dei Costi
5. Altre Tecniche di Stima

#### 8.5.2.7 Competenze Esperte dello Scrum Guidance Body

Descritte nella sezione 8.4.2.7.

Con riferimento a questo processo, le Competenze Esperte dello Scrum Guidance Body possono essere collegate a norme o regolamenti documentati; oppure a standard o best practice per la creazione delle Epic. Può anche essere presente un team di esperti della materia che può assistere il Product Owner nel processo *Creare il Prioritized Product Backlog*. Questo team può essere formato da Analisti di Business, Capi Progettisti, Sviluppatori Senior, Esperti di Scrum, o altre persone con esperienza. Questo gruppo di esperti di solito non è lo stesso team che resterà fisso a lavorare su un progetto specifico, in quanto tendono a muoversi da uno progetto all'altro durante la 'fase di vendita' o 'fase zero' con i clienti o gli utenti.

### 8.5.3 Output

#### 8.5.3.1 Prioritized Product Backlog\*

Il Product Owner sviluppa un Prioritized Product Backlog che contiene un elenco prioritizzato dei requisiti di business e di progetto scritti sotto forma di Epic, le quali sono User Story di alto livello. Il Prioritized Product Backlog si basa su tre fattori principali: valore, rischio o incertezza e dipendenze. Viene chiamato anche Risk Adjusted Product Backlog, in quanto include i rischi identificati e valutati relativi al progetto. Comprende

anche tutti i Cambiamenti Approvati che possono essere prioritizzati in modo adeguato nel Prioritized Product Backlog (così come spiegato nel capitolo 6, sezione 6.3.1).

- **Valore**—È una responsabilità del Product Owner garantire la consegna in primo luogo di quei prodotti che forniscono il valore di business più alto. Anche un prodotto di grandissimo valore può non far parte del primo rilascio se ce ne sono altri di valore ancora più alto che bastano per un primo rilascio.
- **Rischio e Incertezza**—Maggiore è l'incertezza, più rischioso è il progetto. Per questo è importante assegnare la priorità più alta ai prodotti più rischiosi del Prioritized Product Backlog. I prodotti che portano con sé il livello di rischio maggiore richiederanno inoltre delle azioni di mitigazione del rischio. L'assegnazione di una priorità di esecuzione a queste azioni di mitigazione del rischio si traduce in un Risk Adjusted Product Backlog. Trattare i rischi nelle prime fasi del progetto non garantisce il successo del progetto, ma aumenta la capacità del team di fronteggiare il rischio stesso. Questo aspetto è spiegato nel capitolo 7, sezione 7.4.3.
- **Dipendenze**—Normalmente non è possibile creare un Prioritized Product Backlog nel quale non vi siano dipendenze fra le User Story. I requisiti funzionali spesso dipendono da altri requisiti funzionali e anche non funzionali. Queste dipendenze possono incidere sul modo di prioritizzare le User Story all'interno del Prioritized Product Backlog. Due dei modi più comuni di risolvere le dipendenze sono dividere una singola User Story in più parti oppure unificare le User Story interdipendenti.
- **Stime**—Nel Prioritized Product Backlog sono disponibili anche le stime di alto livello delle Epic.

### 8.5.3.2 Done Criteria\*

I Done Criteria sono un complesso di regole valide per tutte le User Story. È fondamentale pervenire ad una definizione chiara di ciò che può ritenersi 'Done', cioè 'Fatto', perché questo consente di eliminare eventuali ambiguità nei requisiti e aiuta il team a conformarsi alle norme di qualità obbligatorie. Questa definizione chiara viene utilizzata per creare appunto i Done Criteria, che sono un output del processo *Creare il Prioritized Product Backlog*. Una User Story si considera 'Done' (cioè fatta) quando è illustrata al e approvata dal Product Owner, che la giudica sulla base dei Done Criteria e dei Criteri di Accettazione della User Story.

*Esempio di Done Criteria:*

Progetto: Ideare le nuove varianti di una famosa auto sportiva presso la LRA Ltd.

## Done Criteria:

- La progettazione è approvata dalla divisione di Eccellenza Tecnica.
- Il prototipo passa tutti i test della galleria del vento ordinati dalla divisione Aerodinamica.
- La progettazione è approvata per la produzione dalla divisione Proprietà Intellettuale.
- Le aspettative di sicurezza della progettazione sono confermate dal rapporto sulla Sicurezza della Progettazione della Divisione Sicurezza.
- Il rapporto sulla Stima dei Costi di progettazione è approvato dalla divisione Finanze.

## 8.6 Eseguire la Pianificazione dei Rilasci

La Figura 8-14 mostra tutti gli input, gli strumenti e gli output del processo *Eseguire la Pianificazione dei Rilasci*.



Figura 8-14: Eseguire la Pianificazione dei Rilasci—Input, Strumenti e Output

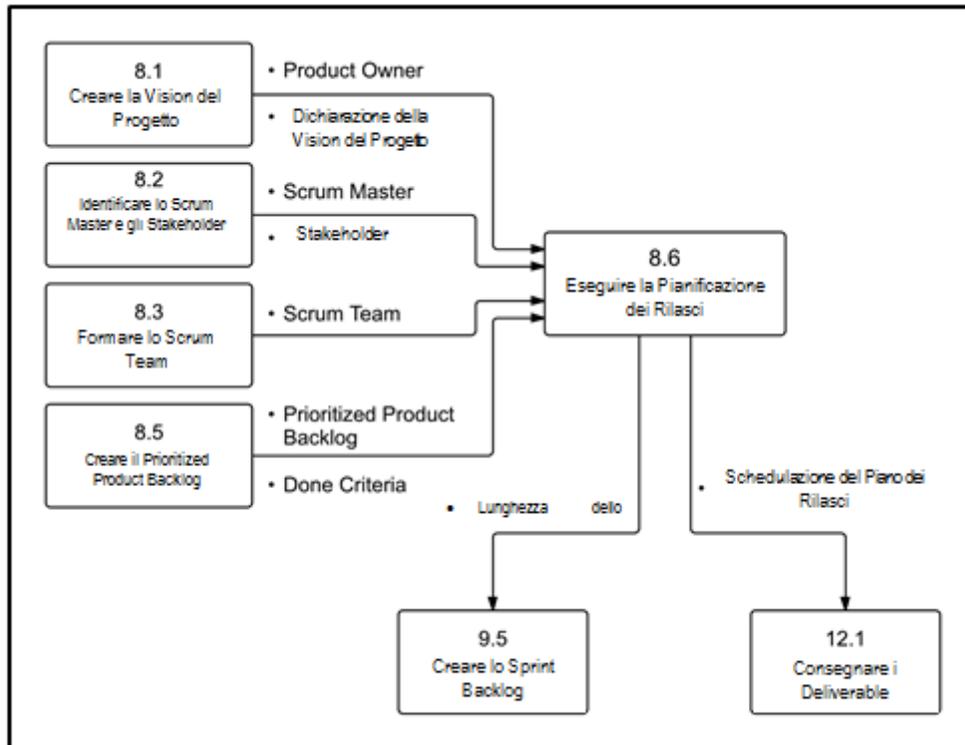


Figura 8-15: Eseguire la Pianificazione dei Rilasci—Diagramma di Flusso dei Dati

## 8.6.1 Input

### 8.6.1.1 Scrum Core Team\*

Descritto nella sezione 8.4.1.1.

### 8.6.1.2 Stakeholder\*

Descritti nella sezione 8.2.3.2.

### 8.6.1.3 Dichiarazione della Vision del Progetto\*

Descritta nella sezione 8.1.3.2.

### 8.6.1.4 Prioritized Product Backlog\*

Descritto nella sezione 8.5.3.1.

#### **8.6.1.5 Done Criteria\***

Descritti nella sezione 8.5.3.2.

#### **8.6.1.6 Program Product Owner**

Descritto nella sezione 8.1.1.2.

#### **8.6.1.7 Program Scrum Master**

Descritto nella sezione 8.1.1.3.

#### **8.6.1.8 Chief Product Owner**

Descritto nella sezione 8.1.1.5.

#### **8.6.1.9 Program Product Backlog**

Descritto nella sezione 8.1.1.6.

#### **8.6.1.10 Requisiti di Business**

Descritti nella sezione 8.5.1.7.

#### **8.6.1.11 Calendario delle Festività**

È importante per lo Scrum Team tenere traccia delle date chiave e della disponibilità di tutti i membri del team. Questo si può ottenere con l'uso di un calendario condiviso che fornisce informazioni sulle festività ufficiali, sulle ferie, sui viaggi pianificati, sugli eventi, ecc. Questo calendario aiuterà il team a pianificare ed eseguire gli Sprint.

#### **8.6.1.12 Raccomandazioni dello Scrum Guidance Body**

Descritte nella sezione 8.1.1.12.

Con riferimento al processo *Eseguire la Pianificazione dei Rilasci*, le Raccomandazioni dello Scrum Guidance Body possono collegarsi a norme, regolamenti, standard e best practice per lo sviluppo del Piano dei Rilasci. Il Guidance Body può rappresentare l'autorità più adatta a definire le linee guida relative al valore di business, alle aspettative sui rilasci, alle strategie di distribuzione, alla qualità e alla sicurezza.

## 8.6.2 Strumenti

### 8.6.2.1 Sessioni di Pianificazione dei Rilasci\*

Per sviluppare un Piano dei Rilasci vengono condotte delle Sessioni di Pianificazione dei Rilasci. Il piano stabilisce quando saranno consegnati al cliente i vari gruppi di funzionalità o prodotti utilizzabili. In Scrum, l'obiettivo principale di un Release Planning Meeting è quello di permettere allo Scrum Team di avere una visione d'insieme dei rilasci e del programma di consegna del prodotto che il team sta sviluppando, in modo da potersi allineare alle aspettative del Product Owner e degli stakeholder pertinenti (primo fra tutti lo sponsor del progetto).

Molte organizzazioni hanno una strategia riguardo al rilascio dei prodotti. Alcune preferiscono la distribuzione continua, nella quale si fa un rilascio dopo la creazione di una specifica funzionalità utilizzabile. Altre preferiscono la distribuzione per fasi, nella quale i rilasci vengono fatti a intervalli predefiniti. A seconda della strategia adottata dall'organizzazione, le Sessioni di Pianificazione dei Rilasci nei progetti possono essere basate sulle funzionalità, nel qual caso l'obiettivo è consegnare un rilascio ogni volta che viene sviluppato un gruppo prestabilito di funzionalità; oppure la pianificazione può essere basata sulle date, nel qual caso il rilascio avviene in date prestabilite.

Dal momento che il framework Scrum promuove il processo decisionale iterativo e basato sulle informazioni raccolte piuttosto che la pianificazione anticipata di dettaglio praticata nello stile tradizionale di project management a cascata, le Sessioni di Pianificazione dei Rilasci non devono produrre un Piano dei Rilasci dettagliato per l'intero progetto. Tale piano può essere aggiornato continuamente man mano che sono disponibili informazioni rilevanti.

### 8.6.2.2 Metodi di Prioritizzazione dei Rilasci\*

Per sviluppare un Piano dei Rilasci vengono utilizzati dei Metodi di Prioritizzazione dei Rilasci. Questi metodi sono propri di ciascun settore industriale e organizzazione e sono di solito stabiliti dal gruppo dirigente dell'organizzazione.

## 8.6.3 Output

### 8.6.3.1 Schedulazione del Piano dei Rilasci\*

Una Schedulazione del Piano dei Rilasci è uno degli output fondamentali del processo *Eseguire la Pianificazione dei Rilasci*. Tale schedulazione stabilisce quali deliverable devono essere rilasciati ai clienti, insieme agli intervalli pianificati e alle date dei rilasci. Non è detto che vi sia un rilascio programmato alla fine di ogni Sprint. A volte, un rilascio può essere previsto dopo il completamento di un gruppo di Sprint. A seconda della strategia adottata dall'organizzazione, le sessioni di Pianificazione dei Rilasci nei progetti possono essere basate sulle funzionalità, nel qual caso l'obiettivo è effettuare una consegna ogni volta che viene sviluppato un gruppo prestabilito di funzionalità; oppure la pianificazione può essere basata sulle date,

nel qual caso il rilascio avviene in date prestabilite. Il deliverable dovrebbe essere rilasciato quando offre al cliente un valore di business sufficiente.

### **8.6.3.2 Lunghezza dello Sprint\***

Il Product Owner e lo Scrum Team fissano la Lunghezza dello Sprint dello specifico progetto basandosi sui vari input, tra cui i requisiti di business e la Schedulazione del Piano dei rilasci. Una volta che è stata stabilita, la Lunghezza dello Sprint rimane invariata per tutto il progetto.

Tuttavia, la Lunghezza dello Sprint può essere modificata se e come il Product Owner e lo Scrum Team ritengono opportuno. Nelle prime fasi del progetto è ancora possibile fare delle prove per trovare la Lunghezza dello Sprint ideale. Nel prosieguo del progetto, una modifica della Lunghezza dello Sprint di norma sta a significare che è possibile ridurla grazie a miglioramenti verificatisi nell'ambiente di progetto.

Uno Sprint può avere una durata predeterminata da 1 e 6 settimane. Tuttavia, per ottenere il massimo beneficio da un progetto Scrum, è sempre consigliabile fissare la durata in 4 settimane, salvo che per progetti con requisiti molto stabili, nei quali la durata degli Sprint può essere estesa fino a 6 settimane.

L'impatto dei Cambiamenti Attesi sulla Lunghezza dello Sprint viene descritto nel capitolo 6, sezione 6.5.1.

### **8.6.3.3 Clienti Target del Rilascio**

Non tutti i rilasci saranno rivolti a tutti gli stakeholder o a tutti gli utenti. Gli stakeholder possono scegliere di limitare determinati rilasci ad un sottoinsieme di utenti. Il Piano dei Rilasci deve specificare i clienti target (cioè destinatari) del singolo rilascio.

### **8.6.3.4 Prioritized Product Backlog Rifinito**

In questo processo si può rifinire il Prioritized Product Backlog, sviluppato nel processo *Creare il Prioritized Product Backlog*. Dopo le Sessioni di Pianificazione dei Rilasci, condotte dallo Scrum Team con gli stakeholder, le User Story del Prioritized Product Backlog possono infatti risultare più chiare.

## 8.7 Diagramma di Flusso dei Dati della Fase

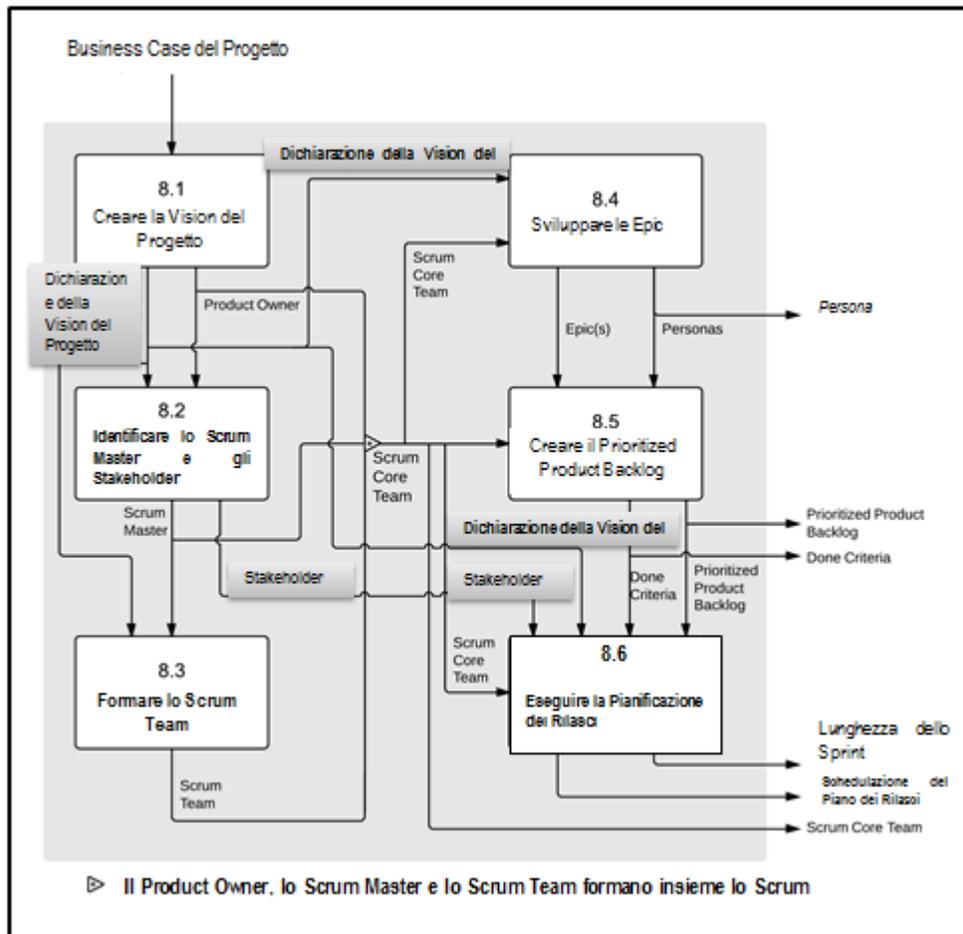


Figura 8-16: Fase di Inizio—Diagramma di Flusso dei Dati

## 9. PIANIFICAZIONE E STIMA

La fase di Pianificazione e Stima riguarda i processi relativi alle attività di pianificazione e stima, che sono: *Creare le User Story, Approvare, Stimare e Prendere in Carico le User Story, Creare le Attività, Stimare le Attività e Creare lo Sprint Backlog.*

La fase di *Pianificazione e Stima*, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Per facilitare la migliore applicazione del framework Scrum, questo capitolo identifica gli input, gli strumenti e gli output di ciascun processo qualificandoli come “obbligatori” o “opzionali”. Gli input, gli strumenti e gli output contrassegnati da un asterisco (\*) sono obbligatori, mentre quelli senza asterisco sono opzionali.

Si raccomanda allo Scrum Team e a tutti coloro che vengono introdotti al framework e ai processi Scrum di concentrarsi soprattutto sugli input, gli strumenti e gli output obbligatori; i Product Owner, gli Scrum Master e altri professionisti di Scrum più esperti dovrebbero invece sforzarsi di raggiungere una conoscenza più approfondita delle informazioni contenute nell'intero capitolo. È altresì importante avere chiaro che, sebbene tutti i processi siano definiti singolarmente nella *Guida SBOK™*, non è necessario eseguirli in maniera sequenziale o separatamente. A volte, a seconda dei requisiti specifici di ciascun progetto, potrebbe essere più appropriato unire alcuni processi fra loro.

Questo capitolo è scritto dal punto di vista di un singolo Scrum Team che lavora ad un singolo Sprint per produrre Deliverable potenzialmente consegnabili come parte di un progetto più ampio. Tuttavia, le informazioni fornite sono ugualmente applicabili a interi progetti, programmi e portfolio. Ulteriori informazioni sull'uso di Scrum nei progetti, programmi e portfolio sono disponibili nei capitoli da 2 a 7, che trattano i principi e gli aspetti di Scrum.

La Figura 9-1 fornisce una visione d'insieme dei processi della fase di Pianificazione e Stima, che sono:

**9.1 Creare le User Story**—In questo processo vengono create le User Story e i loro relativi Criteri di Accettazione. Le User Story sono scritte di solito dal Product Owner e sono volte ad assicurare la chiara rappresentazione dei requisiti del cliente e la piena comprensione di tali requisiti da parte di tutti gli stakeholder. Si possono tenere dei Workshop di Scrittura delle User Story per coinvolgere i membri dello

Scrum Team nella creazione delle User Story. Le User Story sono incorporate all'interno del Prioritized Product Backlog.

**9.2 Approvare, Stimare e Prendere in Carico le User Story**—In questo processo il Product Owner approva le User Story di uno Sprint. Dopo di che, lo Scrum Master e lo Scrum Team stimano l'impegno necessario per sviluppare la funzionalità descritta in ciascuna User Story. Infine, lo Scrum Team si impegna a consegnare i requisiti del cliente sotto forma di User Story Approvate, Stimate e Prese in Carico.

**9.3 Creare le Attività**—In questo processo, le User Story Approvate, Stimate e Prese in Carico sono scomposte in specifiche attività e trascritte in una Lista delle Attività. A questo fine viene spesso tenuto un Task Planning Meeting.

**9.4 Stimare le Attività**—In questo processo lo Scrum Core Team, durante i Task Estimation Meeting, stima l'impegno necessario per compiere ciascuna attività della Lista delle Attività. Il risultato di questo processo è una Lista delle Attività con Impegno Stimato.

**9.5 Creare lo Sprint Backlog**—In questo processo lo Scrum Core Team tiene degli Sprint Planning Meeting, durante i quali il gruppo crea uno Sprint Backlog che contiene tutte le attività da completare nello Sprint.

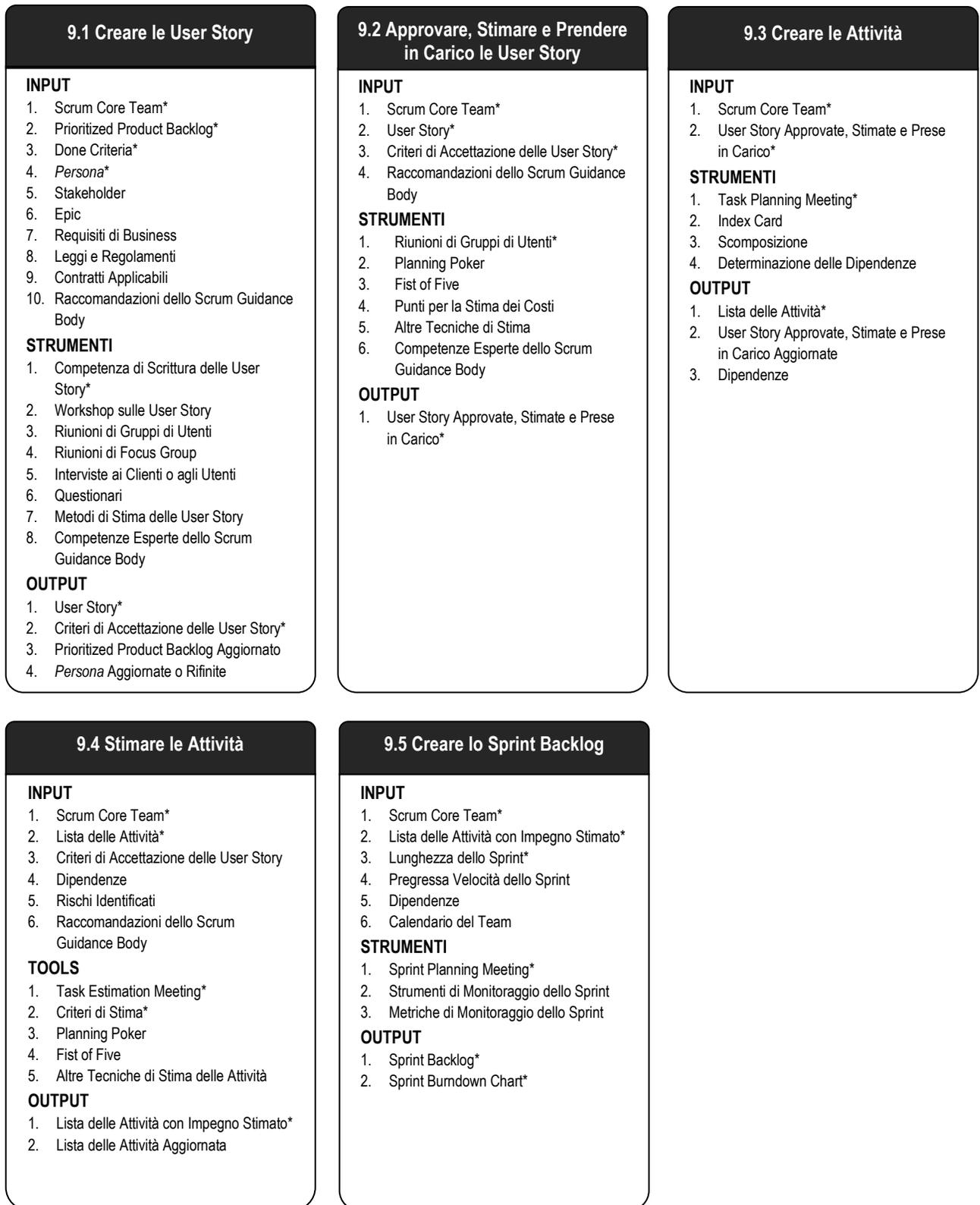


Figura 9-1: Visione d'insieme della Fase di Pianificazione e Stima

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

La Figura 9-2 sotto riportata mostra gli input, gli strumenti e gli output obbligatori dei processi della fase di Pianificazione e Stima.

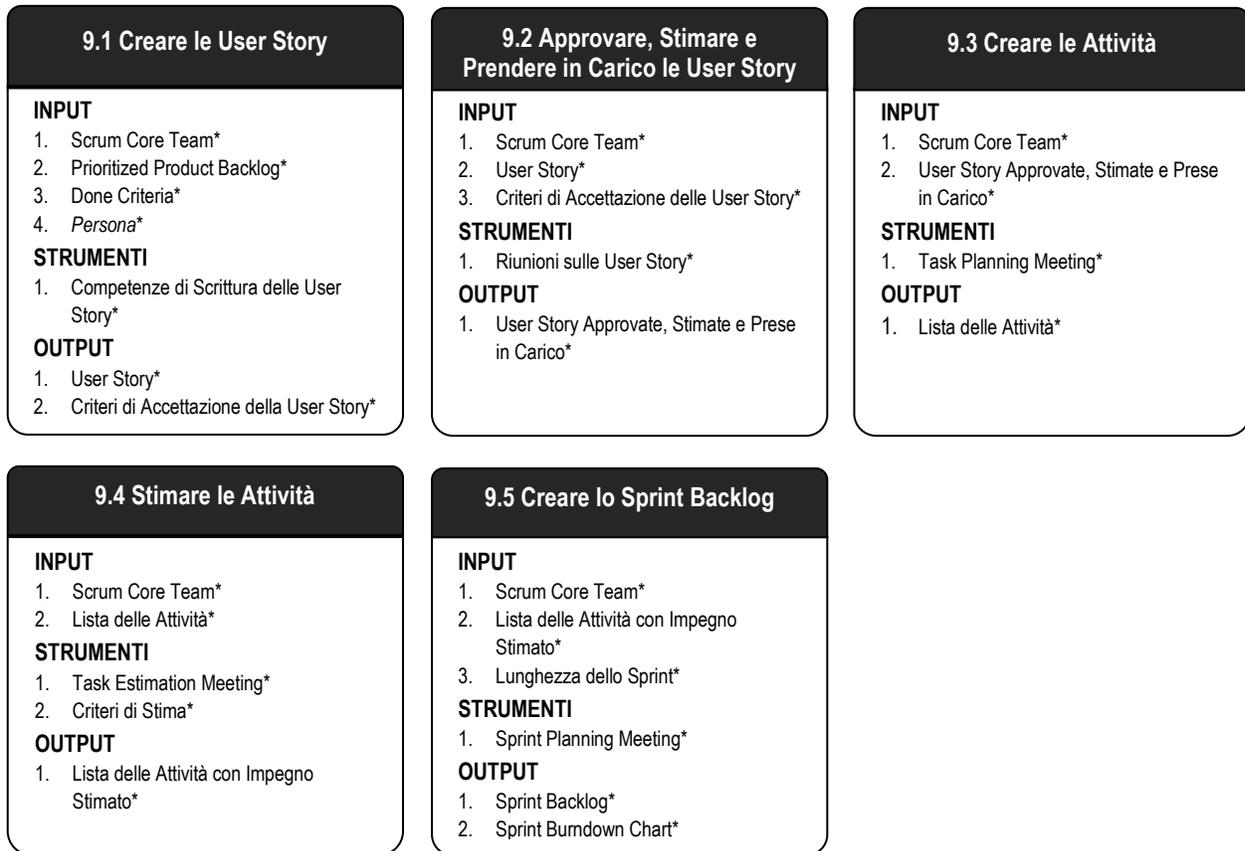


Figura 9-2: Visione d'insieme della Fase di Pianificazione e Stima (Elementi essenziali)

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente

## 9.1 Creare le User Story

La Figura 9-3 mostra tutti gli input, gli strumenti e gli output del processo *Creare le User Story*.

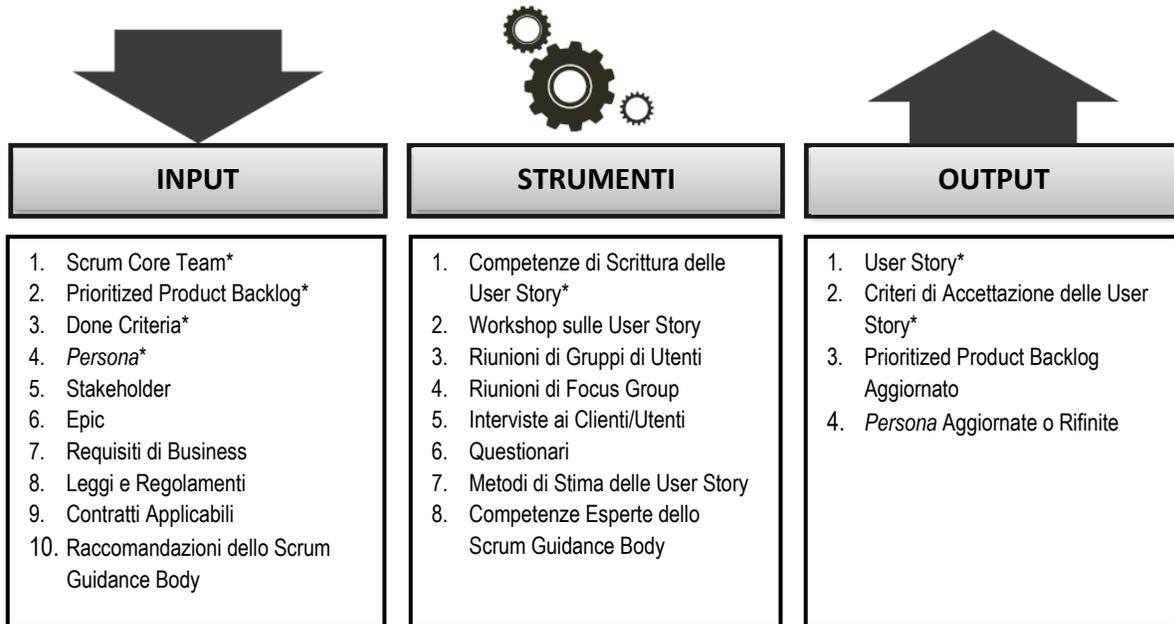


Figura 9-3: Creare le User Story—Input, Strumenti e Output

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

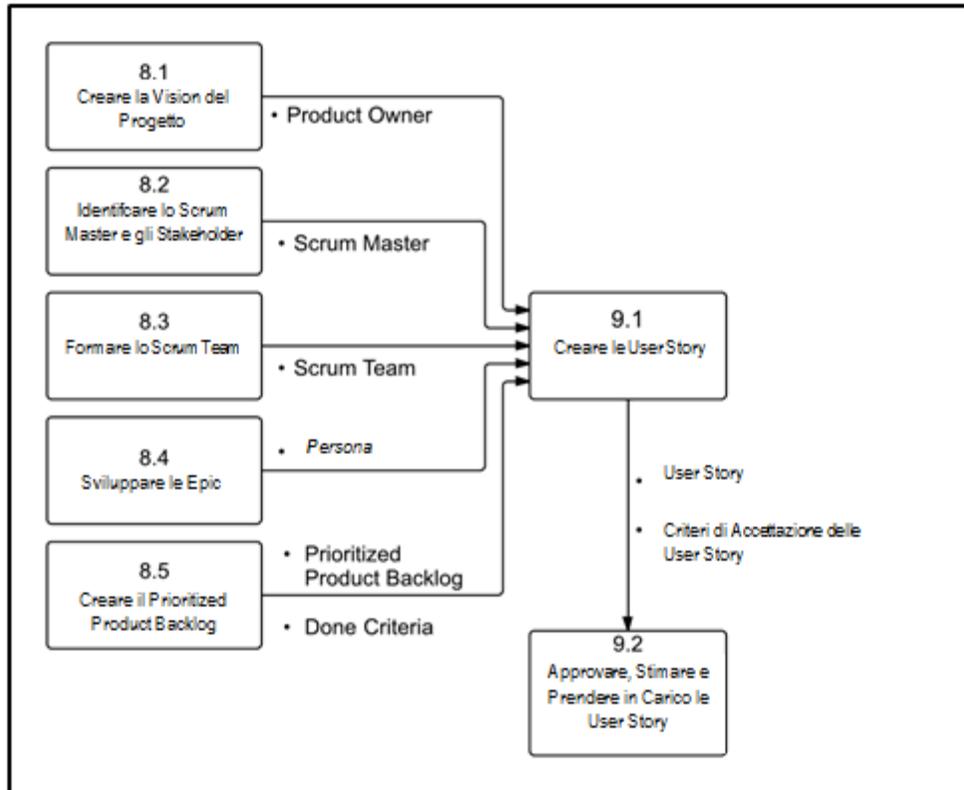


Figura 9-4: Creare le User Story—Diagramma di Flusso dei Dati

## 9.1.1 Input

### 9.1.1.1 Scrum Core Team\*

Descritto nel capitolo 8, sezione 8.4.1.1.

### 9.1.1.2 Prioritized Product Backlog\*

Descritto nel capitolo 8, sezione 8.5.3.1.

### 9.1.1.3 Done Criteria\*

Descritti nel capitolo 8, sezione 8.5.3.2.

### 9.1.1.4 Persona\*

Descritte nel capitolo 8, sezione 8.4.3.2.

#### **9.1.1.5 Stakeholder**

Descritti nel capitolo 8, sezione 8.2.3.2.

#### **9.1.1.6 Epic**

Descritte nel capitolo 8, sezione 8.4.3.1.

#### **9.1.1.7 Requisiti di Business**

Descritti nel capitolo 8, sezione 8.5.1.7.

#### **9.1.1.8 Leggi e Regolamenti**

Descritti nel capitolo 8, sezione 8.4.1.8.

#### **9.1.1.9 Contratti Applicabili**

Descritti nel capitolo 8, sezione 8.4.1.9.

#### **9.1.1.10 Raccomandazioni dello Scrum Guidance Body**

Descritte nel capitolo 8, sezione 8.1.1.12.

Con riferimento al processo *Creare le User Story*, le Raccomandazioni dello Scrum Guidance Body possono contenere informazioni su norme, regolamenti, standard e best practice necessari per creare User Story efficaci.

### **9.1.2 Strumenti**

#### **9.1.2.1 Competenze di Scrittura delle User Story\***

Il Product Owner, basandosi sull'interazione con gli stakeholder, sulla sua conoscenza e competenza di business e sugli input forniti dal team, sviluppa le User Story che andranno a costituire il Prioritized Product Backlog iniziale del progetto. Il Prioritized Product Backlog rappresenta la somma totale di tutto ciò che deve essere completato nel progetto. L'obiettivo di questo esercizio è quello di creare User Story elaborate e rifinite che possono essere approvate, stimate e prese in carico dallo Scrum Team. A volte, il Product Owner può farsi assistere da un Analista di Business per la scrittura delle User Story.

Anche se la responsabilità principale della scrittura delle User Story ricade sul Product Owner, che spesso svolge questo compito da solo, volendo si può tenere un Workshop sulla Scrittura delle User Story.

### 9.1.2.2 Workshop sulle User Story

Descritti nel capitolo 8, sezione 8.4.2.2.

### 9.1.2.3 Riunioni di Gruppi di Utenti

Descritti nel capitolo 8, sezione 8.4.2.1.

### 9.1.2.4 Riunioni di Focus Group

Le Riunioni dei Focus Group rappresentano una tecnica di tipo qualitativo per valutare e comprendere i bisogni e le aspettative degli utenti rispetto ad un prodotto che viene proposto. Per formare il focus group viene selezionato un piccolo gruppo di utenti. Questo gruppo può essere scelto in modo casuale fra una vasta platea di utenti oppure può essere scelto appositamente per rappresentare tutte le principali *Persona* destinatarie del prodotto. Le Riunioni dei Focus Group generalmente si conformano ad uno schema determinato, secondo il quale si pongono al gruppo delle domande che vengono poi discusse fra i partecipanti. Ogni Riunione di Focus Group può avere le sue personali regole di discussione secondo quanto deciso dagli organizzatori. Queste riunioni sono di solito tenute alla presenza di un moderatore.

### 9.1.2.5 Interviste ai Clienti o agli Utenti

Descritte nel capitolo 8, sezione 8.4.2.4.

### 9.1.2.6 Questionari

Descritti nel capitolo 8, sezione 8.4.2.5.

### 9.1.2.7 Metodi di Stima delle User Story

Per creare stime di alto livello delle Epic al momento di dare vita al Prioritized Product Backlog si possono utilizzare tutti gli strumenti usati per il processo *Approvare, Stimare e Prendere in Carico le User Story* (così come descritti nella sezione 9.2.2). Alcuni importanti strumenti sono:

1. Riunioni di Gruppi di Utenti
2. Planning Poker
3. Fist of Five
4. Punti per la Stima dei Costi
5. Altre Tecniche di Stima

### 9.1.2.8 Competenze Esperte dello Scrum Guidance Body

Descritte nel capitolo 8, sezione 8.4.2.7.

Con riferimento al processo *Creare le User Story*, le Competenze Esperte dello Scrum Guidance Body Expertise possono essere collegate a norme o regolamenti documentati; oppure a standard o best practice per la creazione delle User Story. Può anche essere presente un team di esperti della materia che può assistere il Product Owner o fornirgli delle indicazioni su come creare le User Story. Questo team può essere formato da Analisti di Business, Capi Progettisti, Sviluppatori Senior, Esperti di Scrum, o altre persone con esperienza. Questo gruppo di esperti di solito non è lo stesso team che resterà fisso a lavorare su un progetto specifico, in quanto tendono a muoversi da un progetto all'altro per fornire istruzioni ai vari Scrum Team, se necessario.

## 9.1.3 Output

### 9.1.3.1 User Story\*

Le User Story si conformano ad una specifica struttura predefinita e costituiscono un modo semplice di documentare i requisiti e le funzionalità desiderate dall'utente finale. Una User Story ci racconta tre cose riguardo ai requisiti: Chi, Cosa e Perché. I requisiti espressi nelle User Story sono dichiarazioni brevi, semplici e di facile comprensione. Il formato standard predefinito si traduce in un miglioramento della comunicazione fra gli stakeholder e in stime più precise da parte del team. Alcune User Story possono essere troppo grandi da gestire all'interno di un solo Sprint. Queste ampie User Story sono spesso chiamate Epic. Una volta che un'Epic viene scelta dal Prioritized Product Backlog per essere completata in uno Sprint imminente, viene ulteriormente scomposta in User Story più ridotte.

Il Prioritized Product Backlog è un elenco dinamico che viene continuamente aggiornato a causa dell'attività di ripriorizzazione e dell'introduzione, dell'aggiornamento, della rifinitura e a volte della cancellazione delle User Story. Questi aggiornamenti del backlog sono di norma il risultato di cambiamenti dei requisiti di business.

Per saperne di più sul Prioritized Product Backlog si può fare riferimento anche al capitolo 8, sezione 8.5.3.1.

*Formato della User Story:*

Nella mia qualità di <ruolo/persona>, dovrei essere in grado di <requisito> in modo da <beneficio>.

