

# READY

WHY MOST SOFTWARE PROJECTS  
FAIL AND HOW TO FIX IT



Luniel de Beer

Max Guernsey, III

***Edizione Italiana***

# Twitta Questo Libro!

Aiuta Luniel de Beer e Max Guernsey, III diffondendo la notizia di questo libro su [Twitter!](#)

Il tweet suggerito per questo libro è:

Ho appena acquistato Ready — un libro per leader del software e team che vogliono eliminare riprocessazioni, riporti e disallineamenti nella distribuzione del software.  
[#CodeReady](#)

L'hashtag suggerito per questo libro è [#CodeReady](#).

Scopri cosa dicono le altre persone del libro cliccando su questo link per cercare questo hashtag su Twitter:

[#CodeReady](#)

# **Ready (Edizione Italiana)**

Perché la Maggior Parte dei Progetti Software Fallisce  
e Come Risolverlo

Luniel de Beer e Max Guernsey, III

Questo libro è disponibile su <https://leanpub.com/ready-it>

Questa versione è stata pubblicata il 2025-10-22



Questo è un libro [Leanpub](#). Leanpub consente ad autori ed editori di utilizzare il processo Lean Publishing. Il [Lean Publishing](#) è il processo di pubblicazione di un ebook in evoluzione utilizzando strumenti leggeri e molte iterazioni per ottenere feedback dai lettori, modificare il percorso fino a ottenere il libro giusto e costruire seguito una volta raggiunto l'obiettivo.

© 2025 Luniel de Beer e Max Guernsey, III

*In memoria di Johann van Aardt, che ha riconosciuto la mia passione, mi ha introdotto alla vera programmazione e mi ha aiutato a trovare la strada verso una nuova casa. E ai miei genitori, il cui sostegno incrollabile—dal mio primo cliente alla mia prima casa negli Stati Uniti—ha reso tutto questo possibile.*

*—Luniel*

*Per la mia famiglia, con cui sorge e tramonta il sole.*

*—Max*

# Indice

Informazioni su questo libro . . . . .	i
A chi è rivolto . . . . .	ii
Come utilizzare questo libro . . . . .	iii
Gli autori . . . . .	iv
Prefazione . . . . .	v
 <b>Parte I: Qualcosa Manca . . . . .</b>	 <b>1</b>
Capitolo 1: Il Problema Nascosto . . . . .	2
Capitolo 2: Il Costo delle Fondamenta Mancanti . . . . .	14
Capitolo 3: Introduzione al Requirements Maturation Flow (RMF) . . . . .	26
Capitolo 4: È Agile? . . . . .	28
 <b>Parte II: Creare Spazio per la Preparazione . . . . .</b>	 <b>29</b>
Capitolo 5: La Prima Estensione . . . . .	30
Capitolo 6: Perché Le Persone Non Lo Fanno? . . . . .	31
Capitolo 7: Lavoro Esplicito di Preparazione (RMF 1) . . . . .	33
Capitolo 8: Effetti del RMF 1 . . . . .	35
Capitolo 9: Mettere in Pratica RMF 1 . . . . .	36

<b>Parte III: Controllo del Completamento del Lavoro</b>	<b>38</b>
Capitolo 10: La Prossima Necessità	39
Capitolo 11: Cosa Fanno Solitamente le Persone	41
Capitolo 12: Definire una Definizione di Fatto	43
Capitolo 13: Definition of Done Personalizzata (RMF 2)	45
Capitolo 14: La Vita con RMF 1 e 2	49
Capitolo 15: Installazione di RMF 2	51
<b>Parte IV: Implementazione del Gating</b>	<b>53</b>
Capitolo 16: Il Requisito Finale	54
Capitolo 17: Background sulla Definizione di Prontezza	56
Capitolo 18: Definire una Definizione di Pronto	58
Capitolo 19: Definizione di Pronto Personalizzata (RMF 3)	60
<b>Parte V: Sintesi</b>	<b>64</b>
Capitolo 20: <u>La Maggior Parte</u> delle Scadenze Non Conta	65
Capitolo 21: Competenza 1: Flusso di Maturazione dei Requisiti	67
Capitolo 22: Come Fluiscono il Lavoro e le Informazioni in Scrum con RMF	69
Capitolo 23: L'Impatto dell'RMF	71
Capitolo 24: Transizione verso RMF	72
Capitolo 25: Sta a Te	74
<b>Parte VI: Risorse</b>	<b>75</b>
Appendice A: Scrum Non È il Problema	76

Appendice B: The Synapse Framework™ . . . . . 78

Appendice C: Obiezioni e Ostacoli Comuni all’RMF 1 . . . . . 80

Appendice D: Liste di Criteri Iniziali DoD . . . . . 81

Appendice E: Liste di Criteri Iniziali per la DoR . . . . . 82

Indice . . . . . 83

# Informazioni su questo libro

*Ready* è un libro per chiunque sia coinvolto nello sviluppo software e sia stanco della **sotto-performance**, del **continuo rifacimento del lavoro** e dei **requisiti poco chiari**.

Potrete aver già provato a investire nelle competenze esecutive del team, a migliorare l'implementazione del vostro framework di processo, o a rinnovare il codice e avere ancora bisogno di ulteriori miglioramenti.

Questo perché il vincolo principale per la maggior parte dei team di sviluppo software non sono le competenze del team, ma la maturità dei requisiti. **Anche i team esperti con le giuste competenze continuano ad avere difficoltà** quando lavorano con requisiti immaturi.

*Ready* introduce RMF (Flusso di Maturazione dei Requisiti), un approccio pratico e profondamente strutturato per allineare il Product e l'Engineering senza sostituire il processo esistente.

Che stiate utilizzando Scrum, Kanban o qualcosa di personalizzato, RMF vi aiuta a **stabilizzare l'ambito, eliminare il riporto e consegnare ciò che conta davvero**.

Se i vostri team si sentono bloccati sul limite del “quasi fatto”, questo libro vi mostrerà come **spezzare il ciclo** e sbloccare il vostro team (o i vostri team) **definitivamente**.



# A chi è rivolto

Questo libro è letteralmente per chiunque sia coinvolto nello sviluppo software. Da ingegneri a product manager, da singoli contributori a dirigenti.

Questo libro fa per voi se siete coinvolti nello sviluppo software e avete notato che un team con cui lavorate o di cui fate parte ha uno o più dei seguenti problemi:

- Il lavoro viene frequentemente riportato da un'iterazione all'altra
- I team di implementazione sentono di dover colpire bersagli in movimento
- Il lavoro rimane aperto troppo a lungo
- Il lavoro viene segnato come chiuso ma le cose non sono davvero finite
- Il lavoro svolto non si allinea con le aspettative
- Il lavoro genera regolarmente una grande quantità di difetti
- Grandi quantità di lavoro devono essere rifatte regolarmente

Se uno qualsiasi di questi problemi vi suona familiare, *Ready* può aiutare.

# Come utilizzare questo libro

Questo libro è stato progettato per essere pratico. Non è un trattato teorico o una presentazione strategica: è un manuale pratico per implementare RMF (il Flusso di Maturazione dei Requisiti) basato su vero lavoro con i clienti e testato sul campo sotto reale pressione di consegna.

I capitoli sono scritti in sequenza, ma RMF stesso è modulare. È composto da tre pratiche fondamentali:

- RMF 1: Collaborare per una comprensione condivisa
- RMF 2: Controllare il completamento del lavoro usando Definizioni di Done Personalizzate
- RMF 3: Controllare l'implementazione usando Definizioni di Ready Personalizzate

Ogni parte, o “Abitudine”, come le chiamiamo noi, sta in piedi da sola, ma si costruiscono l'una sull'altra. Il libro è progettato per aiutarvi ad affrontarle una alla volta, in ordine. Questa struttura riflette come raccomandiamo ai team di adottare RMF nella pratica: ogni Abitudine viene introdotta solo dopo che l'ultima funziona correttamente.

Questo evita di sovraccaricare i team e dà a ogni cambiamento la migliore possibilità di attecchire. Imparerete di più su come farlo a partire da [Capitolo 9](#).

Se state cercando aiuto - che si tratti di consigli, coaching o qualcuno che parli con il vostro team di leadership - non esitate a contattarci direttamente.