*Esempio di User Story:*

Nella mia qualità di Amministratore del Database, dovrei essere in grado di tornare indietro di un determinato numero di aggiornamenti del database in modo da poter ripristinare la versione del database desiderata.

### 9.1.3.2 Criteri di Accettazione delle User Story\*

Ciascuna User Story ha dei Criteri di Accettazione ad essa associati. Le User Story sono soggettive, per cui i Criteri di Accettazione forniscono l'obiettività necessaria per poter considerare la User Story come 'Done' (fatta) o 'not Done' (incompiuta) in sede di Revisione dello Sprint. I Criteri di Accettazione chiariscono al team cosa ci si aspetta da una User Story, eliminano eventuali ambiguità presenti nei requisiti e aiutano ad allinearsi alle aspettative. Il Product Owner stabilisce e comunica i Criteri di Accettazione allo Scrum Team. Durante gli Sprint Review Meeting, i Criteri di Accettazione forniscono al Product Owner il contesto per decidere se una User Story è stata completata in modo soddisfacente. È importante – ed è una responsabilità dello Scrum Master – fare in modo che il Product Owner non modifichi nel mezzo di un Sprint i Criteri di Accettazione di una User Story presa in carico.

### 9.1.3.3 Prioritized Product Backlog Aggiornato

Il Prioritized Product Backlog creato nel processo *Creare il Prioritized Product Backlog* viene aggiornato con le informazioni relative alle User Story, alle Epic, alle stime delle User Story e ai Criteri di Accettazione delle User Story.

Il Prioritized Product Backlog è descritto nel capitolo 8, sezione 8.5.3.1.

### 9.1.3.4 Persona Aggiornate o Rifinite

Le *Persona* sono create inizialmente nel processo *Sviluppare le Epic*. Nello scrivere le User Story, lo Scrum Team può decidere che alcune delle *Persona* iniziali sono inadeguate e richiedono un miglioramento. Se è necessario procedere al miglioramento delle *Persona*, lo si fa normalmente verso la fine del processo *Creare le User Story*.

Le *Persona* sono descritte nel capitolo 8, sezione 8.4.3.2.

## 9.2 Approvare, Stimare e Prendere in Carico le User Story

La Figura 9-5 mostra tutti gli input, gli strumenti e gli output del processo *Approvare, Stimare e Prendere in Carico le User Story*.



Figura 9-5: Approvare, Stimare e Prendere in Carico le User Story—Input, Strumenti e Output

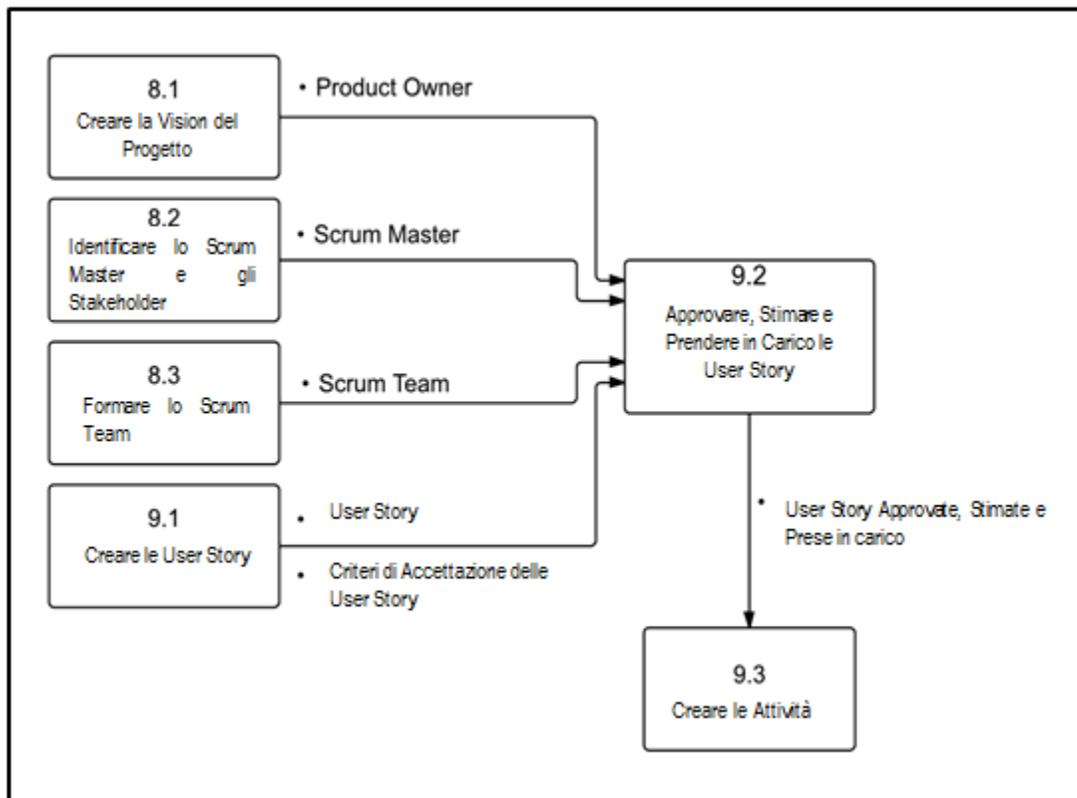


Figura 9-6: Approvare, Stimare e Prendere in Carico le User Story—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 9.2.1 Input

### 9.2.1.1 Scrum Core Team\*

Descritto nel capitolo 8, sezione 8.4.1.1.

### 9.2.1.2 User Story\*

Descritte nella sezione 9.1.3.1.

I processi *Creare il Prioritized Product Backlog* e *Creare le User Story* producono delle stime di alto livello delle User Story. Queste stime saranno usate dal Product Owner per redigere un elenco di User Story approvate che saranno sottoposte a stime più accurate da parte dello Scrum Team. Le User Story così stimate sono quindi prese in carico dallo Scrum Team per essere completate nello Sprint.

### 9.2.1.3 Criteri di Accettazione delle User Story\*

Descritti nella sezione 9.1.3.2.

### 9.2.1.4 Raccomandazioni dello Scrum Guidance Body

Descritti nel capitolo 8 sezione 8.1.1.12.

Le Raccomandazioni dello Scrum Guidance Body per il processo *Approvare, Stimare e Prendere in Carico le User Story* possono contenere informazioni su norme, regolamenti, standard e best practice necessarie per una efficace esecuzione del processo.

## 9.2.2 Strumenti

### 9.2.2.1 Riunioni di Gruppi di Utenti\*

Descritte nel capitolo 8, sezione 8.4.2.1.

### 9.2.2.2 Planning Poker

Il Planning Poker, chiamato anche Estimation Poker, è una tecnica di stima che utilizza il consenso per stimare le dimensioni relative delle User Story o l'impegno necessario per crearle.

Nel Planning Poker, viene assegnato un mazzo di carte a ciascun membro del team. Ogni carta è numerata secondo una sequenza e i numeri rappresentano la complessità del problema, in termini di tempo o di impegno, per come stimato dal membro del team. Il Product Owner sceglie una User Story dal Prioritized

Product Backlog e la presenta al team. I membri dello Scrum Team valutano la User Story e cercano di capirla più a fondo prima di fornire la propria stima per il suo sviluppo. In seguito, ognuno prende dal mazzo una carta che rappresenta la propria stima per quella User Story. Se la maggioranza di tutti i membri del team sceglie la stessa carta allora la stima indicata da quella carta sarà la stima stabilita per quella User Story. Se non c'è accordo, i membri del team discutono le ragioni della scelta delle differenti carte o stime. Dopo questa discussione scelgono di nuovo le carte. Questa sequenza si ripete fino a quando sono ben compresi i presupposti, sono risolte le incomprensioni e si raggiunge il consenso unanime o un accordo.

Il Planning Poker promuove una maggiore interazione e una migliore comunicazione fra i partecipanti. Questa tecnica facilita il pensiero indipendente dei partecipanti, evitando così il fenomeno del pensiero di gruppo.

### 9.2.2.3 Fist of Five

Il Fist of Five è un meccanismo semplice e veloce per raggiungere il consenso in un gruppo e guidare la discussione. Dopo una discussione iniziale su una data proposta o su una decisione in sospeso, si chiede a ciascun membro del team di votare su una scala da 1 a 5 utilizzando le proprie dita. Questa tecnica è preziosa non solo per costruire il consenso ma anche per guidare la discussione, perché a ogni membro del team viene chiesto di spiegare le ragioni della propria votazione. Viene inoltre data loro l'opportunità di esternare eventuali problemi o preoccupazioni. Dopo che il team ha discusso l'argomento, verrà presa una decisione collettiva.

Il numero di dita utilizzate per votare indica il livello di accordo e il desiderio di approfondire la discussione:

1. Un dito: Sono in totale disaccordo con la conclusione del gruppo e ho preoccupazioni importanti.
2. Due dita: Sono in disaccordo con la conclusione del gruppo e vorrei discutere questioni secondarie.
3. Tre dita: Non sono sicuro e vorrei associarmi alla conclusione che riceve il consenso del gruppo.
4. Quattro dita: Sono d'accordo con la conclusione del gruppo e vorrei discutere alcune questioni secondarie.
5. Cinque dita: Sono completamente d'accordo con la conclusione del gruppo.

### 9.2.2.4 Punti per la Stima dei Costi

La stima dei costi può essere fatta utilizzando unità relative (ad esempio, le stime dell'impegno) invece che unità assolute (cioè gli effettivi costi sostenuti). Per stimare il costo di implementazione di una User Story, lo Scrum Team può utilizzare gli story point. In questo caso, il costo stimato di ciascuna attività sarà espresso in story point, invece che in unità monetarie. Perché questo metodo abbia successo, lo Scrum Team deve individuare una User Story baseline a cui tutti i membri del team possano fare riferimento. Una volta identificata questa baseline, tutte le stime di costo delle User Story devono essere fatte in rapporto ad essa. Queste stime rimangono fisse per tutto lo Sprint, in quanto si suppone che i team non cambino nel corso di uno Sprint.

## 9.2.2.5 Altre Tecniche di Stima

### 9.2.2.5.1 Wideband Delphi

La Wideband Delphi è una tecnica di stima di gruppo che serve a stabilire quanto lavoro si deve eseguire e quanto tempo servirà per completarlo. I singoli individui di un team forniscono anonimamente delle stime per ciascuna funzionalità e le stime iniziali vengono segnate su un grafico. Dopo di che il team discute i fattori che hanno influenzato le stime di ognuno e si procede con un secondo giro di stime. Questo processo viene ripetuto fino a quando le stime dei singoli sono vicine fra loro e si può quindi raggiungere un consenso sulla stima finale.

Il planning poker (descritto nella sezione 9.2.2.2) è un esempio di tecnica Wideband Delphi. È inoltre importante notare che l'input individuale raccolto attraverso un particolare meccanismo evita il pensiero di gruppo. Dopo di che gli input individuali vengono utilizzati per una decisione di gruppo.

### 9.2.2.5.2 Relative Sizing/Story Point

Oltre ad essere usati per la stima dei costi, gli story point possono essere utilizzati anche per stimare la dimensione totale di una User Story o funzionalità. Questo approccio prevede l'assegnazione di un valore in story point sulla base di una valutazione complessiva della dimensione di una User Story considerando il rischio, la quantità di impegno richiesta e il livello di complessità. Questa valutazione sarà eseguita dallo Scrum Team, che assegnerà un valore in story point. Una volta eseguita la valutazione su una delle User Story del Prioritized Product Backlog, lo Scrum Team può valutare le altre User Story in rapporto alla prima. Per esempio, una funzionalità con un valore di 2 story point deve avere una difficoltà di completamento doppia rispetto ad una funzionalità che ha 1 story point; una funzionalità da 3 story point deve avere una difficoltà di completamento tre volte superiore a quella da 1 story point.

### 9.2.2.5.3 Stima per Affinità

La Stima per Affinità è una tecnica utilizzata per stimare velocemente un gran numero di User Story. Servendosi di post-it o delle index card e di nastro adesivo, il team attacca le User Story su una parete o su un'altra superficie, in ordine crescente di grandezza. Per fare questo, ciascun membro del team inizia con un sottoinsieme di User Story prese dal complessivo Prioritized Product Backlog, da collocare in base alla dimensione relativa. Questa collocazione iniziale viene fatta in silenzio. Dopo che tutti hanno posizionato le proprie User Story sulla parete, il team rivede tutte le collocazioni e può spostare le User Story da un posto all'altro a seconda dei casi. Questa seconda parte dell'esercizio implica una discussione. Alla fine, il Product Owner indicherà sulla parete alcune categorie di grandezza. Queste categorie possono essere 'piccola', 'media' o 'grande' oppure possono essere numerate con i valori in story point per indicare la dimensione relativa. Il team sposterà quindi le User Story all'interno di queste categorie, come passo finale del procedimento. Alcuni dei benefici fondamentali di questo approccio sono la grande trasparenza del processo, la visibilità da parte di tutti e la facilità di esecuzione.

#### 9.2.2.5.4 Intervallo di Stima

Le stime dei progetti dovrebbero essere presentate sotto forma di intervalli. I numeri esatti possono dare l'impressione di una maggiore accuratezza, ma in realtà può non essere così. Infatti, le stime sono per definizione non esattamente precise. Gli intervalli di stima dovrebbero basarsi sul livello di fiducia che il team ripone in ciascuna stima. L'intervallo può essere stretto quando il team è fiducioso e ampio quando il team lo è un po' meno.

#### 9.2.2.6 Competenze Esperte dello Scrum Guidance Body

Descritte nel capitolo 8, sezione 8.4.2.7.

Durante questo processo possono nascere dei conflitti aventi ad oggetto le stime riguardanti il completamento di determinate User Story, perché le opinioni dei membri del team possono differire e anche perché il team può non avere ancora abbastanza esperienza per stimare gli Sprint. In queste situazioni, l'esperienza e le competenze esperte del Guidance Body possono aiutare a risolvere i conflitti.

### 9.2.3 Output

#### 9.2.3.1 User Story Approvate, Stimate e Prese in Carico\*

Le User Story che costituiscono un input di questo processo sono corredate da stime di alto livello prodotte dai processi *Creare il Prioritized Product Backlog* e *Creare le User Story*. Queste stime saranno utilizzate dal Product Owner per approvare le User Story dello Sprint.

Si deve notare che il Product Owner ha la responsabilità di fare in modo che le User Story approvate consegnino valore e soddisfino i bisogni e i requisiti degli stakeholder del progetto. Una volta approvate, le User Story sono stimate dal team con le varie tecniche di stima illustrate in questa sezione. Dopo la stima, il team prende in carico un sottoinsieme di User Story approvate e stimate, che costituiscono quanto il team ritiene di poter completare nello Sprint successivo. Queste sono le User Story Approvate, Stimate e Prese in Carico che entreranno a far parte dello Sprint Backlog.

Anche se il Product Owner approva le User Story iniziali di uno Sprint, la decisione finale su quali specifiche User Story (fra quelle approvate) scegliere per lo Sprint spetta allo Scrum Team. Lo Scrum Team (di concerto con il Product Owner, se necessario) definisce a quali User Story lavorerà durante lo Sprint.

## 9.3 Creare le Attività

La Figura 9-7 mostra tutti gli input, gli strumenti e gli output del processo *Creare le Attività*.



Figura 9-7: Creare le Attività—Input, Strumenti e Output

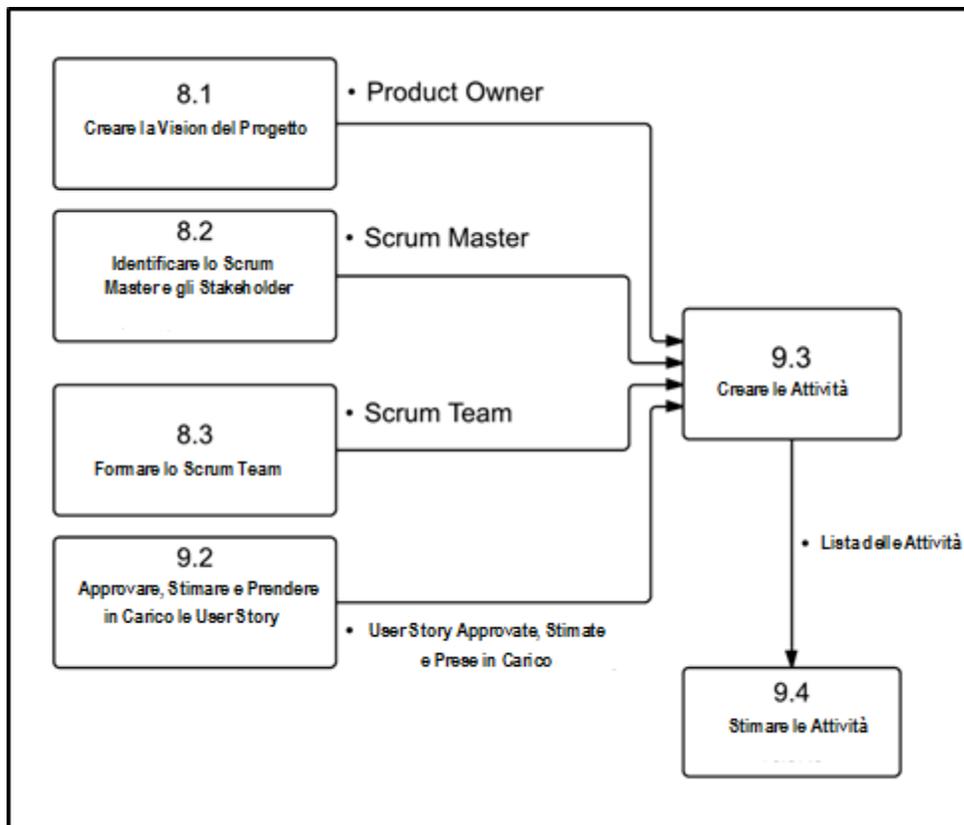


Figura 9-8: Creare le Attività—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 9.3.1 Input

### 9.3.1.1 Scrum Core Team\*

Descritto nel capitolo 8, sezione 8.4.1.1.

### 9.3.1.2 User Story Approvate, Stimate e Prese in Carico\*

Descritte nella sezione 9.2.3.1.

## 9.3.2 Strumenti

### 9.3.2.1 Task Planning Meeting\*

Lo Scrum Team si riunisce nei Task Planning Meeting per pianificare il lavoro da compiere nello Sprint. Il team rivede le User Story prese in carico poste in cima al Prioritized Product Backlog. Il Product Owner partecipa a questa riunione, nel caso siano necessari chiarimenti sulle User Story incluse nel Prioritized Product Backlog e per aiutare il team a prendere decisioni progettuali. Per aiutare il gruppo a rimanere in argomento, si dovrebbe stabilire una durata predeterminata per questa riunione, con una lunghezza standard contenuta in due ore per ogni settimana di durata dello Sprint. Questo aiuta a prevenire la tendenza a soffermarsi su discussioni che dovrebbero effettivamente essere aperte in altri tipi di riunioni, come il Release Planning o lo Sprint Review Meeting. Alla fine della riunione, l'intero Scrum Team si sarà preso l'impegno pieno a consegnare nello Sprint un sottoinsieme di User Story del Prioritized Product Backlog.

Il Task Planning Meeting è di solito suddiviso in due parti, ognuna delle quali ha uno scopo specifico e un'agenda generale (si veda la Figura 9-9)

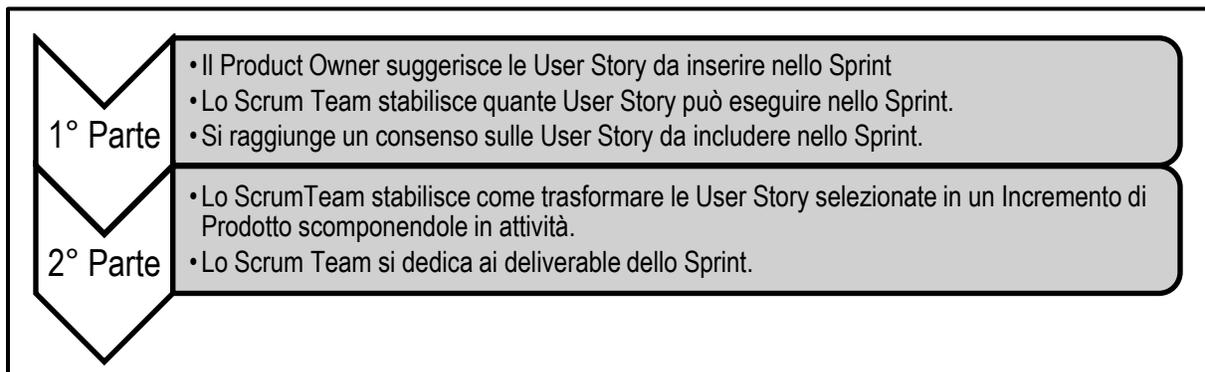


Figura 9-9: Task Planning Meeting

I Task Planning Meeting possono a volte essere anche chiamati "Sprint Planning Meeting" – tali riunioni possono essere anche combinate con i Task Estimation Meeting descritti nella sezione 9.4.2.1.

### 9.3.2.2 Index Card

In Scrum le User Story sono scritte su delle piccole Index Card. Sulle carte, che possono essere utilizzate dallo Scrum Team per collaborare e discutere, vengono documentati solo i dettagli essenziali. Queste Index Card, spesso definite Story Card, aumentano la visibilità e la trasparenza e facilitano la scoperta precoce di eventuali problemi che possono insorgere.

### 9.3.2.3 Scomposizione

La Scomposizione è uno strumento per mezzo del quale le attività di alto livello sono scomposte in attività più dettagliate e di livello più basso. Le User Story sono scomposte in attività dai membri dello Scrum Team. Le User Story del Prioritized Product Backlog devono essere scomposte fino a un livello sufficiente a fornire allo Scrum Team informazioni adeguate per procedere alla creazione dei deliverable partendo dalle Attività elencate nella Lista delle Attività.

### 9.3.2.4 Determinazione delle Dipendenze

Una volta che lo Scrum Team ha scelto le User Story per un dato Sprint, deve poi prendere in considerazione le eventuali dipendenze, incluse quelle relative alla disponibilità delle persone, oltre a eventuali dipendenze di carattere tecnico. Una documentazione appropriata delle dipendenze aiuta gli Scrum Team a stabilire l'ordine relativo di esecuzione delle Attività per la creazione dei Deliverable dello Sprint. Le dipendenze evidenziano inoltre la relazione e interazione fra le Attività sia all'interno dello Scrum Team che sta lavorando ad un dato Sprint, sia con gli altri Scrum Team del progetto.

Ci sono molti tipi di dipendenze: obbligatorie e discrezionali, interne e esterne, o una combinazione di queste. Ad esempio, una dipendenza può essere contemporaneamente obbligatoria ed esterna.

- **Dipendenze obbligatorie**—Dipendenze che sono inerenti alla natura del lavoro, come un limite fisico, o che possono diventare tali in virtù di obbligazioni contrattuali o di requisiti legali. Per esempio i lavori sul primo piano non possono iniziare fino a quando non sono completate le fondamenta dell'edificio. Le dipendenze obbligatorie sono comunemente definite anche come hard logic.
- **Dipendenze discrezionali**—Dipendenze che sono inserite nel flusso di lavoro per scelta. Di solito, le dipendenze discrezionali sono stabilite dallo Scrum Team in base alle esperienze passate o alle best practice di un particolare campo o settore. Per esempio, il team può decidere di completare un'attività prima di iniziare a lavorare su di un'altra perché ciò costituisce una best practice, ma non è una pratica necessaria. Sempre per esempio, il team può scegliere di costruire i telai delle porte e delle finestre prima che sia posizionata la struttura completa della parete.
- **Dipendenze esterne**—Le dipendenze esterne sono quelle relative ad attività, azioni o prodotti che sono al di fuori dell'ambito del lavoro che deve eseguire lo Scrum Team, ma sono necessari per

completare un'attività di progetto o per creare un deliverable del progetto. Le dipendenze esterne di norma fuoriescono dal controllo dello Scrum Team. Ad esempio, se lo Scrum Team non ha la responsabilità di approvvigionare i materiali richiesti per la costruzione delle pareti, allora quei materiali e le attività relative al loro approvvigionamento sono considerate dipendenze esterne.

- **Dipendenze interne**—Le dipendenze interne sono quelle che intercorrono fra attività, prodotti o azioni che sono sotto il controllo dello Scrum Team. Per esempio, la realizzazione della muratura o del cartongesso deve essere completata prima di poter far iniziare la pittura della parete. Questo è un esempio di dipendenza interna, in quanto entrambe le attività fanno parte del progetto. In questo caso, si tratta di una dipendenza anche obbligatoria perché si basa su una limitazione di carattere materiale. Non è possibile pitturare la parete prima di aver realizzato la muratura.

### 9.3.3 Output

#### 9.3.3.1 Lista delle Attività\*

Si tratta di un elenco completo che comprende tutte le attività che lo Scrum Team si è impegnato a compiere nello Sprint corrente. Contiene la descrizione di ciascuna attività insieme alle stime prodotte durante il processo *Creare le Attività*. La Lista delle Attività deve includere gli eventuali impegni necessari per il testing e l'integrazione, in modo che l'Incremento di Prodotto frutto dello Sprint possa essere integrato con successo nei deliverable prodotti dagli Sprint precedenti.

Anche se le attività sono spesso basate sulle azioni da compiere, il livello di dettaglio fino a cui le attività vengono scomposte è deciso dallo Scrum Team.

#### 9.3.3.2 User Story Approvate, Stimate e Prese in Carico Aggiornate

Durante questo processo le User Story vengono aggiornate. Gli aggiornamenti possono riguardare anche le revisioni delle stime originarie delle User Story sulla base della creazione delle attività e dei fattori di complessità discussi durante lo Sprint Planning Meeting. Le User Story Approvate, Stimate e Prese in Carico sono descritte nella sezione 9.2.3.1.

#### 9.3.3.3 Dipendenze

Le Dipendenze descrivono la relazione e l'interazione fra diverse attività di un progetto e possono essere classificate come obbligatorie o discrezionali, oppure come interne o esterne, come esposto nella sezione 9.3.2.4.

Ci sono svariati modi di identificare, definire e presentare le attività e le loro dipendenze. Due metodi diffusi sono l'utilizzo di diagrammi di flusso del prodotto e di grafici di Gantt.

## 9.4 Stimare le Attività

La Figura 9-10 mostra tutti gli input, gli strumenti e gli output del processo *Stimare le Attività*.



Figura 9-10: Stimare le Attività—Input, Strumenti e Output

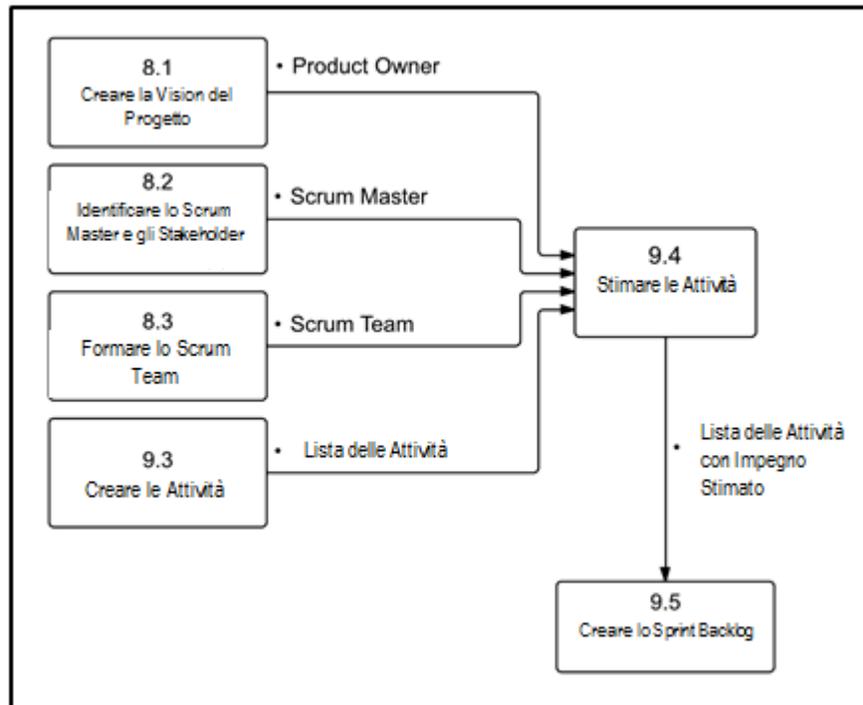


Figura 9-11: Stimare le Attività—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indica un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 9.4.1 Input

### 9.4.1.1 Scrum Core Team\*

Descritto nel capitolo 8, sezione 8.4.1.1.

### 9.4.1.2 Lista delle Attività\*

Descritta nella sezione 9.3.3.1.

### 9.4.1.3 Criteri di Accettazione delle User Story

Descritti nella sezione 9.1.3.2.

Lo Scrum Team deve assicurarsi che i Criteri di Accettazione definiti siano appropriati per le User Story e chiariscano bene i requisiti. I test di accettazione attengono alla capacità del deliverable completato di soddisfare i propri Criteri di Accettazione. Questo fornisce al Product Owner le informazioni che lo aiutano a decidere se approvare o rifiutare il Deliverable.

Nello sviluppare i Criteri di Accettazione della User Story, si devono tenere presenti i seguenti principi:

- I Criteri di Accettazione non devono essere vaghi, ambigui o troppo generici.
- I Criteri di Accettazione definiti devono mettere in grado il team di poter verificare che i risultati siano allineati con gli obiettivi specifici e generali dell'organizzazione dello sponsor.

### 9.4.1.4 Dipendenze

Descritte nella sezione 9.3.3.3.

### 9.4.1.5 Rischi Identificati

Descritti nel capitolo 8, sezione 8.4.3.4.

### 9.4.1.6 Raccomandazioni dello Scrum Guidance Body

Descritte nel capitolo 8, sezione 8.1.1.12.

Le Raccomandazioni dello Scrum Guidance Body per il processo *Stimare le Attività* possono riguardare informazioni su norme, regolamenti, standard e best practice necessari per stimare in modo efficace le attività incluse nella Lista delle Attività.

## 9.4.2 Strumenti

### 9.4.2.1 Task Estimation Meeting\*

I Task Estimation Meeting consentono allo Scrum Team di stimare l'impegno necessario per completare un'attività o un insieme di attività e per stimare l'impegno delle persone e le altre risorse necessarie per eseguire le attività all'interno di un dato Sprint. Durante i Task Estimation Meeting i membri dello Scrum Team usano la Lista delle Attività per stimare la durata e l'impegno necessario per completare le User Story nello Sprint.

Uno dei benefici chiave di questa tecnica è che consente al team di avere una visione condivisa delle User Story e dei requisiti e di poter quindi stimare in modo affidabile l'impegno necessario. Le informazioni sviluppate nei Task Estimation Meeting vengono inserite nella Lista delle Attività con Impegno Stimato e sono utilizzate per stabilire la velocità dello Sprint.

In questo workshop, lo Scrum Team può usare varie tecniche come la scomposizione, il giudizio di esperti, la stima per analogia e la stima parametrica.

I Task Estimation Meeting vengono a volte chiamati anche "Sprint Planning Meeting" – queste riunioni possono essere anche combinate con i Task Planning Meeting descritti nella sezione 9.3.2.1.

9

### 9.4.2.2 Criteri di Stima\*

L'obiettivo principale dell'utilizzo dei Criteri di Stima è mantenere ferme le grandezze di stima relative e ridurre al minimo la necessità di ripetere le stime. I Criteri di Stima possono essere espressi in molti modi, fra i quali due esempi tipici sono gli story point e il tempo ideale. Per esempio, un tempo ideale descrive di norma il numero di ore che un membro dello Scrum Team dedica esclusivamente al lavoro di sviluppo dei deliverable del progetto, escludendo tutto il tempo speso in altre attività o in lavoro esterno al progetto. I Criteri di Stima rendono più facile allo Scrum Team stimare l'impegno occorrente e gli permettono, quando necessario, di valutare e affrontare le inefficienze.

### 9.4.2.3 Planning Poker

Descritto nella sezione 9.2.2.2.

### 9.4.2.4 Fist of Five

Descritto nella sezione 9.2.2.3.

### 9.4.2.5 Altre Tecniche di Stima delle Attività

Descritte nella sezione 9.2.2.5.

## 9.4.3 Output

### 9.4.3.1 Lista delle Attività con Impegno Stimato\*

La Lista delle Attività con Impegno Stimato è un elenco di attività associate alle User Story Prese in Carico incluse in uno Sprint. Normalmente l'accuratezza delle stime varia a seconda delle competenze del team. L'impegno stimato è espresso in base ai Criteri di Stima concordati dal team. La Lista delle Attività con Impegno Stimato è utilizzata dallo Scrum Team durante gli Sprint Planning Meeting per creare lo Sprint Backlog e lo Sprint Burndown Chart. Questa lista è utilizzata anche durante la Pianificazione dello Sprint, per stabilire quando il team deve ridurre il proprio impegno di lavoro o, viceversa, può farsi carico di User Story aggiuntive.

### 9.4.3.2 Lista delle Attività Aggiornata

La Lista delle Attività, sviluppata come parte del processo *Creare le Attività*, contiene le stime iniziali delle User Story, che devono essere riviste sulla base delle attività di stima più dettagliate compiute nel processo *Stimare le Attività*. Ci possono essere anche delle ripetizioni di stime a seguito di una revisione di precedenti Sprint, o del cambiamento della comprensione collettiva delle User Story e dei requisiti da parte dello Scrum Team.

## 9.5 Creare lo Sprint Backlog

La Figura 9-12 mostra tutti gli input, gli strumenti e gli output del processo *Creare lo Sprint Backlog*.



Figura 9-12: Creare lo Sprint Backlog—Input, Strumenti e Output

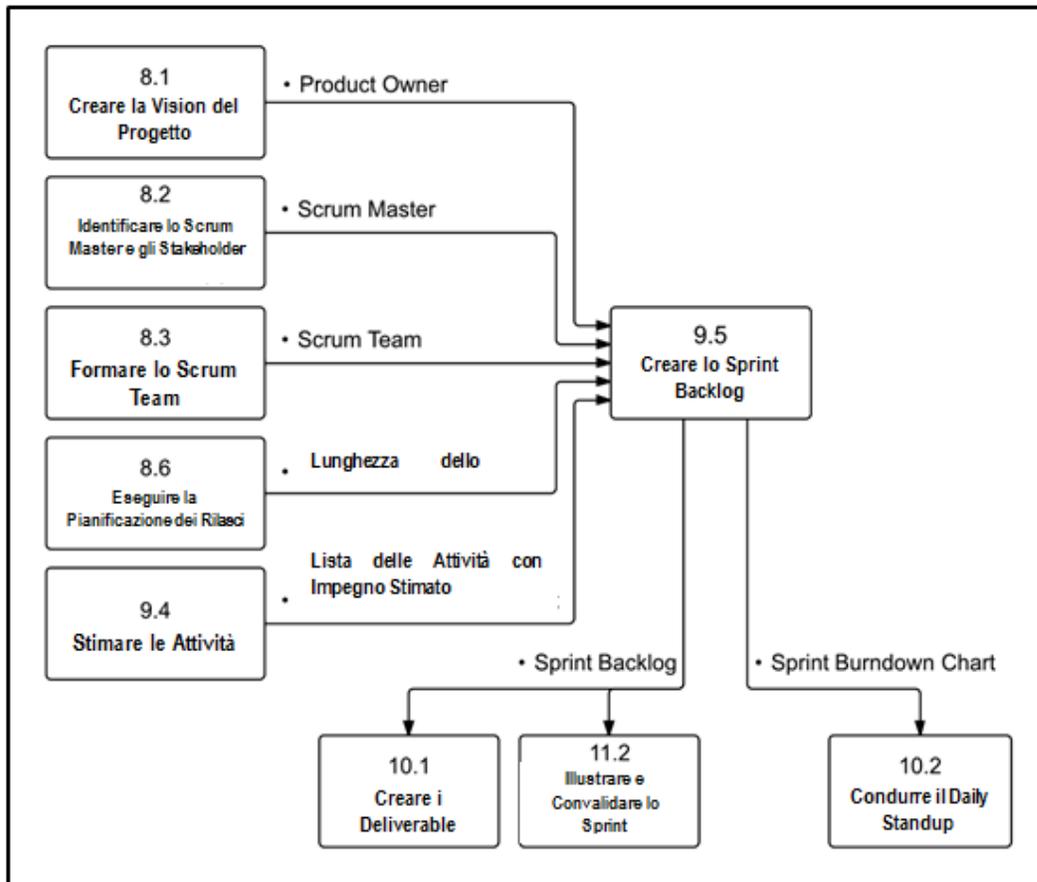


Figura 9-13: Creare lo Sprint Backlog—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente

## **9.5.1 Input**

### **9.5.1.1 Scrum Core Team\***

Descritto nel capitolo 8, sezione 8.4.1.1.

### **9.5.1.2 Lista delle Attività con Impegno Stimato\***

Descritta nella sezione 9.4.3.1.

### **9.5.1.3 Lunghezza dello Sprint\***

Descritta nel capitolo 8, sezione 8.6.3.2.

### **9.5.1.4 Velocità dei Precedenti Sprint**

La Velocità dello Sprint rappresenta la velocità alla quale il team può completare il lavoro in uno Sprint. Di solito è espressa nelle stesse unità utilizzate per la stima, di norma gli story point o il tempo ideale. Viene mantenuta una registrazione della Velocità dello Sprint del team per ciascuno Sprint, da utilizzare come riferimento per gli Sprint successivi. La Velocità degli Sprint diventa il fattore più importante per determinare la quantità di lavoro che il team può prendere in carico in un successivo Sprint. Eventuali cambiamenti di situazione o di condizioni rispetto all'ultimo Sprint saranno tenuti in considerazione per assicurare una stima accurata della velocità dello Sprint a venire.

### **9.5.1.5 Dipendenze**

Descritte nella sezione 9.3.3.3.

### **9.5.1.6 Calendario del Team**

Un Calendario del Team contiene informazioni sulla disponibilità dei membri del team, fra cui quelle relative alle ferie dei dipendenti, ai permessi, ad eventi importanti e alle festività.

Uno degli obiettivi principali dell'utilizzo di un Calendario del Team è quello di monitorare ciò a cui sta lavorando ciascun membro del team lungo tutto il corso del progetto. Questo strumento aiuta il team non solo a pianificare e ad eseguire gli Sprint in maniera efficiente, ma anche ad allineare gli Sprint con le date dei rilasci.

## 9.5.2 Strumenti

### 9.5.2.1 Sprint Planning Meeting\*

Durante gli Sprint Planning Meeting, le User Story che sono approvate, stimate e prese in carico durante il processo *Approvare, Stimare e Prendere in Carico le User Story* vengono discusse dallo Scrum Team. Ciascun membro dello Scrum Team utilizza inoltre la Lista delle Attività con Impegno Stimato per scegliere le attività a cui pianifica di lavorare nello Sprint, in base alle proprie capacità e alla propria esperienza. Durante gli Sprint Planning Meeting, lo Scrum Team crea anche lo Sprint Backlog e lo Sprint Burndown Chart, utilizzando le User Story e la Lista delle Attività con Impegno Stimato.