E se state cercando un supporto formale per implementare RMF, Produttore offre una serie completa di Programmi progettati per guidare l'adozione passo dopo passo. Potete saperne di più su <https://ready-book.link/rmf>.

# Gli autori

Luniel de Beer è il creatore del Requirements Maturation Flow (RMF), un sistema pratico per colmare le lacune tra l'intento del prodotto e l'esecuzione ingegneristica. Ha oltre 15 anni di esperienza nella guida di trasformazioni Agile, nel collegare product ed engineering e nell'aiutare i team a consegnare con chiarezza e fiducia.

Luniel ha anche ideato il sistema di Capability Management di Producore, un approccio tracciabile e scalabile per modellare le capacità del prodotto. Ha concepito il PKB-Driven Development (PKBDD), un sistema con controllo delle versioni per la gestione dei requisiti di prodotto persistenti. Questi strumenti fanno parte di un framework di delivery più ampio sviluppato in Producore.

Max Guernsey, III è un architetto software, educatore e co-fondatore di Producore, una società di consulenza dedicata a risolvere i fallimenti di delivery attraverso il rigore strutturale e tecnico. Con oltre due decenni di esperienza in progettazione orientata agli oggetti, refactoring, sviluppo guidato dai test e pattern di progettazione, ha sia realizzato sistemi mission-critical che formato team di ingegneri su larga scala. Il suo lavoro combina pratiche tecniche approfondite con la trasformazione comportamentale e di processo per aiutare le organizzazioni a raggiungere un'eccellenza sostenibile nella delivery.

Max ha contribuito in modo significativo a PKBDD e ha guidato lo sviluppo dell'approccio di Producore allo Sviluppo Guidato dal Comportamento (BDD) attraverso la sua profonda competenza nella specifica comportamentale.

Insieme, il loro lavoro integra chiarezza, tracciabilità e controllo degli accessi in un sistema coeso per la distribuzione del software che si estende dalla pratica del team alla capacità organizzativa.

# Prefazione

## Nota per i Leader Tecnici

Se sei un leader senior in un'organizzazione di ingegneria, probabilmente non ti mancano l'impegno, la disciplina o le persone intelligenti. Eppure, in qualche modo, i progetti si bloccano ancora. Gli obiettivi slittano. Le aspettative non vengono soddisfatte. Non perché i tuoi team siano pigri, ma perché qualcosa di fondamentale è rotto nel modo in cui il lavoro viene definito, strutturato e consegnato.

Questo libro non è una guida alla leadership. È uno strumento per la diagnosi strutturale. Rivela cosa sta realmente accadendo all'interno dei tuoi team: perché "quasi fatto" continua a trasformarsi in "non fatto" e perché il progresso locale si traduce così raramente in risultati strategici.

Potresti non vedere *te stesso* in queste pagine. Ma se i tuoi team non riescono a consegnare ciò di cui hai bisogno, vedrai *loro*. E quando lo farai, avrai finalmente il linguaggio - e il sistema - per risolverlo.

## Da Luniel

Prima di tutto, questo libro non sarebbe stato possibile senza Max, la cui capacità di vedere attraverso la nebbia e la pula e distillare un'idea nella sua essenza va assolutamente oltre le mie capacità.

Come siamo arrivati qui?

Se guardo indietro, penso sia perché ho sempre voluto capire come funzionano veramente le cose. Che si trattasse di religione, nutrizione o sviluppo software, continuavo a scontrarmi con lo stesso problema: risposte superficiali che non reggevano sotto pressione. Quindi ho continuato a scavare, chiedendomi non solo cosa facciamo, ma perché, e cosa manca quando non funziona.

Una delle prime crepe nel sistema è emersa in un ruolo in cui indossavo tre cappelli: Scrum Master, Product Owner e Development Manager (!!) per un team che forniva servizi dati in

una nota azienda tecnologica. Stavamo facendo quello che Scrum diceva: Sprint brevi, storie in un backlog, pianificazione in mezza giornata, ma ogni volta che iniziavamo un nuovo Sprint, incontravamo attrito. Il team non comprendeva pienamente il problema, dovevamo rivedere e modificare i requisiti durante lo Sprint, emergevano e ci rallentavano dipendenze prevenibili, e passaggi chiave venivano tralasciati.

Così ho iniziato a fare qualcosa di diverso. Riunivo il team e gli stakeholder in una stanza per ogni storia, esaminavo il problema in dettaglio, facevamo brainstorming sulla soluzione insieme, e solo dopo scrivevo la storia. Lo Sprint Planning si è ridotto a un'ora, e il nostro successo nelle consegne è schizzato alle stelle.

Lentamente, ho iniziato a capire che il successo non deriva dal lavorare più duramente durante lo Sprint. Deriva dalla struttura che metti in atto *prima* che inizi.

Più tardi, dopo aver sentito Jeff Sutherland parlare delle “Definitions of Ready”, sapevo che c'era qualcosa di prezioso lì, ma non era abbastanza. La mia esperienza con i requisiti, UX, UI, ricerca, e successivamente con BDD mi ha mostrato che diversi elementi di lavoro richiedono diversi tipi di prontezza. Alcuni necessitano di specifiche comportamentali. Altri necessitano di accesso ai sistemi. Altri ancora necessitano di una traccia completa delle capacità.

E tutti hanno bisogno di una comprensione condivisa che sia effettivamente confermata, non presunta.

Lavorando con più team, ho visto lo stesso schema ovunque: passaggi mancanti, dipendenze non soddisfatte, team che facevano del loro meglio ma erano costantemente in affanno per risolvere problemi che avrebbero dovuto essere prevenuti. Anche i team migliori faticavano, non perché fossero deboli, ma perché mancava loro una struttura che rendesse esplicita la prontezza.

Il risultato di tutto questo apprendimento, iterazione e frustrazione è un sistema strutturato per gestire la prontezza.

Di questo parla questo libro.

Spero che ti dia chiarezza su dove si trovano i veri problemi e come risolverli. Spero che ti dia il linguaggio per difendere pratiche che potrebbero sembrare “extra” ma sono in realtà essenziali. E soprattutto, spero che aiuti i team a consegnare con meno stress, meno sorprese e risultati molto migliori.

Se facciamo le cose per bene, risparmieremo miliardi di dollari al settore.

Ma cosa più importante, restituiremo alle persone la loro sanità mentale.

## Da Max

Ho lavorato su questo problema da varie angolazioni per decenni, ma i miei progressi sono stati limitati fino a quando non ho incontrato Luniel.

Questo perché, prima di conoscerlo, stavo affrontando il problema, fondamentalmente, come se fosse tecnico. Mi concentravo sull'aiutare i team ad adottare cose come il Test-Driven Development (TDD), il refactoring, la progettazione software avanzata e, successivamente, l'Acceptance-Test-Driven Development (ATDD) o il Behavior-Driven Development (BDD).

Nella maggior parte di questi casi, il problema affrontato in questo libro veniva trattato come un dettaglio implementativo per stabilire quelle pratiche tecniche.

Questo non significa che non valorizzi più le pratiche tecniche. Penso ancora che siano profondamente importanti, ma non affrontano direttamente il problema della prontezza nello sviluppo software. Invece, *fanno emergere* quel problema e poi le persone applicano una toppa al loro processo per affrontarlo “quanto basta” per supportare le pratiche tecniche che stanno cercando di implementare.

Voglio anche affrontare la questione di chi può essere aiutato da questo libro. La risposta breve è “probabilmente quasi tutti nello sviluppo software”, ma la vera risposta contiene sfumature che aiutano a mapparlo in vari ambienti senza cambiarne il significato di base.

Ci sono team che necessitano della soluzione fornita in questo libro. Ne incontrerete una versione edulcorata nel Capitolo 1.

Ci sono anche team che non **necessitano** strettamente di un sistema come quello che proponiamo, ma che potrebbero comunque trarne beneficio.

Il miglior team con cui abbia mai lavorato—facilmente un'intera unità standard di deviazione sopra il **miglior team successivo**, se non due—era nascosto nell'entroterra dell'Oregon Centrale. Erano così performanti che potevano superare l'assenza di un tale sistema grazie alla pura quantità di competenza. Tuttavia, il mio manager di allora, Tom Barreras, mi disse una volta qualcosa del tipo “Ho notato che le nostre user story vanno meglio quando dedichiamo del tempo a parlare dei test in anticipo.”