### 9.5.2.2 Strumenti di Monitoraggio dello Sprint

È importante monitorare lo stato di avanzamento di uno Sprint e sapere dove si trova lo Scrum Team in termini di completamento delle attività dello Sprint Backlog. Per monitorare il lavoro di uno Sprint si possono usare tanti tipi di strumenti, ma uno dei più comuni è una Scrumboard, nota anche come task board o progress chart. La Scrumboard è divisa in tre sezioni: Da fare (a volte chiamata 'Lavoro Non Iniziato'), Lavoro in Corso, e Lavoro Completato. Per dare conto dell'esatto stato del lavoro vengono collocati nella categoria appropriata dei post-it che rappresentano ciascuno un'attività o User Story. Questi post-it sono spostati nella categoria successiva man mano che il lavoro progredisce.

### 9.5.2.3 Metriche di Monitoraggio dello Sprint

Le metriche utilizzate nei progetti Scrum sono la velocità, il valore di business consegnato e il numero di story.

*Velocità*—rappresenta il numero di User Story o il numero di funzionalità consegnate in un singolo Sprint.

*Valore di business consegnato*—misura il valore delle User Story consegnate dal punto di vista del business.

*Numero di story*—si riferisce a quante User Story sono consegnate come parte di un singolo Sprint. Può essere espresso in termini di conteggio semplice o conteggio ponderato.

## 9.5.3 Output

### 9.5.3.1 Sprint Backlog\*

L'elenco delle attività che devono essere eseguite dallo Scrum Team nello Sprint a venire è chiamato Sprint Backlog.

È pratica abituale rappresentare lo Sprint Backlog su una Scrumboard o task board, che offre una rappresentazione costantemente visibile dello stato delle User Story del backlog. Nello Sprint Backlog sono inclusi anche gli eventuali rischi associati alle varie attività. Nello Sprint Backlog saranno inoltre incluse come attività le eventuali azioni di mitigazione individuate per trattare i rischi identificati.

Una volta che lo Sprint Backlog è ultimato e preso in carico dallo Scrum Team, non dovrebbero essere aggiunte nuove User Story; tuttavia, può essere necessario aggiungere attività che potrebbero essere state dimenticate o trascurate dalle User Story prese in carico. Se durante uno Sprint emergono nuovi requisiti, saranno aggiunti al Prioritized Product Backlog e inclusi in uno Sprint futuro.

### 9.5.3.2 Sprint Burndown Chart\*

Lo Sprint Burndown Chart è un grafico che illustra la quantità di lavoro che rimane da fare nello Sprint in corso. Lo Sprint Burndown Chart iniziale è accompagnato da un piano di burndown. Lo Sprint Burndown Chart deve essere aggiornato al termine di ciascuna giornata, man mano che il lavoro è completato. Questo grafico mostra i progressi compiuti dallo Scrum Team e permette anche di scoprire le stime che si sono rivelate non corrette. Se lo Sprint Burndown Chart mostra che lo Scrum Team non è sulla strada giusta per finire in tempo le attività dello Sprint, lo Scrum Master deve identificare le situazioni che ostacolano o impediscono di completare il lavoro con successo e provare a rimuoverle.

Un grafico collegato è lo Sprint Burnup Chart. A differenza dello Sprint Burndown Chart, che mostra la quantità di lavoro rimanente, lo Sprint Burnup Chart illustra il lavoro completato come parte dello Sprint.

## 9.6 Diagramma di Flusso dei Dati della Fase

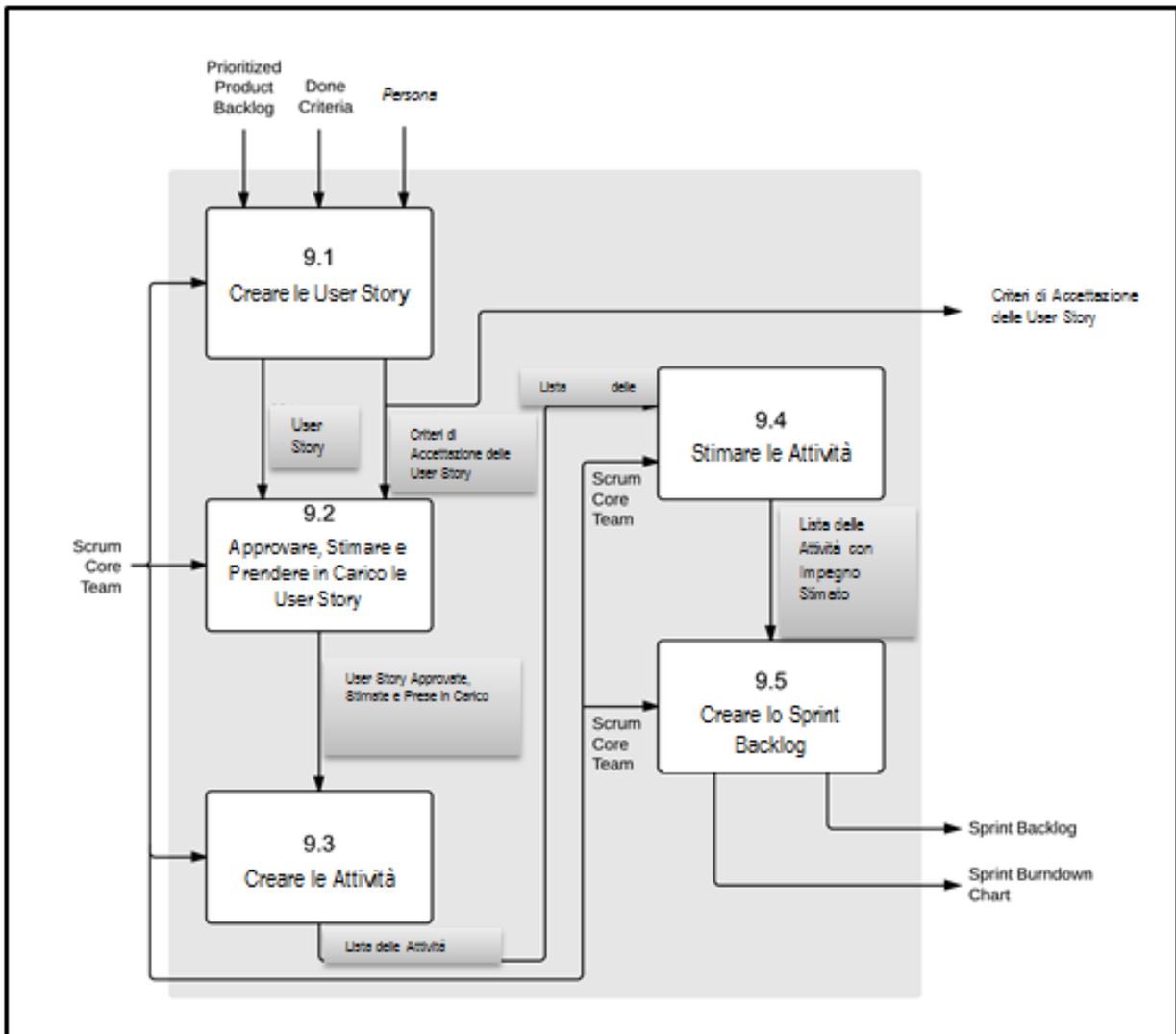


Figura 9-14: Fase di Pianificazione e Stima—Diagramma di Flusso dei Dati

## 10. IMPLEMENTAZIONE

La fase di Implementazione riguarda l'esecuzione delle attività e delle azioni che servono a creare un prodotto di progetto. Queste attività attengono alla creazione di vari deliverable, alla conduzione dei Daily Standup Meeting e alla messa a punto (vale a dire, revisione, ottimizzazione e regolare aggiornamento) del Product Backlog a intervalli regolari.

La fase di *Implementazione*, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Per facilitare la migliore applicazione del framework Scrum, questo capitolo identifica gli input, gli strumenti e gli output di ciascun processo qualificandoli come “obbligatori” o “opzionali”. Gli input, gli strumenti e gli output contrassegnati da un asterisco (\*) sono obbligatori, mentre quelli senza asterisco sono opzionali.

Si raccomanda allo Scrum Team e a tutti coloro che vengono introdotti al framework e ai processi Scrum di concentrarsi soprattutto sugli input, gli strumenti e gli output obbligatori; i Product Owner, gli Scrum Master e altri professionisti di Scrum più esperti dovrebbero invece sforzarsi di raggiungere una conoscenza più approfondita delle informazioni contenute nell'intero capitolo. È altresì importante avere chiaro che, sebbene tutti i processi siano definiti singolarmente nella *Guida SBOK™*, non è necessario eseguirli in maniera sequenziale o separatamente. A volte, a seconda dei requisiti specifici di ciascun progetto, potrebbe essere più appropriato unire alcuni processi fra loro.

Questo capitolo è scritto dal punto di vista di un singolo Scrum Team che lavora ad un singolo Sprint per produrre Deliverable potenzialmente consegnabili come parte di un progetto più ampio. Tuttavia, le informazioni fornite sono ugualmente applicabili a interi progetti, programmi e portfolio. Ulteriori informazioni sull'uso di Scrum nei progetti, programmi e portfolio sono disponibili nei capitoli da 2 a 7, che trattano i principi e gli aspetti di Scrum.

La Figura 10-1 fornisce una visione d'insieme dei processi della fase di Implementazione, che sono:

**10.1 Creare i Deliverable**—In questo processo, lo Scrum Team lavora alle attività dello Sprint Backlog per creare i Deliverable dello Sprint. Spesso viene utilizzata una Scrumboard per monitorare il lavoro e le attività che si stanno eseguendo. Le questioni o i problemi che lo Scrum Team si trova ad affrontare possono essere annotati in un Registro degli Impedimenti.

**10.2 Condurre il Daily Standup**—In questo processo viene condotta tutti i giorni una riunione estremamente mirata, di durata predeterminata, chiamata Daily Standup Meeting. Questo rappresenta per lo Scrum Team il luogo di discussione per un aggiornamento sui reciproci progressi e sugli eventuali impedimenti che i singoli membri stanno riscontrando.

**10.3 Mettere a Punto il Prioritized Product Backlog**—L'oggetto di questo processo è il continuo aggiornamento e mantenimento del Prioritized Product Backlog. Può essere tenuto un Prioritized Product Backlog Review Meeting, nel corso del quale tutti i cambiamenti e gli aggiornamenti del backlog vengono discussi e, se del caso, incorporati nel Prioritized Product Backlog.

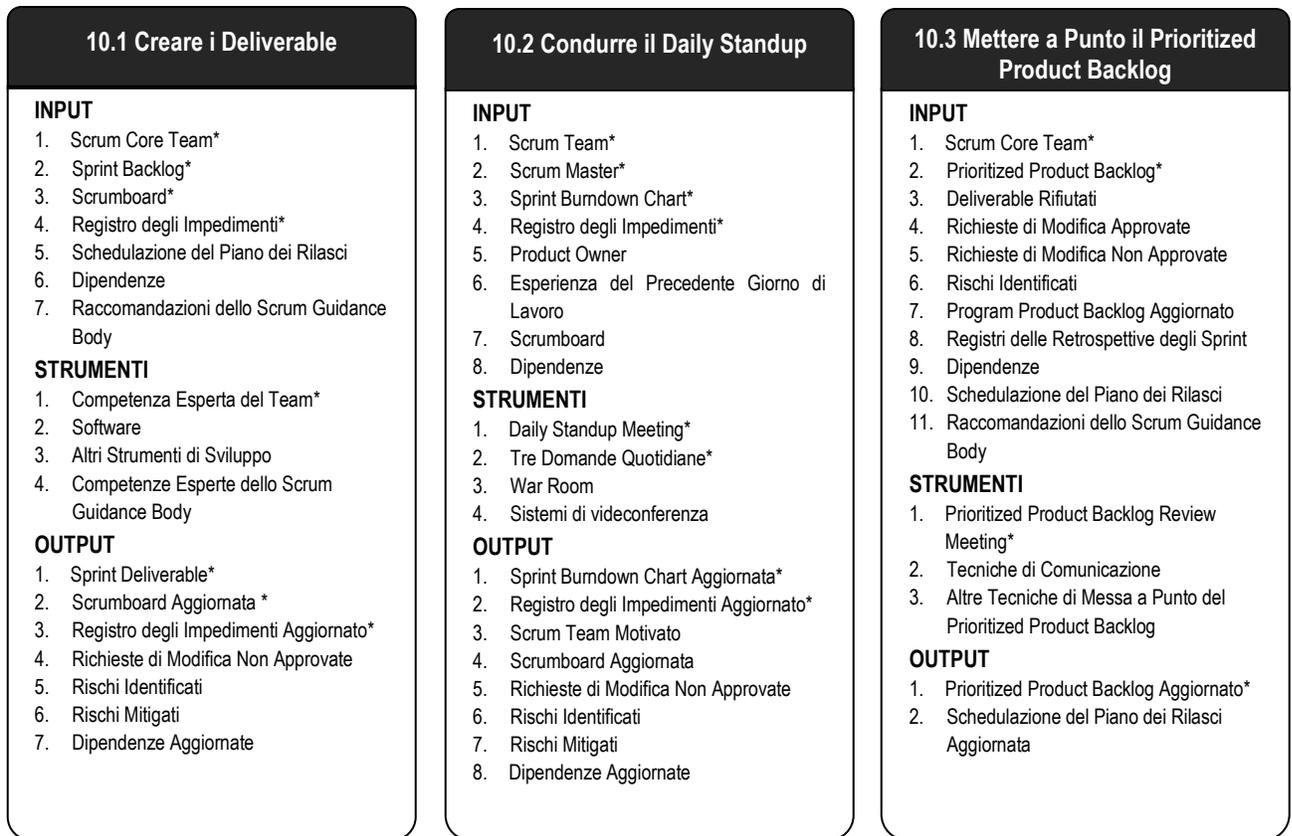


Figura 10-1: Visione d'insieme della Fase di Implementazione

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

La Figura 10-2 qui sotto mostra gli input, gli strumenti e gli output obbligatori dei processi della fase di Implementazione.

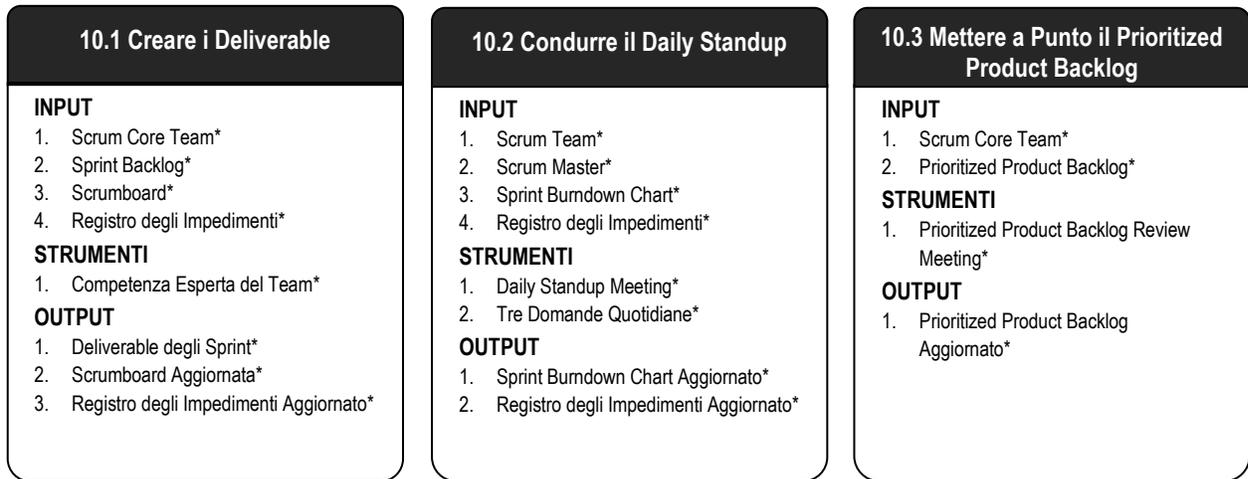


Figura 10-2: Visione d'insieme della Fase di Implementazione (Elementi essenziali)

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 10.1 Creare i Deliverable

La Figura 10-3 mostra tutti gli input, gli strumenti e gli output del processo *Creare i Deliverable*.



Figura 10-3: Creare i Deliverable—Input, Strumenti e Output

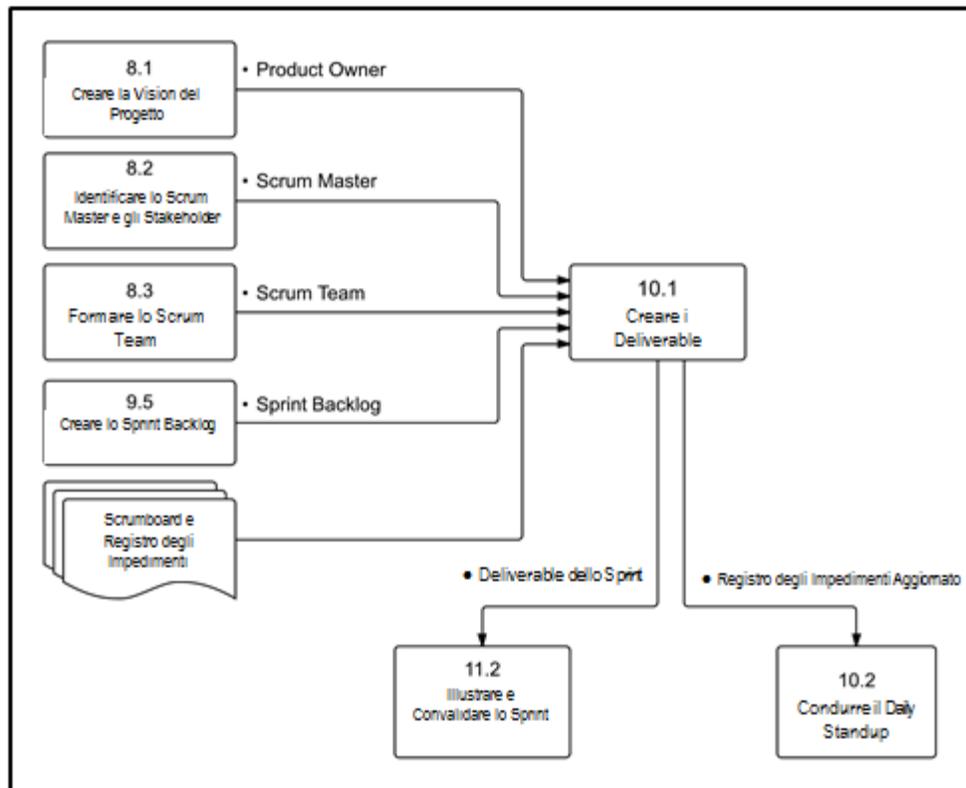


Figura 10-4: Creare i Deliverable—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.



Lo Scrum Team deve apportare i necessari cambiamenti o aggiunte alla Scrumboard, in modo che questa fornisca le informazioni e il controllo visivi riguardo al lavoro in corso, come concordato e preso in carico dal team.

#### **10.1.1.4 Registro degli Impedimenti\***

Un impedimento è un qualsiasi ostacolo o complicazione che riduce la produttività dello Scrum Team. Gli impedimenti devono essere identificati, risolti e rimossi se si vuole che il team continui a lavorare in modo efficace. Gli impedimenti possono essere interni al team, come un flusso di lavoro inefficiente o una carenza di comunicazione, oppure esterni. Alcuni esempi di impedimenti esterni possono essere le questioni relative alle licenze software o i requisiti di documentazione superflui. Il framework Scrum, con la sua intrinseca trasparenza, facilita la veloce e agevole identificazione degli impedimenti. Il non riuscire a identificare o a gestire gli impedimenti può costare molto. Gli impedimenti devono essere registrati in modo formale dallo Scrum Master in un Registro degli Impedimenti e se del caso possono essere discussi durante i Daily Standup Meeting e gli Sprint Review Meeting.

#### **10.1.1.5 Schedulazione del Piano dei Rilasci**

Descritta nel capitolo 8, sezione 8.6.3.1.

#### **10.1.1.6 Dipendenze**

Descritte nel capitolo 9, sezione 9.3.3.3.

#### **10.1.1.7 Raccomandazioni dello Scrum Guidance Body**

Descritte nel capitolo 8, sezione 8.1.1.12.

Le Raccomandazioni dello Scrum Guidance Body per il processo *Creare i Deliverable* possono riguardare le best practice per l'efficace creazione dei deliverable, compresi i metodi preferibili per realizzare le revisioni, i test, la documentazione, ecc.

## 10.1.2 Strumenti

### 10.1.2.1 Competenza Esperta del Team\*

Questo strumento si riferisce alla capacità collettiva dei membri dello Scrum Team di capire le User Story e le attività dello Sprint Backlog per poter creare i deliverable finali. La Competenza Esperta del Team è usata per valutare gli input necessari all'esecuzione del lavoro pianificato del progetto. Questo giudizio e competenza sono applicati a tutti gli aspetti tecnici e di gestione del progetto durante il processo *Creare i Deliverable*. I membri dello Scrum Team hanno l'autorità e la responsabilità di stabilire i mezzi migliori per trasformare gli elementi del Prioritized Product Backlog in prodotti finiti, senza bisogno di coinvolgere alcuno stakeholder esterno al team. Se necessario, è possibile fare ricorso alle ulteriori competenze esperte dello Scrum Guidance Body.

### 10.1.2.2 Software

Per la schedulazione, la raccolta delle informazioni e la loro distribuzione possono essere utilizzati strumenti di Software Automatizzato. Nei progetti in cui gli Scrum Team non sono co-ubicati sono inoltre essenziali gli strumenti di collaborazione virtuale. Sono disponibili molti strumenti software automatizzati, che permettono il monitoraggio dello stato di avanzamento, la raccolta dei dati e la loro distribuzione e contribuiscono a velocizzare i processi.

### 10.1.2.3 Altri Strumenti di Sviluppo

In base ai requisiti specifici del progetto e alle specifiche del settore industriale, si possono utilizzare altri strumenti di sviluppo.

#### 1. Refactoring

Il Refactoring è uno strumento proprio dei progetti software. Lo scopo di questa tecnica è quello di migliorare la manutenibilità del codice esistente e di renderlo più semplice, più conciso e più flessibile. Fare il refactoring significa migliorare la progettazione del codice esistente senza cambiarne il comportamento. Ciò implica:

- L'eliminazione del codice ripetitivo o ridondante
- La scomposizione dei metodi e delle funzioni in routine più piccole
- La definizione chiara delle variabili e dei nomi dei metodi
- La semplificazione del disegno progettuale del codice
- Rendere il codice più facile da capire e da modificare

Il refactoring abituale ottimizza un po' alla volta il disegno progettuale del codice, in un certo periodo di tempo. In definitiva, il refactoring si traduce in un codice più pulito, più manutenibile, preservandone nel contempo tutte le funzionalità.

## 2. Schemi di Progettazione

Gli Schemi di Progettazione (Design Patterns) forniscono un modo per registrare formalmente la soluzione ad un problema di progettazione tecnica in uno specifico campo di competenza. Questi schemi registrano sia il processo utilizzato che la soluzione concreta da poter riutilizzare successivamente per migliorare il processo decisionale e la produttività.

### 10.1.2.4 Competenze Esperte dello Scrum Guidance Body

Descritte nel capitolo 8, sezione 8.4.2.7.

La Competenze Esperte dello Scrum Guidance Body riguardo ai processi *Creare i Deliverable* e *Approvare, Stimare e Prendere in Carico le User Story* può essere collegata a norme o regolamenti documentati, a linee guida di sviluppo, oppure a standard o best practice (ad esempio indicazioni su come eseguire revisioni o test). Può anche essere presente un team di esperti della materia che può fornire allo Scrum Team indicazioni per la creazione dei deliverable. Questo team può essere formato da Capi Progettisti, Sviluppatori Senior, Esperti della Sicurezza, o altre persone con esperienza.

### 10.1.3 Output

#### 10.1.3.1 Deliverable dello Sprint\*

Alla fine di ogni Sprint viene completato un incremento di prodotto o deliverable. Il deliverable deve possedere tutte le caratteristiche e le funzionalità definite nelle User Story dello Sprint e deve essere stato testato con successo.

#### 10.1.3.2 Scrumboard Aggiornata\*

La Scrumboard viene aggiornata regolarmente, man mano che il team completa delle attività. Tuttavia, la Scrumboard sarà azzerata o ripulita alla fine dello Sprint e ne sarà creata una nuova per lo Sprint successivo.

#### 10.1.3.3 Registro degli Impedimenti Aggiornato

Descritto nella sezione 10.1.1.4.

#### 10.1.3.4 Richieste di Modifica Non Approvate

Descritte nel capitolo 8, sezione 8.4.1.6.

### 10.1.3.5 Rischi Identificati

Descritti nel capitolo 8, sezione 8.4.3.4.

### 10.1.3.6 Rischi Mitigati

Lo Scrum Team mette in pratica le azioni di mitigazione che sono state definite per trattare i Rischi precedentemente Identificati man mano che esegue il lavoro di creazione dei deliverable conformemente alle User Story del Product Backlog. Per tutta la durata del processo *Creare i Deliverable*, il team documenta gli eventuali nuovi Rischi Identificati e le azioni di mitigazione adottate. La registrazione dei rischi di progetto è un documento vitale, che viene aggiornato continuamente dal team nel corso del progetto per riflettere lo stato attuale di tutti i rischi.

Ulteriori informazioni sulla Gestione dei Rischi sono fornite nel capitolo 7, sezione 7.4.3.

### 10.1.3.7 Dipendenze Aggiornate

Descritte nel capitolo 9, sezione 9.3.3.3.

## 10.2 Condurre il Daily Standup

La Figura 10-6 mostra tutti gli input, gli strumenti e gli output del processo *Condurre il Daily Standup*.

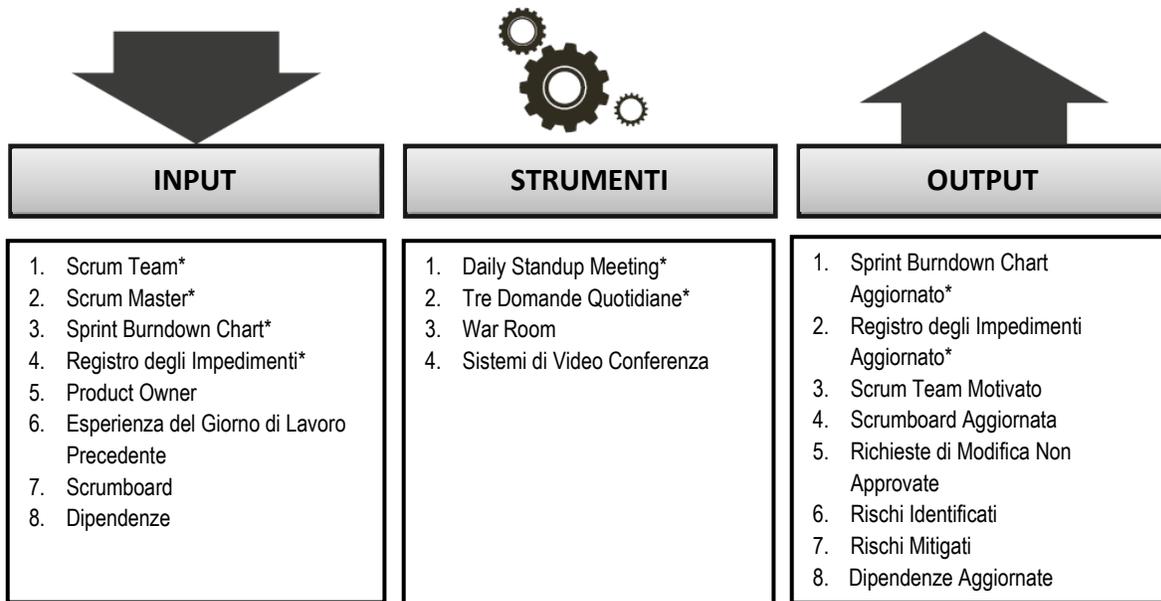


Figura 10-6: Condurre il Daily Standup—Input, Strumenti e Output

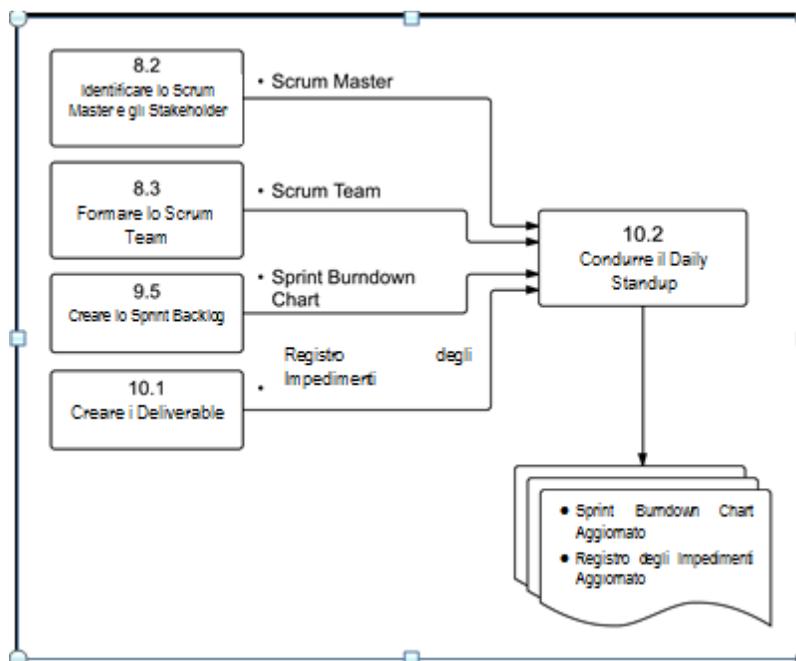


Figura 10-7: Conduct Daily Standup—Data Flow Diagram

## 10.2.1 Input

### 10.2.1.1 Scrum Team\*

Descritto nel capitolo 8, sezione 8.3.3.1.

### 10.2.1.2 Scrum Master\*

Descritto nel capitolo 8, sezione 8.2.3.1.

### 10.2.1.3 Sprint Burndown Chart\*

Descritto nel capitolo 9, sezione 9.5.3.2.

### 10.2.1.4 Registro degli Impedimenti\*

Descritto nella sezione 10.1.1.4.

### 10.2.1.5 Product Owner

Descritto nel capitolo 8, sezione 8.1.3.1.

### 10.2.1.6 Esperienza del Giorno di Lavoro Precedente

Nel Daily Standup Meeting i membri dello Scrum Team aggiornano i propri colleghi sullo stato del lavoro. Questa sessione viene chiamata Standup perché i partecipanti stanno in piedi per tutta la durata della riunione. I membri del team discutono i risultati conseguiti e l'esperienza fatta nel giorno di lavoro precedente. Questa esperienza rappresenta un input importante per il Daily Standup Meeting.

### 10.2.1.7 Scrumboard

Descritta nella sezione 10.1.1.3.

### 10.2.1.8 Dipendenze

Descritte nel capitolo 9, sezione 9.3.3.3.

## 10.2.2 Strumenti

### 10.2.2.1 Daily Standup Meeting\*

Il Daily Standup Meeting è una breve riunione quotidiana, che ha una durata predeterminata di 15 minuti. I membri del team si riuniscono per fare un resoconto sui loro progressi nello Sprint e per pianificare le attività del giorno. La durata della riunione è molto breve e ci si aspetta che partecipino tutti i membri dello Scrum Team. Tuttavia, la riunione non viene cancellata o rinviata se uno o più membri non possono partecipare.

Nel corso della riunione, ciascun membro dello Scrum Team risponde alle Tre Domande Quotidiane di cui si parla nella sezione 10.2.2.2. Le discussioni fra lo Scrum Master e il team o fra alcuni membri dello Scrum Team sono incoraggiate, ma devono avvenire dopo la riunione per garantire la brevità del Daily Standup Meeting.

### 10.2.2.2 Tre Domande Quotidiane\*

Nel Daily Standup Meeting, facilitato dallo Scrum Master, ciascun membro dello Scrum Team fornisce informazioni sotto forma di risposte a tre domande specifiche:

- Che cosa ho completato ieri?
- Cosa completerò oggi?
- Quali impedimenti o ostacoli sto eventualmente incontrando in questo momento?

Focalizzandosi su queste tre domande, l'intero team può farsi un'idea precisa dello stato del lavoro. Ogni tanto possono essere discussi anche altri elementi, ma questo dibattito viene ridotto al minimo, alla luce della durata predeterminata che caratterizza il tipo di riunione.

È fortemente raccomandato che, per quanto possibile, le risposte alle prime due domande vengano fornite dai membri del team in termini quantificabili, anziché come lunghe risposte qualitative. I membri del team possono organizzare ulteriori riunioni dopo il Daily Standup Meeting per trattare i temi che necessitano di un approfondimento della discussione.

### 10.2.2.3 War Room

In Scrum è preferibile la co-ubicazione del team, vale a dire che tutti i membri lavorino nel medesimo luogo. Il termine che si usa di solito per descrivere questo luogo è la War Room. Di norma è progettata in modo tale che i membri del team possono muoversi intorno liberamente, lavorare e comunicare facilmente perché si trovano a stretto contatto tra loro. Di solito nella stanza sono messi a disposizione index card, post-it e altri strumenti di interazione umana a bassa tecnologia, per facilitare il flusso di lavoro, la collaborazione e il problem solving.

A volte la stanza è rumorosa a causa delle conversazioni del team, ma queste conversazioni contribuiscono ai progressi del team. Una War Room ben fatta è priva di scomparti e permette all'intero team di stare

insieme garantendo la comunicazione faccia a faccia, che genera team building e apertura. La War Room è inoltre ideale per condurre i Daily Standup Meeting.

Possono passare per la War Room anche stakeholder e membri di altri Scrum Team per discutere di questioni importanti.

#### **10.2.2.4 Sistemi di Video Conferenza**

In situazioni reali, può non sempre essere possibile la co-ubicazione dell'intero Scrum Team. In questi casi, diventa assolutamente essenziale usare strumenti di video conferenza per consentire la comunicazione faccia a faccia.

### **10.2.3 Output**

#### **10.2.3.1 Sprint Burndown Chart Aggiornato\***

Descritto nel capitolo 9, sezione 9.5.3.2.

#### **10.2.3.2 Registro degli Impedimenti Aggiornato\***

Descritto nella sezione 10.1.1.4.

#### **10.2.3.3 Scrum Team Motivato**

I Daily Standup Meeting diffondono l'idea che ogni membro del team è importante e dà un contributo significativo, il che migliora il morale del singolo e del team. Questo, insieme al concetto di auto-organizzazione dei team, migliora la motivazione complessiva e sfocia in un aumento delle prestazioni del team e in un miglioramento della qualità dei deliverable prodotti.

Lo Scrum Team è descritto nel capitolo 8, sezione 8.3.3.1.

#### **10.2.3.4 Scrumboard Aggiornata**

Descritta nella sezione 10.1.1.3.

#### **10.2.3.5 Richieste di Modifica Non Approvate**

Descritte nel capitolo 8, sezione 8.4.1.6.

### **10.2.3.6 Rischi Identificati**

Descritti nel capitolo 8, sezione 8.4.3.4.

### **10.2.3.7 Rischi Mitigati**

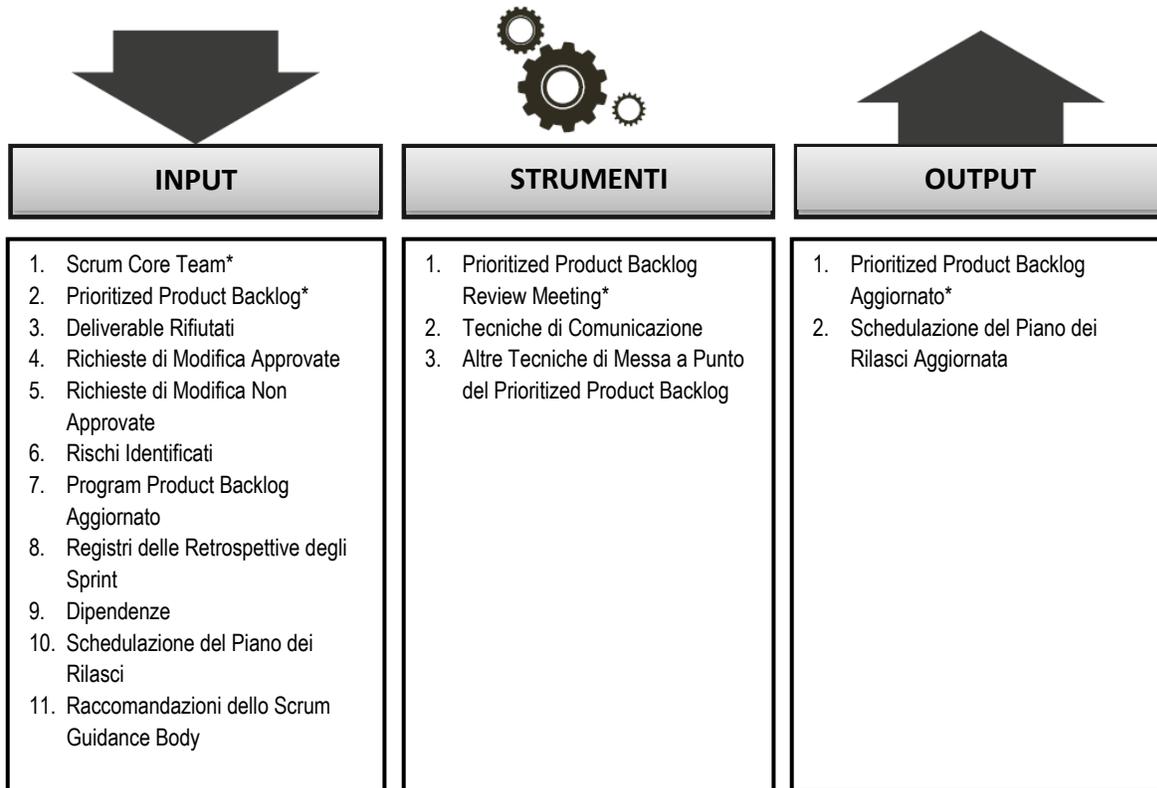
Descritti nella sezione 10.1.3.6.

### **10.2.3.8 Dipendenze Aggiornate**

Descritte nel capitolo 9, sezione 9.3.3.3.

## 10.3 Mettere a Punto il Prioritized Product Backlog

La Figura 10-8 mostra tutti gli input, gli strumenti e gli output del processo *Mettere a Punto il Prioritized Product Backlog*.



**Figura 10-8: Mettere a Punto il Prioritized Product Backlog—Input, Strumenti e Output**

*Nota:* Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente

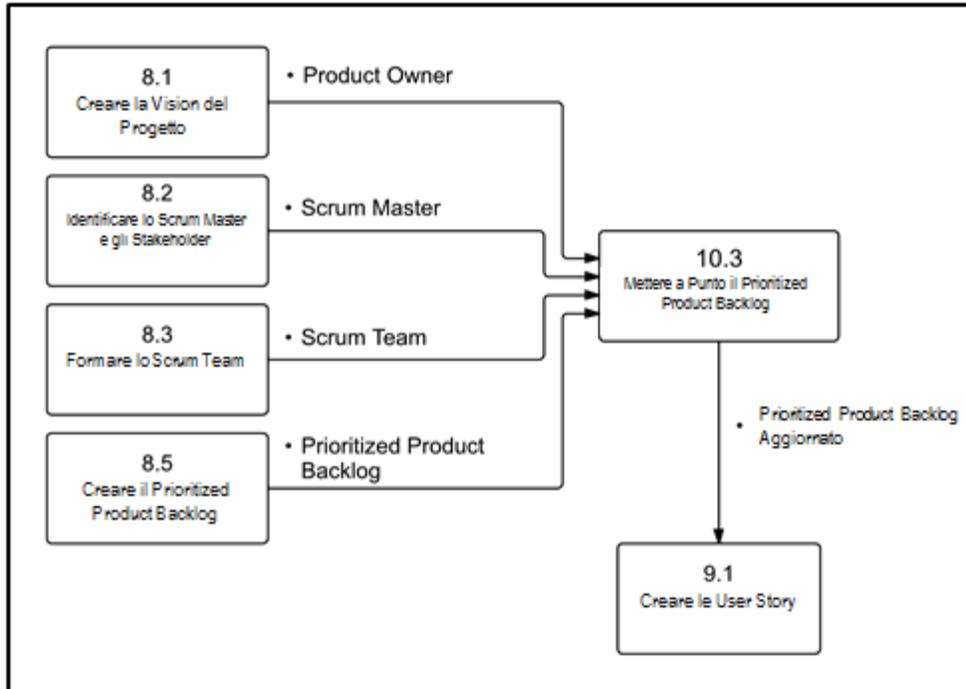


Figura 10-9: Mettere a Punto il Prioritized Product Backlog—Diagramma di Flusso dei Dati

Note: Asterisks (\*) denote a "mandatory" input, tool, or output for the corresponding process.

## 10.3.1 Input

### 10.3.1.1 Scrum Core Team\*

Descritto nel capitolo 8, sezioni 8.1.3.1, 8.2.3.1 e 8.3.3.1.

### 10.3.1.2 Prioritized Product Backlog\*

Descritto nel capitolo 8, sezione 8.5.3.1.

### 10.3.1.3 Deliverable Rifiutati

Nei casi in cui un deliverable non soddisfa i Criteri di Accettazione, viene considerato come Deliverable Rifiutato. Di norma i Deliverable Rifiutati non vengono tenuti in un elenco separato. Rimangono semplicemente nel Prioritized Product Backlog e non vengono segnati come fatti, di modo che possono essere ripriorizzati nel processo *Mettere a Punto il Prioritized Product Backlog* e quindi essere presi in considerazione per lo sviluppo nello Sprint successivo.

#### **10.3.1.4 Richieste di Modifica Approvate**

Descritte nel capitolo 8, sezione 8.4.1.5.

#### **10.3.1.5 Richieste di Modifica Non Approvate**

Descritte nel capitolo 8, sezione 8.4.1.6.

#### **10.3.1.6 Rischi Identificati**

Descritti nel capitolo 8, sezione 8.4.3.4.

#### **10.3.1.7 Program Product Backlog Aggiornato**

Analogamente al Project Product Backlog, anche il Program Product Backlog può subire una messa a punto periodica per incorporare i cambiamenti e i nuovi requisiti. I cambiamenti al Program Product Backlog possono derivare da cambiamenti delle condizioni sia interne che esterne. Le condizioni esterne potrebbero riguardare modifiche degli scenari di business, tendenze tecnologiche, o requisiti di conformità alla legge. I fattori interni che incidono sul Program Product Backlog potrebbero essere collegati a modifiche delle strategie o delle politiche dell'organizzazione, a Rischi Identificati o ad altri fattori. I cambiamenti dei requisiti del Program Product Backlog spesso impattano i Project Product Backlog dei progetti sottostanti, per cui devono essere tenuti in considerazione durante il processo *Mettere a Punto il Prioritized Product Backlog*.

#### **10.3.1.8 Registro delle Retrospective degli Sprint**

Descritto nel capitolo 11, sezione 11.3.3.4.

#### **10.3.1.9 Dipendenze**

Descritte nel capitolo 9, sezione 9.3.3.3.

#### **10.3.1.10 Schedulazione del Piano dei Rilasci**

Descritta nel capitolo 8, sezione 8.6.3.1.

#### **10.3.1.11 Raccomandazioni dello Scrum Guidance Body**

Descritte nel capitolo 8, sezione 8.1.1.12.

Le Raccomandazioni dello Scrum Guidance Body per il processo *Mettere a Punto il Prioritized Product Backlog* possono riguardare le best practice su come riuscire in modo sistematico a capire e a raccogliere i requisiti dagli Stakeholder e dagli Scrum Team e poi priorizzarli in modo appropriato nel Product Backlog e comunicare gli aggiornamenti a tutte le persone pertinenti coinvolte nel progetto Scrum.

## 10.3.2 Strumenti

### 10.3.2.1 Prioritized Product Backlog Review Meeting\*

Il Product Owner può tenere più riunioni separate con gli Stakeholder pertinenti, con lo Scrum Master e con lo Scrum Team, per essere sicuro di avere sufficienti informazioni per poter procedere all'aggiornamento del Prioritized Product Backlog durante il processo *Mettere a Punto il Prioritized Product Backlog*.

L'intento dei Prioritized Product Backlog Review Meeting è fare in modo che le User Story e i Criteri di Accettazione siano compresi e scritti in modo appropriato dal Product Owner, così da riflettere i reali requisiti e priorità dello stakeholder (cliente); che le User Story siano comprese da tutto lo Scrum Team e, infine che le User Story con priorità alta siano ben rifinite, in modo da poter essere correttamente stimate e quindi prese in carico dallo Scrum Team. I Prioritized Product Backlog Review Meeting assicurano inoltre la rimozione delle User Story inutili e l'incorporazione nel Prioritized Product Backlog di tutte le Richieste di Modifica Approvate e dei Rischi Identificati.

### 10.3.2.2 Tecniche di Comunicazione

Scrum promuove una comunicazione accurata ed efficace prima di tutto attraverso la co-ubicazione dello Scrum Team. Scrum favorisce inoltre le interazioni informali faccia a faccia rispetto alle comunicazioni scritte formali. Quando si deve dislocare uno Scrum Team in più luoghi, lo Scrum Master deve fare in modo che siano disponibili tecniche di comunicazione efficaci, in modo che il team possa auto-organizzarsi e lavorare efficacemente.

### 10.3.2.3 Altre Tecniche di Messa a Punto del Prioritized Product Backlog

Alcuni altri strumenti per la messa a punto del Prioritized Product Backlog includono molti di quelli utilizzati nei seguenti processi:

- *Sviluppare le Epic*—Descritto nel capitolo 8, sezione 8.4.2.
- *Creare il Prioritized Product Backlog*—Descritto nel capitolo 8, sezione 8.5.2.
- *Eseguire la Pianificazione dei Rilasci*—Descritto nel capitolo 8, sezione 8.6.2.
- *Creare le User Story*—Descritto nel capitolo 9, sezione 9.1.2.
- *Approvare, Stimare e Prendere in Carico le User Story*—Descritto nel capitolo 9, sezione 9.2.2.
- *Creare le Attività*—Descritto nel capitolo 9, sezione 9.3.2.
- *Stimare le Attività*—Descritto nel capitolo 9, sezione 9.4.2.

### 10.3.3 Output

#### 10.3.3.1 Prioritized Product Backlog Aggiornato\*

Descritto nel capitolo 8, sezione 8.5.3.1.

Il Prioritized Product Backlog può essere aggiornato con nuove User Story, nuove Richieste di Modifica, nuovi Rischi Identificati, User Story Aggiornate o riprioritizzazioni delle User Story esistenti.

#### 10.3.3.2 Schedulazione del Piano dei Rilasci Aggiornata

Descritta nel capitolo 8, sezione 8.6.3.1.

La Schedulazione del Piano dei Rilasci può essere aggiornata per riflettere l'impatto di User Story nuove o modificate inserite nel Prioritized Product Backlog.

## 10.4 Diagramma di Flusso dei Dati della Fase

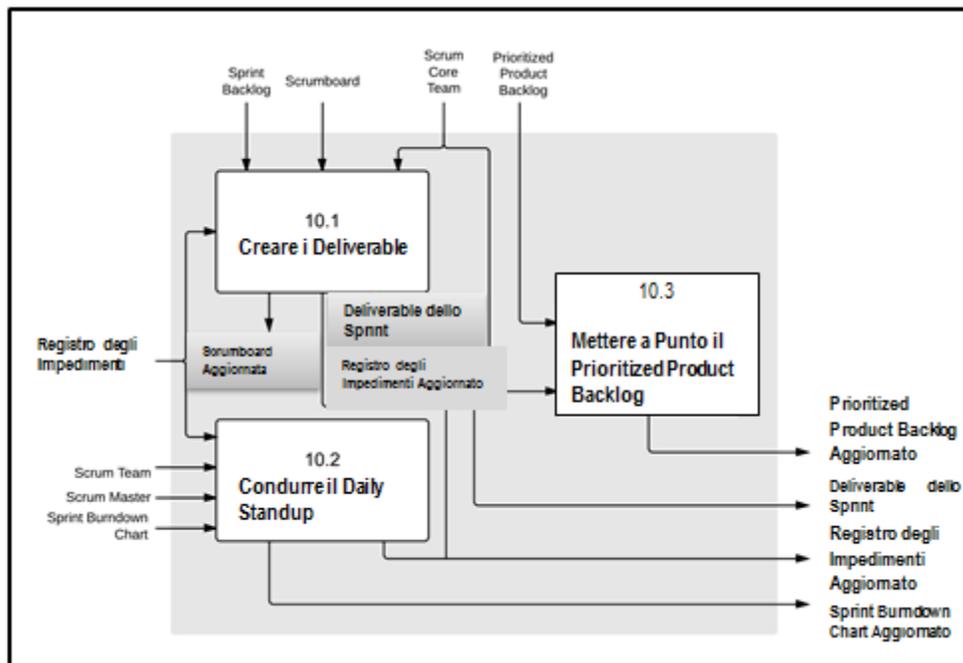


Figura 10-10: Fase di Implementazione—Diagramma di Flusso dei Dati

## 11. REVISIONE E RETROSPETTIVA

La Fase di Revisione e Retrospeffiva riguarda la revisione dei deliverable e del lavoro che è stato fatto, nonché la determinazione dei modi per migliorare le pratiche e i metodi usati per eseguire il lavoro di progetto. Nelle organizzazioni di grandi dimensioni i processi della fase *Revisione e Retrospeffiva* possono includere anche la convocazione degli Scrum of Scrums Meeting.

La Fase di *Revisione e Retrospeffiva*, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Per facilitare la migliore applicazione del framework Scrum, questo capitolo identifica gli input, gli strumenti e gli output di ciascun processo qualificandoli come “obbligatori” o “opzionali”. Gli input, gli strumenti e gli output contrassegnati da un asterisco (\*) sono obbligatori, mentre quelli senza asterisco sono opzionali.

Si raccomanda allo Scrum Team e a tutti coloro che vengono introdotti al framework e ai processi Scrum di concentrarsi soprattutto sugli input, gli strumenti e gli output obbligatori; i Product Owner, gli Scrum Master, e altri professionisti di Scrum più esperti dovrebbero invece sforzarsi di raggiungere una conoscenza più approfondita delle informazioni contenute nell'intero capitolo. È altresì importante avere chiaro che, sebbene tutti i processi siano definiti singolarmente nella *Guida SBOK™*, non è necessario eseguirli in maniera sequenziale o separatamente. A volte, a seconda dei requisiti specifici di ciascun progetto, potrebbe essere più appropriato unire alcuni processi fra loro.

Questo capitolo è scritto dal punto di vista di un singolo Scrum Team che lavora ad un singolo Sprint per produrre Deliverable potenzialmente consegnabili come parte di un progetto più ampio. Tuttavia, le informazioni fornite sono ugualmente applicabili a interi progetti, programmi e portfolio. Ulteriori informazioni sull'uso di Scrum nei progetti, programmi e portfolio sono disponibili nei capitoli da 2 a 7, che trattano i principi e gli aspetti di Scrum.

La Figura 11-1 fornisce una visione d'insieme dei processi della fase di Revisione e Retrospeffiva, che sono:

**11.1 Convocare lo Scrum degli Scrum**—In questo processo, i rappresentanti degli Scrum Team convocano gli Scrum of Scrums (SoS) Meeting a intervalli prestabiliti o ogni volta che sia necessario per collaborare e per avere un allineamento sui rispettivi stati di avanzamento, gli impedimenti e le dipendenze fra i vari team. Questo processo è appropriato solo per i progetti di grandi dimensioni, in cui sono coinvolti molti Scrum Team.

**11.2 Illustrare e Convalidare lo Sprint**—In questo processo, lo Scrum Team illustra i deliverable dello Sprint al Product Owner e agli stakeholder pertinenti nel corso di uno Sprint Review Meeting. Lo scopo di questa riunione è di assicurarsi l’approvazione e l’accettazione del prodotto o servizio da parte del Product Owner.

**11.3 Retrospettiva dello Sprint**—In questo processo, lo Scrum Master e lo Scrum Team si riuniscono per discutere le lezioni apprese nel corso dello Sprint. Queste informazioni vengono documentate come lezioni apprese che possono essere applicate in Sprint successivi. Spesso, da questa discussione possono scaturire dei Miglioramenti Fattibili Concordati o un aggiornamento delle Raccomandazioni dello Scrum Guidance Body.

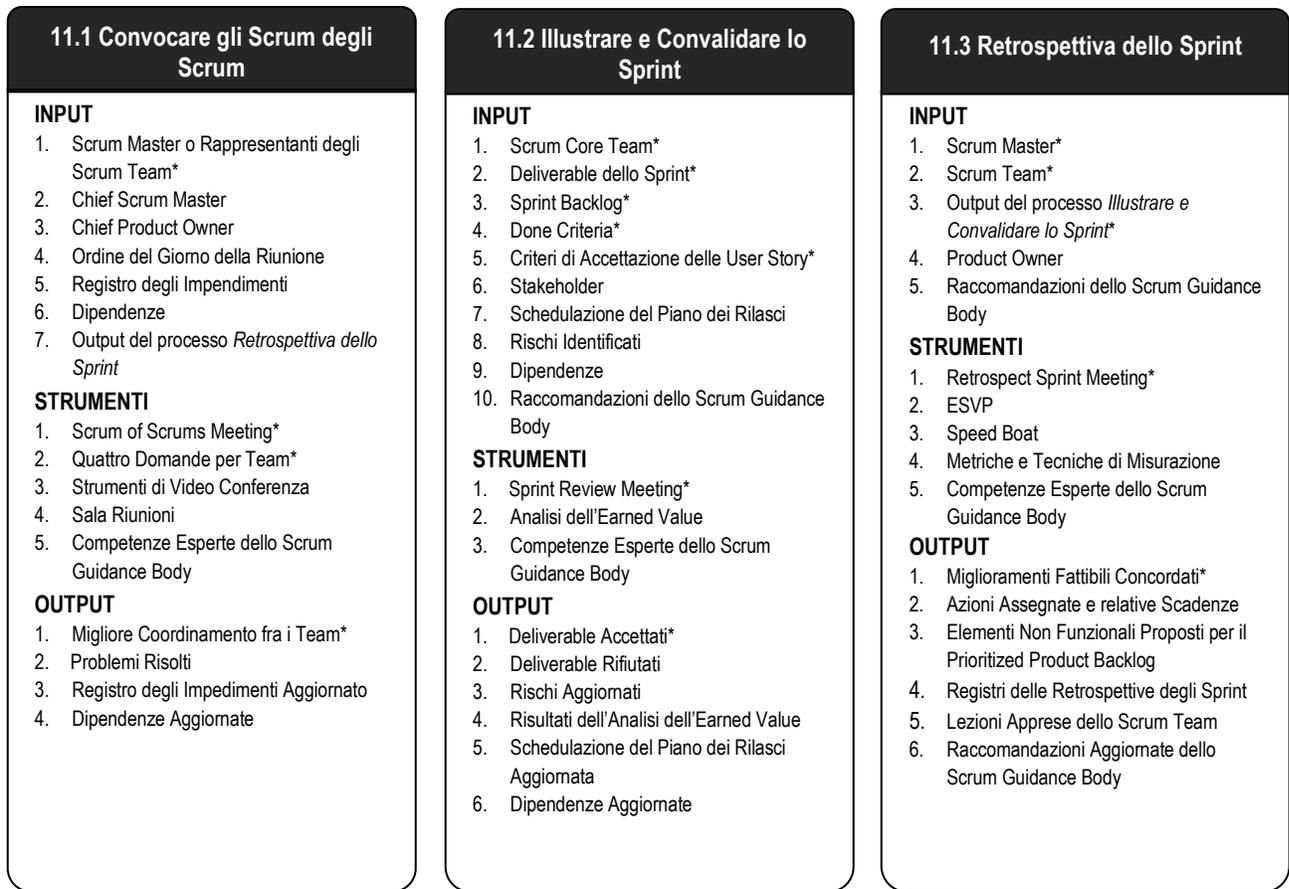
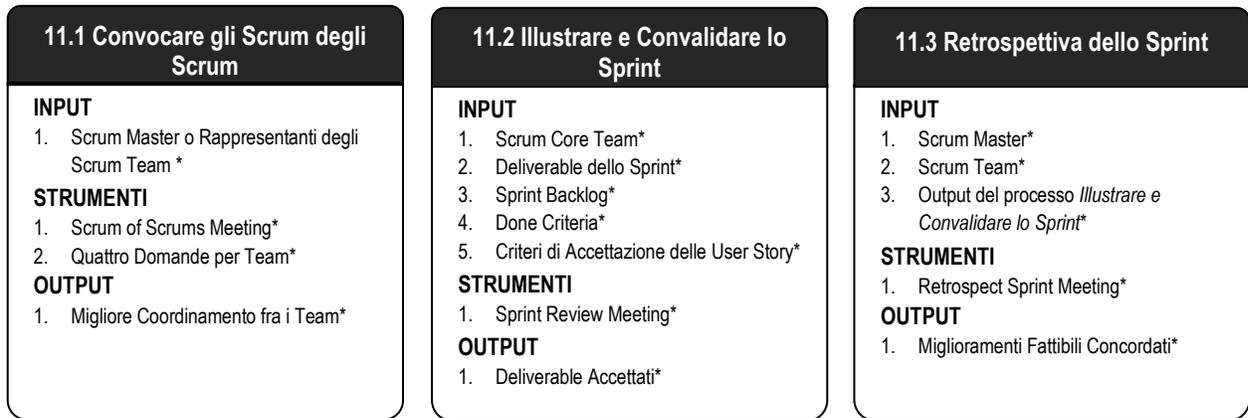


Figura 11-1: Visione d'Insieme della Fase di Revisione e Retrospettiva

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

La Figura 11-2 qui sotto mostra gli input, gli strumenti e gli output obbligatori dei processi della Fase di Revisione e Retrospeffiva.



**Figura 11-2: Visione d'Insieme della Fase di Revisione e Retrospeffiva (Elementi essenziali)**

*Nota:* Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 11.1 Convocare lo Scrum degli Scrum

La Figura 11-3 mostra tutti gli input, gli strumenti e gli output del processo *Convocare lo Scrum degli Scrum*.

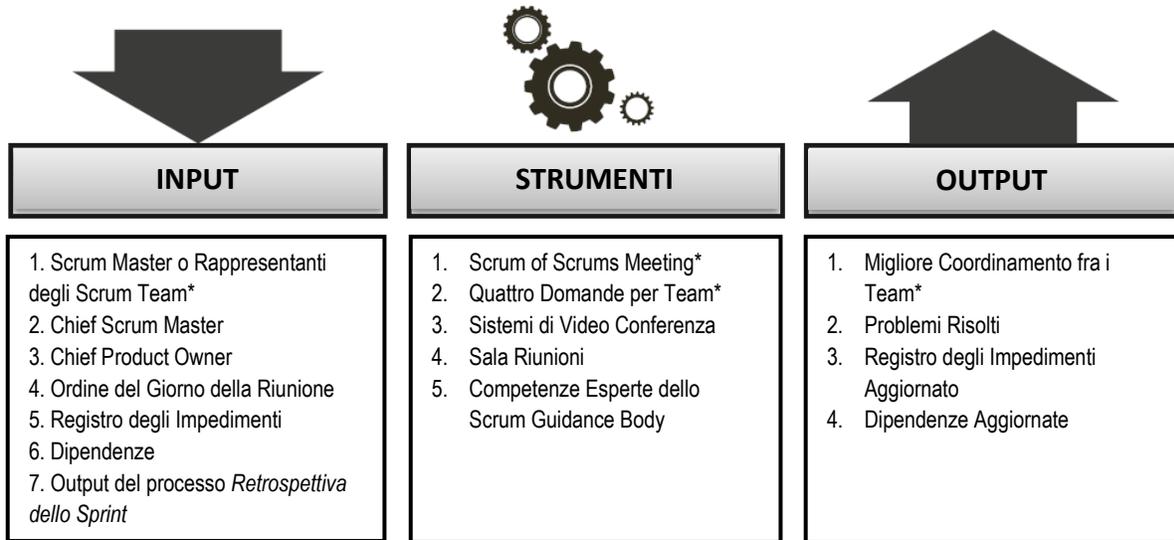


Figura 11-3: Convocare lo Scrum degli Scrum—Input, Strumenti e Output

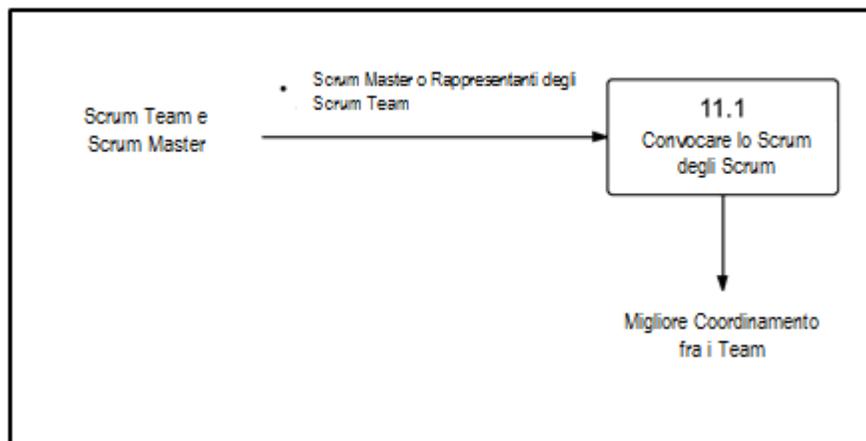


Figura 11-4: Convocare lo Scrum degli Scrum—Diagramma di Flusso dei Dati

*Nota:* Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 11.1.1 Input

### 11.1.1.1 Scrum Master o Rappresentanti degli Scrum Team\*

Di norma, un membro di ciascuno Scrum Team rappresenterà il proprio team nello Scrum of Scrums (SoS) Meeting. Nella maggior parte dei casi si tratta dello Scrum Master, ma a volte può essere anche qualcun altro a rappresentare il team. È possibile che sia nominata una sola specifica persona come rappresentante del team in tutti gli SoS Meeting, oppure il rappresentante può cambiare nel tempo, in base a chi è più adatto a ricoprire il ruolo in virtù delle problematiche e delle circostanze del momento. Ogni persona che partecipa alla riunione deve avere le conoscenze tecniche per poter identificare i casi in cui i team potrebbero essere reciprocamente causa di impedimenti o ritardi.

### 11.1.1.2 Chief Scrum Master

Descritto nel capitolo 8, sezione 8.2.1.6.

### 11.1.1.3 Chief Product Owner

Descritto nel capitolo 8, sezione 8.1.1.5.

### 11.1.1.4 Ordine del Giorno della Riunione

Lo scopo principale dello Scrum of Scrums (SoS) Meeting è di comunicare lo stato di avanzamento fra molti team. Il Chief Scrum Master (o qualsiasi Scrum Master che faciliti il SoS Meeting) può rendere noto un ordine del giorno prima della riunione. Questo consente ai singoli team di esaminare i punti all'ordine del giorno in vista del SoS Meeting. Devono essere indicati tutti gli eventuali impedimenti incontrati da un team che possono in qualche modo incidere anche sugli altri team, in modo da poterli convogliare nel SoS Meeting. Inoltre, se un team si rende conto di un problema di vaste dimensioni, di un cambiamento o di un rischio che possono impattare gli altri team, dovrebbe comunicarlo nel SoS Meeting.

### 11.1.1.5 Registro degli Impedimenti

Descritto nel capitolo 10, sezione 10.1.1.4.

### 11.1.1.6 Dipendenze

Descritte nel capitolo 9, sezione 9.3.3.3.

### 11.1.1.7 Output del processo Retrospektiva dello Sprint

Gli output del processo *Retrospektiva dello Sprint* possono impattare molti Scrum Team e possono essere utilizzati come input per un efficace Scrum of Scrums (SoS) Meeting.

## 11.1.2 Strumenti

### 11.1.2.1 Scrum of Scrums Meeting\*

Si tratta di riunioni preferibilmente brevi (ma di solito non di durata predeterminata, per consentire una maggiore condivisione delle informazioni fra i vari team) nelle quali i rappresentanti di ciascuno Scrum Team si riuniscono per condividere la situazione dei rispettivi team. Lo Scrum of Scrums (SoS) Meeting si tiene a intervalli prestabiliti, oppure quando richiesto dagli Scrum Team, per facilitare la condivisione delle informazioni fra Scrum Team differenti. I problemi, le dipendenze e i rischi che impattano su molteplici Scrum Team possono essere monitorati da vicino, il che aiuta i vari team che lavorano ad un progetto di grandi dimensioni a coordinare e integrare il proprio lavoro in modo migliore. È responsabilità del Chief Scrum Master (o di un altro Scrum Master che facilita i SoS Meeting) fare in modo che tutti i rappresentanti abbiano un ambiente favorevole alla condivisione aperta e onesta delle informazioni, incluso il feedback da parte dei rappresentanti degli altri team. Per i progetti più grandi, che coinvolgono un numero significativo di team, possono essere convocati più livelli di queste riunioni per condividere la situazione dei rispettivi team.

Il SoS Meeting è descritto in maggiore dettaglio nel capitolo 3, sezione 3.7.2.

### 11.1.2.2 Quattro Domande per Team\*

A turno, il rappresentante di ciascuno Scrum Team fornirà aggiornamenti riguardanti il proprio team. Questi aggiornamenti sono di solito forniti sotto forma di risposte a quattro specifiche domande.

- 1) A cosa ha lavorato il mio team nel tempo trascorso dalla precedente riunione?
- 2) Che cosa farà il mio team fino alla prossima riunione?
- 3) Cosa contavano gli altri team che il nostro team finisse e che invece è rimasto incompiuto?
- 4) Cosa sta pianificando di fare il nostro team che potrebbe avere un impatto sugli altri team?

Le risposte a queste quattro domande forniscono le informazioni che permettono a ciascun team di capire in modo chiaro lo stato del lavoro di tutti gli altri team.

### 11.1.2.3 Sistemi di Video Conferenza

Descritti nel capitolo 10, sezione 10.2.2.4.

È molto probabile che lo Scrum of Scrums (SoS) Meeting non venga effettuato faccia a faccia. La video conferenza è di solito necessaria per i progetti di grandi dimensioni, nei quali c'è una maggiore possibilità di team distribuiti.

### 11.1.2.4 Sala Riunioni

È consigliabile che per il SoS Meeting sia resa disponibile una sala conferenze dedicata, dove tutti i Rappresentanti degli Scrum Team possano sentirsi a loro agio.

### 11.1.2.5 Competenze Esperte dello Scrum Guidance Body

Descritte anche nel capitolo 8, sezione 8.4.2.7.

Con riferimento al processo *Convocare lo Scrum degli Scrum*, le Competenze Esperte dello Scrum Guidance Body possono riferirsi a best practice documentate relative a come condurre gli Scrum of Scrum (SoS) Meeting e a come incorporare i suggerimenti provenienti da questo tipo di riunioni nel lavoro di progetto dei singoli Scrum Team. Ci può anche essere un team di esperti della materia che può aiutare il Chief Scrum Master a facilitare il SoS Meeting.

## 11.1.3 Output

### 11.1.3.1 Migliore Coordinamento fra i Team\*

Lo Scrum of Scrums (SoS) Meeting agevola il coordinamento del lavoro fra molteplici Scrum Team. Questo è importante soprattutto quando ci sono delle attività che comportano delle interdipendenze fra i vari team. Le incompatibilità e le discrepanze tra il lavoro e i deliverable di team differenti vengono alla luce rapidamente. Questo forum dà inoltre ai team l'opportunità di presentare i propri risultati e di fornire il proprio feedback agli altri team. Grazie all'utilizzo del SoS Meeting, si ha una collaborazione diffusa nell'organizzazione, invece di persone che lavorano in team chiusi preoccupati prima di tutto delle loro responsabilità individuali.

### 11.1.3.2 Questioni Risolte

Lo Scrum of Scrums (SoS) Meeting è un forum dove i membri degli Scrum Team hanno la possibilità di discutere in modo trasparente i problemi che impattano sul loro progetto. La necessità di consegnare ogni Sprint rispettando i tempi forza i team ad affrontare subito attivamente questo tipo di problemi, invece di rinviare la ricerca di una soluzione. Questa tempestiva discussione e risoluzione dei problemi all'interno del SoS Meeting migliora notevolmente il coordinamento fra i diversi Scrum Team e riduce inoltre la necessità di riprogettazione e rilavorazione. Vengono mitigati anche i rischi collegati alle dipendenze e ai tempi di consegna.

Il SoS Meeting è descritto in maggiore dettaglio nel capitolo 3, sezione 3.7.2.1.

### 11.1.3.3 Registro degli Impedimenti Aggiornato

Descritto nel capitolo 10, sezione 10.1.3.3.

### 11.1.3.4 Dipendenze Aggiornate

Descritte nel capitolo 9, sezione 9.3.3.3.

## 11.2 Illustrare e Convalidare lo Sprint

La Figura 11-5 mostra tutti gli input, gli strumenti e gli output del processo *Illustrare e Convalidare lo Sprint*.

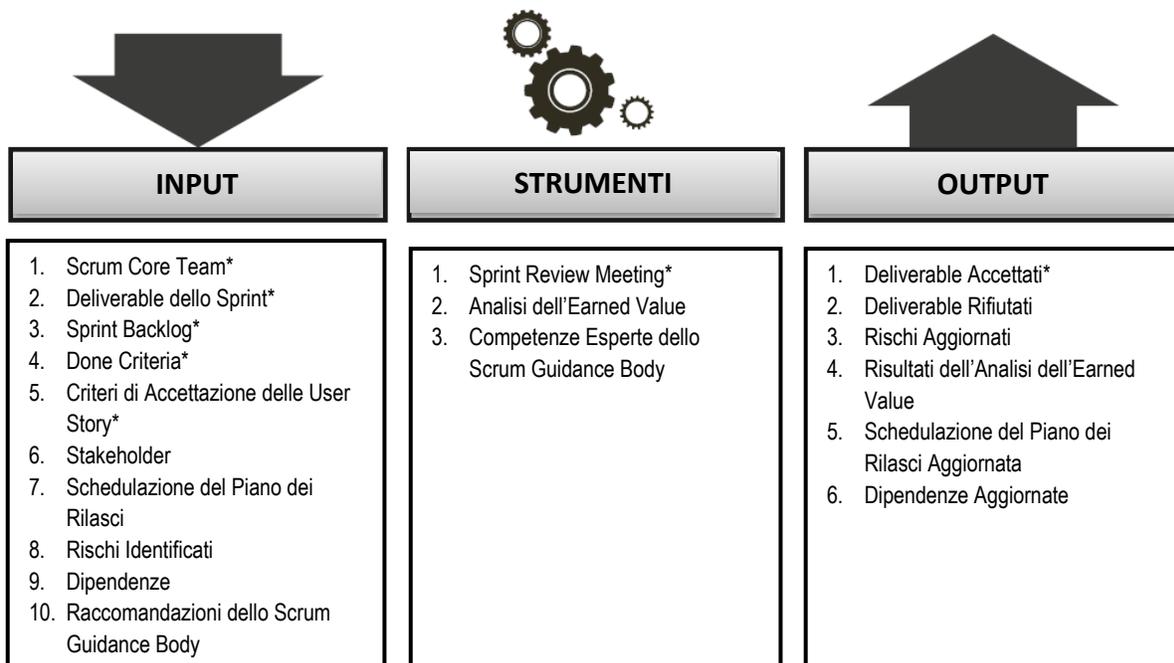


Figura 11-5: Illustrare e Convalidare lo Sprint—Input, Strumenti e Output

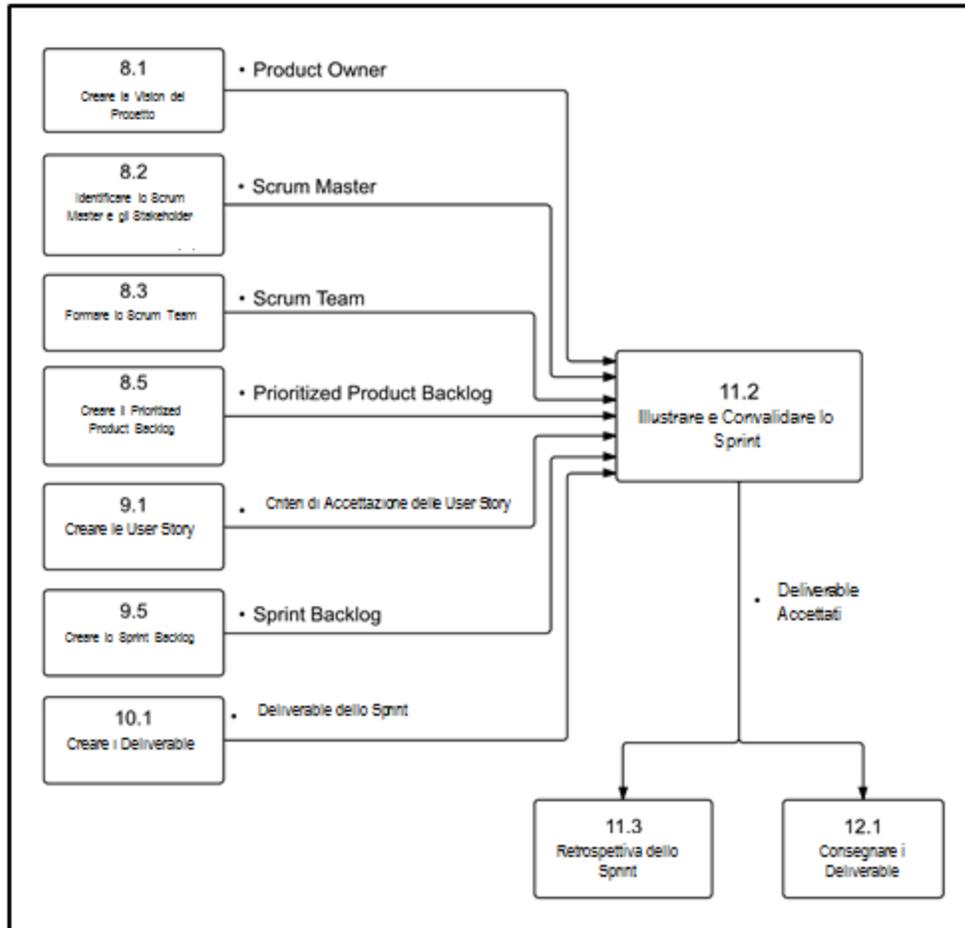


Figura 11-6: Illustrare e Convalidare lo Sprint—Diagramma di Flusso dei Dati

## 11.2.1 Input

### 11.2.1.1 Scrum Core Team\*

Descritto nel capitolo 8, sezione 8.4.1.1.

### 11.2.1.2 Deliverable dello Sprint\*

Descritti nel capitolo 10, sezione 10.1.3.1.

### 11.2.1.3 Sprint Backlog\*

Descritto nel capitolo 9, sezione 9.5.3.1.

#### **11.2.1.4 Done Criteria\***

Descritto nel capitolo 8, sezione 8.5.3.2.

#### **11.2.1.5 Criteri di Accettazione delle User Story\***

Descritti nel capitolo 9, sezione 9.4.1.3.

#### **11.2.1.6 Stakeholder**

Descritti nel capitolo 8, sezione 8.2.3.2.

#### **11.2.1.7 Schedulazione del Piano dei Rilasci**

Descritta nel capitolo 8, sezione 8.6.3.1.

#### **11.2.1.8 Rischi Identificati**

Descritti nel capitolo 8, sezione 8.4.3.4.

#### **11.2.1.9 Dipendenze**

Descritte nel capitolo 9, sezione 9.3.3.3.

#### **11.2.1.10 Raccomandazioni dello Scrum Guidance Body**

Descritte nel capitolo 8, sezione 8.1.1.12.

Con riferimento al processo *Illustrare e Convalidare lo Sprint*, le Raccomandazioni dello Scrum Guidance Body possono riguardare best practice su come condurre gli Sprint Review Meeting e valutare i risultati dell'Analisi dell'Earned Value. Vi possono inoltre essere delle indicazioni su come condividere le esperienze con altri Scrum Core Team e anche con altri Scrum Team del progetto.

### **11.2.2 Strumenti**

#### **11.2.2.1 Sprint Review Meeting\***

I membri dello Scrum Core Team e gli Stakeholder pertinenti partecipano agli Sprint Review Meeting per accettare i deliverable che soddisfano i Criteri di Accettazione delle singole User Story e per rifiutare i deliverable non accettabili. Queste riunioni sono convocate alla fine di ogni Sprint. Lo Scrum Team illustra i risultati dello Sprint, incluse le nuove funzionalità o i nuovi prodotti creati. Questo offre l'opportunità al

Product Owner e agli Stakeholder di esaminare ciò che è stato completato fino a quel momento e di stabilire se devono essere apportati eventuali cambiamenti al progetto o ai processi negli Sprint successivi.

### 11.2.2.2 Analisi dell'Earned Value

Descritta nel capitolo 4, sezione 4.6.1.

### 11.2.2.3 Competenze Esperte dello Scrum Guidance Body

Descritte nel capitolo 8, sezione 8.4.2.7.

Con riferimento al processo *Illustrare e Convalidare lo Sprint*, le Competenze Esperte dello Scrum Guidance Body possono riguardare best practice documentate su come condurre gli Sprint Review Meeting. Potrebbero inoltre essere disponibili degli esperti che possono fornire indicazioni su come facilitare al meglio uno Sprint Review Meeting.

## 11.2.3 Output

### 11.2.3.1 Deliverable Accettati\*

I Deliverable che soddisfano i Criteri di Accettazione delle singole User Story sono accettati dal Product Owner. L'obiettivo di uno Sprint è quello di creare deliverable potenzialmente consegnabili, o incrementi di prodotto, che soddisfano i Criteri di Accettazione definiti dal cliente e dal Product Owner. Questi sono appunto considerati Deliverable Accettati, che possono essere rilasciati al cliente, se così desidera. Si mantiene una lista dei Deliverable Accettati e la si aggiorna dopo ogni Sprint Review Meeting. Se un deliverable non soddisfa i Criteri di Accettazione stabiliti, non si considera accettato e di norma sarà riportato in uno Sprint successivo per correggere tutti gli eventuali problemi. Una situazione di questo tipo è altamente indesiderabile, perché l'obiettivo di ogni Sprint è che i deliverable soddisfino i criteri di accettazione.

### 11.2.3.2 Deliverable Rifiutati

I Deliverable che non soddisfano i Criteri di Accettazione sono rifiutati. Le User Story associate con tali Deliverable Rifiutati vengono aggiunti al Prioritized Product Backlog, in modo da poter prendere in considerazione tali deliverable come parte di uno Sprint successivo.

### 11.2.3.3 Rischi Aggiornati

Descritti nel capitolo 8, sezione 8.4.3.4.