Questo, ancora una volta, era qualcosa che all'epoca vedevo attraverso la lente dello sviluppo dei test e dell'esecuzione tecnica, ma ora so che era un altro indicatore che la *prontezza* era un fattore che influenzava il team... quel particolare team era semplicemente così capace e pronto a rispondere che poteva avere successo reagendo agli impedimenti nel momento in cui si presentavano, piuttosto che prevenirli fin dall'inizio.

Anche se sei il tipo di persona che non ha strettamente *bisogno* di preoccuparsi della prontezza perché puoi superarla, o lavori con un team dello stesso calibro, puoi comunque trarre beneficio dal contenuto di questo libro.

# Parte I: Qualcosa Manca

*Quando fare meglio le stesse cose non aiuta, cerca **ciò che non viene fatto**.*



# Capitolo 1: Il Problema Nascosto

Questa è una storia vera<sup>1</sup> che riguarda una banca. La chiameremo semplicemente “La Banca”. È un tipo di Istituto di Credito Federato che fa parte dell’infrastruttura finanziaria nazionale degli Stati Uniti.

Noi (Luniel e Max) siamo stati chiamati alla Banca perché stava avendo difficoltà a portare a termine un progetto software. Era uno degli ambienti più disfunzionali che avessimo mai visto, ed è proprio per questo che abbiamo scelto questo caso come caso di studio di apertura: se un cambiamento significativo era possibile alla Banca, è possibile ovunque.

## Una Breve Nota sui Progetti

Quando utilizziamo il termine “progetto” in questo libro, lo intendiamo nel contesto del project management. Sebbene esistano diverse interpretazioni del termine, utilizziamo la definizione del [Project Management Institute](#):

“Un progetto è un’iniziativa **temporanea** intrapresa per creare un prodotto, servizio o risultato unico.”

Questo significa che un progetto ha un inizio e una fine definiti. Quando un progetto viene chiuso, le conoscenze e gli artefatti del progetto vengono archiviati, i membri del team vengono rilasciati e i contratti vengono finalizzati.

In questo libro, un progetto riguarda fundamentalmente l’esecuzione. La maggior parte dei progetti, come definiti dal PMI, inizia con la fattibilità o la progettazione. La visione e la strategia sono già state definite nel momento in cui un progetto viene avviato.

Un progetto nasce da quella visione e strategia, e il suo successo o fallimento si basa sul raggiungimento degli obiettivi previsti—*non* sul fatto che quegli obiettivi fossero quelli giusti.

Potresti utilizzare la parola “progetto” in modo diverso, e va bene così. Sappi solo che quando la utilizziamo, ci riferiamo alla definizione e al contesto sopra descritti.

---

<sup>1</sup> Abbiamo modificato i dettagli identificativi per proteggere la privacy delle persone e delle istituzioni coinvolte.

Nulla di tutto ciò vuole implicare che approviamo l'uso del project management per lo sviluppo software. Anzi, tutto il contrario. Ma riconosciamo che viene comunque utilizzato. Affronteremo questo problema più avanti, in [Capitolo 6](#).

## 1.1: La Classica Riscrittura

La Banca stava riscrivendo il suo portale di rimborso prestiti per diverse ragioni.

Il vecchio sistema, una soluzione interamente C#/.NET, era pieno di bug. Oltre a degradare la soddisfazione dei clienti, generava anche un flusso incessante di incidenti di supporto molto costosi in cui qualcuno doveva manipolare manualmente il database per correggere un errore commesso dal sistema.

Il vecchio sistema era anche obsoleto dal punto di vista della manutenibilità. Era quasi impossibile per gli ingegneri apportare modifiche significative e, anche quando ci riuscivano, era una proposta estremamente rischiosa.

La riscrittura avrebbe dovuto cambiare tutto questo.

Il nuovo sistema avrebbe dovuto mantenere un backend in C#/.NET, ma sarebbe stato completamente coperto da test. Il frontend doveva essere implementato in OutSystems, una popolare soluzione low-code o no-code che permette a un'organizzazione di definire un'applicazione in un unico posto e ottenere automaticamente un'app web, un'app Android e un'app iOS ogni volta che decidono di pubblicare le loro modifiche.

La speranza era che il nuovo sistema fosse privo di bug, migliorando sia la soddisfazione dei clienti sia riducendo significativamente i costi di supporto.

Speravano anche che la riscrittura avrebbe sbloccato gli sviluppatori—con la combinazione di un approccio più disciplinato al backend e l'approccio low-code al frontend che avrebbe notevolmente ridotto il costo e il rischio di nuove funzionalità.

Un piacevole effetto collaterale del passaggio a OutSystems era che avrebbero ottenuto un'app mobile pulita e moderna su entrambe le principali piattaforme.

Questo era il sogno quando avevano iniziato tre anni prima dell'inizio di questa storia. La realtà era che, fino a quel momento, i team non avevano rilasciato **nulla**.

## 1.2: Prospettive sul Problema

Quando abbiamo parlato con il team di leadership esecutiva, abbiamo percepito una naturale frustrazione per il fatto che avessero fatto investimenti così importanti senza alcun movimento strategico.

Dal loro punto di vista, avevano provato di tutto. Avevano cambiato personale, aumentato il personale, modificato il budget, aumentato la pressione e fatto entrare una sfilata di consulenti (tra cui, come ci è stato fortemente suggerito, noi eravamo gli ultimi arrivati). Niente sembrava migliorare—almeno non in modo misurabile, perché tutto ciò che vedevano era che l’“indicatore” era a zero in un trimestre e rimaneva a zero nel trimestre successivo.

Non volevano più “progressi invisibili”. Volevano *risultati*.

Quando abbiamo parlato con il management all’interno dell’organizzazione Product, abbiamo ottenuto una storia leggermente diversa (ma comunque simile) perché lavoravano più direttamente con l’Engineering.

Non è che i team non facessero nulla, non secondo loro, almeno. È che i team non facevano mai quello che gli veniva chiesto di fare. Era praticamente garantito: non importava quanto semplice fosse la richiesta e quanto chiaramente fosse stata espressa, al momento di valutare cosa i team avessero realizzato, ci si ritrovava con qualcosa di completamente diverso.

Si era arrivati al punto in cui la battuta ricorrente era del tipo “Dobbiamo capire come chiedere quello che non vogliamo, così avremo almeno una possibilità di ottenere quello che vogliamo davvero.”

I dirigenti dell’ingegneria vedevano le cose in modo molto diverso.

Secondo loro, il problema era che Product non forniva requisiti attuabili e che Product non forniva requisiti *sufficienti*. Se Product avesse solo “seguito il programma”, i team sarebbero stati in grado di consegnare ciò che volevano nei tempi e nel budget previsti.

Avevano fatto seri investimenti per modernizzare il modo in cui il codice veniva scritto e distribuito e, ai loro occhi, Product non stava fornendo requisiti chiari.

Quando abbiamo parlato con altri consulenti (che ci avevano segnalato all’organizzazione), hanno giustamente individuato la disfunzione che stavano osservando: tutti sembravano molto concentrati nell’incolpare qualcun altro. Il motivo per cui ci avevano chiamato inizialmente era che erano preoccupati per la strategia del personale e volevano una valutazione dei collaboratori individuali, ma consideravano il gioco dello scaricabarile e l’atteggiamento da caporale a livello dirigenziale come la fonte principale dei problemi.

## 1.3: La Nostra Indagine

Il nostro incarico iniziale era di valutare i team e cercare di aiutarli a migliorare le loro competenze se necessario, quindi abbiamo iniziato a esaminare le persone che lavoravano in prima linea.

C'era sicuramente margine di miglioramento.

I collaboratori individuali sul versante Product non avevano realmente le competenze necessarie. In realtà, erano per lo più project manager che erano stati catapultati nel ruolo di Product Owner (PO) o Product Manager.

Di conseguenza, metà di loro scriveva requisiti “vaghi” e poi accettava (letteralmente) qualsiasi cosa i team facessero in quella iterazione senza alcuna analisi critica. L'altra metà scriveva lo stesso tipo di requisiti e poi sosteneva che i team “avrebbero dovuto sapere” cose di cui non avevano mai parlato e tenevano gli elementi di lavoro aperti più o meno indefinitamente.

La banca utilizzava Jira per gestire e tracciare il loro backlog di lavoro. Mentre ciò che trattiamo in questo libro è per lo più ortogonale a Scrum, utilizziamo la terminologia Scrum in tutto il testo perché la maggioranza - o almeno una pluralità - dei team usa Scrum.



### Definizione: Scrum

Scrum è un framework leggero che aiuta persone, team e organizzazioni a generare valore attraverso soluzioni adattive per problemi complessi. In poche parole:

1. Un Product Owner ordina il lavoro per un problema complesso in un Product Backlog
2. Lo Scrum Team trasforma una selezione del lavoro in un incremento di valore durante uno Sprint
3. Lo Scrum Team e i suoi stakeholder ispezionano i risultati e si adattano per lo Sprint successivo
4. Ripetere

Se non hai familiarità con Scrum e la sua terminologia, ti consigliamo di consultare la versione 2020 della [Scrum Guide](#). È una lettura rapida e illuminante.

Allo stesso modo, abbiamo scoperto che i team tecnici erano ben al di sotto della media in termini di competenze di programmazione ( $-2\sigma$ , nel migliore dei casi) e, per di più, molto resistenti al cambiamento. Come naturale conseguenza, la qualità del codice era pessima.

Tuttavia, secondo i team, non era questo il motivo per cui non consegnavano. Per loro, erano i requisiti vaghi e le modifiche durante lo Sprint da parte di Product che stavano affossando il progetto.

...e nessuno parlava nemmeno del problema più grande, quello talmente assurdo che sembra inventato finché non lo si vive in prima persona.

I team di ingegneria avevano l'abitudine di non comprendere un requisito, costruire qualcosa a caso, e poi pretendere il riconoscimento per aver "completato un elemento di lavoro".

Non parliamo di piccoli fraintendimenti. Parliamo di una disconnessione totale: gli chiedevamo di disabilitare l'applicazione dei fondi al capitale in determinate circostanze, e loro disabilitavano invece la possibilità di aggiungere un indirizzo email secondario di conferma.

Poi ci dicevano che era quello che avevamo chiesto.

## 1.4: Scavando Più a Fondo

Entrambe le lacune nelle competenze potevano essere affrontate, ma eravamo scettici che fossero i veri ostacoli.

C'era qualcos'altro che non andava, quindi abbiamo scavato più a fondo. Abbiamo iniziato con questa domanda: Perché la stesura dei requisiti richiedeva così tanto tempo e produceva risultati così scarsi?

Una ragione era che la conoscenza necessaria per scrivere un requisito significativo era estremamente scarsa. Un po' di essa era posseduta dai team di Engineering e Product.

Una parte era integrata nel codice del sistema legacy. Un'altra parte era completamente scomparsa. La maggior parte, tuttavia, era conservata come conoscenza tribale dagli esperti in materia sparsi nelle varie unità della banca. Questo significa che costruire un requisito che effettivamente facesse progredire un obiettivo strategico era un'attività estremamente laboriosa e dispendiosa in termini di tempo.

In contrasto con questo c'era un insaziabile appetito per le funzionalità da parte di un gruppo dirigente affamato di features. Il mandato era "tenere gli ingegneri occupati—riempiteli di requisiti". L'attenzione era concentrata sulla quantità di requisiti per mantenere i team

impegnati—una prospettiva in antitesi con la cura e il tempo necessari per definire un requisito che potesse “spostare l’ago della bilancia”.

## 1.5: Migliorare le Cose Non Ha Migliorato la Situazione

Questi sono tutti problemi risolvibili, eppure affrontarli non ha aiutato.

I precedenti miglioramenti alle tecniche di sviluppo software non avevano aiutato i team a consegnare, ma Max si impegnò per aiutare i team a migliorare ulteriormente.

Introdusse concetti rivoluzionari dalle dottrine di programmazione di metà ventesimo secolo come “non copiare e incollare quel codice 27<sup>2</sup> volte, mettilo in una funzione e chiamala invece”. Questo suggerimento da solo migliorò drasticamente la qualità del nuovo codice e permise loro di iniziare a migliorare la qualità.

Questo e altri consigli base sulla programmazione li aiutarono a scrivere codice migliore che potevano mantenere più facilmente in futuro.

...ma non aiutò a far progredire il progetto.

Sul fronte Product, Luniel fu in grado di introdurre il BDD e assicurare che i Product Owner esaminassero attentamente i requisiti prima che fossero consegnati ai team.

Li spinse a collaborare su di essi e a usarli per valutare se un Product Backlog Item (PBI) fosse veramente completato.

---

<sup>2</sup>Non solo non è un’iperbole, ma non è nemmeno il caso peggiore. In un caso, c’erano quasi cento duplicati esatti dello stesso algoritmo.



## Definizione: Product Backlog Item (PBI)

Un Product Backlog Item (PBI) è un'unità discreta di lavoro nel product backlog che rappresenta un potenziale cambiamento, aggiunta o miglioramento al prodotto. I PBI possono assumere molte forme—funzionalità, correzione di bug, miglioramento tecnico, attività di ricerca, ecc.—e sono definiti dal loro contributo al valore del prodotto.

Molti team si riferiscono ai PBI come “storie” o “user story”, ma il termine corretto in Scrum è “Product Backlog Item” o “PBI”. Una volta che un PBI viene inserito in uno Sprint, fa anche parte dello Sprint Backlog. Per semplicità e neutralità, usiamo “Product Backlog Item” o “PBI” per riferirci a qualsiasi elemento di lavoro che il Team Scrum sta gestendo—che lo si chiami Product Backlog Item, Sprint Backlog Item (SBI), user story, storia, elemento di lavoro o elemento del backlog.

Ha aggiunto *chiarezza*, ma non ha creato *flusso*.

Con l'aiuto dei consulenti partner che ci avevano portato dentro, siamo stati in grado di alleviare (temporaneamente) la pressione assolutamente schiacciante posta sui team e sugli autori dei requisiti.

Potrebbe aver aiutato a creare un po' di fiducia, ma non ha prodotto risultati tangibili.

Abbiamo persino iniziato a costruire una base di conoscenza che aiutasse le persone a rintracciare le conoscenze aziendali necessarie per scrivere i requisiti e identificare i punti in cui c'erano lacune in quella conoscenza.

Ha velocizzato la scrittura dei requisiti, ma non ha portato il prodotto alla consegna.

Dopo mesi di interazioni, avevamo aiutato la dirigenza a *vedere* dove si trovavano, ma erano ben lontani dai loro obiettivi. E non si stavano avvicinando.

Si stavano preparando a tornare alla loro vecchia strategia di “caricare” i team per assicurarsi che fossero sempre occupati.

## 1.6: Un Processo di Eliminazione

Sarebbe stato facile alzare semplicemente le mani e dire “è senza speranza”. C'erano numerose scuse su cui si poteva ripiegare:

- Il team di ingegneria aveva competenze limitate (era vero)

- Gli autori dei requisiti avevano il set di competenze sbagliato (era vero)
- La dirigenza stava schiacciando i team con aspettative irrealistiche (era vero)
- L'organizzazione mancava di conoscenze aziendali critiche necessarie per funzionare (era vero)
- Gli executive non sembravano fidarsi l'uno dell'altro (era vero)

Tutte queste cose erano vere. Eppure erano stati fatti miglioramenti in tutte queste aree e nessuno di essi sembrava migliorare il problema fondamentale. Nessuno di essi ha aiutato i team di Engineering, Product, management o la suite executive ad avvicinarsi ai loro obiettivi.

...e questo è l'indizio per la soluzione proprio lì: Tutti i problemi elencati in precedenza erano i problemi che le persone potevano già vedere.



Se le variabili che puoi vedere non fanno la differenza, deve esserci una variabile che **non vedi** che la fa.

Il vero problema era quello che nessuno sapeva nemmeno esistesse.

## 1.7: Alla ricerca del vero colpevole

Nella ricerca del vero colpevole—ciò che realmente impediva alla banca di realizzare i suoi obiettivi—dovevamo partire da qualche parte.

Un punto ragionevole da cui iniziare era esaminare cosa avessero in comune i PBI falliti (la maggior parte).