### 11.2.3.4 Risultati dell'Analisi dell'Earned Value

Descritti nel capitolo 4, sezione 4.6.1.

### 11.2.3.5 Schedulazione del Piano dei Rilasci Aggiornata

Descritta nel capitolo 10, sezione 10.3.3.2.

### 11.2.3.6 Dipendenze Aggiornate

Descritte nel capitolo 9, sezione 9.3.3.3.

## 11.3 Retrospektiva dello Sprint

La Figura 11-7 mostra tutti gli input, gli strumenti e gli output del processo *Retrospektiva dello Sprint*.

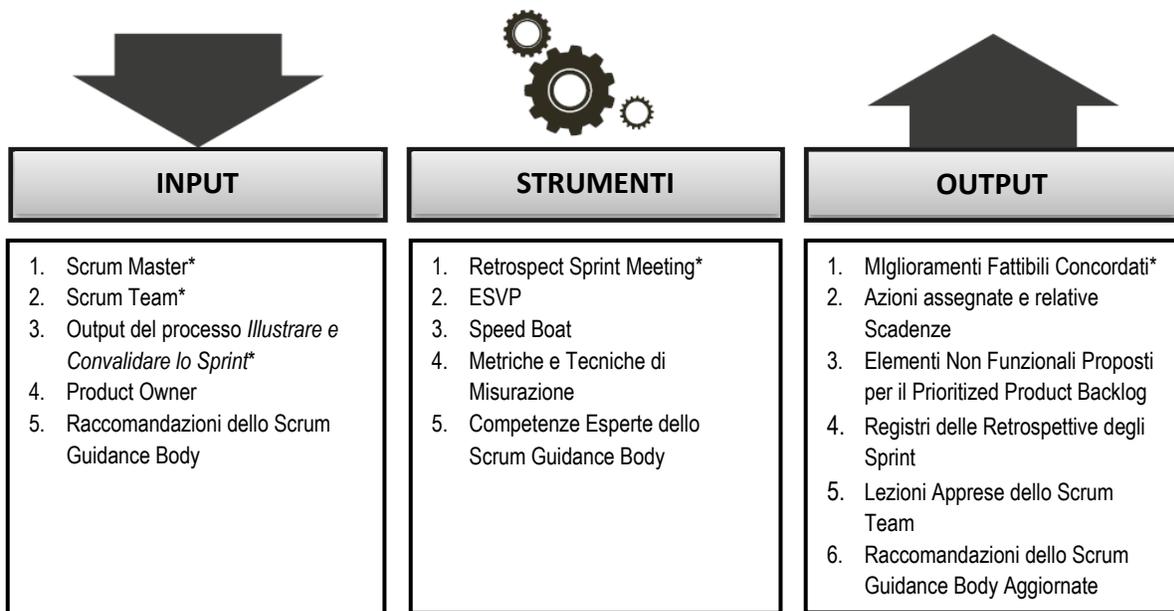


Figura 11-7: Retrospektiva dello Sprint—Input, Strumenti e Output

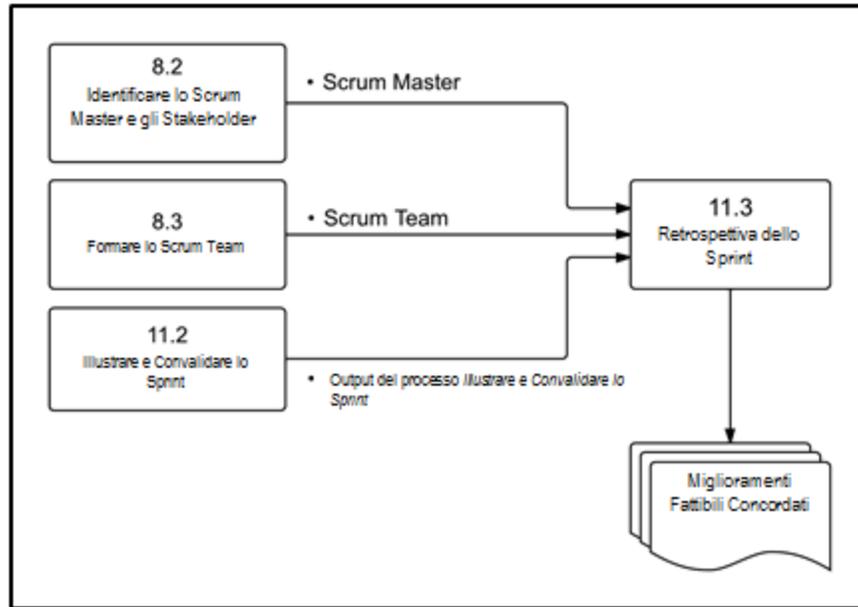


Figura 11-8: Retrospectiva dello Sprint—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 11.3.1 Input

### 11.3.1.1 Scrum Master\*

Descritto nel capitolo 8, sezione 8.2.3.1.

### 11.3.1.2 Scrum Team\*

Descritto nel capitolo 8, sezione 8.3.3.1.

### 11.3.1.3 Output del processo *Illustrare e Convalidare lo Sprint*\*

Descritti nella sezione 11.2.3.

Gli output del processo *Illustrare e Convalidare lo Sprint* forniscono informazioni preziose durante l'esecuzione del processo *Retrospectiva dello Sprint*.

### 11.3.1.4 Product Owner

Descritto nel capitolo 8, sezione 8.1.3.1.

### 11.3.1.5 Raccomandazioni dello Scrum Guidance Body

Lo Scrum Guidance Body può fornire linee guida per la conduzione dei Retrospect Sprint Meeting, compresi suggerimenti sugli strumenti da utilizzare e sulla documentazione o sui deliverable attesi da queste riunioni.

## 11.3.2 Strumenti

### 11.3.2.1 Retrospect Sprint Meeting\*

Il Retrospect Sprint Meeting è un elemento importante del concetto di “ispezione-adattamento” del framework Scrum e costituisce il passo finale di uno Sprint. Tutti i membri dello Scrum Team partecipano alla riunione, che viene facilitata o moderata dallo Scrum Master. È consigliata, ma non obbligatoria, la partecipazione del Product Owner. Uno dei membri del team funge da segretario verbalizzante e documenta le discussioni e i punti delle azioni future. È fondamentale tenere questa riunione in un ambiente aperto e rilassato per incoraggiare la piena partecipazione di tutti i membri del team. Le discussioni del Retrospect Sprint Meeting riguardano sia ciò che è andato storto sia ciò che è andato bene. L'obiettivo primario della riunione è quello di identificare tre elementi specifici:

- 1) Cose che il team desidera continuare a fare: best practice
- 2) Cose che il team desidera iniziare a fare: miglioramenti dei processi
- 3) Cose che il team desidera non fare più: problemi e ‘colli di bottiglia’ dei processi.

Queste aree vengono discusse e viene creato un elenco di Miglioramenti Fattibili Concordati.

### 11.3.2.2 Explorer—Shopper—Vacationer—Prisoner (ESVP)

Questo esercizio può essere eseguito all'inizio del Retrospect Sprint Meeting per capire l'atteggiamento mentale dei partecipanti e stabilire il tono della riunione. Viene chiesto ai presenti di indicare in forma anonima quale profilo rappresenta meglio come si sentono rispetto al dover prendere parte alla riunione.

- Explorer (Esploratore)—Vuole partecipare e imparare tutto ciò che viene discusso nella retrospettiva
- Shopper (Acquirente)—Vuole ascoltare tutto e scegliere cosa portarsi via dalla retrospettiva
- Vacationer (Turista)—Vuole rilassarsi e comportarsi da turista nella retrospettiva
- Prisoner (Prigioniero)—Vuole essere altrove e sta partecipando alla riunione perché è necessario

Lo Scrum Master raccoglie quindi le risposte, predispone le informazioni e le condivide con il gruppo.

### 11.3.2.3 Speed Boat

La Speed Boat è una tecnica che può essere utilizzata per condurre il Retrospect Sprint Meeting. I membri del team fanno la parte dell'equipaggio di un motoscafo (Speed Boat). L'imbarcazione deve raggiungere

un'isola, che rappresenta simbolicamente la Vision del Progetto. I partecipanti utilizzano dei post-it per registrare i motori e le ancore. I motori li aiutano a raggiungere l'isola, mentre le ancore impediscono loro di raggiungerla. Questo esercizio ha una durata prefissata di pochi minuti. Una volta che sono stati documentati tutti i punti, le informazioni sono raccolte, discusse e priorizzate mediante un processo di votazione. Vengono quindi riconosciuti i motori e pianificate delle azioni di mitigazione per le ancore, in base alla priorità.

#### 11.3.2.4 Metriche e Tecniche di Misurazione

Per misurare le prestazioni del team nell'attuale Sprint e confrontarle con quelle degli Sprint precedenti possono essere utilizzate varie metriche. Alcuni esempi di queste metriche sono:

- La velocità del team—Numero di story point completati in un dato Sprint
- Tasso di successo di ciò che è stato completato—Percentuale di story point che sono stati completati (Done) rispetto a quelli presi in carico
- Efficacia delle Stime—Numero o percentuale di deviazioni fra il tempo stimato e quello realmente impiegato per le attività e le User Story
- Rivedere le valutazioni di feedback—Si può richiedere il feedback da parte degli Stakeholder utilizzando valutazioni quantitative o qualitative, che forniscono una misura delle prestazioni del team.
- Valutazioni del morale del team—Risultati delle auto-valutazioni del morale dei membri del team
- Feedback fra pari—Meccanismi di feedback a 360 gradi possono essere utilizzati per sollecitare critiche costruttive e opinioni sulle prestazioni del team
- Stato di avanzamento rispetto al rilascio o al lancio—Valore di business fornito da ogni rilascio, oltre al valore rappresentato dallo stato di avanzamento attuale rispetto ad un rilascio. Questo contribuisce alla motivazione del team e al livello di soddisfazione lavorativa.

#### 11.3.2.5 Competenze Esperte dello Scrum Guidance Body

Descritte anche nel capitolo 8, sezione 8.4.2.7.

Con riferimento al processo *Retrospectiva dello Sprint*, le Competenze Esperte dello Scrum Guidance Body possono riguardare le best practice per la conduzione dei Retrospect Sprint Meeting. Ci possono essere anche alcuni esperti che potrebbero contribuire a fornire indicazioni su come utilizzare gli strumenti nel processo *Retrospectiva dello Sprint* per realizzare Miglioramenti Fattibili Concordati per gli Sprint futuri.

### 11.3.3 Output

#### 11.3.3.1 Miglioramenti Fattibili Concordati\*

I Miglioramenti Fattibili Concordati sono il principale output del processo *Retrospektiva dello Sprint*. Rappresentano l'elenco delle iniziative perseguibili messe a punto dal team per affrontare i problemi e migliorare i processi al fine di aumentare le proprie prestazioni negli Sprint a venire.

#### 11.3.3.2 Azioni Assegnate e relative Scadenze

Una volta che i Miglioramenti Fattibili Concordati sono stati elaborati e rifiniti, lo Scrum Team può prendere in considerazione le azioni che servono ad attuare i miglioramenti. Ogni singola azione avrà una scadenza definita per il suo completamento.

#### 11.3.3.3 Elementi Non Funzionali Proposti per il Prioritized Product Backlog

Il Prioritized Product Backlog sviluppato inizialmente si basa sulle User Story e sulle funzionalità richieste. Spesso nelle prime fasi del progetto i requisiti non funzionali non sono ancora pienamente definiti e possono emergere durante lo Sprint Review o il Retrospect Sprint Meeting. Questi elementi devono essere aggiunti al Prioritized Product Backlog non appena vengono alla luce. Alcuni esempi di requisiti non funzionali sono i tempi di risposta, le limitazioni di capacità e le problematiche legate alla sicurezza.

#### 11.3.3.4 Registri delle Retrospektive degli Sprint

Il Registro della Retrospektiva dello Sprint è una registrazione delle opinioni, delle discussioni e delle iniziative perseguibili emerse in un Retrospect Sprint Meeting. Lo Scrum Master può facilitare la creazione di questo registro con gli input provenienti dai membri dello Scrum Core Team. L'insieme di tutti i Registri delle Retrospektive degli Sprint diventa il diario di progetto e dettaglia i successi, le questioni, i problemi del progetto e le soluzioni. I registri sono documenti pubblici a disposizione di tutti coloro che fanno parte dell'organizzazione.

#### 11.3.3.5 Lezioni Apprese dello Scrum Team

Ci si aspetta che lo Scrum Team, a motivo della sua auto-organizzazione e del potere che gli è stato conferito, sia capace di imparare dagli errori commessi durante uno Sprint. Queste lezioni apprese aiutano i team a migliorare le proprie prestazioni negli Sprint a venire e possono essere anche documentate all'interno delle Raccomandazioni dello Scrum Guidance Body per essere condivise con altri Scrum Team.

Da uno Sprint possono scaturire moltissime lezioni apprese positive. Queste lezioni apprese positive rappresentano una parte fondamentale della retrospektiva e dovrebbero essere adeguatamente condivise

all'interno del team e con lo Scrum Guidance Body, poiché i team lavorano per il continuo auto-miglioramento.

### 11.3.3.6 Raccomandazioni dello Scrum Guidance Body Aggiornate

Da un Retrospect Sprint Meeting possono scaturire suggerimenti per rivedere o migliorare le Raccomandazioni dello Scrum Guidance Body. Se il Guidance Body accetta questi suggerimenti, questi saranno incorporati come aggiornamenti della documentazione dello Scrum Guidance Body.

## 11.4 Diagramma di Flusso dei Dati della Fase

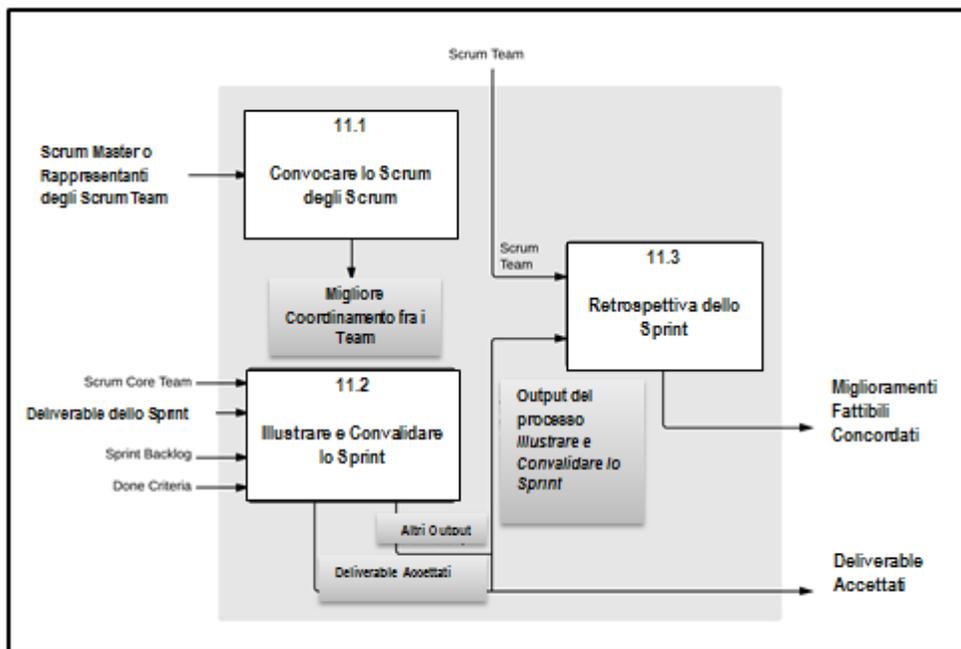


Figura 11-9: Fase di Revisione e Retrospectiva—Diagramma di Flusso dei Dati

## 12. RILASCIO

La fase di Rilascio pone l'accento sulla consegna dei Deliverable Accettati al cliente e sull'identificazione, documentazione e interiorizzazione delle lezioni apprese durante il progetto.

La fase di *Rilascio*, così come definita nella *Guida al Corpo di Conoscenze di Scrum (Guida SBOK™)*, si applica alle situazioni di seguito elencate:

- Portfolio, programmi e/o progetti di qualsiasi settore industriale
- Prodotti, servizi, o qualsiasi altro risultato da consegnare agli stakeholder
- Progetti di qualsiasi dimensione o complessità

Il termine “prodotto” nella *Guida SBOK™* si può riferire ad un prodotto, servizio, o altro deliverable. Scrum può essere applicato in maniera efficace a qualsiasi progetto di qualunque settore industriale – dai piccoli progetti o team di appena sei membri fino ai progetti grandi e complessi che arrivano a diverse centinaia di membri del team.

Per facilitare la migliore applicazione del framework Scrum, questo capitolo identifica gli input, gli strumenti e gli output di ciascun processo qualificandoli come “obbligatori” o “opzionali”. Gli input, gli strumenti e gli output contrassegnati da un asterisco (\*) sono obbligatori, mentre quelli senza asterisco sono opzionali.

Si raccomanda allo Scrum Team e a tutti coloro che vengono introdotti al framework e ai processi Scrum di concentrarsi soprattutto sugli input, gli strumenti e gli output obbligatori; i Product Owner, gli Scrum Master, e altri professionisti di Scrum più esperti dovrebbero invece sforzarsi di raggiungere una conoscenza più approfondita delle informazioni contenute nell'intero capitolo. È altresì importante avere chiaro che, sebbene tutti i processi siano definiti singolarmente nella *Guida SBOK™*, non è necessario eseguirli in maniera sequenziale o separatamente. A volte, a seconda dei requisiti specifici di ciascun progetto, potrebbe essere più appropriato unire alcuni processi fra loro.

Questo capitolo è scritto dal punto di vista di un singolo Scrum Team che lavora ad un singolo Sprint per produrre Deliverable potenzialmente consegnabili come parte di un progetto più ampio. Tuttavia, le informazioni fornite sono ugualmente applicabili a interi progetti, programmi e portfolio. Ulteriori informazioni sull'uso di Scrum nei progetti, programmi e portfolio sono disponibili nei capitoli da 2 a 7, che trattano i principi e gli aspetti di Scrum.

La Figura 12-1 fornisce una visione d'insieme dei processi della fase di Rilascio, che sono:

**12.1 Consegnare i Deliverable**—In questo processo i Deliverable Accettati sono consegnati o trasferiti agli stakeholder pertinenti. Un Accordo formale sui Deliverable Funzionanti documenta il buon esito dello Sprint.

**12.2 Retrospettiva del Progetto**—In questo processo, che completa il progetto, gli stakeholder dell'organizzazione e i membri dello Scrum Core Team si riuniscono per fare la retrospettiva del progetto e identificare, documentare e interiorizzare le lezioni apprese. Spesso queste lezioni portano alla documentazione di Miglioramenti Fattibili Concordati, da implementare in futuri progetti.

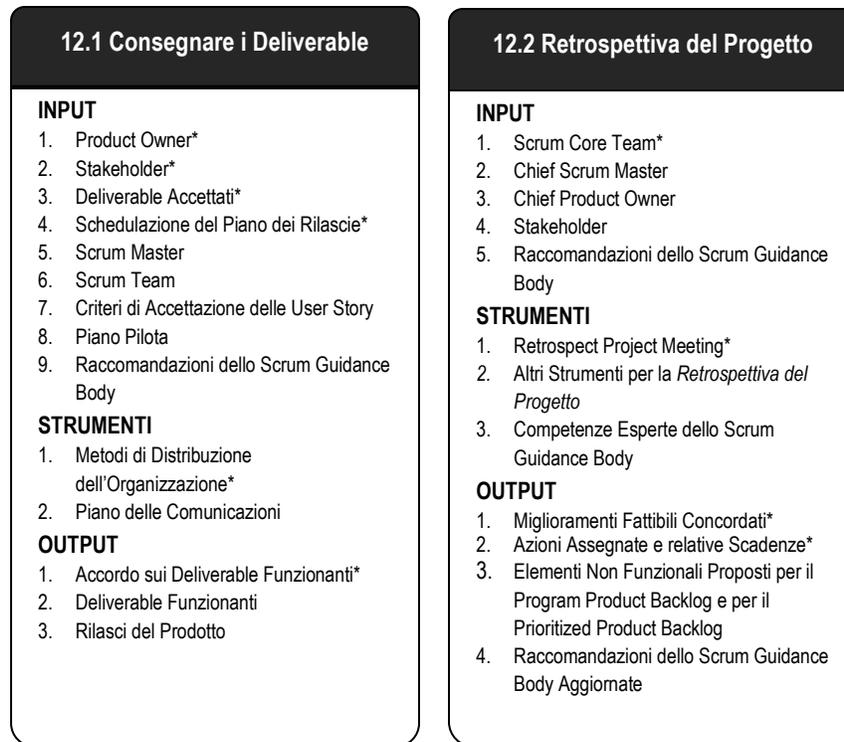


Figura 12-1: Visione d'insieme della Fase di Rilascio

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

La Figura 12-2 qui sotto mostra gli input, gli strumenti e gli output obbligatori dei processi della Fase di Rilascio.

12

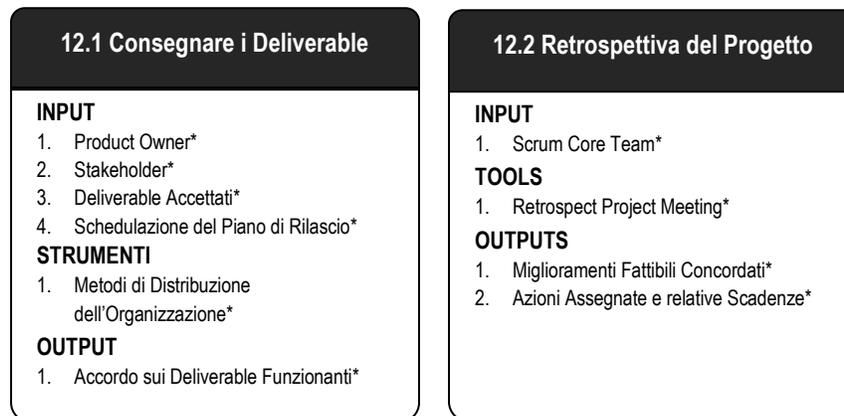


Figura 12-2: Visione d'insieme della fase di Rilascio (Elementi essenziali)

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 12.1 Consegnare i Deliverable

La Figura 12-3 mostra tutti gli input, gli strumenti e gli output del processo *Consegnare i Deliverable*.

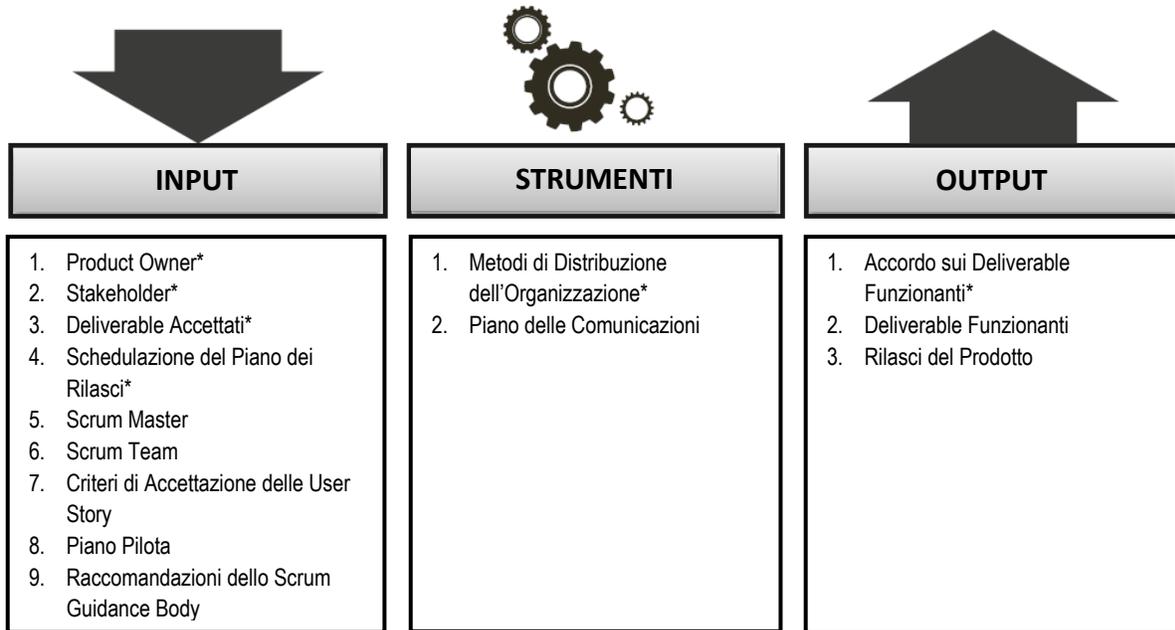


Figura 12-3: Consegnare i Deliverable—Input, Strumenti e Output

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo

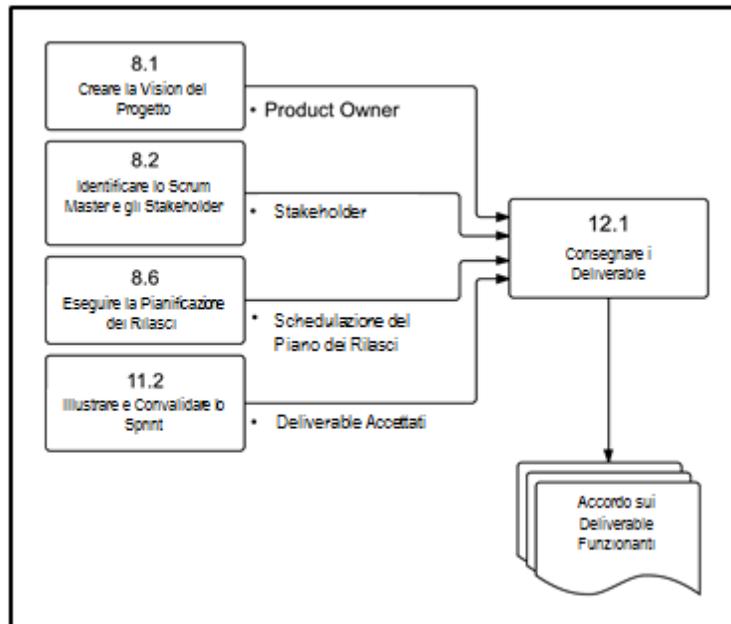


Figura 12-4: Consegnare i Deliverable—Diagramma di Flusso dei Prodotti

## **12.1.1 Input**

### **12.1.1.1 Product Owner\***

Descritto nel capitolo 8, sezione 8.1.3.1.

### **12.1.1.2 Stakeholder\***

Descritti nel capitolo 8, sezione 8.2.3.2.

### **12.1.1.3 Deliverable Accettati\***

Descritti nel capitolo 11, sezione 11.2.3.1.

### **12.1.1.4 Schedulazione del Piano dei Rilasci**

Descritta nel capitolo 8, sezione 8.6.3.1.

### **12.1.1.5 Scrum Master**

Descritto nel capitolo 8, sezione 8.2.3.1.

### **12.1.1.6 Scrum Team**

Descritto nel capitolo 8, sezione 8.3.3.1.

### **12.1.1.7 Criteri di Accettazione delle User Story**

Descritti nel capitolo 9, sezione 9.1.3.2.

### **12.1.1.8 Piano Pilota**

Un Piano Pilota è un input opzionale che può essere utilizzato per tracciare nel dettaglio una distribuzione pilota. Nel Piano Pilota sono specificati e condivisi con gli stakeholder l'ambito e gli obiettivi della distribuzione, la base di utenti target della distribuzione, un programma di distribuzione, i piani di transizione, la preparazione necessaria per gli utenti, i criteri di valutazione della distribuzione ed altri elementi chiave legati alla distribuzione.

### **12.1.1.9 Raccomandazioni dello Scrum Guidance Body**

Descritte nel capitolo 8, sezione 8.1.1.12.

Con riferimento al processo *Consegnare i Deliverable*, lo Scrum Guidance Body può fornire raccomandazioni e linee guida riguardo alla distribuzione dei prodotti. Si tratta di best practice che dovrebbero essere tenute presenti quando si distribuisce un prodotto al cliente al fine di massimizzare il valore consegnato.

## **12.1.2 Strumenti**

### **12.1.2.1 Metodi di Distribuzione dell'Organizzazione\***

I meccanismi di distribuzione delle varie organizzazioni tendono a differenziarsi in base al settore industriale, agli utenti target e al posizionamento. A seconda del prodotto da consegnare, la distribuzione può avvenire da remoto o può invece implicare la spedizione fisica o il trasferimento di un articolo. Poiché la distribuzione implica tendenzialmente un livello di rischio elevato, di solito le organizzazioni possiedono meccanismi di distribuzione ben definiti e consolidati, muniti di processi dettagliati per assicurare la conformità a tutti gli standard applicabili e alle misure di garanzia della qualità. Tra questi possono essere inclusi autorizzazioni da parte di specifici rappresentanti del management, meccanismi di approvazione dell'utente, nonché linee guida riguardo alle funzionalità minime di un rilascio.

### **12.1.2.2 Piano delle Comunicazioni**

In molti progetti esiste un Piano delle Comunicazioni. Questo piano specifica le registrazioni che devono essere create e mantenute durante tutto il progetto. Per veicolare agli stakeholder le informazioni di progetto importanti viene utilizzata una vasta gamma di metodi. Il Piano delle Comunicazioni definisce questi metodi e le responsabilità relative alle varie attività di comunicazione. Quando i Deliverable sono testati, le attività di testing vengono comunicate per mezzo del Piano delle Comunicazioni come stabilito dal Product Owner e dallo sponsor. Un meccanismo di comunicazione diffuso è costituito da un'unità di visualizzazione che raffigura le informazioni importanti in un formato di facile interpretazione, posizionata in un luogo accessibile e mantenuta aggiornata con le informazioni più attuali.

## 12.1.3 Output

### 12.1.3.1 Accordo sui Deliverable Funzionanti\*

I Deliverable che soddisfano i Criteri di Accettazione ricevono la formale autorizzazione del business e l'approvazione del cliente o dello sponsor. Per vedersi riconosciuto il ricavo è fondamentale ottenere l'accettazione formale da parte del cliente e la responsabilità del suo ottenimento sarà definita dalle politiche aziendali, non ricadendo necessariamente sul Product Owner.

### 12.1.3.2 Deliverable Funzionanti

Questo output rappresenta il Deliverable consegnabile finale per il quale è stato deliberato il progetto. Man mano che sono creati, i nuovi incrementi di prodotto vengono continuamente integrati negli incrementi precedenti, di modo che in ogni momento nel corso del progetto è disponibile un prodotto potenzialmente consegnabile.

### 12.1.3.3 Rilasci di Prodotto

I Rilasci di Prodotto dovrebbero includere:

- **Contenuto del Rilascio**—Consiste delle informazioni essenziali riguardanti i deliverable che possono aiutare il Team di Supporto ai Clienti.
- **Note di Rilascio**—Le Note di Rilascio attengono ai criteri di spedizione esterni o rivolti al mercato relativi al prodotto da consegnare.

## 12.2 Retrospeffiva del Progetto

La Figura 12-5 mostra tutti gli input, gli strumenti e gli output del processo *Retrospeffiva del Progetto*.

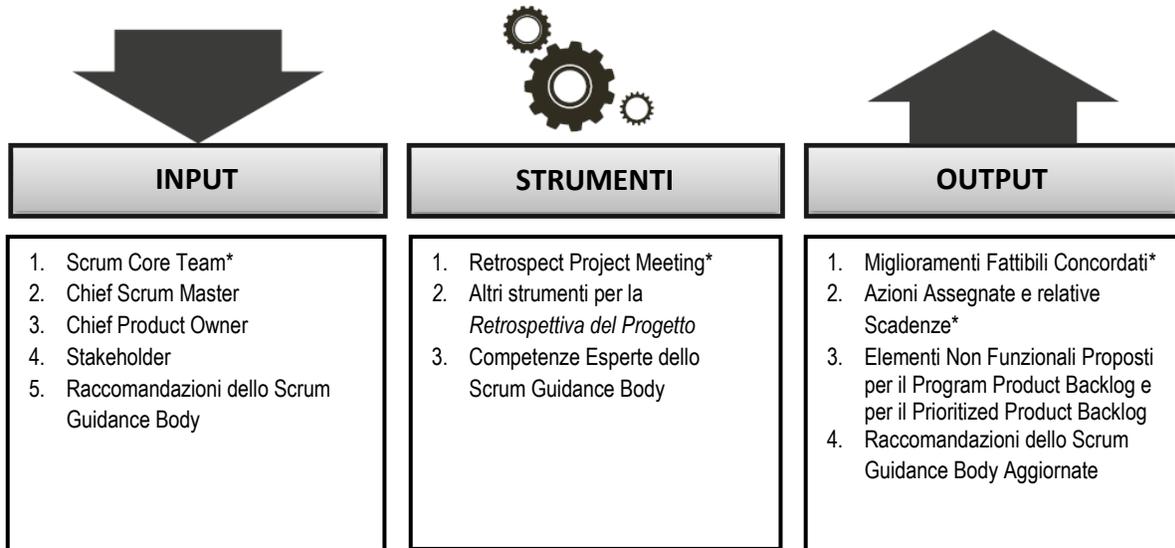


Figura 12-5: Retrospeffiva del Progetto—Input, Strumenti e Output

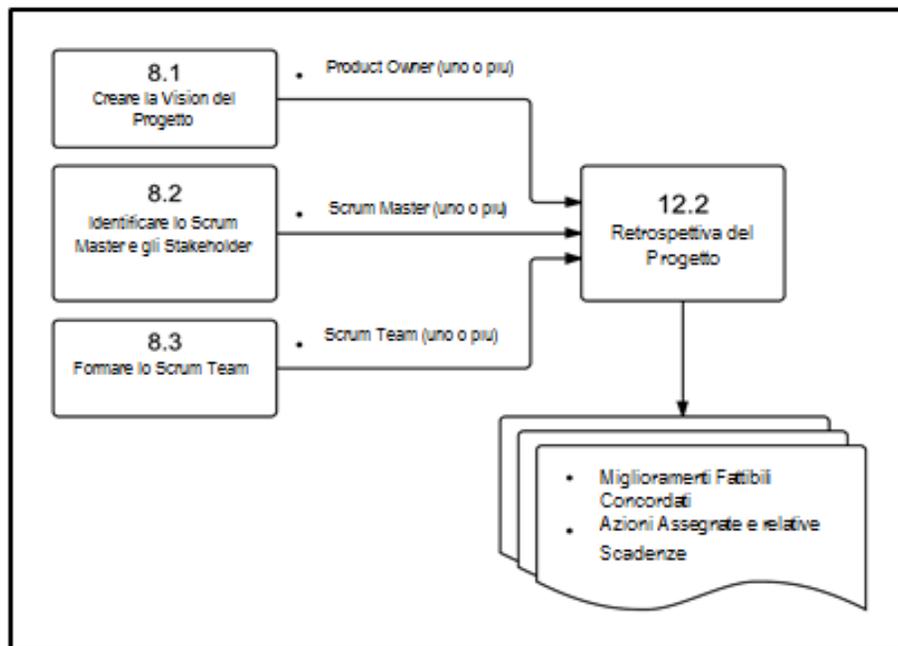


Figura 12-6: Retrospeffiva del Progetto—Diagramma di Flusso dei Dati

Nota: Gli asterischi (\*) indicano un input, uno strumento o un output "obbligatorio" del processo corrispondente.

## 12.2.1 Input

### 12.2.1.1 Scrum Core Team\*

Descritti nel capitolo 8, sezione 8.4.1.1.

### 12.2.1.2 Chief Scrum Master

Descritto nel capitolo 8, sezione 8.2.1.6.

### 12.2.1.3 Chief Product Owner

Descritto nel capitolo 8, sezione 8.1.1.5.

### 12.2.1.4 Stakeholder

Descritti nel capitolo 8, sezione 8.2.3.2.

### 12.2.1.5 Raccomandazioni dello Scrum Guidance Body

Descritte nel capitolo 8, sezione 8.1.1.12.

Con riferimento al processo *Retrospectiva del Progetto*, le Raccomandazioni dello Scrum Guidance Body possono riguardare un archivio di template interni a supporto di progetti futuri, nonché indicazioni sulla conduzione del Retrospect Project Meeting. La guida fornita può riferirsi a procedure amministrative, audit, valutazioni e criteri di transizione del progetto. Spesso le raccomandazioni riguardano anche come l'organizzazione manterrà la knowledge base delle lezioni apprese e le informazioni provenienti da tutti i progetti.

## 12.2.2 Strumenti

### 12.2.2.1 Retrospect Project Meeting\*

Il Retrospect Project Meeting è una riunione volta a stabilire i modi in cui la collaborazione e l'efficacia del team possono essere migliorate nei progetti a venire. Sono inoltre discussi gli aspetti positivi, quelli negativi e le potenziali opportunità di miglioramento. Questa riunione non ha una durata predeterminata e può essere tenuta sia in presenza che in forma virtuale. Ad essa prendono parte il Team di Progetto, il Chief Scrum Master, il Chief Product Owner e gli Stakeholder. Durante la riunione vengono documentate le lezioni apprese e i partecipanti cercano opportunità per migliorare i processi e prendono in esame le inefficienze.

### 12.2.2.2 Altri Strumenti per la Retrospettiva del Progetto

Alcuni degli strumenti utilizzati nel processo *Retrospettiva dello Sprint* possono essere utilizzati anche in questo processo. Alcuni esempi sono:

- L'esercizio Explorer—Shopper—Vacationer—Prisoner (ESVP)
- La Speed Boat
- Le Metriche e Tecniche di Misurazione

### 12.2.2.3 Competenze Esperte dello Scrum Guidance Body

Discusse nel capitolo 8, sezione 8.4.2.7.

Con riferimento al processo *Retrospettiva del Progetto*, la responsabilità principale dello Scrum Guidance Body è fare in modo che le lezioni apprese in ciascun progetto non vadano perse e sia incorporate nell'organizzazione.

Oltre a ciò, un guidance body può fornire competenze esperte utili per il processo *Retrospettiva del Progetto* riguardo a molte aree, fra cui la Qualità, le Risorse Umane e Scrum. Nelle Raccomandazioni dello Scrum Guidance Body vi possono poi essere suggerimenti su come condurre il Retrospect Project Meeting.

## 12.2.3 Output

### 12.2.3.1 Miglioramenti Fattibili Concordati\*

Descritti nel capitolo 11, sezione 11.3.3.1.

### 12.2.3.2 Azioni Assegnate e relative Scadenze\*

Descritte nel capitolo 11, sezione 11.3.3.2.

### 12.2.3.3 Elementi Non Funzionali Proposti per il Program Product Backlog e per il Prioritized Product Backlog

Il Program Product Backlog o il Prioritized Product Backlog iniziali si basano sulle User Story e sulle funzionalità richieste. Spesso, nelle fasi iniziali del progetto i requisiti non funzionali possono non essere pienamente definiti e possono emergere durante lo Sprint Review, il Retrospect Sprint o il Retrospect Project Meeting. Questi elementi devono essere aggiunti al Program Product Backlog (per il programma) e al Prioritized Product Backlog (per il progetto) non appena vengono alla luce. Alcuni esempi di requisiti non funzionali sono i tempi di risposta, i limiti di capacità, nonché le problematiche legate alla sicurezza.

### 12.2.3.4 Raccomandazioni dello Scrum Guidance Body Aggiornate

Descritte nel capitolo 8, sezione 8.1.1.12 e nel capitolo 11, sezione 11.3.3.5.

## 12.3 Diagramma di Flusso dei Dati della Fase

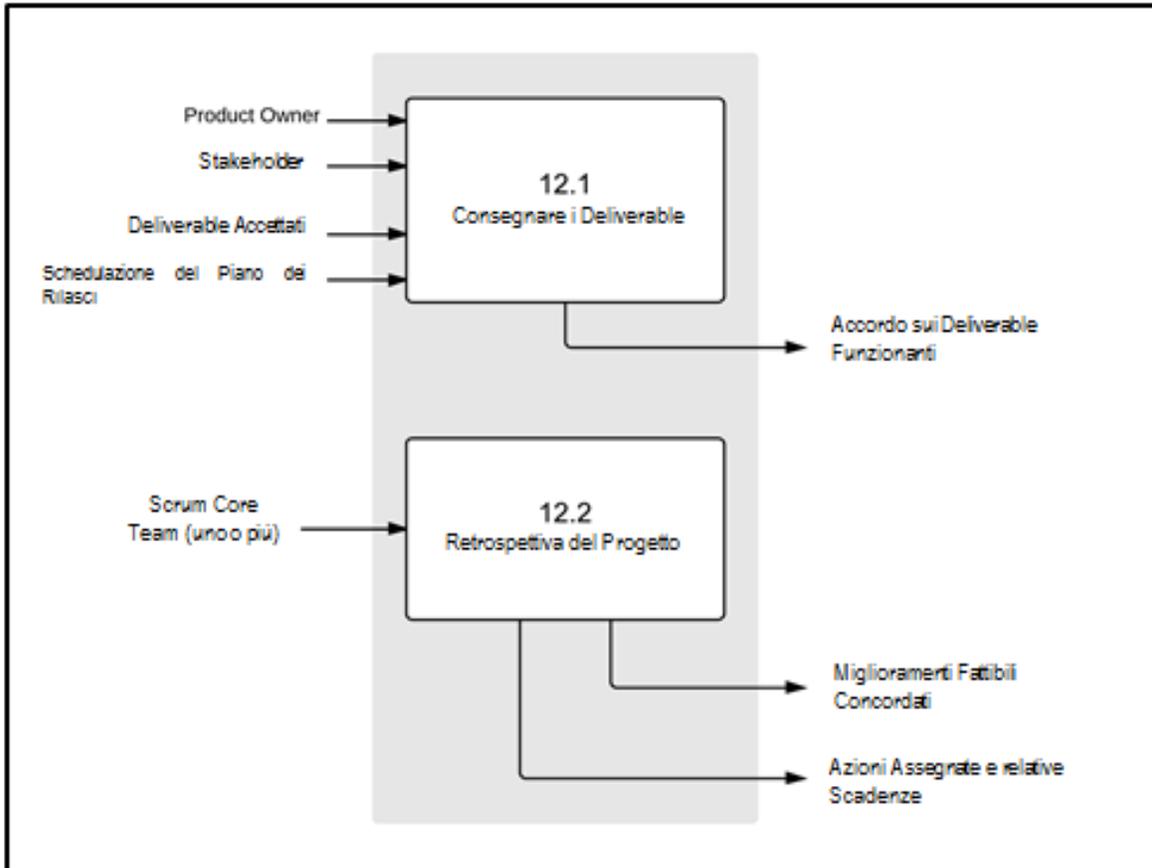


Figura 12-7: Fase di Rilascio—Diagramma di Flusso dei Dati

## APPENDICE A. PANORAMICA DI AGILE

### A.1 Introduzione

Questa appendice ha lo scopo di far conoscere ai lettori i concetti di sviluppo Agile e le varie metodologie Agile.

L'appendice si compone delle seguenti sezioni:

**A.2 Panoramica**—Questa sezione spiega la definizione e i fattori sottesi alla crescita di Agile.

**A.3 Manifesto Agile**—Questa sezione presenta il *Manifesto Agile*, i suoi principi e *La Dichiarazione di Interdipendenza* per fornire il contesto storico di Agile.

**A.4 Metodi Agile**—Questa sezione offre una breve panoramica di specifiche metodologie Agile, ed in particolare:

- Lean Kanban
- Extreme Programming
- Metodi Crystal Methods
- Metodi Dynamic Systems Development
- Feature Driven Development
- Test Driven Development
- Adaptive Software Development
- Agile Unified Process
- Domain Driven Development

## A.2 Panoramica

Il termine “agile” si riferisce generalmente alla capacità di muoversi o rispondere velocemente e facilmente, all’essere pronti. In qualsiasi tipo di disciplina di gestione, la qualità agile dovrebbe quindi essere qualcosa a cui è giusto puntare. Il project management agile comporta specificamente la capacità di adattarsi durante la creazione di un prodotto, servizio o altro risultato.

È importante avere chiaro che, se da un lato i metodi di sviluppo Agile sono altamente adattivi, è però necessario considerare anche la stabilità nei loro processi adattivi.

### A.2.1 La Crescita di Agile

I rapidi cambiamenti della tecnologia, delle domande del mercato e delle aspettative hanno comportato sfide crescenti per lo sviluppo dei prodotti e servizi con modelli tradizionali di project management. Questo ha spianato la strada alla concettualizzazione e alla implementazione di metodi e valori Agile in molte organizzazioni. I modelli di sviluppo Agile hanno risolto i limiti, propri dei modelli tradizionali di project management, che le organizzazioni andavano riscontrando riguardo alla soddisfazione delle sempre crescenti domande e aspettative dell’ambiente. Dal momento che i modelli tradizionali di project management pongono generalmente l’accento sulla pianificazione anticipata e sulla conformità al piano una volta che questo è divenuto baseline, tali modelli non hanno avuto successo nel soddisfare la realtà dei frequenti cambiamenti ambientali.

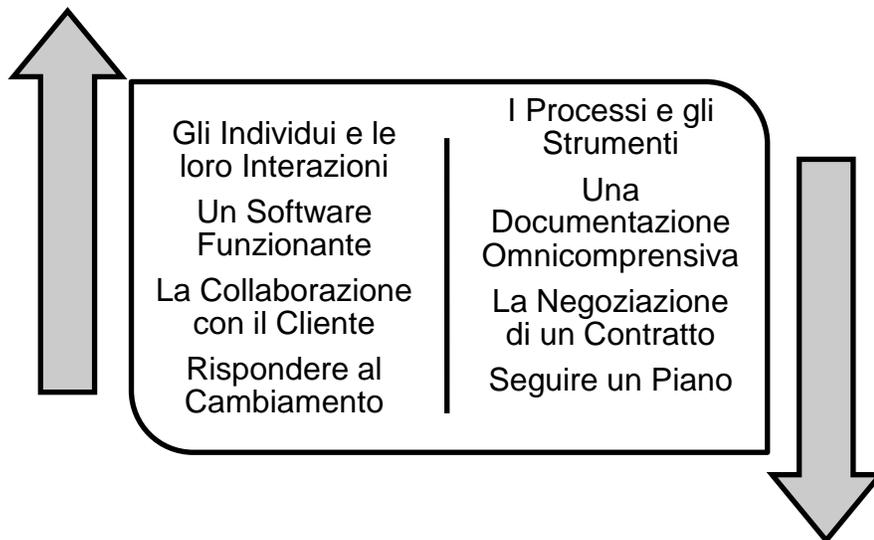
Agile fa affidamento sulla pianificazione di tipo adattivo e sullo sviluppo e la consegna iterativi. Si concentra soprattutto sul valore costituito dalle persone che eseguono il lavoro in modo efficace. Anche se le metodologie adattive e incrementalmente esistono fin dagli anni ’50, solo le metodologie che si conformano al *Manifesto Agile* sono generalmente considerate come veramente “agili”.

## A.3 Il Manifesto Agile

Nel Febbraio 2001, un gruppo di 17 guru dell'informatica, sviluppatori software e manager organizzarono un ritiro per discutere metodi leggeri di sviluppo del software. Queste persone costituirono l'*Alleanza Agile* e le discussioni tenutesi in quegli incontri si tradussero più tardi in un *Manifesto dello Sviluppo Agile del Software*. Il Manifesto è stato scritto da Fowler e Highsmith nel 2001 e poi sottoscritto da tutti i partecipanti per fissare le linee guida basilari di qualsiasi metodologia Agile.

Lo scopo del *Manifesto Agile* è stato esposto nei termini che seguono:

*Stiamo scoprendo i modi migliori per sviluppare il software mettendoli in atto e aiutando gli altri a farlo.  
Attraverso questo lavoro siamo arrivati a valorizzare:*



*Ciò sta a significare che, anche se nelle voci di destra c'è valore, noi diamo più valore alle voci che stanno a sinistra.*

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Autorizzazione alla riproduzione concessa dagli autori sopra indicati mediante comunicazione nel sito <http://agilemanifesto.org/>.

I quattro 'compromessi' messi in evidenza dal Manifesto Agile sono sviluppati come segue:

**1. Le persone e le loro interazioni prima dei processi e degli strumenti**

Sebbene i processi e gli strumenti di lavoro aiutino a completare con successo un progetto, sono le persone che si impegnano, partecipano, implementano un progetto e stabiliscono quali processi e strumenti utilizzare. Gli attori chiave di qualsiasi progetto sono perciò le persone, per questo l'attenzione dovrebbe essere posta su di loro e sulle loro interazioni, piuttosto che su processi e strumenti complicati.

**2. Un software funzionante prima di una documentazione omnicomprensiva**

Anche se la documentazione è necessaria e utile per qualsiasi progetto, molti team si focalizzano sulla raccolta e la registrazione di descrizioni qualitative e quantitative dei deliverable, mentre il valore reale viene rilasciato al cliente prima di tutto sotto forma di un software funzionante. Pertanto, il punto focale di Agile è la consegna di un software funzionante in forma incrementale durante l'intero ciclo di vita del prodotto, piuttosto che la produzione di una documentazione dettagliata.

**3. La collaborazione con il cliente prima della negoziazione di un contratto**

Tradizionalmente, i clienti sono visti come dei giocatori esterni coinvolti principalmente all'inizio e alla fine del ciclo di vita di un prodotto e le cui relazioni sono basate sui contratti e sul loro adempimento. Agile crede invece in un approccio di condivisione del valore, nel quale i clienti sono visti come collaboratori. Il team di sviluppo e il cliente lavorano insieme per elaborare e sviluppare il prodotto.

**4. Rispondere al Cambiamento prima di Seguire un Piano**

Nel mercato attuale, nel quale le esigenze dei clienti, le tecnologie disponibili e i modelli di business sono in continuo cambiamento, è essenziale un approccio allo sviluppo del prodotto di tipo adattivo che consenta l'incorporazione del cambiamento e rapidi cicli di vita di sviluppo del prodotto, piuttosto che porre l'attenzione sulla necessità di seguire piani basati su dati potenzialmente obsoleti.

### A.3.1 I Principi del Manifesto Agile

I 12 principi del Manifesto Agile di Fowler e Highsmith (2001) sono:

1. La nostra massima priorità è soddisfare il cliente attraverso il rilascio anticipato e continuo di software di valore.
2. I cambiamenti dei requisiti sono bene accetti, anche a stadi avanzati dello sviluppo. I processi Agile sfruttano il cambiamento come vantaggio competitivo per il cliente.
3. Consegnare un software funzionante a cadenze ravvicinate, da un paio di settimane a un paio di mesi, con una preferenza per l'intervallo più breve.
4. Le persone del business e gli sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.
5. Costruire progetti intorno a persone motivate, offrire loro l'ambiente ed il supporto di cui hanno bisogno e fidarsi della loro capacità di portare a compimento il lavoro.
6. Il metodo più efficiente ed efficace di veicolare le informazioni verso e all'interno di un team di sviluppo è la conversazione faccia a faccia.
7. Il software funzionante è la principale misura dell'avanzamento del lavoro.
8. I processi Agile promuovono lo sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere un ritmo costante a tempo indeterminato.
9. L'attenzione continua verso l'eccellenza tecnica e la buona progettazione aumenta l'agilità.
10. La semplicità - cioè l'arte di massimizzare la quantità di lavoro non fatto - è essenziale.
11. Le architetture, i requisiti e le progettazioni migliori vengono fuori dai team auto-organizzati.
12. Ad intervalli regolari, il team riflette su come diventare più efficace e poi regola e modifica il suo comportamento di conseguenza.

### A.3.2 Dichiarazione di Interdipendenza

La *Dichiarazione di Interdipendenza* del project management Agile è stata scritta all'inizio del 2005 da un gruppo di 15 capi progetto come integrazione del *Manifesto Agile*. Essa elenca sei valori di gestione necessari per rafforzare una mentalità di sviluppo Agile, in particolar modo quando si gestiscono progetti complessi e caratterizzati dall'incertezza.

La dichiarazione sottolinea che i team di progetto, i clienti e gli altri stakeholder sono fra loro interdipendenti e connessi e devono prendere atto di questo dato per avere successo. Anche gli stessi valori sono fra loro interdipendenti.

*Noi ...*

**Aumentiamo il ritorno sull'investimento** concentrando la nostra attenzione sul flusso continuo di valore.

**Consegniamo risultati affidabili** coinvolgendo i clienti con frequenti interazioni e condivisione della responsabilità.

**Ci aspettiamo l'incertezza** e la gestiamo grazie alle iterazioni, all'anticipazione e all'adattamento.

**Diamo sfogo alla creatività e all'innovazione** riconoscendo che le persone sono la massima fonte di valore e creando un ambiente dove esse possono fare la differenza.

**Diamo sostegno alle prestazioni** propugnando la responsabilità condivisa e di gruppo per i risultati ottenuti e per l'efficacia del team.

**Miglioriamo l'efficacia e l'affidabilità** grazie a strategie, processi e pratiche contestualizzati.

Anderson. D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., e altri, 2005

## A.4 Metodi Agile

Negli anni '90 e nei primi anni 2000 ha avuto origine e si è potenziata una serie di metodologie Agile. Se da un lato queste metodologie si differenziano sotto molteplici aspetti, dall'altro ciò che le accomuna è la loro adesione al *Manifesto Agile*.

Nel prosieguo sono brevemente trattati i seguenti metodi Agile:

1. Lean Kanban
2. Extreme Programming (XP)
3. Metodi Crystal
4. Dynamic Systems Development Methods (DSDM)
5. Feature Driven Development (FDD)
6. Test Driven Development (TDD)
7. Adaptive Software Development (ASD)
8. Agile Unified Process (AUP)
9. Domain-Driven Design (DDD)

### A.4.1 Lean Kanban

Il concetto Lean ottimizza il sistema di un'organizzazione volto alla produzione di risultati di valore in base alle proprie risorse, ai propri bisogni e alle alternative disponibili, riducendo nel contempo gli sprechi. Gli sprechi potrebbero derivare dalla realizzazione del prodotto sbagliato, dal non essere riusciti a imparare, oppure da pratiche che bloccano il processo. Poiché questi fattori sono per loro natura dinamici, un'organizzazione snella valuta il proprio intero sistema e affina continuamente i propri processi. Le fondamenta di Lean consistono nel fatto che la riduzione della lunghezza di ciascun ciclo (cioè un'iterazione) porta ad un aumento della produttività grazie alla riduzione dei ritardi, aiuta a scoprire precocemente gli errori, e conseguentemente riduce la quantità totale di impegno necessario per portare a termine un'attività. I principi del software Lean sono stati applicati con successo allo sviluppo del software.

Kanban significa letteralmente “cartellone” o “manifesto” e sposa l'utilizzo di supporti visivi per coadiuvare e monitorare la produzione. Il concetto è stato introdotto da Taiichi Ohno, considerato il padre dei Sistemi di Produzione Toyota (TPS). L'uso di supporti visivi è realmente efficace ed è diventato una pratica comune. Esempi di questo tipo di supporti sono le task card, le Scrumboard e i Burndown Chart. Questi metodi si sono guadagnati l'attenzione per il fatto di essere praticati in Toyota, azienda leader nella gestione dei processi. Il metodo Lean Kanban integra l'uso dei sistemi di visualizzazione prescritti da Kanban con i principi di Lean, dando vita ad un sistema di gestione del processo visivo, incrementale ed evolutivo.

## A.4.2 Extreme Programming

L'Extreme Programming (XP), che ha avuto origine in Chrysler, si è potenziato negli anni '90. XP rende possibile impedire che il costo delle modifiche al software aumenti drasticamente con il tempo. Gli attributi chiave dell'XP sono lo sviluppo incrementale, la schedulazione flessibile, i codici di test automatizzati, la comunicazione orale, la progettazione tecnica in continua evoluzione, la stretta collaborazione e il vincolo di tutte le persone coinvolte alle motivazioni di lungo e di breve termine.

XP dà valore alla comunicazione, al feedback, alla semplicità e al coraggio. I vari ruoli dell'approccio XP comprendono il cliente, lo sviluppatore, il controllore e l'istruttore. Questo metodo prescrive varie pratiche di codifica, di sviluppo e di business, oltre a eventi e artefatti per realizzare uno sviluppo efficace ed efficiente. L'XP è stato adottato in maniera massiccia proprio per le sue pratiche di ingegneria ben definite.

## A.4.3 Metodi Crystal

Le metodologie di sviluppo del software Crystal sono state presentate da Alistair Cockburn nei primi anni '90. I Crystal si propongono come metodi incentrati sulle persone, leggeri e facili da adattare. Dal momento che le persone sono fondamentali, i processi e gli strumenti di sviluppo non sono fissi, ma sono piuttosto adattati alle esigenze e alle caratteristiche specifiche del progetto. Per stabilire la variante di un progetto si utilizza lo spettro dei colori. Fattori come il benessere, il denaro per spese voluttuarie, il denaro per spese essenziali e la sopravvivenza svolgono un ruolo fondamentale nel determinare il "peso" della metodologia, che viene rappresentato con i vari colori dello spettro. La famiglia Crystal si divide in Crystal Clear, Crystal Yellow, Crystal Orange, Crystal Orange Web, Crystal Red, Crystal Maroon, Crystal Diamond e Crystal Sapphire.

Tutti i metodi Crystal prevedono quattro ruoli—executive sponsor, capo progettista, sviluppatori e utenti esperti. I Metodi Crystal raccomandano varie strategie e tecniche per raggiungere l'agilità. Il ciclo di un progetto Crystal è costituito da sviluppo del project charter, ciclo di consegna e chiusura.

## A.4.4 Dynamic Systems Development Methods (DSDM)

Il framework dei Dynamic Systems Development Methods (DSDM) fu pubblicato inizialmente nel 1995 ed è gestito dal Consorzio DSDM. Il DSDM stabilisce in partenza la qualità e l'impegno in termini di costo e di tempo e adatta i deliverable del progetto per soddisfare i criteri fissati, prioritizzandoli in base alle categorie "Must have" (Essenziale), "Should have" (Preferibile) "Could have" (Potenziale) e "Won't have" (Attualmente Non Realizzabile) – utilizzando quindi la tecnica di prioritizzazione MoSCoW. Il DSDM è un metodo orientato al sistema che prevede sei distinte fasi—Pre-progetto, Fattibilità, Fondamenti, Esplorazione e Ingegneria, Distribuzione, Valutazione dei Benefici.

Una versione successiva del DSDM, nota come DSDM Atern, presentata nel 2007, si concentra sia sulla prioritizzazione dei deliverable che sulla collaborazione costante dell'utente o del cliente. La versione più nuova del metodo si ispira alla Sterna Artica e lo rende un framework di sviluppo software incentrato sullo sviluppatore, volto alla consegna di funzionalità di progetto di valore per l'utente e di qualità controllata, nel rispetto dei tempi e dei costi previsti.

#### **A.4.5 Feature Driven Development (FDD)**

Il Feature Driven Development (FDD) è stato presentato da Jeff De Luca nel 1997 e opera in base al principio che un progetto va completato scomponendolo in piccole funzioni di valore per il cliente che possono essere consegnate in meno di due settimane di tempo. L'FDD ha due principi fondamentali—lo sviluppo del software è un'attività umana e lo sviluppo del software è una funzionalità che ha valore per il cliente.

L'FDD definisce sei ruoli principali —il Project Manager, il Chief Architect, il Development Manager, i Chief Programmer, i Class Owner e i Domain Expert, oltre a una serie di ruoli di supporto. Il processo FDD è iterativo e consiste nello sviluppo di un modello generale, nella creazione di un elenco di funzionalità e infine nella pianificazione, progettazione e realizzazione basata sulle funzionalità.

#### **A.4.6 Test Driven Development (TDD)**

A volte noto come Test-First Development, il Test Driven Development è stato presentato da Kent Beck, uno degli ideatori dell'Extreme Programming (XP). Il Test Driven Development è un metodo di sviluppo del software che implica prima la scrittura del codice di test automatizzato e dopo lo sviluppo della minima quantità di codice necessaria a passare quel test. L'intero progetto è scomposto in piccole funzionalità che hanno un valore per il cliente e che devono essere sviluppate nel più breve ciclo di sviluppo possibile. I test sono scritti sulla base dei requisiti e delle specifiche del cliente. I test progettati nella fase predetta sono utilizzati per progettare e scrivere il codice di produzione.

Il TDD può essere classificato in due livelli: l'Acceptance TDD (ATDD) che richiede un test di accettazione distinto e il Developer TDD (DTDD) che implica la scrittura di un unico test di sviluppo. Il TDD è divenuto popolare a causa dei tanti vantaggi che offre, come i risultati rapidi e affidabili, il feedback costante e la riduzione del tempo di debug.

### A.4.7 Adaptive Software Development (ASD)

L'Adaptive Software Development (ASD) ha avuto origine dal lavoro di rapido sviluppo delle applicazioni ad opera di Jim Highsmith e Sam Bayer. I punti salienti dell'ASD sono il costante adattamento dei processi al lavoro che si sta portando avanti in quel momento, la previsione di soluzioni ai problemi che emergono nei progetti di grandi dimensioni e lo sviluppo iterativo e incrementale accompagnato dalla continua prototipazione.

Essendo un approccio di sviluppo basato sul rischio e tollerante al cambiamento, l'ASD ritiene che un piano non può ammettere incertezze e rischi, perché questo indica un piano imperfetto e non riuscito. L'ASD si basa sulle funzionalità e sugli obiettivi. La prima fase di sviluppo dell'ASD è *Supporte* (al contrario di Pianificare), seguita dalle fasi *Collaborare* e *Apprendere*.

### A.4.8 Agile Unified Process (AUP)

L'Agile Unified Process (AUP) è un'evoluzione del Rational Unified Process di IBM. Sviluppato da Scott Ambler, l'AUP combina le tecniche Agile provate e testate dalle industrie, come il Test Driven Development (TDD), la Modellazione Agile, la gestione agile del cambiamento e il refactoring del database, per consegnare un prodotto funzionante della migliore qualità.

L'AUP modella i suoi processi e tecniche sui valori della Semplicità, dell'Agilità, della Personalizzazione, dell'Auto-organizzazione, dell'Indipendenza degli strumenti e della focalizzazione sulle attività ad alto valore. I principi e i valori dell'AUP vengono messi in atto nelle fasi di Inizio, Elaborazione, Costruzione e Transizione.

### A.4.9 Domain-Driven Design (DDD)

Il Domain-Driven Design è un approccio di sviluppo Agile pensato per la gestione di progettazioni complesse, la cui implementazione è collegata a un modello in evoluzione. È stato ideato da Eric Evans nel 2004 e ruota attorno alla progettazione di un dominio centrale. Il "dominio" è definito come un'area di attività alla quale l'utente applica un programma o una funzionalità. Molte di queste aree sono raggruppate e viene progettato un modello. Il modello consiste in un sistema di astrazioni che possono essere usate per concepire il progetto complessivo e risolvere i problemi relativi ai domini raggruppati. I valori fondamentali del DDD sono la progettazione basata su modelli orientati al dominio, il linguaggio valido per tutti (ubiquitous language) e il contesto delimitato.

Nel DDD, si stabilisce l'ubiquitous language e si modella il dominio. Dopo di che seguono la progettazione, lo sviluppo e il testing. Il modello di dominio viene perfezionato e sottoposto a refactoring fino a quando non è soddisfacente.

## APPENDICE B. AUTORI E REVISORI DELLA GUIDA SBOK™

Questa appendice elenca i nomi di coloro che hanno contribuito allo sviluppo e alla produzione della *Guida SBOK™*.

SCRUMstudy™ ringrazia tutte queste persone per il loro continuo supporto e riconosce pienamente il loro contributo allo sviluppo della *Guida SBOK™*.

### B.1 Autore Principale

Tridibesh Satpathy

### B.2 Coautori ed Esperti della Materia

R-A Alves

Winfried Hackmann

Quincy D. Jordan

Gaynell Malone

J. Drew Nations

Buddy Peacock

Karen Lyncook

Jaimie M. Rush

Elizabeth Lynne Warren

Ruth Kim

Mehul Doshi

Gaurav Garg

Ajey Grandhem

Sayan Guha

Vinay Jagannath

Deepak Ramaswamy

Ahmed Touseefullah Siddiqui

## **B.3 Revisori e Team di Correzione**

Corey T. Bailey  
Sohini Banerjee  
Vince Belanger  
Bobbie Green  
Magaline D. Harvey  
Ravneet Kaur  
Robert Lamb  
Mimi LaRaque  
Melissa Lauro  
Richard Mather  
Lachlan McGurk  
Madhuresh Kumar Mishra  
Neha Mishra  
Yogaraj Mudalgi  
Jose Nunez  
Obi Nwaojigba  
Bryan Lee Perez  
James Pruitt  
Charles J. Quansah  
Frank Quinteros  
Nadra Rafee  
Tommie L. Sherrill  
Barbara Siefken  
Sandra A. Strech  
Frances Mary Jo Tessler  
Chrys Thorsen  
Mike Tomaszewski  
Ron Villmow

## BIBLIOGRAFIA

Anderson, D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., DeCarlo, D., Fitzgerald, D., Highsmith, J., Jepsen, O., Lindstrom, L., Little, T., McDonald, K., Pixton, P., Smith, P., and Wysocki, R. (2005) "Dichiarazione di Interdipendenza" accessibile da Settembre 2013, <http://www.pmdoi.org/>.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001) "Manifesto dello Sviluppo Agile del Software," accessibile da Settembre 2013, <http://agilemanifesto.org/>.

Fellers, G. (1994) *perché le Cose Vanno Storte: La Filosofia di Deming in Dodici Sessioni di Dieci Minuti*. Gretna, LA: Pelican Publishing.

Greenleaf, R. K. (1977) *Servant Leadership: Viaggio nella Natura del Potere Legittimo e e nella Grandezza*. Mahwah, NJ: Paulist Press.

Kano, N., Seraku, N., Takahashi, F., and Tsuji, S. (1984) "Qualità Desiderabile e Qualità Obbligatoria", *Qualità*, 14 (2): 39–48.

Leffingwell, D. and Widrig, D. (2003) *Gestire i Requisiti del Software: Approccio a un Caso d'Uso, II ed.* Boston: Addison-Wesley.

Maslow, A. H. (1943) "Teoria sulla Motivazione Umana", *Psychological Review*, 50 (4): 370–396.

McGregor, D. (1960) *Il Lato Umano dell'Impresa*. New York: McGraw-Hill.

Patton, J. (2005) "Tutto Sta in Come lo Dividi", *Better Software*, January: 16–40.

Spears, L. C. (2010) "Carattere e Servant Leadership: Le Dieci Caratteristiche di Efficaci Leader di Cura", *The Journal of Virtues & Leadership*, 1 (1): 25–30.

Takeuchi, H. and Nonaka, I. (1986) "il Nuovo Gioco di Sviluppo di un Nuovo Prodotto", *Harvard Business Review*, Gennaio-Febbraio: 137–146.

# GLOSSARIO

## **100-Point Method - Metodo dei 100 Punti**

Il Metodo dei 100 Punti è stato sviluppato da Dean Leffingwell e Don Widrig (2003). Esso consiste nel mettere a disposizione del cliente 100 punti che può utilizzare per votare per le caratteristiche che ritiene essere le più importanti.

## **Accepted Deliverables - Deliverable Accettati**

I Deliverable che soddisfano i Criteri di Accettazione delle User Story sono accettati dal Product Owner. Questi sono considerati come Deliverable Accettati che possono essere rilasciati al cliente se lo desidera.

## **Adaptation - Adattamento**

L'adattamento avviene quando lo Scrum Core Team e gli Stakeholder apprendono attraverso la trasparenza e l'ispezione e si adattano apportando miglioramenti al proprio lavoro.

## **Affinity Estimation - Stima per Affinità**

La Stima per Affinità è una tecnica utilizzata per stimare velocemente un grande numero di User Story utilizzando delle categorie. Le categorie possono essere piccola, media o grande, oppure possono essere numerate utilizzando i valori degli story point per indicarne la dimensione relativa. Alcuni benefici chiave di questo approccio sono che il processo è molto trasparente, visibile a tutti, e facile da eseguire.

## **Agreed Actionable Improvements - Miglioramenti Fattibili Concordati**

I Miglioramenti Fattibili Concordati sono il principale output del processo *Retrospectiva dello Sprint*. Rappresentano l'elenco delle iniziative perseguibili messe a punto dal team per affrontare i problemi e migliorare i processi al fine di aumentare le proprie prestazioni negli Sprint a venire.

## **Approve, Estimate and Commit User Stories - Approvare, Stimare e Prendere in Carico le User Story**

In questo processo il Product Owner approva le User Story di uno Sprint. Successivamente, lo Scrum Master e lo Scrum Team stimano l'impegno necessario allo sviluppo della funzionalità descritta in ciascuna User Story. Infine lo Scrum Team si impegna a consegnare i requisiti del cliente sotto forma di User Story Approvate, Stimate e Prese in Carico.

### **Approved Change Requests - Richieste di Modifica Approvate**

Le Richieste di Modifica Approvate sono modifiche delle quali è stata approvata l'inclusione nel Prioritized Product Backlog. A volte, le Richieste di Modifica Approvate possono provenire dai program o portfolio manager ed in tal caso costituiscono input da aggiungere alla lista dei cambiamenti di progetto approvati da implementare in Sprint futuri.

### **Approved, Estimated, and Committed User Stories- User Story Approvate, Stimate e Prese in Carico**

Le User Story che costituiscono un input di questo processo hanno stime di alto livello provenienti dai processi *Creare il Prioritized Product Backlog* e *Creare le User Story*. Queste stime sono utilizzate dal Product Owner per approvare le User Story dello Sprint. Una volta approvate, le User Story sono stimate dal team avvalendosi di varie tecniche di stima. Dopo aver proceduto alla stima, il team prende in carico, nell'ambito delle User Story approvate e stimate, quel gruppo che ritiene di poter completare nel prossimo Sprint. Queste rappresentano le User Story Approvate, Stimate e Prese in Carico, che entreranno a far parte dello Sprint Backlog.

### **Assertive Leader - Leader Assertivo**

I leader assertivi affrontano le questioni e mostrano fiducia per stabilire la propria autorità con il rispetto.

### **Assigned Action Items and Due Dates - Azioni Assegnate e relative Scadenze**

Una volta che i Miglioramenti Fattibili Concordati sono stati elaborati e rifiniti, lo Scrum Team può prendere in considerazione le azioni che servono ad attuare i miglioramenti. Ogni singola azione avrà una scadenza definita per il suo completamento.

### **Autocratic Leader - Leader Autocratico**

I leader autocratici prendono decisioni per conto proprio, consentendo ai membri del team un intervento o una discussione minima, se non nulla, prima dell'emissione di una decisione. Questo stile di leadership dovrebbe essere utilizzato solo in rarissime occasioni.

### **Automated Software Tools - Strumenti Software Automatizzati**

Gli Strumenti Software Automatizzati sono strumenti software utilizzati per la schedulazione, la raccolta delle informazioni e la loro distribuzione.

### **Better Team Coordination - Migliore Coordinamento fra i Team**

Lo Scrum of Scrums Meeting facilita il coordinamento del lavoro fra molteplici Scrum Team. Questo è particolarmente importante quando ci sono attività che comportano delle interdipendenze fra i vari team. Eventuali incompatibilità e discrepanze fra il lavoro e i deliverable di team diversi vengono alla luce rapidamente. Questo forum offre inoltre ai team l'opportunità di presentare i propri risultati positivi e di fornire un feedback agli altri team.

**Brainstorming - Brainstorming**

Sessioni nelle quali gli stakeholder pertinenti e i membri dello Scrum Core Team condividono in maniera aperta le idee attraverso discussioni e sessioni di condivisione della conoscenza, di solito condotte da un facilitatore.

**Business Justification - Giustificazione Commerciale**

La Giustificazione Commerciale illustra le ragioni per intraprendere un progetto. Questa tematica risponde alla domanda "Perché c'è bisogno di questo progetto?" La Giustificazione Commerciale guida tutto il processo decisionale relativo ad un progetto.

**Business Needs - Esigenze di Business**

Le esigenze di business sono i risultati che ci attende vengano realizzati dal progetto, come documentati nella Dichiarazione della Vision del Progetto.

**Business Requirements - Requisiti di Business**

I Requisiti di Business definiscono cosa deve essere consegnato per realizzare le esigenze di business e fornire valore agli stakeholder. L'insieme di tutti i dati ottenuti attraverso i vari strumenti, come le interviste agli utenti o ai clienti, i questionari, le JAD session, la Gap Analysis, l'Analisi SWOT e altre riunioni, aiuta ad avere una visione migliore dei requisiti di business e a creare il Prioritized Product Backlog.

**Change Request(s) - Richieste di Modifica**

Le richieste di apportare cambiamenti sono di solito presentate come Richieste di Modifica. Le Richieste di Modifica rimangono nello stato di non approvate fino al momento in cui sono formalmente approvate.

**Chief Product Owner - Chief Product Owner**

Nel caso di progetti di grandi dimensioni, il Chief Product Owner prepara e mantiene il Prioritized Product Backlog complessivo del progetto. Lui o lei coordina il lavoro dei Product Owner dei vari Scrum Team. I Product Owner, a loro volta, gestiscono le loro rispettive parti del Prioritized Product Backlog.

**Chief Scrum Master – Chief Scrum Master**

Nel caso di progetti di grandi dimensioni, il Chief Scrum Master ha la responsabilità di fare da moderatore dello Scrum of Scrums (SoS) Meeting e di rimuovere gli impedimenti che riguardano più team.

**Coaching/Supportive Leader - Leader Coach/Di Supporto**

I leader coach e di supporto danno istruzioni e poi supportano e monitorano i membri del team attraverso l'ascolto, l'assistenza, l'incoraggiamento, mostrando inoltre un atteggiamento positivo durante i momenti di incertezza.

**Collaboration - Collaborazione**

In Scrum la Collaborazione si riferisce allo Scrum Core Team che lavora insieme e si interfaccia con gli stakeholder per creare e convalidare i deliverable del progetto e realizzare gli obiettivi delineati nella Vision del Progetto. Si ha collaborazione quando un team lavora insieme per trarre vantaggio dai reciproci contributi allo scopo di produrre qualcosa di più grande.

**Collaboration Plan - Piano di Collaborazione**

La Collaborazione è un elemento estremamente importante per Scrum e il Piano di Collaborazione delinea come i diversi organi decisionali, stakeholder e membri del team si impegnano e collaborano gli uni con gli altri.

**Colocation - Co-ubicazione**

La Co-ubicazione è la situazione in cui tutti i membri dello Scrum Core Team si trovano fisicamente nel medesimo posto di lavoro, sfruttando così i vantaggi di un migliore coordinamento, soluzione dei problemi, condivisione della conoscenza e apprendimento.

**Communication Plan - Piano delle Comunicazioni**

Questo piano specifica le registrazioni che devono essere create e mantenute nel corso del progetto. Per veicolare agli stakeholder le informazioni di progetto importanti, viene utilizzata una vasta gamma di metodi. Il Piano delle Comunicazioni definisce questi metodi e le responsabilità relative alle varie attività di comunicazione.

**Company Mission - Mission Aziendale**

La Mission Aziendale fornisce un quadro per la formulazione delle strategie di un'azienda o organizzazione e guida il processo decisionale complessivo.

**Company Vision - Vision Aziendale**

La comprensione della Vision Aziendale aiuta il progetto a mantenersi focalizzato sugli obiettivi dell'organizzazione e sul possibile futuro dell'azienda. Il Product Owner può ricavare indicazioni e direttive dalla Vision Aziendale per creare la Dichiarazione della Vision del Progetto.

**Conduct Daily Standup - Condurre il Daily Standup**

Nel processo *Condurre il Daily Standup* viene tenuta ogni giorno una riunione estremamente mirata e di durata predeterminata. Questa riunione si chiama Daily Standup Meeting e rappresenta per lo Scrum Team il luogo di discussione per un aggiornamento reciproco sullo stato di avanzamento e sugli eventuali impedimenti che ognuno sta riscontrando.

**Conduct Release Planning - Eseguire la Pianificazione dei Rilasci**

In questo processo, lo Scrum Core Team rivede le User Story di alto livello contenute nel Prioritized Product Backlog per sviluppare una Schedulazione del Piano dei Rilasci, che è essenzialmente un programma di distribuzione per fasi che può essere condiviso con gli Stakeholder. In questo processo viene stabilita anche la Lunghezza degli Sprint.

**Conflict Management - Gestione dei Conflitti**

Le tecniche di Gestione dei Conflitti sono utilizzate dai membri del team per gestire i conflitti che possono insorgere durante un progetto Scrum. Le fonti di conflitto sono spesso legate alle schedulazioni, alle priorità, alla catena gerarchica del reporting, a questioni tecniche, a procedure, a problemi caratteriali e per finire ai costi.

**Continuous Improvement - Miglioramento Continuo**

Il Miglioramento Continuo è un approccio di Scrum in base al quale il team impara dall'esperienza e dal coinvolgimento degli stakeholder per mantenere costantemente aggiornato il Prioritized Product Backlog con gli eventuali cambiamenti dei requisiti.

**Continuous Value Justification - Giustificazione Continua del Valore**

La Giustificazione Continua del Valore si riferisce al sottoporre a valutazione costante il valore di business per stabilire se continui a sussistere la giustificazione o fattibilità dell'esecuzione del progetto.

**Convene Scrum of Scrums - Convocare lo Scrum degli Scrum**

In questo processo gli Scrum Master o i rappresentanti dei vari Scrum Team si riuniscono per partecipare agli Scrum of Scrums Meetings a intervalli prestabiliti, o ogni volta che sia necessario per collaborare e monitorare i rispettivi stati di avanzamento, impedimenti e le dipendenze fra i team.

**Core Role(s) - Ruoli Core**

I Ruoli Core sono quei ruoli che sono richiesti obbligatoriamente per dare vita al prodotto del progetto, sono totalmente dedicati al progetto e sono in definitiva responsabili del successo di ciascuno Sprint del progetto e del progetto nel suo insieme.

**Create Deliverables - Creare i Deliverable**

Il processo *Creare i Deliverable* è quello nel quale lo Scrum Team lavora alle attività dello Sprint Backlog per creare i Deliverable dello Sprint.

**Create Prioritized Product Backlog - Creare il Prioritized Product Backlog**

In questo processo le Epic sono rifinite ed elaborate e quindi priorizzate per dare vita al Prioritized Product Backlog del progetto. In questo momento sono fissati anche i Done Criteria.

**Create Project Vision - Creare la Vision del Progetto**

In questo processo si rivede il Business Case del Progetto per formulare la Dichiarazione della Vision del Progetto, che servirà da ispirazione e costituirà il punto di riferimento per l'intero progetto. Il Product Owner viene identificato nel corso di questo processo.