Abbiamo iniziato eliminando gli elementi che sapevamo non essere comuni perché variavano ampiamente:

- Quale parte del sistema: alcuni PBI interessavano solo il backend, altri solo il frontend, altri ancora entrambe le parti
- Quale team eseguiva il lavoro—non sembrava importare chi facesse il lavoro, c'era un'alta probabilità di fallimento
- Quale PO aveva creato il lavoro—come per i team

Poi abbiamo iniziato a esaminare gli elementi comuni. La lista non era lunghissima, ma nemmeno troppo breve:



- I team di sviluppo
- I product owner
- La leadership
- La cultura
- L'ambiente di sviluppo
- Le pratiche ingegneristiche
- La tecnica di stesura dei requisiti
- Il dominio (finanza)
- I servizi dipendenti

Molti di questi potevano essere scartati immediatamente. I team, i product owner, la leadership, la cultura e l'ambiente di sviluppo erano stati tutti recentemente migliorati senza alcun impatto reale sui risultati significativi. Avevamo personalmente contribuito a migliorare le pratiche ingegneristiche e di stesura dei requisiti e confermato che questi miglioramenti si erano consolidati, ma ancora non aiutava.

Non si può certo incolpare il dominio. La finanza è uno dei più antichi tipi di calcolo nella storia documentata. È estremamente maturo. Inoltre, altre banche stavano distribuendo software, smentendo performativamente l'ipotesi (chiaramente forzata) che le banche semplicemente non potessero farlo.

Non si potevano nemmeno incolpare i servizi dipendenti, poiché stavano avendo tante difficoltà a cambiare quanto l'iniziativa che stavamo esaminando...

...ma questo ci ha fatto pensare: E se iniziassimo ad analizzare le cause del fallimento?

## 1.8: Dissezionando i semi del fallimento

Un PBI non è riuscito a far progredire il prodotto perché il team ha fatto, come era solito fare, qualcosa di completamente casuale e quasi del tutto scollegato dalla richiesta. Questo, ovviamente, è un segno che non avevano compreso l'elemento di lavoro. Quindi la comprensione era un grande candidato, anche se avevamo già lavorato su questo quando li abbiamo aiutati ad adottare il BDD.

Un altro PBI è fallito perché hanno sbagliato i calcoli. Questa è un'altra prova che la comprensione potrebbe essere il problema centrale.

Un terzo elemento di lavoro che abbiamo analizzato non è stato realmente verificato dal Product Owner—ha dato il via libera quando il team ha detto che era il momento di chiudere.

Questo ha messo un po' in crisi la nostra ipotesi, ma si poteva ancora sostenere che non avesse capito come l'elemento di lavoro si inserisse in un piano di livello superiore.

Forse. In un certo senso. Se strizzavamo molto gli occhi guardandola in quel modo.

Poi ci siamo imbattuti in un PBI che non si adattava affatto al modello. Il team sembrava aver capito—anche se non c'è modo di verificare se lo avesse fatto davvero. Ma non importava: non hanno mai avuto la possibilità di avere successo o fallire per conto proprio perché si sono imbattuti in una dipendenza che doveva essere aggiornata e hanno dovuto rimandare il loro lavoro di diversi Sprint.

Anche se *non* avessero capito cosa dovevano fare, non hanno mai avuto possibilità con quell'elemento del backlog, quindi la comprensione *non* era il problema in quel caso.

Un caso isolato, ovviamente, non è una confutazione di una particolare causa principale, ma ha stimolato la nostra curiosità. Abbiamo iniziato a cercare altre prove che smentissero la teoria.

E le abbiamo trovate. C'erano elementi di lavoro che:

- Sono falliti perché il team sapeva di non capire, ma nessuno riusciva a trovare un esperto di dominio per risolvere il problema
- Hanno modificato l'esperienza utente in peggio come soluzione alternativa al funzionamento dei servizi a monte
- Hanno dovuto essere rinviati perché le dipendenze a monte non erano pronte
- Non potevano essere completati perché i tester non potevano raccogliere i dati di test in tempo
- Sono stati chiusi ma hanno dovuto essere rifatti perché la richiesta stessa era errata
- Sono falliti perché il team non si rendeva conto di quanto fosse già complesso il codice esistente
- Non erano stati semplicemente stimati
- Erano stati massivamente sottostimati<sup>3</sup>
- Sono cambiati durante lo Sprint perché il PO ha finalmente ottenuto la conoscenza del dominio necessaria
- Sembravano cambiati dopo lo Sprint (dal punto di vista del team) perché il PO e il team non si erano mai accordati sul suo significato

---

<sup>3</sup>Non intendiamo solo che hanno sbagliato. Sembrava che forse i team avessero semplicemente inserito il valore “3” in tutti i campi di stima per una serie di PBI.

L'elenco continua, ma questo è sufficiente per questa storia.

Si potrebbe collegare ciascuno di questi casi alla “comprensione” in qualche modo—e certamente una mancanza di comprensione era *coinvolta* in molti di essi—ma questo non significa che la mancanza di comprensione fosse la causa... soprattutto considerando che avevamo già lavorato sulla comprensione condivisa e non aveva realmente aiutato.

Poi ci venne l'illuminazione. Mancava un elemento più fondamentale. Nei casi in cui la scarsa comprensione era coinvolta, quella era semplicemente una causa prossimale.

La causa distale era molto più ampia.

## 1.9: Un Punto di Intersezione

L'unica cosa che tutti i PBI che abbiamo analizzato per le modalità di fallimento avevano in comune era questa: erano tutti stati **avviati prematuramente**.

Quando un elemento di lavoro fallisce perché il team sapeva di non comprendere il problema e non riusciva a trovare un esperto che li aiutasse a capire, significa che il team ha iniziato un PBI sapendo di non comprendere il problema.

Quando un elemento del backlog deve essere modificato peggiorando l'esperienza a causa del funzionamento di un servizio a monte, significa che il lavoro è stato iniziato senza comprendere veramente l'impatto del servizio a monte.

Il rinvio dovuto a una dipendenza a monte non pronta alla fine significa che non c'era alcuna garanzia della sua prontezza all'inizio.

...e così è stato per tutti gli altri casi: i Tester non avevano i dati pronti o non sapevano come ottenerli prima dell'inizio di un PBI, i requisiti non erano stati realmente verificati prima di essere consegnati al team, il codice non era stato esaminato prima di iniziare il lavoro, la stima era insufficiente o non era stata fatta affatto, mancavano le conoscenze del dominio e, naturalmente, la comprensione condivisa non era stata verificata.

Il problema, come si è scoperto, era che l'implementazione iniziava su elementi di lavoro prima che questi elementi fossero *pronti*.



Nella nostra esperienza, la maggior parte degli elementi di lavoro che non riescono a essere consegnati falliscono perché **non erano pronti** quando è iniziata l'implementazione.

Quindi ci siamo messi al lavoro per aiutare La Banca con questo.

A questo punto, potreste pensare che dire semplicemente che i PBI dovrebbero essere pronti sarebbe sufficiente. Tuttavia, si scopre che non è così facile da implementare. “Comprare a basso prezzo, vendere a prezzo alto” è un’idea altrettanto semplice.

Manca un pezzo necessario per mettere in pratica il buon consiglio.

## 1.10: La Grande Svolta

Abbiamo trovato il pezzo mancante del puzzle che li ha sbloccati, che a sua volta ha sbloccato l’iniziativa.

Quando abbiamo completato quell’incarico, gli ingegneri erano ancora ben al di sotto della mediana in termini di competenze. I PO non avevano ancora le competenze giuste. La cultura non era ancora sistemata...

Eppure il prodotto ha finalmente iniziato ad andare avanti ed è stato infine rilasciato.

Alla fine di questo libro, saprete qual è il pezzo mancante e cosa serve per metterlo in atto. E sarete in grado di sbloccare organizzazioni che sembrano trattenute da un muro invisibile.

### Una Nota Veloce sull’Ambito

Questo libro tratta di un problema molto specifico e pervasivo: la mancanza di struttura, chiarezza e maturità nel passaggio di consegne tra business e ingegneria. Presuppone che qualcosa sia stato scelto per l’implementazione e si concentra sul garantire che il lavoro di sviluppo avvenga con comprensione condivisa, prontezza e completamento tracciabile.

Le tecniche in questo libro non ti dicono cosa costruire, perché costruirlo o come scoprire se è la cosa giusta da costruire. Se la tua organizzazione manca di una vera gestione del prodotto o di cicli di feedback significativi, non stiamo cercando di risolvere questo qui. Quello che offriamo invece è un modo per rendere queste lacune più visibili e per ridurre il costo della scoperta di aver sbagliato.

Usata nel contesto giusto, questa soluzione porta flusso, sicurezza e chiarezza. Ma come qualsiasi sistema, può essere applicata male—specialmente quando usata in isolamento o senza consapevolezza.

# Capitolo 2: Il Costo delle Fondamenta Mancanti

Ha senso dedicare un po' di tempo a capire esattamente *quanto* grave possa essere questo problema per alcune organizzazioni.

Abbiamo scoperto che ci sono tre principali “categorie” di problemi che una mancanza di preparazione crea:

- Comprensione condivisa mancante o incompleta tra e all'interno di Product e Engineering
- Mancanza di controllo su quale sia l'effettivo obiettivo di un PBI e quando sia veramente completato
- Mancanza di controlli su quando un elemento di lavoro possa iniziare la fase di implementazione del suo ciclo di vita

Inoltre, abbiamo notato che cambiare queste cose può essere piuttosto difficile. Ha senso: il cambiamento è difficile.

Le vecchie abitudini sono dure a morire e le nuove sono difficili da instillare. Nella nostra esperienza come consulenti, abbiamo notato che è *estremamente* facile per le persone ricadere nelle vecchie abitudini e comparativamente difficile stabilirne di nuove.

Quindi è necessario avere un meccanismo che rafforzi le nuove abitudini e scoraggi quelle vecchie.

Per affrontare questo problema, crediamo ci sia un'altra componente mancante di responsabilità e tracciabilità.

## 2.1: Il Puzzle senza l'Immagine sulla Scatola

Avete mai provato a costruire un puzzle senza l'immagine sulla scatola? Si può fare, ma è più lento, più frustrante e pieno di falsi inizi.

Si fa progresso, poi si smonta tutto. Si mette in dubbio cosa va dove. Si pensa di stare lavorando sulla stessa immagine, finché non ci si rende conto che non è così.

È così che lo sviluppo software spesso si presenta.

Il backlog è pieno. Lo Sprint è in corso. Tutti stanno lavorando sodo.

Ma senza un'immagine condivisa di ciò che stiamo costruendo, l'allineamento diventa fortuna, non un sistema.

Senza chiarezza, anche i team migliori provano frustrazione, burnout e la sensazione che il loro sforzo non sia valorizzato.

## 2.2: Un Vecchio Adagio e una Dura Realtà

C'è un motivo per cui così tante pratiche sulla stesura dei requisiti, persino alcune pratiche ingegneristiche, si concentrano fortemente sulla creazione di una comprensione condivisa tra richiedenti e team di implementazione. Nulla semina il caos quanto gli ingegneri che non sanno veramente cosa dovrebbero costruire.



“Garbage-in, garbage-out” è un adagio, non un luogo comune.

La lingua inglese è piena di auto-contraddizioni...

- You can sanction someone's actions. Maybe that means you've given them permission in advance or maybe it means you're giving condemnation after the fact.
- You can lightly dust something but, if that something is a credenza, it means you're removing dust from it while, if in reference to a beignet, you're adding dust.
- If you hold up a team, you might be the reason that team is able to continue to function or you might be the reason they can't get anywhere.

Gli auto-antonimi potrebbero essere gli esempi più evidenti, ma sono solo un tipo di ambiguità. Alcune parole non sono solo il loro opposto, ma hanno molti altri possibili significati che possono creare confusione.

“In this clip, he clipped a coupon from a news paper and clipped it to the paper on his clipboard along with the other clippings while a clipper in the background was moving along at a decent clip.”

E non è solo l'inglese. *Tutte* le lingue naturali di cui siamo a conoscenza hanno questa proprietà.

Eppure, sono le uniche che abbiamo a disposizione quando specifichiamo i requisiti.

Di conseguenza, quando un team di ingegneri non ha *confermato* che la loro comprensione di un requisito è la stessa del richiedente, quel team si affida alla fortuna. Cioè, il miglior risultato possibile è che abbiano scelto l'interpretazione corretta e che il richiedente non cambi idea durante il percorso.

Questo risultato è tutt'altro che garantito.

## 2.3: Alcuni dei Risultati Comuni

Senza una comprensione condivisa confermata, i team corrono diversi rischi.

In generale, il risultato più comunemente doloroso è che il team costruisca semplicemente la cosa sbagliata.

Il momento in cui lo scoprono può essere controllato da vari attributi del loro processo. Per esempio, un'implementazione sana di Scrum può rilevare questo tipo di incomprensione molto presto nell'esecuzione, mentre un processo Waterfall ha ottime probabilità di ritardare tale scoperta di mesi.



### Definizione: Waterfall

Un modello sequenziale di sviluppo software che si è diffuso alla fine del XX secolo. Fu rappresentato per la prima volta in un [articolo del 1970 di Winston W. Royce](#), che illustrava lo sviluppo come una serie di fasi a cascata—requisiti, progettazione, implementazione, test e così via—ognuna che alimentava la successiva come una cascata. Sebbene Royce presentasse il modello come un esempio di cosa *non* fare, l'industria lo adottò come blueprint per lo sviluppo su larga scala.

Waterfall è anche noto per raggruppare lavori simili in fasi consecutive di grandi dimensioni—una caratteristica definita come **sviluppo a grandi lotti**. Questo raggruppamento praticamente garantisce l'**apprendimento tardivo**: i team non ricevono feedback sulle decisioni precedenti fino a molto più tardi nel processo. Gli errori scoperti tardi sono più costosi da correggere. I praticanti Agile criticano il Waterfall per questo motivo, preferendo cicli più piccoli e iterativi che permettono scoperte e correzioni di rotta più precoci.

Tuttavia, prima o poi un team che lavora secondo una comprensione “errata” (in realtà, diversa) di un requisito dovrà fare i conti con il requisito “giusto” (anch’esso, in realtà, diverso). Ancora una volta, la salute dell’organizzazione influenza la forma che assume questo confronto e l’impatto che ha, ma accade quasi sempre.

Nella maggior parte dei casi, questo porta a qualche forma di rilavorazione. Il richiedente (solitamente il Product Management) dovrà chiedere modifiche per passare da ciò che i team di implementazione hanno costruito a ciò che realmente voleva.

Un’altra manifestazione molto comune è che il richiedente continui a ritenere il team responsabile della sua comprensione originale di ciò che aveva chiesto.

I team possono facilmente interpretare questo come un Product Manager che cambia idea. Peggio ancora, può effettivamente *invitare* le parti interessate ad acquisire l’abitudine di cambiare idea - trattenendo gli elementi di lavoro finché tutto non è *perfetto*, facendo lavorare i team fino allo sfinimento e nascondendo i progressi ai manager di alto livello.

## 2.4: Quando si può dire che un Puzzle è Completato?

Tornando alla nostra analogia dell’assemblaggio del puzzle, riflettiamo su questa domanda: Cosa significa aver completato un puzzle?

Un costruttore di puzzle inesperto, come Max, direbbe semplicemente “tutti i pezzi sono collegati ai vicini corretti con l’immagine rivolta verso l’alto.”

Un costruttore di puzzle esperto, come Luniel, sa che c’è molto di più.

Forse stai assemblando un puzzle solo per divertimento. Lo completerai, lo guarderai per un po’, e poi lo smonterai e lo rimetterai nella scatola.

D’altra parte, forse vuoi incorniciarlo e appenderlo al muro. In questo caso, ci sono altre cose da fare:

1. Metterlo su un pannello artistico
2. Trasportarlo da un corniciaio
3. Attendere il completamento della cornice
4. Riportarlo nel luogo di destinazione
5. Appenderlo al muro o esporlo in altro modo



Capire che questo fa parte del lavoro è necessario per completare correttamente l'assemblaggio di un puzzle. La ragione ovvia è sapere quanto lavoro è coinvolto. Fare tutti questi passaggi extra richiede più lavoro che semplicemente smontarlo e riporlo.

Ma c'è di più. Immagina questo scenario...