**Create Sprint Backlog - Creare lo Sprint Backlog**

In questo processo lo Scrum Core Team tiene gli Sprint Planning Meeting, nel corso dei quali il gruppo crea uno Sprint Backlog che contiene tutte le attività da completare nello Sprint.

**Create Tasks - Creare le Attività**

In questo processo le User Story Approvate, Stimate e Prese in Carico sono scomposte in specifiche attività e inserite in una Lista delle Attività. A tale scopo viene spesso tenuto un Task Planning Meeting.

**Create User Stories - Creare le User Story**

In questo processo si creano le User Story e i relativi Criteri di Accettazione. Le User Story sono di solito scritte dal Product Owner e sono concepite per assicurare la chiara rappresentazione dei requisiti del cliente e la possibilità di una loro piena comprensione da parte degli stakeholder.

**Cumulative Flow Diagram (CFD) - Diagramma di Flusso Cumulativo (CFD)**

Un Diagramma di Flusso Cumulativo (CFD) è un utile strumento per il reporting e il monitoraggio delle prestazioni del progetto. Fornisce una semplice rappresentazione visiva dello stato di avanzamento del progetto in un determinato momento temporale. Viene di solito utilizzato per fornire una situazione più a grandi linee del progetto nel suo complesso e non gli aggiornamenti quotidiani provenienti dai singoli Sprint.

**Customer - Cliente**

Il Cliente è la persona o l'organizzazione che acquista il prodotto, il servizio o altro risultato del progetto. Per ciascuna organizzazione, a seconda del progetto, ci possono essere sia clienti interni (cioè all'interno della stessa organizzazione), che clienti esterni (cioè, al di fuori dell'organizzazione).

**Customer Value-based Prioritization - Prioritizzazione basata sul Valore per il Cliente**

La Prioritizzazione basata sul Valore per il Cliente dà primaria importanza al cliente e si sforza di implementare per prime le User Story che hanno il valore più elevato. Queste User Story di alto valore vengono identificate e spostate in cima al Prioritized Product Backlog.

**Daily Standup Meeting - Daily Standup Meeting**

Il Daily Standup Meeting è una breve riunione quotidiana, che ha una durata predeterminata di 15 minuti. I membri del team si riuniscono per fare un resoconto sui loro progressi rispondendo alle seguenti tre domande:

4. Cosa ho completato ieri?
5. Cosa completerò oggi?
6. Quali eventuali impedimenti o ostacoli sto riscontrando attualmente?

**Decomposition - Scomposizione**

La scomposizione è uno strumento per mezzo del quale le attività di alto livello sono suddivise in attività di livello più basso, maggiormente dettagliate. Le User Story vengono scomposte in attività dai membri dello Scrum Team. Le User Story contenute nel Prioritized Product Backlog dovrebbero essere scomposte fino ad un livello sufficiente a fornire allo Scrum Team informazioni adeguate per procedere alla creazione dei deliverable partendo dalle Attività elencate nella Lista delle Attività.

**Delegating Leader - Leader Delegante**

I leader deleganti sono coinvolti nella maggior parte delle decisioni; tuttavia, essi delegano alcune responsabilità di pianificazione e decisionali ai membri del team, soprattutto se questi hanno competenze specifiche per gestire le attività loro assegnate. Questo stile di leadership è appropriato nelle situazioni in cui il leader è in sintonia con i dettagli specifici del progetto e quando il tempo a disposizione è limitato.

**Demonstrate and Validate Sprint - Illustrare e Convalidare lo Sprint**

In questo processo lo Scrum Team, nel corso di uno Sprint Review Meeting, illustra i Deliverable dello Sprint al Product Owner e agli stakeholder pertinenti.

**Dependency Determination - Determinazione delle Dipendenze**

Dopo aver selezionato le User Story per un dato Sprint, lo Scrum Team prende in esame le eventuali dipendenze, comprese quelle relative alla disponibilità delle persone, oltre a eventuali dipendenze di carattere tecnico. Documentare le dipendenze in maniera adeguata aiuta gli Scrum Team a stabilire l'ordine relativo nel quale devono essere eseguite le attività per creare i Deliverable dello Sprint. Le dipendenze mettono inoltre in evidenza la relazione e l'interazione fra le varie attività sia all'interno dello Scrum Team che sta lavorando ad un dato Sprint, sia rispetto agli altri Scrum Team del progetto.

### **Design Patterns - Schemi di Progettazione**

Gli Schemi di Progettazione (Design Patterns) forniscono un modo per registrare formalmente la soluzione ad un problema di progettazione tecnica in uno specifico campo di competenza. Questi schemi registrano sia il processo utilizzato che la soluzione concreta da poter riutilizzare successivamente per migliorare il processo decisionale e la produttività.

### **Develop Epic(s) - Sviluppare le Epic**

In questo processo, la Dichiarazione della Vision del Progetto funge da base per lo sviluppo delle Epic. Per *Sviluppare le Epic* si possono tenere delle Riunioni di Gruppi di Utenti.

### **Development in Phases Contract - Contratto di Sviluppo per Fasi**

Questo contratto rende i fondi disponibili ogni mese o ogni trimestre, dopo che è stato completato con successo un rilascio. Questa tipologia offre un incentivo sia al cliente che al fornitore e assicura che il rischio monetario del cliente sia limitato a quel particolare periodo di tempo, in quanto i rilasci non andati a buon fine non vengono finanziati.

### **Directing Leader - Leader Direttivo**

I leader direttivi impartiscono ai membri del team istruzioni riguardo a quali attività siano necessarie, quando dovranno essere eseguite e in che modo dovranno essere eseguite.

### **Discretionary Dependencies - Dipendenze Discrezionali**

Le Dipendenze Discrezionali sono dipendenze che vengono inserite nel flusso di lavoro per scelta. Di solito le dipendenze discrezionali sono stabilite dallo Scrum Team sulla base delle esperienze passate o delle best practice di un particolare campo o settore.

### **Done Criteria – Done Criteria**

I Done Criteria sono un insieme di regole applicabili a tutte le User Story. Una definizione chiara del termine 'Done' (Fatto) rappresenta un fattore cruciale, in quanto rimuove le ambiguità dai requisiti e aiuta il team a rispettare le norme di qualità obbligatorie. Questa definizione chiara viene utilizzata per elaborare appunto i Done Criteria, che sono un output del processo *Creare il Prioritized Product Backlog*. Una User Story si considera fatta (done) quando è stata illustrata al e approvata dal Product Owner, che la giudica sulla base dei Done Criteria e dei Criteri di Accettazione della User Story.

### **Earned Value Analysis - Analisi dell'Earned Value**

L'Analisi dell'Earned Value analizza le prestazioni effettive del progetto rispetto a quelle pianificate in un dato momento temporale. Questo metodo misura gli scostamenti attuali delle prestazioni relative ai tempi e ai costi del progetto e fa una previsione del costo finale sulla base delle prestazioni attuali identificate.

**Effort Estimated Task List - Lista delle Attività con Impegno Stimato**

La Lista delle Attività con Impegno Stimato è un elenco di attività associate alle User Story prese in carico inserite in uno Sprint. L'impegno stimato viene espresso secondo i criteri di stima concordati dal team. La Lista delle Attività con Impegno Stimato viene utilizzata dallo Scrum Team durante gli Sprint Planning Meeting per creare lo Sprint Backlog e lo Sprint Burndown Chart.

**Empirical Process Control - Controllo Empirico di Processo**

Un modello di Controllo Empirico di Processo aiuta a prendere decisioni basate sull'osservazione e sulla sperimentazione piuttosto che su una dettagliata pianificazione anticipata. Tale modello fa affidamento sulle tre idee principali di trasparenza, ispezione e adattamento.

**Epic(s) - Epic**

Le Epic sono scritte nelle fasi iniziali del progetto quando la maggior parte delle User Story sono costituite da funzionalità di alto livello o da descrizioni di prodotti e i requisiti sono definiti solo in linea generale. Si tratta di User Story estese e non rifinite inserite nel Prioritized Product Backlog.

**Estimate Range - Intervallo di Stima**

Le stime dei progetti dovrebbero essere presentate sotto forma di intervalli. I numeri esatti possono dare l'impressione di una maggiore accuratezza ma in realtà non sempre è così. Infatti, le stime sono per definizione non esattamente precise. Gli intervalli di stima dovrebbero basarsi sul livello di fiducia che il team ripone in ciascuna stima. L'intervallo può essere stretto quando il team è fiducioso e ampio quando il team lo è un po' meno.

**Estimate Tasks process - Stimare le Attività**

In questo processo, lo Scrum Core Team, nel corso di un Workshop sulla Stima delle Attività, procede a stimare l'impegno necessario per il compimento di ciascuna delle attività incluse nella Lista delle Attività. L'output di questo processo è una Lista delle Attività con Impegno Stimato.

**Estimation Criteria - Criteri di Stima**

L'obiettivo principale dell'utilizzo dei Criteri di Stima è mantenere ferme le grandezze di stima relative e ridurre al minimo la necessità di ripetere le stime. I Criteri di Stima possono essere espressi in molti modi, fra i quali due esempi tipici sono gli story point e il tempo ideale.

**Expected Monetary Value - Valore Monetario Atteso**

Si tratta di una tecnica di valutazione del rischio nella quale l'impatto finanziario potenziale di un rischio viene stabilito in base al Valore Monetario Atteso (Expected Monetary Value – EMV). L'EMV si calcola moltiplicando l'impatto monetario per la probabilità del rischio, secondo quanto indicato dal cliente.

## **Explorer—Shopper—Vacationer—Prisoner (ESVP)-Explorer—Shopper—Vacationer—Prisoner (ESVP)**

Questo esercizio può essere eseguito all'inizio del Retrospect Sprint Meeting per capire l'atteggiamento mentale dei partecipanti e stabilire il tono della riunione. Viene chiesto ai presenti di indicare in forma anonima quale profilo rappresenta meglio come si sentono rispetto al dover prendere parte alla riunione.

## **External dependencies - Dipendenze Esterne**

Le dipendenze esterne sono quelle relative ad attività, azioni o prodotti che sono al di fuori dell'ambito del lavoro che deve eseguire lo Scrum Team, ma sono necessari per completare un'attività di progetto o per creare un deliverable del progetto. Le dipendenze esterne di norma fuoriescono dal controllo dello Scrum Team.

## **Fist of Five - Fist of Five**

Il Fist of Five è un meccanismo semplice e veloce per raggiungere il consenso in un gruppo e guidare la discussione. Dopo una discussione iniziale su una data proposta o su una decisione in sospeso, si chiede a ciascun membro del team di votare su una scala da 1 a 5 utilizzando le proprie dita.

## **Focus Group Meetings - Riunioni di Focus Group**

I focus group riuniscono varie persone in una sessione guidata per raccogliere le loro opinioni, impressioni o valutazioni di un prodotto, di un servizio o di un risultato desiderato. I membri dei focus group possono liberamente farsi delle domande a vicenda e ottenere chiarimenti su particolari argomenti o concetti. Grazie alle domande, alla critica costruttiva e al feedback i focus group portano ad un prodotto di migliore qualità e perciò contribuiscono alla soddisfazione delle aspettative degli utenti.

## **Form Scrum Team - Formare lo Scrum Team**

Durante questo processo vengono identificati i membri dello Scrum Team. Di solito la responsabilità della scelta dei membri del team fa capo principalmente al Product Owner, il quale però spesso si avvale della collaborazione dello Scrum Master per compiere questa attività.

## **Forming Stage - Fase Forming**

La fase Forming è il primo stadio della costituzione di un team, spesso vissuta come una fase divertente in quanto è tutto nuovo ed il team non ha ancora incontrato alcuna difficoltà riguardo al progetto.

## **Four Questions per Team - Quattro Domande per Team**

Un insieme di domande formulate in ogni Scrum of Scrums (SoS) Meeting. Il rappresentante di ciascuno Scrum Team fornirà aggiornamenti riguardanti il proprio team, di solito sotto forma di risposte a quattro specifiche domande.

1. A cosa ha lavorato il mio team nel tempo trascorso dalla precedente riunione?
2. Che cosa farà il mio team fino alla prossima riunione?
3. Cosa contavano gli altri team che il nostro team finisse e che invece è rimasto incompiuto?

4. Cosa sta pianificando di fare il nostro team che potrebbe avere un impatto sugli altri team?

**Gap Analysis - Gap Analysis**

La Gap Analysis è una tecnica utilizzata per confrontare lo stato effettivo attuale con quello desiderato e per stabilire come colmare il divario fra i due.

**Groom Prioritized Product Backlog - Mettere a Punto il Prioritized Product Backlog**

*Mettere a Punto il Prioritized Product Backlog* è il processo attraverso il quale il Prioritized Product Backlog è continuamente aggiornato e mantenuto.

**Identify Scrum Master and Stakeholder(s) process - Identificare lo Scrum Master e gli Stakeholder**

In questo processo vengono identificati lo Scrum Master e gli stakeholder facendo uso di Criteri di Selezione specifici.

**Impediment - Impedimento**

Un impedimento è un qualsiasi ostacolo o complicazione che riduce la produttività dello Scrum Team.

**Implement Phase - Fase di Implementazione**

La Fase di Implementazione include i processi relativi all'esecuzione delle attività e delle azioni che servono a creare il prodotto di un progetto.

**Incentive and Penalty Contract - Contratto a Incentivo e Penale**

Questo contratto si basa sull'accordo che il fornitore sarà ricompensato con un incentivo finanziario se i prodotti del progetto verranno consegnati nei tempi previsti, ma sarà soggetto al pagamento di penali se la consegna avverrà in ritardo.

**Incremental Delivery Contract - Contratto a Consegna Incrementale**

Questo contratto prevede punti di controllo a intervalli regolari. Tale forma aiuta il cliente o gli stakeholder a prendere decisioni sullo sviluppo del prodotto periodicamente nel corso del progetto, in corrispondenza di ciascun punto di controllo. Il cliente può accettare lo sviluppo del prodotto, oppure decidere di fermare lo sviluppo del prodotto, o infine richiedere modifiche al prodotto.

**Index Cards - Index Card**

Le Index Card, spesso definite Story Card, sono utilizzate per monitorare le User Story nel corso del progetto. Questo aumenta la visibilità e la trasparenza e facilita la scoperta precoce di eventuali problemi che possono insorgere.

**Initiate phase - Fase di Inizio**

Questa fase è costituita dai processi relativi all'inizio di un progetto: *Creare la Vision del Progetto*, *Identificare lo Scrum Master e gli Stakeholder*, *Formare lo Scrum Team*, *Sviluppare le Epic*, *Creare il Prioritized Product Backlog* e *Eseguire la Pianificazione dei Rilasci*.

**Inspection - Ispezione**

L'Ispezione si riferisce al monitoraggio necessario per seguire il controllo empirico di processo, al fine di assicurare che i deliverable del progetto rispettino i requisiti.

**Internal Dependencies - Dipendenze Interne**

Le Dipendenze Interne sono quelle dipendenze fra attività, prodotti o azioni che si trovano sotto il controllo dello Scrum Team e all'interno dell'ambito del lavoro che deve essere eseguito dallo Scrum Team.

**Internal Rate of Return (IRR) - Tasso Interno di Rendimento (IRR)**

Il Tasso Interno di Rendimento (IRR) è un tasso di sconto su un investimento che rende il valore attuale dei flussi monetari in entrata pari al valore attuale dei flussi monetari in uscita per valutare il tasso di rendimento di un progetto. Quando si confrontano fra loro più progetti, di norma quello che ha un IRR più alto è migliore.

**Issues - Questioni**

Le questioni sono generalmente eventi certi e ben definiti che stanno accadendo attualmente nel progetto, per cui non c'è alcun bisogno di eseguire una valutazione delle probabilità come per un rischio.

**Iterative Delivery - Consegna Iterativa**

La consegna Iterativa è la consegna progressiva di valore al cliente.

**JAD Sessions - JAD Session**

Una Joint Application Design (JAD) Session è una tecnica di raccolta dei requisiti. Si tratta di un workshop facilitato molto strutturato che accelera il processo *Creare la Vision del Progetto*, in quanto consente agli Stakeholder e agli altri organi decisionali di pervenire ad un consenso in merito all'ambito, agli obiettivi e ad altre specifiche del progetto.

**Joint Venture Contract - Contratto in Joint Venture**

Questo contratto viene generalmente utilizzato quando due o più parti si associano per eseguire il lavoro di un progetto. Tutte le parti coinvolte nel progetto realizzeranno un Ritorno sull'Investimento poiché i ricavi o benefici generati saranno condivisi fra i partecipanti.

**Kano Analysis - Analisi di Kano**

L'Analisi di Kano è stata sviluppata da Noriaki Kano nel 1984 e consiste nel classificare le funzionalità o i requisiti in quattro categorie basate sulle preferenze del cliente:

1. Exciters/Delighters (cioè che provocano eccitazione e meraviglia)

2. Satisfiers (cioè che soddisfano)
3. Dissatisfiers (cioè che non soddisfano)
4. Indifferent (cioè che sono indifferenti)

**Laissez Faire Leader - Leader Laissez Faire**

Uno stile di leadership nel quale il team viene lasciato per lo più senza sorveglianza, per cui il leader non interferisce nelle attività lavorative quotidiane del team stesso. Questo stile porta spesso ad una situazione di anarchia.

**Length of Sprint - Lunghezza dello Sprint**

Sulla base dei vari input, compresi i requisiti di business e la Schedulazione del Piano dei Rilasci, il Product Owner e lo Scrum Team stabiliscono la lunghezza degli Sprint del progetto. Una volta decisa, la lunghezza dello Sprint rimane di solito invariata per tutto il progetto.

La Lunghezza dello Sprint rappresenta la durata degli Sprint stabilita per un progetto.

**Mandatory Dependencies - Dipendenze Obbligatorie**

Queste dipendenze sono inerenti alla natura del lavoro, come un limite fisico, oppure possono diventare tali in virtù di obbligazioni contrattuali o di requisiti legali.

**Market Study - Studio di Mercato**

Lo Studio di Mercato si riferisce alla ricerca organizzata, alla raccolta, alla collazione e all'analisi dei dati relativi alle preferenze dei clienti riguardo ai prodotti. Spesso riguarda dati dettagliati sulle tendenze del mercato, sulla segmentazione del mercato e sui processi di marketing.

**Minimum Acceptance Criteria - Criteri di Accettazione Minimi**

I Criteri di Accettazione Minimi sono dichiarati dall'unità di business. Gli stessi diventano poi parte dei Criteri di Accettazione di tutte le User Story di quella unità di business. Qualsiasi funzionalità definita dall'unità di business deve soddisfare questi Criteri di Accettazione Minimi per essere accettata dal rispettivo Product Owner.

**Mitigated Risks - Rischi Mitigati**

I Rischi Mitigati si riferiscono ai rischi che sono stati trattati o mitigati con successo dallo Scrum Team durante il progetto.

**Monopoly Money - Monopoly Money**

Questa tecnica consiste nel consegnare ai clienti una quantità di "soldi del Monopoli" o "soldi falsi" pari al budget del progetto, chiedendo loro di distribuirli fra le User Story prese in esame. In questo modo, il cliente fa una scala di priorità in base a ciò che è disposto a pagare per ciascuna User Story.

**MoSCoW Prioritization - Prioritizzazione MoSCoW**

Lo schema di prioritizzazione MoSCoW prende il nome dalle prime lettere delle frasi “Must have” (cioè “Deve avere”), “Should have” (cioè “Dovrebbe avere”), “Could have” (cioè “Potrebbe avere”), e “Won’t have” (cioè “Non avrà”). Questo metodo di prioritizzazione è generalmente più efficace degli schemi semplici. Le categorie sono in ordine decrescente di priorità, dove le funzionalità del tipo “Must have” sono quelle senza le quali il prodotto non avrà valore e quelle del tipo “Won’t have” sono quelle che, anche se sarebbe bello avere, non è necessario includere.

**Net Present Value (NPV) - Valore Netto Attuale (NPV)**

Il Valore Netto Attuale (NPV) è un metodo usato per determinare il valore netto presente di un beneficio finanziario futuro, dato come assunto un determinato tasso di inflazione o di interesse.

**Non-core role - Ruolo Non-core**

I ruoli non-core (non essenziali) sono quelli che non sono obbligatoriamente richiesti per il progetto Scrum. Possono includere i membri del team che hanno un interesse nel progetto, che non hanno un ruolo formale all’interno del team di progetto, che possono interfacciarsi con il team, ma non essere necessariamente responsabili per il successo del progetto.

**Norming stage - Fase Norming**

La terza fase di formazione del team nella quale il team inizia a maturare, chiarisce le proprie differenze interne e cerca soluzioni per lavorare insieme. È considerato un periodo di assestamento.

**Number of Stories - Numero di Story**

Il numero di Story si riferisce al numero di User Story che sono consegnate come parte di un singolo Sprint. Può essere espresso in termini di conteggio semplice o conteggio ponderato.

**Opportunities - Opportunità**

I Rischi che si presumono avere un impatto positivo sul progetto si definiscono opportunità.

**Opportunity Cost - Costo Opportunità**

Il costo opportunità si riferisce alla migliore opzione di business o al miglior progetto che è stato scartato a favore di quello prescelto.

**Organizational Deployment Methods - Metodi di Distribuzione dell’Organizzazione**

I meccanismi di distribuzione delle varie organizzazioni tendono a differenziarsi in base al settore industriale, agli utenti target e al posizionamento. A seconda del prodotto da consegnare, la distribuzione può avvenire da remoto o può invece implicare la spedizione fisica o il trasferimento di un articolo.

### **Organizational Resource Matrix - Matrice Organizzativa delle Risorse**

La Matrice Organizzativa delle Risorse è una rappresentazione gerarchica di una combinazione di una struttura organizzativa funzionale e di una struttura organizzativa progettuale. Le organizzazioni a matrice mettono insieme i membri del team di un progetto prendendoli da vari dipartimenti funzionali, come tecnologia dell'informazione, finanze, marketing, vendite, produzione, e altri dipartimenti, creando così team interfunzionali.

### **Paired Comparison - Comparazione a Coppie**

La Comparazione a Coppie è una tecnica nella quale si prepara una lista di tutte le User Story presenti nel Prioritized Product Backlog. Successivamente ciascuna User Story viene presa singolarmente e comparata con le altre User Story della lista, una alla volta. Ogni volta che sono messe a confronto due User Story, viene presa una decisione riguardo a quale delle due sia più importante. Grazie a questo procedimento, si può produrre una lista prioritizzata di User Story.

### **Pareto Analysis - Analisi di Pareto**

Questa tecnica di valutazione del rischio consiste nel classificare i rischi in base alla magnitudine. Ciò aiuta lo Scrum Team a trattare i rischi secondo l'ordine di potenziale impatto che gli stessi hanno sul progetto.

### **PDCA/PDSA cycle - Ciclo PDCA/PDSA**

Il Ciclo Plan-Do-Check-Act—noto anche come Ciclo di Deming o di Shewhart—fu sviluppato dal Dott. W. Edwards Deming, considerato il padre del moderno controllo di qualità, e dal Dott. Walter A. Shewhart. Deming modificò successivamente il ciclo Plan-Do-Check-Act in Plan-Do-Study-Act (PDSA) perché ritenne che il termine 'Study' – Studia – mettesse in luce il concetto di analisi piuttosto che semplicemente quello di ispezione, come suggerito dal termine 'Check' – Controlla. Sia Scrum che il Ciclo PDCA di Deming/Shewhart sono metodi iterativi che si focalizzano sul miglioramento continuo.

### **Performing stage - Fase Performing**

La fase finale di formazione del team nella quale il team raggiunge il suo massimo di coesione ed opera al suo più alto livello in termini di prestazioni. I membri si sono trasformati in un team efficiente di professionisti alla pari che sono uniformemente produttivi.

### **Personas - Persona**

Le *Persona* sono personaggi immaginari molto dettagliati, che rappresentano la maggioranza degli utenti e degli altri stakeholder che non necessariamente utilizzano direttamente il prodotto finale. Le *Persona* sono create per identificare i bisogni della base di utenti target.

**Piloting Plan - Piano Pilota**

Un Piano Pilota può essere utilizzato per tracciare nel dettaglio una distribuzione pilota. Nel Piano Pilota sono specificati e condivisi con gli stakeholder l'ambito e gli obiettivi della distribuzione, la base di utenti target della distribuzione, un programma di distribuzione, i piani di transizione, la preparazione dell'utente necessaria, i criteri di valutazione della distribuzione, ed altri elementi chiave legati alla distribuzione.

**Plan and Estimate phase - Fase di Pianificazione e Stima**

La Fase di Pianificazione e Stima è costituita dai processi legati alla pianificazione e stima delle attività, che sono *Creare le User Story*; *Approvare, Stimare e Prendere in Carico le User Story*; *Creare le Attività*; *Stimare le Attività*; e infine *Creare lo Sprint Backlog*.

**Planning for Value - Pianificazione Incentrata sul Valore**

La Pianificazione Incentrata sul Valore si riferisce alla giustificazione e alla conferma del valore del progetto. L'onere di decidere come si crea il valore ricade sugli stakeholder (sponsor, clienti e/o utenti), mentre lo Scrum Team si concentra su cosa deve essere sviluppato.

**Planning Poker - Planning Poker**

Il Planning Poker, chiamato anche Estimation Poker, è una tecnica di stima che bilancia il pensiero di gruppo e il pensiero individuale per stimare le dimensioni relative delle User Story o l'impegno necessario per crearle.

**Points for Cost Estimating - Punti per la Stima dei Costi**

La stima dei costi può essere fatta utilizzando unità relative (ad esempio, le stime dell'impegno) invece che unità assolute (cioè gli effettivi costi sostenuti). Per stimare il costo di implementazione di una User Story, lo Scrum Team può utilizzare gli story point. In questo caso, il costo stimato di ciascuna attività sarà espresso in story point, anziché in unità monetarie.

**Portfolio - Portfolio**

Un portfolio è un gruppo di programmi collegati, con l'obiettivo di consegnare risultati di business come definiti nella Dichiarazione della Vision del Portfolio. Il Prioritized Portfolio Backlog incorpora i Prioritized Program Backlog di tutti i programmi del portfolio.

**Portfolio Product Owner - Portfolio Product Owner**

Il Portfolio Product Owner definisce gli obiettivi strategici e le priorità del portfolio.

**Portfolio Scrum Master - Portfolio Scrum Master**

Il Portfolio Scrum Master risolve i problemi, rimuove gli impedimenti, facilita e conduce le riunioni di portfolio.

**Prioritization - Prioritizzazione**

La prioritizzazione può essere definita come la determinazione dell'ordine delle cose e la separazione fra ciò che deve essere fatto subito e ciò che può essere fatto in un secondo momento.

**Prioritized Product Backlog - Prioritized Product Backlog**

Il Prioritized Product Backlog è un singolo documento sui requisiti che definisce l'ambito del progetto attraverso una lista prioritizzata delle caratteristiche del prodotto o servizio che il progetto deve consegnare.

**Probability Impact Grid - Griglia di Probabilità e Impatto**

Una griglia nella quale i Rischi sono valutati in base alla probabilità di accadimento e al potenziale impatto sugli obiettivi del progetto. In genere viene assegnato un punteggio numerico sia alla probabilità che all'impatto in maniera indipendente. I due valori sono quindi moltiplicati fra loro per ricavarne un punteggio di gravità del rischio, che può essere utilizzato per prioritizzare i rischi.

**Probability Trees - Alberi delle Probabilità**

Gli eventi potenziali sono rappresentati in un diagramma che ha un ramo per ogni possibile risultato degli eventi stessi. La probabilità di ogni possibile risultato viene indicata sul ramo appropriato e questi valori possono essere usati per calcolare il complessivo impatto della materializzazione dei rischi in un progetto.

**Product - Prodotto**

Il termine "prodotto" nella *Guida SBOK™* può fare riferimento ad un prodotto, servizio o altro deliverable che fornisce valore al cliente.

**Prioritized Product Backlog Review Meeting - Prioritized Product Backlog Review Meeting**

Un Product Backlog Review Meeting (chiamato anche Sessione di Messa a Punto del Prioritized Product Backlog) è una riunione formale che si tiene durante il processo *Mettere a Punto il Prioritized Product Backlog* e che aiuta lo Scrum Team a rivedere il Prioritized Product Backlog e a ottenere il consenso sullo stesso.

### **Product Owner - Product Owner**

Il Product Owner è il soggetto responsabile di trarre il massimo valore di business dal progetto. Lui o lei ha la responsabilità di esprimere al meglio i requisiti del cliente e di mantenere salda la giustificazione commerciale del progetto.

### **Program - Programma**

Un programma è un gruppo di progetti collegati, con l'obiettivo di consegnare i risultati di business definiti nella Dichiarazione della Vision del programma. Il Prioritized Program Backlog incorpora i Prioritized Product Backlog di tutti i progetti del programma stesso.

### **Program and Portfolio Risks - Rischi di Programma e di Portfolio**

Rischi relativi a un portfolio o programma che avranno un impatto anche sui progetti che fanno parte rispettivamente del portfolio o del programma.

### **Program Product Owner - Program Product Owner**

Il Program Product Owner definisce gli obiettivi strategici e le priorità del programma.

### **Program Scrum Master - Program Scrum Master**

Il Program Scrum Master risolve i problemi, rimuove gli impedimenti, facilita e conduce le riunioni del programma.

### **Project - Progetto**

Un progetto è un'impresa collaborativa volta sia a creare nuovi prodotti o servizi sia a consegnare risultati, così come definiti nella Dichiarazione della Vision del Progetto. I progetti sono abitualmente condizionati da vincoli di tempo, costi, ambito, qualità, persone e capacità organizzative.

### **Project Benefits - Benefici del Progetto**

I benefici del progetto includono tutti miglioramenti misurabili di un prodotto, servizio o risultato che possono essere ottenuti attraverso il positivo completamento di un progetto.

### **Project Budget - Budget di Progetto**

Il Budget di Progetto è un documento finanziario che include i costi delle persone, dei materiali e altre spese relative a un progetto. Il budget di progetto è approvato generalmente dagli sponsor per assicurare la disponibilità di fondi sufficienti.

**Project Charter - Project Charter**

Un project charter è una dichiarazione ufficiale degli obiettivi e dei risultati desiderati del progetto. In molte organizzazioni, il Project Charter è il documento che autorizza ufficialmente e formalmente il progetto, conferendo per iscritto al team l'autorità per iniziare il lavoro di progetto.

**Project Costs - Costi del Progetto**

I costi del progetto sono rappresentati dall'investimento e dagli altri costi di sviluppo di un progetto.

**Project Reasoning - Motivi del Progetto**

I motivi del progetto sono rappresentati da tutti i fattori che rendono necessario il progetto, siano essi positivi o negativi, frutto di una scelta o meno (ad esempio, scarsa capacità di soddisfare la domanda esistente e quella prevista, calo della soddisfazione del cliente, bassi profitti, requisiti legali, ecc.).

**Project Timescales - Tempistiche del Progetto**

Le tempistiche riflettono la lunghezza o durata di un progetto. Le tempistiche collegate al business case comprendono anche il tempo durante il quale saranno realizzati i benefici del progetto.

**Project Vision Meeting - Riunione sulla Vision del Progetto**

Ad una Riunione sulla Vision del Progetto partecipano gli Stakeholder del Programma, il Program Product Owner, il Program Scrum Master e il Chief Product Owner. Questa riunione aiuta ad identificare il contesto di business, i requisiti di business e le aspettative degli stakeholder, in modo da poter sviluppare una Dichiarazione della Vision del Progetto realmente efficace.

**Project Vision Statement - Dichiarazione della Vision del Progetto**

L'output fondamentale del processo *Creare la Vision del Progetto* è una Dichiarazione della Vision del Progetto ben strutturata. Una buona Vision del Progetto spiega l'esigenza di business che il progetto è destinato a soddisfare piuttosto che come la soddisferà.

**Proposed Non-Functional Items for Product Backlog - Elementi Non Funzionali Proposti per il Product Backlog**

I requisiti non funzionali possono non essere ancora pienamente definiti nelle fasi iniziali del progetto e possono emergere durante lo Sprint Review o il Retrospect Sprint Meeting. Questi elementi devono essere aggiunti al Prioritized Product Backlog non appena vengono alla luce.

**Quality - Qualità**

La qualità è definita come l'abilità del prodotto completato o dei deliverable di soddisfare i Criteri di Accettazione e realizzare il valore di business atteso dal cliente.

**Quality Assurance - Garanzia di Qualità**

La garanzia di qualità si riferisce alla valutazione dei processi e degli standard che governano la gestione della qualità in un progetto per confermarne la persistente adeguatezza. Le attività di garanzia della qualità sono eseguite come parte del lavoro di progetto.

**Quality Control - Controllo di Qualità**

Il controllo di qualità si riferisce alla esecuzione delle attività pianificate relative alla qualità da parte dello Scrum Team nel processo di creazione di deliverable che sono potenzialmente consegnabili. Tale controllo include anche l'apprendimento da ciascun insieme di attività completate, al fine di realizzare il miglioramento continuo.

**Quality Management - Gestione della Qualità**

In Scrum la gestione della qualità consente ai clienti di acquisire in anticipo la consapevolezza dei problemi del progetto e li aiuta a rendersi conto se un progetto stia o meno lavorando nel loro interesse. La gestione della qualità in Scrum è facilitata da tre attività fra loro correlate:

1. Pianificazione della qualità
2. Controllo della qualità
3. Garanzia della qualità

**Quality Planning - Pianificazione della Qualità**

La Pianificazione della Qualità si riferisce all'identificazione e alla definizione del prodotto che si richiede ad uno Sprint e al progetto, insieme ai Criteri di Accettazione, ai metodi di sviluppo da adottare e alle responsabilità chiave dei membri dello Scrum Team riguardo alla qualità.

**Refactoring - Refactoring**

Il refactoring è uno strumento proprio dei progetti software. Lo scopo di questa tecnica è quello di migliorare la manutenibilità del codice esistente e di renderlo più semplice, più conciso e più flessibile. Fare il refactoring significa migliorare la progettazione del codice esistente senza cambiarne il comportamento. Ciò comporta:

- L'eliminazione del codice ripetitivo o ridondante
- La scomposizione dei metodi e delle funzioni in routine più piccole

- La definizione chiara delle variabili e dei nomi dei metodi
- La semplificazione del disegno progettuale del codice
- Rendere il codice più facile da capire e da modificare

### **Rejected Deliverables - Deliverable Rifiutati**

I Deliverable Rifiutati sono quelli che non soddisfano i Criteri di Accettazione. Viene tenuto un elenco dei Deliverable Rifiutati che viene aggiornato dopo ogni Sprint Review Meeting con tutti i deliverable che non sono stati accettati.

### **Relative Prioritization Ranking - Relative Prioritization Ranking**

Il Relative Prioritization Ranking (cioè Classifica di Priorità Relativa) è una semplice elencazione di User Story in ordine di priorità. Rappresenta un metodo efficace per stabilire le User Story desiderate per ciascuna iterazione o rilascio del prodotto o servizio.

### **Relative Sizing/Story Point - Relative Sizing/Story Points**

Oltre ad essere usati per la stima dei costi, gli story point possono essere usati anche per stimare la dimensione totale di una User Story o funzionalità. Questo approccio prevede l'assegnazione di un valore in story point sulla base di una valutazione complessiva della dimensione di una User Story, considerando il rischio, la quantità di impegno richiesta e il livello di complessità.

### **Release Content - Contenuto del Rilascio**

Consiste delle informazioni essenziali riguardanti i deliverable che possono aiutare il Team di Supporto ai Clienti.

### **Release Notes - Note di Rilascio**

Le Note di Rilascio attengono ai criteri di spedizione esterni o rivolti al mercato relativi al prodotto da consegnare.

### **Release Planning Schedule - Schedulazione del Piano dei Rilasci**

Una Schedulazione del Piano dei Rilasci è uno degli output fondamentali del processo *Eeguire la Pianificazione dei Rilasci*. Questo tipo di Schedulazione stabilisce quali deliverable devono essere rilasciati ai clienti, insieme agli intervalli pianificati e alle date dei rilasci. Non è detto che vi sia un rilascio programmato alla fine di ogni Sprint.

**Release Planning Sessions Release Planning Sessions - Sessioni di Pianificazione dei Rilasci**

L'obiettivo principale delle Sessioni di Pianificazione dei Rilasci è quello di creare una Schedulazione del Piano dei Rilasci e di permettere allo Scrum Team di avere una visione d'insieme dei rilasci e del programma di consegna del prodotto che il team sta sviluppando, in modo da potersi allineare alle aspettative del Product Owner e degli Stakeholder pertinenti.

**Release Prioritization Methods - Metodi di Prioritizzazione dei Rilasci**

Per sviluppare un Piano dei Rilasci vengono utilizzati dei Metodi di Prioritizzazione dei Rilasci. Questi metodi sono propri di ciascun settore industriale e organizzazione e sono di solito stabiliti dal gruppo dirigente di un'organizzazione.

**Resolved Issues - Questioni Risolte**

Negli Scrum of Scrums (SoS) Meeting i membri degli Scrum Team hanno la possibilità di discutere in modo trasparente i problemi che impattano sul loro progetto. Questa tempestiva discussione e risoluzione dei problemi all'interno del SoS Meeting migliora notevolmente il coordinamento fra i diversi Scrum Team e riduce inoltre la necessità di riprogettazione e rilavorazione.

**Retrospect Project - Retrospettiva del Progetto**

In questo processo, che completa il progetto, gli stakeholder dell'organizzazione e i membri dello Scrum Core Team si riuniscono per esaminare retrospettivamente il progetto e identificare, documentare ed assimilare le lezioni apprese. Queste lezioni portano spesso alla documentazione di Miglioramenti Fattibili Concordati, da implementare in progetti futuri.

**Retrospect Project Meeting - Retrospect Project Meeting**

Il Retrospect Project Meeting è una riunione volta a stabilire i modi in cui la collaborazione e l'efficacia del team possono essere migliorate nei progetti a venire. Sono inoltre discussi gli aspetti positivi, quelli negativi e le potenziali opportunità di miglioramento. Questa riunione non ha una durata predeterminata e può essere tenuta sia in presenza che in forma virtuale.

**Retrospect Sprint - Retrospettiva dello Sprint**

In questo processo lo Scrum Master e lo Scrum Team si riuniscono per discutere le lezioni apprese nel corso dello Sprint. Queste informazioni sono registrate come lezioni apprese che possono essere applicate ai futuri Sprint.

**Retrospect Sprint Log(s) - Registri delle Retrospective degli Sprint**

Il Registro della Retrospectiva dello Sprint è una registrazione delle opinioni, delle discussioni e delle iniziative perseguibili emerse in un Retrospect Sprint Meeting. Lo Scrum Master può facilitare la creazione di questo registro con gli input provenienti dai membri dello Scrum Core Team.

**Retrospect Sprint Meeting - Retrospect Sprint Meeting**

Il Retrospect Sprint Meeting ha una durata predeterminata di 4 ore per uno Sprint di un mese e viene eseguito come parte del processo *Retrospectiva dello Sprint*. Durante questo incontro, lo Scrum Team si riunisce per rivedere e riflettere sullo Sprint precedente in termini di processi seguiti, strumenti impiegati, meccanismi di collaborazione e comunicazione ed altri aspetti di rilievo del progetto.

**Return on Investment (ROI) - Ritorno sull'Investimento (ROI)**

Il Ritorno sull'Investimento (ROI) quando utilizzato per la giustificazione del progetto, valuta l'utile netto atteso che deve essere ottenuto da un progetto. Si calcola sottraendo i costi o gli investimenti previsti di un progetto dal ricavo atteso e dividendo poi il risultato (profitto netto) per i costi previsti, al fine di ottenere un tasso di rendimento.