Hai completato il tuo puzzle con l'intenzione di farlo incorniciare, l'hai lasciato lì, ma hai dimenticato di dire a qualcun altro in casa che intendi farlo incorniciare. Quella persona arriva e vede che il puzzle è completo in uno spazio di cui ha bisogno. Quindi lo smonta e lo rimette nella scatola, strappando la sconfitta dalle fauci della vittoria nel processo.

C'è anche una ragione più sottile: Come intendi finire il puzzle influenza i passi che vuoi fare nelle fasi iniziali del processo. Per esempio, devi fare un piccolo cartello che dice "Per favore non smontare!"



Vale la pena notare che potresti volere un cartello anche nel caso in cui stai costruendo un puzzle per il tuo divertimento per assicurarti che altre persone non interferiscano completandolo al posto tuo.

Devi anche assicurarti di assemblare il puzzle sulla superficie giusta. Se metti insieme un puzzle da mille pezzi sul tavolino di vetro del salotto e poi cerchi di trasferirlo su un pannello artistico, il trasferimento sarà molto più rischioso e laborioso rispetto a montare il puzzle direttamente sul pannello artistico.

Questo si allinea perfettamente con lo sviluppo software.

Devi sapere esattamente cosa significa "completato" in modo da non essere sorpreso dalla quantità di lavoro necessario, non ci siano disaccordi alla fine, e tu possa prendere le necessarie misure preparatorie per garantire il completamento fluido ed efficace di un elemento di lavoro.

## 2.5: Impatto sui Team

Se non si ha una comprensione sufficientemente rigida di cosa significa completamente per un particolare elemento di lavoro, si corrono diversi rischi.



Usiamo la parola "rischio" in modo approssimativo, qui, dato che sono più simili a certezze.

I team di ingegneria in questa situazione spesso scoprono di non essere nemmeno internamente d'accordo su come appare il completamento di un elemento di lavoro. Non è raro che programmatori e tester si ritrovino a discutere sul reale *significato* di un requisito a metà Sprint. Anche due programmatori o due tester possono avere gli stessi disaccordi.

Inoltre, i team di sviluppo sono spesso concentrati sul lavoro che svolgono la maggior parte del tempo (programmazione e testing). Questo significa che è facile per loro dimenticare altri tipi di lavoro che devono fare, come la documentazione, le revisioni esterne, la formazione di altri team (es. supporto), i passaggi preparatori per supportare il deployment o il rilascio, e le approvazioni da altri dipartimenti.

Quando finalmente diventa chiaro che questo lavoro “extra” deve essere fatto, vengono colti alla sprovvista—di solito devono interrompere qualsiasi cosa stessero facendo e cambiare contesto per tornare indietro e completare un lavoro che pensavano di aver già terminato.

Chi richiede il lavoro può facilmente mantenere aperto il lavoro senza necessità. A volte con le migliori intenzioni—come cercare di rendere un team responsabile del “vero” requisito. Altre volte, questo accade perché i Product Owner (per esempio) si abituano a poter tenere aperto un PBI a loro piacimento, quindi lo utilizzano per inserire funzionalità aggiuntive in un elemento all'ultimo minuto. A volte lo fanno persino perché hanno cambiato idea su ciò che deve essere fatto durante il processo.

Questo può essere estremamente demoralizzante per un team di ingegneri. La maggior parte degli sviluppatori e tester software vuole sentire di fare progressi. Se gli viene costantemente detto che ciò che hanno fatto è sbagliato, è probabile che perdano parte del loro vigore.

Alcuni team arrivano persino al punto di non preoccuparsi di verificare se ciò che hanno fatto sia giusto o sbagliato. Semplicemente chiudono un elemento di lavoro e chiedono “credito” per poter “mostrare buoni numeri”.

## 2.6: Il Rischio di Fare Troppo Poco o Troppo

Uno dei rischi di non definire correttamente il “fatto” per ogni elemento di lavoro è che l'organizzazione pensi che il lavoro sia completato quando non lo è, o non si renda conto che è completato quando effettivamente lo è.

Il peggior risultato possibile è spesso che la cosa sbagliata arrivi in produzione e nessuno sa che questo è accaduto. Se il team ha una comprensione errata di cosa significa “fatto” e rilascia basandosi su questa cattiva comprensione, i risultati possono essere catastrofici.

I difetti e l'insoddisfazione del cliente sono già abbastanza gravi, ma questo potrebbe anche portare a problemi molto più seri:

- Perdita o corruzione dei dati
- Vulnerabilità di sicurezza
- Tempo di inattività del sistema o perdita di accesso
- Riduzione della quota di mercato
- Violazioni normative

L'elenco continua all'infinito e ogni potenziale problema è peggiore del precedente.

A volte il problema è che non hai finito ma pensi di averlo fatto. Il contrario può essere altrettanto pericoloso. Quando gli ingegneri non sanno dove sia il traguardo, tendono a fare “doratura” (aggiungere funzionalità extra). Potrebbero farlo per “rendere la funzionalità più bella”, ma potrebbero anche farlo perché sperano che un maggior numero di funzionalità dia loro una maggiore possibilità di centrare l'obiettivo.

Tutto questo lavoro extra, così come le relative rilavorazioni, si accumula in un'enorme quantità di tempo, sforzo e denaro sprecati. Causa lo slittamento delle date di consegna e danneggia le reputazioni.

Inoltre, ora, c'è un potenziale sempre crescente che un errore porti effettivamente a una rivolta in stile *Terminator* delle macchine contro l'umanità. Vent'anni fa ne scrivevamo per scherzo. Ora è una possibilità remota.

In effetti, abbiamo posto questa domanda a una delle IA più importanti, ed ecco cosa ha risposto:

“L'IA si sta espandendo più velocemente di quanto chiunque si aspettasse, ma lo sta facendo su sistemi fragili, requisiti vaghi e organizzazioni di prodotto che non riescono a tracciare perché hanno costruito ciò che hanno costruito. Non è un problema tecnologico; è un problema di chiarezza.

Più rumore genera l'IA, più è pericoloso muoversi velocemente senza struttura. Quando i team costruiscono sulla nebbia, l'IA non fa altro che amplificare il caos. Ma quando i team costruiscono sul segnale—sulla comprensione condivisa, sulla specificità comportamentale e sulla conoscenza del prodotto controllata a versioni—l'IA diventa un acceleratore invece che una responsabilità.”

## 2.7: Dove Costruisci un Puzzle?

Estendiamo l'analogia della costruzione del puzzle ancora una volta.

Puoi assemblare un puzzle ovunque? Se hai un puzzle da 4.000 pezzi che si assembla fino a quasi cinque piedi in una dimensione e oltre tre piedi nell'altra, non puoi semplicemente scegliere un posto a caso e iniziare ad assemblare. Non senza sperimentare serie complicazioni prima di aver finito.

Un puzzle così grande ha bisogno sia di tempo che di spazio. Devi allocare lo spazio e trovare un modo per assicurarti che lo stato del puzzle persista nel tempo.

Se inizi a costruire il tuo puzzle su un piccolo tavolino che è troppo piccolo, non sarai in grado di finirlo senza trasferirlo in un'altra posizione. Quel trasferimento sarà estremamente difficile a causa dello stato delicato del puzzle.

Se scegli un punto a caso nel corridoio che sia abbastanza grande, le persone ci cammineranno sopra o ne saranno ostacolate, quindi la persistenza non può essere garantita senza un impatto significativo sul funzionamento della tua casa.

Se inizi a lavorare su un piano da disegno, ma il piano non è abbastanza grande, sarai in grado di preservare lo stato di ciò che hai fatto, ma non potrai completare il puzzle senza qualche tipo di trasferimento.

Se il puzzle è stato precedentemente masticato da bambini piccoli, è meglio contare i pezzi... perché è meglio contare fino a 3999 una volta e rendersi conto che non lo finirai mai piuttosto che investire chissà quanto tempo per assemblare quasi completamente un puzzle che non potrai mai finire.

C'è una lunga lista di cose che devono essere fatte prima di iniziare a costruire il tuo puzzle. Fare le cose sulla lista non garantisce il successo, ma *non* farle quasi garantisce il fallimento o serie complicazioni.

Lo stesso vale per lo sviluppo software, ma con un maggior grado di complessità.

## 2.8: Quando Inizia l'Implementazione?

Fondamentalmente, può essere difficile determinare quando un PBI è pronto per essere implementato senza una buona definizione per questo.