**Risk - Rischio**

Il rischio è definito come un evento o un insieme di eventi incerti che possono impattare gli obiettivi di un progetto e potrebbero contribuire al suo successo o fallimento.

**Risk Appetite - Propensione al Rischio**

Si riferisce alla quantità di incertezza che gli stakeholder o l'organizzazione sono disposti ad assumersi.

**Risk Assessment - Valutazione del Rischio**

La valutazione del rischio attiene alla misurazione e stima dei rischi identificati.

**Risk Attitude - Attitudine al Rischio**

In sostanza, l'atteggiamento degli stakeholder verso il rischio determina la quantità di rischio che gli stakeholder considerano accettabile. Questo fattore incide in modo determinante sulle loro decisioni riguardo a quando intraprendere azioni per mitigare i rischi potenzialmente sfavorevoli.

### **Risk Averse - Avverso al Rischio**

‘Avverso al Rischio’ è una delle tre categorie dell’Utility Function. Si riferisce ad uno Stakeholder che non è disposto ad accettare il rischio, qualunque sia il beneficio previsto o l’opportunità.

### **Risk Breakdown Structure - Struttura di Scomposizione del Rischio**

In questa struttura, i rischi sono raggruppati in base alla loro categoria o alle caratteristiche comuni. Ad esempio, i rischi possono essere classificati in finanziari, tecnici, o relativi alla sicurezza.

### **Risk Burndown Chart - Risk Burndown Chart**

Un grafico che illustra la gravità cumulativa del rischio di progetto nel tempo. Le probabilità dei vari rischi sono rappresentate graficamente una sopra l’altra per mostrare il rischio cumulativo sull’asse y. L’identificazione e la valutazione iniziale dei rischi del progetto e la creazione del Risk Burndown Chart vengono eseguite nelle prime fasi del progetto.

### **Risk Checklists - Liste di Controllo dei Rischi**

Le liste di controllo dei rischi contengono i punti chiave da considerare nell’attività di identificazione dei rischi, i rischi in cui ci si imbatte di solito nel progetto Scrum, o anche le categorie di rischio di cui il team si dovrebbe occupare.

### **Risk Communication - Comunicazione del Rischio**

La Comunicazione del Rischio attiene alla comunicazione dei risultati dei primi quattro passi della Gestione del Rischio agli Stakeholder appropriati e alla determinazione di come questi percepiscono gli eventi incerti.

### **Risk Identification - Identificazione del Rischio**

L’Identificazione del Rischio è un passo importante della Gestione del Rischio che implica l’uso di varie tecniche per identificare tutti i potenziali rischi.

### **Risk Meeting - Riunione sul Rischio**

I rischi possono essere prioritizzati più facilmente dal Product Owner convocando una riunione dello Scrum Core Team e invitandovi eventualmente anche gli Stakeholder pertinenti.

**Risk Mitigation - Mitigazione del Rischio**

La Mitigazione del Rischio è un passo importante della Gestione del Rischio che riguarda lo sviluppo di una adeguata strategia per trattare uno specifico rischio.

**Risk Neutral - Neutrale al Rischio**

'Neutrale al Rischio' è la categoria dell'Utility Function che si riferisce allo stakeholder che non è né avverso al rischio né alla ricerca del rischio; qualsiasi decisione data non è influenzata dal livello di incertezza del risultato. Quando due possibili scenari portano con sé lo stesso livello di beneficio, lo stakeholder neutrale al rischio non sarà preoccupato del fatto che uno scenario sia più rischioso dell'altro.

**Risk Prioritization - Prioritizzazione del Rischio**

La Prioritizzazione del Rischio è un passo importante della Gestione del Rischio che implica la prioritizzazione dei rischi da inserire nel Prioritized Product Backlog per l'esecuzione di una specifica azione di risposta.

**Risk Prompt Lists - Prompt List dei Rischi**

Le prompt list sono utilizzate per stimolare le riflessioni riguardo alle possibili fonti di rischio. Sono disponibili pubblicamente prompt list relative a vari settori industriali e tipi di progetto.

**Risk Seeking - Cercatore del Rischio**

'Cercatore del Rischio' è la categoria dell'Utility Function che si riferisce ad uno stakeholder che è disponibile ad accettare il rischio anche se realizza un aumento del ritorno o del beneficio del progetto del tutto marginale.

**Risk threshold - Soglia del Rischio**

La soglia del rischio si riferisce al livello di rischio accettabile per l'organizzazione degli stakeholder. Un rischio ricadrà sopra o sotto la soglia di rischio. Se è sotto, allora lo stakeholder o l'organizzazione sono più propensi ad accettare il rischio.

**Risk tolerance - Tolleranza al Rischio**

La tolleranza al rischio indica il grado, la quantità o il volume di rischio che gli stakeholder saranno disposti a sopportare.

**Risk-Based Spike - Risk-Based Spike**

I Risk-Based Spike sono in sostanza esperimenti che comportano la ricerca o la prototipazione per avere una migliore comprensione dei rischi potenziali. In uno spike, viene condotto un esercizio intenso della durata di due o tre giorni (preferibilmente all'inizio di un progetto o prima dei processi *Sviluppare le Epic* o *Creare il Prioritized Product Backlog*) per aiutare il team a stabilire quali eventi incerti potrebbero incidere sul progetto.

**Risks - Rischi**

I rischi sono tutti gli eventi incerti o non pianificati che possono incidere positivamente o negativamente sul progetto.

**Scope - Ambito**

L'ambito di un progetto è la somma totale di tutti gli incrementi di prodotto e del lavoro necessario a sviluppare il prodotto finale.

**Scrum Guidance Body - Scrum Guidance Body**

Lo Scrum Guidance Body (SGB) è un ruolo opzionale. Generalmente consiste in un gruppo di documenti e/o in un gruppo di esperti che si occupano di norma della definizione degli obiettivi relativi alla qualità, ai regolamenti governativi, alla sicurezza e ad altri parametri chiave dell'organizzazione.

**Scrum Guidance Body Expertise - Competenze Esperte dello Scrum Guidance Body**

Le Competenze Esperte dello Scrum Guidance Body riguardano norme e regolamenti documentati, linee guida o standard di sviluppo e best practice.

**Scrum Master - Scrum Master**

Lo Scrum Master è uno dei ruoli dello Scrum Core Team. Questi facilita la creazione dei deliverable del progetto, gestisce i rischi, i cambiamenti e gli impedimenti durante i processi *Condurre il Daily Standup*, *Retrospektiva dello Sprint*, e in altri processi di Scrum.

**Scrum of Scrums Meeting - Scrum of Scrums Meeting**

Lo Scrum of Scrums (SoS) Meeting è una riunione importante quando si applica Scrum a progetti di grandi dimensioni a cui partecipano rappresentanti di tutti i team. Questa riunione è di solito facilitata dal Chief Scrum Master e ha lo scopo di focalizzarsi sulle aree di coordinamento ed integrazione fra i diversi Scrum Team. Viene tenuta ad intervalli regolari o quando richiesto dagli Scrum Team.

**Scrum Team - Scrum Team**

Lo Scrum Team è uno dei ruoli dello Scrum Core Team. Lo Scrum Team lavora alla creazione dei deliverable del progetto e contribuisce alla realizzazione del valore di business per tutti gli stakeholder e per il progetto.

**Scrum Team Lessons Learned - Lezioni Apprese dello Scrum Team**

Ci si aspetta che lo Scrum Team, a motivo della sua auto-organizzazione e del potere che gli è stato conferito, sia capace di imparare dagli errori commessi durante uno Sprint e queste lezioni apprese aiutano i team a migliorare le proprie prestazioni negli Sprint a venire.

**Scrum Team Representatives - Rappresentanti degli Scrum Team**

Un rappresentante nominato dal team per rappresentare il gruppo negli Scrum of Scrums (SoS) Meeting in base a chi è più adatto a ricoprire il ruolo in virtù delle problematiche e delle circostanze del momento.

**Scrumboard - Scrumboard**

La Scrumboard è uno strumento utilizzato dallo Scrum Team per pianificare e monitorare lo stato di avanzamento durante ciascuno Sprint. La Scrumboard contiene quattro colonne per indicare l'avanzamento delle attività stimate di un determinato Sprint: una colonna 'Da fare' per le attività non ancora iniziate, una colonna 'In Corso' per le attività iniziate ma non ancora completate, una colonna 'Test' per le attività completate ma in fase di test, e infine una colonna 'Fatto' per le attività che sono state completate e testate con successo.

**Self-organization - Auto-organizzazione**

Scrum crede che i lavoratori siano auto-motivati e cerchino di responsabilità più grandi. Pertanto, essi realizzano un valore molto maggiore quando sono auto-organizzati.

**Servant Leader - Leader Servitore**

I leader servitori usano l'ascolto, l'empatia, l'impegno e l'intuizione mentre condividono il potere e l'autorità con i membri del team. I leader servitori sono degli steward che raggiungono risultati mettendo al centro dell'attenzione i bisogni del team. Questo stile rappresenta l'incarnazione del ruolo dello Scrum Master.

**Ship Deliverables - Consegnare i Deliverable**

In questo processo i Deliverable Accettati sono consegnati o trasferiti agli Stakeholder pertinenti. Un Accordo formale sui Deliverable Funzionanti documenta il positivo completamento dello Sprint.

**Simple Schemes - Schemi Semplici**

Gli schemi semplici consistono nel classificare gli elementi come Priorità “1”, “2”, “3” o “Alta”, “Media” e “Bassa” e così via. Anche se si tratta di un approccio semplice e trasparente, può causare dei problemi perché spesso c'è la tendenza a classificare tutto come Priorità “1” o “Alta”.

**Skills Requirement Matrix - Matrice dei Requisiti di Competenza**

La Matrice dei Requisiti di Competenza, nota anche come struttura delle competenze, è utilizzata per valutare le lacune di competenza e le necessità di formazione dei membri del team. Una matrice delle competenze mappa le competenze, le capacità e il livello di interesse dei membri del team ad utilizzare quelle competenze e capacità in un determinato progetto. Attraverso l'utilizzo di questa matrice, l'organizzazione può valutare eventuali lacune di competenze da parte dei membri del team e identificare il personale che ha bisogno di ulteriore formazione in un particolare settore o competenza.

**Speed Boat - Speed Boat**

La Speed Boat è una tecnica che può essere utilizzata per condurre il Retrospect Sprint Meeting. I membri del team fanno la parte dell'equipaggio di un motoscafo (Speed Boat). L'imbarcazione deve raggiungere un'isola, che rappresenta simbolicamente la Vision del Progetto. I partecipanti utilizzano dei post-it per registrare i motori e le ancore. I motori li aiutano a raggiungere l'isola, mentre le ancore impediscono loro di raggiungerla. Questo esercizio ha una durata prefissata di pochi minuti.

**Sponsor - Sponsor**

Lo sponsor è la persona o l'organizzazione che fornisce risorse e supporto al progetto. Lo sponsor è anche lo stakeholder al quale alla fine tutti devono rispondere.

**Sprint - Sprint**

Uno Sprint è un'iterazione che ha una durata predeterminata da una a sei settimane, durante le quali lo Scrum Team lavora alla creazione dei deliverable dello Sprint.

**Sprint Backlog - Sprint Backlog**

Lo Sprint Backlog è un elenco delle attività che lo Scrum Team deve eseguire nello Sprint venturo.

**Sprint Burndown Chart - Sprint Burndown Chart**

Lo Sprint Burndown Chart è un grafico che illustra la quantità di lavoro che rimane da fare nello Sprint in corso.

**Sprint Deliverables - Deliverable dello Sprint**

I Deliverable dello Sprint sono gli incrementi di prodotto o i deliverable che sono stati completati al termine di ciascuno Sprint.

**Sprint Planning Meeting - Sprint Planning Meeting**

Lo Sprint Planning Meeting viene condotto all'inizio di un Sprint come parte del processo *Creare lo Sprint Backlog*. Ha una durata predeterminata di otto ore per uno Sprint di un mese e si divide in due parti: Definizione dell'Obiettivo e Stima delle Attività.

**Sprint Review Meeting - Sprint Review Meeting**

Lo Sprint Review Meeting ha una durata predeterminata di quattro ore per uno Sprint di un mese e può essere scalato a seconda della lunghezza dello Sprint. Durante lo Sprint Review Meeting, lo Scrum Team presenta i deliverable dello Sprint corrente al Product Owner, che li può accettare o rifiutare.

**Sprint Tracking Tools - Strumenti di Monitoraggio dello Sprint**

Gli Strumenti di Monitoraggio dello Sprint sono utilizzati per monitorare lo stato di avanzamento di uno Sprint e per sapere a che punto si trova uno Scrum Team in termini di completamento delle attività dello Sprint Backlog. Per monitorare il lavoro di uno Sprint si possono usare tanti tipi di strumenti, ma uno dei più comuni è una Scrumboard, nota anche come task board o progress chart.

**Sprint Velocity - Velocità dello Sprint**

La Velocità dello Sprint rappresenta la velocità alla quale il team può completare il lavoro in uno Sprint. Di solito è espressa nelle stesse unità utilizzate per la stima, di norma gli story point o il tempo ideale.

**Stakeholder(s) - Stakeholder**

Stakeholder è un termine collettivo che ricomprende i clienti, gli utenti e gli sponsor, i quali tutti si interfacciano spesso con il Product Owner, lo Scrum Master e lo Scrum Team per fornire loro input e facilitare la creazione del prodotto, servizio o altro risultato del progetto.

**Storming stage - Fase Storming**

La seconda fase di formazione del team, nella quale il team prova a eseguire il lavoro. Tuttavia, possono verificarsi lotte di potere e spesso c'è caos e confusione fra i membri del team.

**Story Mapping - Story Mapping**

Lo Story Mapping è una tecnica che serve a fornire un profilo visivo del prodotto e dei suoi componenti chiave. Lo Story Mapping, formulato per la prima volta da Jeff Patton nel 2005, è comunemente usato per illustrare le roadmap del prodotto. Le story map raffigurano la sequenza delle iterazioni di sviluppo del prodotto e delineano quali funzionalità saranno incluse nel primo, nel secondo, nel terzo rilascio ed in quelli seguenti.

**Sustainable Pace - Ritmo Sostenibile**

Il Ritmo Sostenibile è il ritmo al quale il team può lavorare e che può mantenere comodamente. Esso si traduce nell'aumento della soddisfazione del personale, nella stabilità e in una maggiore accuratezza delle stime, fattori che portano tutti in definitiva ad una maggiore soddisfazione del cliente.

**SWOT Analysis - Analisi SWOT**

Lo SWOT è un approccio strutturato alla pianificazione del progetto che aiuta a valutare i punti di forza, i punti di debolezza, le opportunità e le minacce relativi ad un progetto. Questo tipo di analisi aiuta a identificare i fattori sia interni che esterni che possono avere un impatto sul progetto.

**Target Customers for Release - Clienti Target del Rilascio**

Non tutti i rilasci saranno rivolti a tutti gli stakeholder o a tutti gli utenti. Gli Stakeholder possono scegliere di limitare determinati rilasci ad un sottoinsieme di utenti. Il Piano dei rilasci specifica i Clienti Target (cioè destinatari) del singolo Rilascio.

**Task Estimation Workshop - Workshop sulla Stima delle Attività**

Il Workshop sulla Stima delle Attività consente allo Scrum Team di stimare l'impegno necessario a completare un'attività o un insieme di attività e a stimare l'impegno delle persone e le altre risorse che servono per portare a termine le attività all'interno di un dato Sprint.

**Task List - Lista delle Attività**

Si tratta di un elenco completo che comprende tutte le attività che lo Scrum Team si è impegnato a compiere nello Sprint corrente. Contiene le descrizioni di ciascuna attività.

**Task Planning Meeting - Task Planning Meeting**

In un Task Planning Meeting lo Scrum Team si riunisce per pianificare il lavoro da compiere nello Sprint e rivede le User Story prese in carico poste in cima al Prioritized Product Backlog. Per aiutare il gruppo a rimanere in argomento, si dovrebbe stabilire una durata predeterminata per questa riunione, con una lunghezza standard contenuta in due ore per ogni settimana di durata dello Sprint.

**Task-Oriented Leader - Leader Orientato alle Attività**

I leader orientati alle attività esigono il completamento delle attività e il rispetto delle scadenze.

**Team Building Plan - Piano di Team Building**

Dal momento che uno Scrum Team è interfunzionale, ogni membro deve partecipare attivamente a tutti gli aspetti del progetto. Lo Scrum Master deve identificare le potenziali questioni che possono riguardare i membri del team e affrontarle in modo rigoroso per mantenere salda l'efficacia del team.

**Team Calendar - Calendario del Team**

Un Calendario del Team contiene informazioni sulla disponibilità dei membri del team, fra cui quelle relative alle ferie dei dipendenti, ai permessi, ad eventi importanti e alle festività.

**Team Expertise - Competenza Esperta del Team**

La Competenza Esperta del Team si riferisce alla capacità dei membri dello Scrum Team di comprendere le User Story e le Attività inserite nello Sprint Backlog allo scopo di creare i deliverable finali. La Competenza Esperta del Team è utilizzata per valutare gli input necessari all'esecuzione del lavoro pianificato per il progetto.

**Technical Debt - Technical Debt**

Il technical debt (chiamato anche debito di progettazione o debito del codice) si riferisce al lavoro a cui il team dà priorità più bassa, omette, o non completa in quanto lavora alla creazione dei deliverable principali associati al prodotto del progetto. Il technical debt si accumula e si paga nel futuro.

**Theory X - Teoria X**

I leader della Teoria X ritengono che i dipendenti siano intrinsecamente privi di motivazione e che se possibile eviteranno il lavoro, legittimando quindi l'uso di uno stile di gestione autoritario.

**Theory Y - Teoria Y**

I leader della Teoria Y ritengono che i dipendenti siano auto-motivati e che cerchino di accettare responsabilità maggiori. La Teoria Y comporta uno stile manageriale più partecipativo.

**Threats - Minacce**

Le minacce sono rischi che possono incidere negativamente sul progetto.

**Three Daily Questions - Tre Domande Quotidiane**

Si tratta delle Tre Domande Quotidiane utilizzate nei Daily Standup Meeting, facilitati dallo Scrum Master, in cui ciascun membro dello Scrum Team fornisce informazioni sotto forma di risposte a tre domande specifiche:

- Che cosa ho completato ieri?
- Cosa completerò oggi?
- Quali impedimenti o ostacoli sto eventualmente incontrando in questo momento?

**Time-boxing - Time-boxing**

Il Time-boxing consiste nella fissazione di brevi periodi di tempo per l'esecuzione del lavoro. Se il lavoro intrapreso non viene completato entro la fine del Time-box, viene trasferito in un altro Time-box. I time-box forniscono la struttura necessaria per i progetti Scrum, che hanno un elemento di incertezza, sono dinamici per natura e sono soggetti a cambiamenti frequenti.

**Transparency - Trasparenza**

La trasparenza permette a chiunque di osservare tutti i risvolti di un qualunque processo Scrum. La condivisione di tutte le informazioni porta ad un ambiente di elevata fiducia.

**Unapproved Change Requests - Richieste di Modifica Non Approvate**

Le richieste di procedere a dei cambiamenti sono di solito presentate come Richieste di Modifica. Le Richieste di Modifica rimangono nello stato di 'non approvate' fino a quando non sono formalmente approvate.

**Updated Program Product Backlog - Program Product Backlog Aggiornato**

Un Program Product Backlog che subisce messe a punto periodiche per incorporare cambiamenti e nuovi requisiti.

**User - Utente**

Gli utenti sono gli individui o l'organizzazione che usa direttamente il prodotto, servizio o altro risultato del progetto. In alcuni casi, clienti e utenti possono coincidere.

**User Group Meetings - Riunioni di Gruppi di Utenti**

Le Riunioni di Gruppi di Utenti coinvolgono gli stakeholder pertinenti, prima di tutto gli utenti e i clienti del prodotto. Queste riunioni forniscono allo Scrum Core Team informazioni di prima mano riguardo alle aspettative degli utenti. Ciò aiuta a formulare i Criteri di Accettazione del prodotto e offre idee preziose per lo sviluppo delle Epic.

### **User Stories - User Story**

Le User Story si conformano ad una specifica struttura predefinita e costituiscono un modo semplice di documentare i requisiti e le funzionalità desiderate dell'utente finale. I requisiti espressi nelle User Story sono dichiarazioni brevi, semplici e di facile comprensione che si traducono in un miglioramento della comunicazione fra gli stakeholder ed in stime più precise da parte del team.

### **User Story Acceptance Criteria - Criteri di Accettazione delle User Story**

Ciascuna User Story ha dei Criteri di Accettazione ad essa associati. Le User Story sono soggettive, per cui i Criteri di Accettazione forniscono l'obiettività necessaria per poter considerare la User Story come 'Done' (fatta) o 'not Done' (cioè incompiuta) in sede di Revisione dello Sprint, chiarendo al team cosa ci si aspetta da una User Story.

### **User Story Workshops - Workshop sulle User Story**

I Workshop sulle User Story vengono tenuti come parte del processo *Sviluppare le Epic*. Queste sessioni sono facilitate dallo Scrum Master. In tali workshop è coinvolto l'intero Scrum Core Team e a volte è auspicabile farvi partecipare anche altri Stakeholder.

### **User Story Writing Expertise - Competenze di Scrittura delle User Story**

Il Product Owner, basandosi sull'interazione con gli stakeholder, sulla sua conoscenza e competenza di business e sugli input forniti dal team, sviluppa le User Story che andranno a costituire il Prioritized Product Backlog iniziale del progetto.

### **Utility Function - Utility Function**

L'Utility Function è un modello utilizzato per misurare la preferenza di rischio degli stakeholder o il loro atteggiamento nei confronti del rischio. Esso definisce il livello o disponibilità degli stakeholder ad accettare il rischio.

### **Value Stream Mapping - Value Stream Mapping**

Il Value Stream Mapping utilizza i diagrammi di flusso per illustrare il flusso delle informazioni necessarie a completare un processo. Questa tecnica può essere usata per snellire un processo aiutando ad individuare gli elementi che non aggiungono valore.

### **Vendor - Venditore**

I venditori sono le persone o le organizzazioni esterne che forniscono prodotti e servizi che non rientrano nelle competenze principali dell'organizzazione di progetto.

### **Voice of the Customer (VOC) - Voce del Cliente (VOC)**

La Voce del Cliente (Voice of the Customer - VOC) può essere definita come i requisiti impliciti ed espliciti del cliente, che devono essere ben compresi prima di progettare un prodotto o servizio. Il Product Owner rappresenta la Voce del Cliente.

### **War Room - War Room**

War Room è il termine comunemente usato per descrivere il luogo dove sono ubicati tutti i membri dello Scrum Team al lavoro. Di norma è progettata in modo tale che i membri del team possono muoversi intorno liberamente, lavorare e comunicare facilmente perché si trovano a stretto contatto tra loro.

### **Wideband Delphi Technique - Tecnica Wideband Delphi**

La Wideband Delphi è una tecnica di stima di gruppo che serve a stabilire quanto lavoro si deve eseguire e quanto tempo servirà per completarlo. I singoli individui di un team forniscono anonimamente delle stime per ciascuna funzionalità e le stime iniziali vengono segnate su un grafico. Dopo di che il team discute i fattori che hanno influenzato le stime di ognuno e si procede con un secondo giro di stime. Questo processo viene ripetuto fino a quando le stime dei singoli sono vicine fra loro e si può quindi raggiungere un consenso sulla stima finale.

### **Working Deliverables - Deliverable Funzionanti**

Questo output rappresenta il deliverable consegnabile finale per il quale è stato deliberato il progetto.

### **Working Deliverables Agreement - Accordo sui Deliverable Funzionanti**

I deliverable che soddisfano i Criteri di Accettazione ricevono la formale autorizzazione del business e l'approvazione del cliente o dello sponsor.

# INDICE

## 1

100-point method, 183

## A

Accepted Deliverables, 264

Actual Cost, 85

**Adaptability**, 4

Adaptation, 25

**Affinity Estimation**, 208

Agreed Actionable Improvements, 269

Applicable Contracts, 173

*Appropriation*, 30

*Approve, Estimate, and Commit User Stories*, 18, 195

inputs, 206

outputs, 209

tools, 206

Approved Change Requests, 108, 172

Approved Changes, 178

Approved, Estimated, and Committed User Stories, 209

*Articulation*, 30

**Aspects**, 7

**Assertive**, 65

Assigned Action Items and Due Dates, 270

**Autocratic**, 65

*Awareness*, 30

## B

Back-up Persons, 169

Better Team Coordination, 258

**Brainstorming**, 131

**Business case**, 75

Business Justification, 14, 69

Business needs, 74

Business Requirements, 181

Business value, 93

*Business value delivered*, 223

## C

Change, 15, 106, 107

Change approval process, 109

Change Requests, 108

**Chief Product Owner**, 13, 47, 151

**Chief Scrum Master**, 13, 49, 158

**Coaching**, 65

**Collaboration**, 11, 22

Collaboration Plan, 169

**Collective ownership**, 5

**Colocated Teams**, 32

Communication Plan, 280

Communication Techniques, 247

Company Vision, 152

*Conduct Daily Standup*, 19, 228

inputs, 239

outputs, 241

tools, 240

*Conduct Release Planning*, 18, 145

inputs, 189

outputs, 191

tools, 190

**Confirm benefits realization**, 75, 88

Conflict Management, 62

Techniques, 62

**Continuous delivery of value**, 4

**Continuous feedback**, 4

**Continuous improvement**, 4

Continuous integration, 101, 116

**Continuous value justification**, 75, 83

*Convene Scrum of Scrums*, 19, 251

inputs, 255

outputs, 258

tools, 256

**Core roles**, 12

Cost Performance Index, 85

Cost Variance, 85

*Create Deliverables*, 19, 227

inputs, 233

outputs, 236

tools, 235

*Create Prioritized Product Backlog*, 18, 145

inputs, 181

outputs, 184

tools, 182

*Create Project Vision*, 18, 144, 148, 156, 170, 179, 187,

198, 205, 210, 215, 220, 231, 238, 243, 254, 259,

266, 277, 283

inputs, 150

outputs, 155

*Create Sprint Backlog*, 19, 195

inputs, 222

- outputs, 225
- Create Tasks*, 19, 195
  - inputs, 211
  - outputs, 214
  - tools, 211
- Create User Stories*, 18, 194
  - inputs, 200
  - outputs, 203
  - tools, 201
- Cross-functional, 115
- Cumulative Flow Diagram (CFD), 87
- Customer centric**, 4
- Customer Value-based Prioritization**, 80, 115
- Customers**, 73

## D

- Daily Standup Meeting**, 35, 240
- Data Flow Diagram
  - Initiate phase, 193
  - Plan and Estimate phase, 226
  - Release phase, 288
  - Review and Retrospect phase, 272
- Decomposition, 212
- Delegating**, 65
- Delighters*, 81
- Demonstrate and Validate Sprint*, 20, 251
  - inputs, 261
  - outputs, 264
  - tools, 262
- Demonstrations, 88
- Dependencies**, 185, 214
- Dependency Determination, 213
- Design Patterns, 236
- Develop Epic(s)*, 18, 145
  - inputs, 172
  - outputs, 177
- Development in Phases Contract**, 174
- Directing**, 65
- Discretionary dependencies**, 213
- Dissatisfiers*, 81
- Distributed Teams**, 23
- Done, 98
- Done Criteria, 98, 185
- Done success rate, 269

## E

- Earned Value, 85
- Earned Value Analysis (EVA), 83

- Effective Deliverables**, 4
- Efficient development process**, 4
- Effort Estimated Task List, 219
- Empirical Process Control**, 11, 22, 23
- Epic(s), 177
- Estimate at Completion, 85
- Estimate Range**, 209
- Estimate Tasks*, 19, 195
  - inputs, 217
  - outputs, 219
  - tools, 218
- Estimate to Complete, 85
- Estimates**, 185
- Estimation Criteria, 218
- Estimation effectiveness, 269
- Exciters*, 81
- Expected Monetary Value (EMV)**, 135
- Expert Advice from HR, 161, 166
- Expert Scrum Master (ESM™), 6
- Explorer – Shopper – Vacationer – Prisoner (ESVP), 268
- External dependencies**, 213
- External stakeholders, 99

## F

- Faster problem resolution**, 4
- Fist of Five, 207
- Focus Group Meetings, 175, 202
- Form Scrum Team*, 18, 145
  - inputs, 165
  - outputs, 168
  - tools, 166
- Forming**, 61
- Four Questions per Team, 256

## G

- Gap Analysis, 154
- Groom Prioritized Product Backlog*, 19, 228, 243
  - inputs, 245
  - outputs, 248
  - tools, 247

## H

- High trust environment**, 5
- High velocity**, 5
- Holiday Calendar, 190
- HR Theories, 61

**I**

Identified Product Owner, 155  
 Identified Risks, 178  
 Identified Scrum Master, 162  
 Identified Scrum Team, 168  
 Identified Stakeholder(s), 162  
*Identify Scrum Master and Stakeholder(s)*, 18, 145, 156  
   inputs, 158  
   outputs, 162  
   tools, 160  
 Impediment Log, 234  
 Implement, 19, 227  
**Incentive and Penalty Contract**, 174  
**Incremental Delivery Contract**, 173  
 Index Cards, 212  
*Indifferent*, 81  
 Initiate, 18, 144  
**Innovative environment**, 5  
 Inspection, 25  
 Integrating Change, 116  
**Internal dependencies**, 213  
 Internal Rate of Return (IRR), 78  
 Internal stakeholders, 99  
 Issues, 129  
**Iterative Development**, 11, 23, 38, 112

**J**

JAD Sessions, 153  
**Joint Venture Contract**, 174

**K**

Kano analysis, 81, 316

**L**

**Laissez Faire**, 65  
 Laws and Regulations, 173  
 Leadership Styles, 65  
 Length of Sprint, 119, 191  
 Lose/Lose, 63  
 Lose/Win, 63

**M**

**Mandatory dependencies**, 213  
 Market Study, 152  
 Maslow's Hierarchy of Needs, 67

Meeting Agenda, 255  
 Meeting Room, 257  
 Metrics and Measuring Techniques, 268  
 Minimum Acceptance Criteria, 97  
 Minimum Marketable Features (MMF), 82  
 Mitigated Risks, 237  
**Monopoly money**, 80  
**MoSCoW prioritization**, 182  
 Motivated Scrum Team, 241  
**Motivation**, 4

**N**

Net Present Value (NPV), 78  
**Non-core roles**, 13  
**Norming**, 61  
*Number of stories*, 223

**O**

Organization, 12  
 Organizational Deployment Methods, 280  
 Organizational Resource Matrix, 159  
 Other Estimation Techniques, 208

**P**

**Paired comparison**, 183  
**Pareto analysis**, 133  
 PDCA Cycle, 102, 319  
 Peer feedback, 269  
 People Availability and Commitment, 159  
 People Costs, 166  
 People Requirements, 159  
 Percent Complete, 85  
**Performing**, 61  
 Personas, 177  
 Personnel Selection, 51  
 Piloting Plan, 279  
 Plan and Estimate, 18, 194  
 Plan-Do-Check-Act (PDCA) Cycle, 102, 319  
 Plan-Do-Study-Act Cycle, 102, 319  
 Planned Value, 85  
 Planning for value, 80  
 Planning Poker, 207  
 Points for Cost Estimation, 207  
**Portfolio**, 52, 56  
 Portfolio Product Owner, 56, 72  
**Portfolio Scrum Master**, 56

## Portfolios

Change, 122

Risk, 139

Previous Project Information, 174

Previous Sprint Velocity, 222

**Principles**, 7

Prioritized Product Backlog, 95, 184

Prioritized Product Backlog Review Meetings, 247

**Probability Impact grid**, 133**Probability trees**, 132**Processes**, 7

Product Backlog Items (PBIs), 122

Product Backlog Review Meeting, 121

**Product Owner**, 12, 72

Product Releases, 282

**Program**, 52, 56

Program and Portfolio Risks, 173

Program Product Backlog, 151

**Program Product Owner**, 56, 72, 150**Program Scrum Master**, 56, 150

Program Stakeholder(s), 151

## Programs

Change, 123

Risk, 139

Progress to release/launch, 269

**Project**, 52

Project Budget, 156

Project Business Case, 150

Project Charter, 155

Project costs, 74

Project Increment Flow, 96

Project reasoning, 74

**Project timescales**, 74

Project Vision Meeting, 153

Project Vision Statement, 155

Proof of Concept, 152

Proposed Non-Functional Items, 271

Prototypes, 88

**Q**

Quality, 15, 91, 92

Quality control, 101, 324

Quality management, 99

Quality planning, 100

Questionnaires, 176

**R**

Refactoring, 235

Refined Prioritized Product Backlog, 192

Relative prioritization ranking, 82

**Relative Sizing / Story Points**, 208

Release, 20, 273

Release Planning Schedule, 191

Release Planning Sessions, 190

Release Prioritization Methods, 191

Resolved Issues, 258

Resource Costs, 161, 168

Resource Requirements, 166

*Retrospect Project*, 20, 273

inputs, 285

outputs, 287

tools, 286

Retrospect Project Meeting, 286

*Retrospect Sprint*, 20, 251

inputs, 267

outputs, 269

tools, 267

Retrospect Sprint Log(s), 271

**Retrospect Sprint Meeting**, 36, 267

Return on Investment (ROI), 77, 327

Review and Retrospect, 19

Review feedback ratings, 269

Risk, 16, 127, 185

Risk appetite, 129

Risk assessment, 132

Risk attitude, 129

Risk averse, 130

**Risk Breakdown Structure (RBS)**, 131

Risk Burndown Chart, 137

**Risk checklists**, 131

Risk communication, 137

Risk identification, 130

techniques, 130

Risk management, 130

procedure, 130

**Risk meeting**, 132

Risk mitigation, 136

Risk neutral, 130

Risk prioritization, 135

Risk prompt lists, 131, 329

Risk seeking, 130

Risk threshold, 129

Risk tolerance, 129

Risk-based spike, 131, 330

Risks, 74

## S

*Satisfiers*, 81

Scalability of Scrum, 5

Schedule Performance Index, 85

Schedule Variance, 85

Scope, 93

Scrum Aspects, 12

**Scrum Core Team**, 113, 172

Scrum Developer Certified (SDC™), 6

**Scrum Guidance Body** (SGB), 13, 73

Scrum Guidance Body Expertise, 176, 184, 202, 209, 236, 257, 262, 269, 286

Scrum Guidance Body Recommendations, 153, 174, 182, 190, 201, 206, 217, 234, 246, 262, 267, 280, 285

**Scrum Master**, 12, 47, 73, 255

Scrum Master Certified (SMC™), 6

Scrum of Scrums, 53

Scrum of Scrums Meeting, 256

Scrum Processes, 17

Scrum Product Owner Certified (SPOC™), 6

**Scrum Team**, 12, 50, 73

Scrum Team Lessons Learned, 271

Scrum Team Representatives, 255

Scrum Team Selection, 166

Scrumboard, 233

SCRUMstudy Agile Master Certified (SAMC™), 6

Selection Criteria, 160

**Self-organization**, 11, 22, 28, 115

**Senior Management**, 114

**Servant Leadership**, 65

*Ship Deliverables*, 20, 273, 277

inputs, 279

outputs, 282

tools, 280

Simple schemes, 80, 332

Simulations, 88

Skills Requirement Matrix, 160

Software, 235

Speed Boat, 268

**Sponsor**, 45, 73

**Sprint**, 35

Sprint Backlog, 225

Sprint Burndown Chart, 225

Sprint Burnup Chart, 225

Sprint Deliverables, 236

**Sprint Planning Meeting**, 35, 223

**Sprint Review Meeting**, 36, 262

Sprint Tracking Metrics, 223

Sprint Tracking Tools, 223

Stability, 110

Stakeholder Involvement, 58

**Stakeholder(s)**, 13, 112

**Storming**, 61

Story Mapping, 83

**Supportive**, 65

**Sustainable pace**, 4, 101

SWOT Analysis, 154

## T

Target Customers for Release, 192

Task Estimation Meetings, 218

Task List, 214

Task Planning Meetings, 211

Task-oriented, 65, 335

Team Building Plan, 169

Team Calendar, 222

Team Expertise, 235

Team morale ratings, 269

Team velocity, 269

Technical debt, 101

**Theory X**, 68

**Theory Y**, 68

Three Daily Questions, 240

**Time-boxing**, 11, 23, 35, 114

Traditional Project Management, 21

Training and Training Costs, 161, 168

**Transparency**, 4, 23

Trial Project, 152

Tuckman's Model of Group Dynamics, 61

## U

Unapproved Change Requests, 108, 172

**Uncertainty**, 185

Updated Approved, Estimated, and Committed User Stories, 214

Updated or Refined Personas, 204

Updated Prioritized Product Backlog, 204, 248

Updated Program Product Backlog, 245

Updated Release Planning Schedule, 248

Updated Scrum Guidance Body Recommendations, 271

Updated Scrumboard, 236

Updated Task List, 219

User Group Meetings, 175

User or Customer Interviews, 176  
User Stories, 203, 206  
User Story Acceptance Criteria, 95, 203, 217  
User Story Estimation Methods, 183, 202  
User Story Prioritization Methods, 182  
User Story Workshops, 175  
User Story Writing Expertise, 201  
**Users**, 45, 73  
*Utility Function*, 129

## V

**Value**, 184  
**Value Stream Mapping**, 80  
**Value-based Prioritization**, 11, 23, 33

**Value-driven Delivery**, 69, 70  
Variance at Completion, 85  
*Velocity*, 223  
**Vendors**, 13, 45  
Video Conferencing, 241, 257  
Voice of the Customer, 47

## W

War Room, 240  
**Wideband Delphi**, 208  
Win/Lose, 64  
Win/Win, 63  
Working Deliverables, 282  
Working Deliverables Agreement, 282

## La guida essenziale per realizzare progetti con successo Scrum

La *Guida SBOK™* è stata sviluppata come un mezzo per creare un punto di riferimento indispensabile per le organizzazioni e i professionisti della gestione del progetto che vogliono implementare Scrum, oltre che per coloro che già lo fanno e che intendono apportare i cambiamenti desiderati ai propri processi. Tale guida si basa sull'esperienza attinta da migliaia di progetti di una grande varietà di organizzazioni e settori industriali. Nel suo sviluppo sono stati presi in considerazione i contributi di molti esperti di Scrum e professionisti della gestione dei progetti.

L'adozione diffusa del framework *Guida SBOK™* dovrebbe aiutare a standardizzare il modo in cui Scrum viene applicato ai progetti dalla globalità delle organizzazioni, oltre ad aiutare a migliorare in modo significativo il Ritorno sull'Investimento. In aggiunta, dovrebbe promuovere una maggiore riflessione e discussione riguardo all'applicazione di Scrum a molti tipi di progetti, e questo a sua volta contribuirà ad una espansione ed arricchimento del corpo di conoscenze e, di conseguenza, a rendere possibili futuri aggiornamenti di questa guida.

La *Guida SBOK™* può essere utilizzata come punto di riferimento e conoscenza sia da coloro che hanno esperienza di Scrum sia dagli altri professionisti dello sviluppo del prodotto e del servizio, oltre che dalle persone che non hanno alcuna precedente esperienza o conoscenza di Scrum o di una metodologia di project management.