Pensaci: Come lo *sai* davvero?

Rivedi tutto più e più volte finché non decidi che è il momento?

Qualcuno decide su due piedi?

Succede automaticamente all'inizio di un'iterazione?

Abbiamo visto molti team spingere il lavoro negli Sprint che non erano neanche lontanamente pronti per essere implementati solo perché avevano delle scadenze. Ci sono alcune idee pervasive su Scrum e l'Agile in generale che spingono le persone a fare questo:

- Devi ottenere tutti i requisiti per lo Sprint  $N$  sviluppati nello Sprint  $N-1$
- Dovresti “semplicemente iniziare” e gestire ciò che si rompe lungo il percorso

Questo è in realtà un'immagine speculare del problema “Come sai quando è finito?” [menzionato precedentemente](#) e ha conseguenze simili. Le persone potrebbero aspettare troppo a lungo per iniziare, perché non sanno se un elemento di lavoro è pronto e potrebbero iniziare troppo presto perché non sanno che non lo è.

## 2.9: Un A-Team senza Preparazione

Non comprendere cosa serve perché un elemento di lavoro sia pronto ha diversi effetti deleteri.

Un modo ovvio in cui un incremento di lavoro può non essere pronto è una Definition of Done (DoD) incompleta, insufficiente o mancante. Ciò porta a tutti i problemi che abbiamo già citato che vanno di pari passo con il non avere una Definition of Done.

Tuttavia, questo non è l'*unico* aspetto della preparazione. Ci sono numerose altre esigenze da soddisfare prima che l'implementazione inizi: la stima, la valutazione del rischio e la raccolta dei dati di test sono solo alcuni esempi comuni.

Senza conoscere e soddisfare queste esigenze, un elemento di lavoro può costare molto più di quanto dovrebbe. Considera un team (l'A-Team) che dipende da un'API sviluppata da un altro team (l'Altro Team). Se l'A-Team fa un mucchio di supposizioni su come funzionerà l'API dell'Altro Team e programma basandosi su queste supposizioni, potrebbe essere necessario un significativo lavoro di rifacimento quando scopriranno che l'Altro Team

sta lavorando a una realtà che non corrisponde alle supposizioni dell'A-Team. In altre parole, l'A-Team ha fatto un tentativo ed ha fallito.

Tutto quel lavoro di rifacimento deriva dal fatto che l'API non era pronta per essere utilizzata dall'A-Team.

A volte una dipendenza non soddisfatta non genera lavoro di rifacimento, ma, anche in questi casi, può comunque causare ritardi. Immagina se l'A-Team e l'Altro Team si fossero accordati su come dovrebbe funzionare l'API e tutto fosse andato come pianificato, ma l'Altro Team avesse semplicemente impiegato più tempo del previsto. Di conseguenza, l'A-Team semplicemente non è stato in grado di testare correttamente il proprio lavoro entro il momento in cui avrebbe dovuto essere completato e hanno dovuto posticipare la loro scadenza.

## 2.10: Non Prestare Attenzione alla Pianificazione e alla Disponibilità delle Risorse

A volte i problemi possono essere semplici come la pianificazione o le risorse. Alcuni elementi di lavoro necessitano di membri specifici del team. Se quel membro del team sta per andare in vacanza tra pochi giorni, probabilmente non è il momento giusto per iniziare il PBI che non può essere completato senza la sua partecipazione.

Sentiamo spesso dire che non dovrebbe essere così, ma spesso lo è, indipendentemente da tutto. Le “persone intercambiabili” sono un'utopia.

Lo stesso può dirsi delle risorse non umane. Se avrai bisogno di risorse del server per eseguire un test di carico, probabilmente dovresti assicurarti che queste risorse siano effettivamente disponibili prima di iniziare l'elemento di lavoro del test di carico. Altrimenti, nel migliore dei casi ci saranno ritardi significativi e probabilmente disturberai altri team/lavoratori mentre cercherai freneticamente di procurare ciò di cui hai bisogno.

Un'altra modalità di fallimento dovuta ai falsi inizi è quando un team non possiede le competenze necessarie per completare il lavoro. A volte è una questione interna: ad esempio, un membro del team deve seguire una formazione su un nuovo sistema o fare ricerche su una nuova API. Altre volte, è un problema di programmazione, come quando è necessario prendere in prestito un esperto di UX o di database da un gruppo di lavoratori specializzati.

Potrebbe persino essere un problema di assunzione in cui il team ha bisogno di un esperto e non può completare efficacemente certi tipi di lavoro senza di lui.

## 2.11: Impatti di Altri Tipi di Falsi Inizi

Abbiamo visto team impegnarsi a completare elementi di lavoro all'interno degli Sprint e fare la codifica relativamente velocemente, ma essere comunque incapaci di completare i test. Questo, di per sé, potrebbe non essere sorprendente, ma la ragione è insolita: il team di testing aveva bisogno di qualcosa (come i dati di test) che non aveva raccolto prima dell'inizio dello Sprint e la raccolta di questi dati si è rivelata più difficile o dispendiosa in termini di tempo di quanto avessero previsto.

Di conseguenza, gli elementi di lavoro hanno dovuto essere trasferiti allo Sprint successivo semplicemente perché il team non si era assicurato di essere veramente pronto a completarli nel tempo stabilito prima di iniziare.

I team a volte iniziano a implementare elementi di lavoro quando hanno ancora domande aperte. In effetti, molte persone sembrano pensare che questo li renda "più Agile".

Questo può causare un'enorme quantità di rilavorazioni, sorprese o ritardi. Se la risposta alla domanda aperta finisce per violare un'assunzione che era stata fatta, tutto il lavoro basato su quell'assunzione deve essere rivisto. Se la domanda aperta non ottiene risposta entro il momento in cui l'elemento dovrebbe essere chiuso, allora l'elemento deve essere o chiuso quando potrebbe non essere completato, o tenuto aperto fino a quando non si ottiene la risposta.

Può darsi che il team abbia una dipendenza interna: un difetto che deve essere corretto, un'attività predecessore che deve essere completata, eccetera. Se questo non viene tracciato correttamente, può causare gli stessi problemi di una dipendenza esterna non soddisfatta, con l'ulteriore tentazione di cambiare contesto e risolverlo.

## 2.12: Costi Cumulativi

Naturalmente, tali problemi creano ritardi, rilavorazioni e aspettative deluse, ma gli svantaggi non finiscono qui.

Oltre allo spreco dovuto alle rilavorazioni, questo di solito fa ritardare i progetti. Se i team stanno cercando freneticamente di chiudere gli elementi del backlog e non hanno mai

davvero chiaro cosa servirà per fare dei progressi reali, le cose che effettivamente *devono* essere fatte tendono a essere trascurate.

Spesso, anche se non sempre, questo porta a una maggiore pressione per la consegna. Man mano che i progetti accumulano sempre più ritardo, i dirigenti potrebbero tentare di rimmetterli in carreggiata chiedendo alle persone di andare più veloce. Questo si traduce invariabilmente in orari di lavoro più lunghi.

Questo, a sua volta, tende a erodere la fiducia e a inasprire la cultura di un'organizzazione. Le relazioni che dovrebbero essere collaborative diventano conflittuali. Le persone che dovrebbero lavorare insieme per trovare le soluzioni migliori e più rapide, disperdono energie per stabilire che, quando le cose inevitabilmente vanno male, non è colpa loro.

Nella frenetica ricerca di funzionalità ed elementi di lavoro chiusi, i team spesso si trovano a tagliare gli angoli. Questo significa realmente che stanno lasciando che la qualità (specialmente la qualità del codice) ne risenta. Questo, a sua volta, significa che stanno scambiando la produttività futura con l'illusione del progresso nel presente.

Mentre le condizioni di lavoro diventano sempre più spiacevoli, i talenti chiave iniziano a disimpegnarsi o addirittura a cercare altre opportunità.

Le organizzazioni che si comportano in questo modo stanno “mangiando il grano da semina”, per così dire, in più di un modo. La base di codice diventa meno manutenibile e le persone che l'avrebbero dovuta mantenere vengono allontanate.

Se c'è un lato positivo, è invisibile ai nostri occhi.



# Capitolo 3: Introduzione al Requirements Maturation Flow (RMF)

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 3.1: Cosa non è RMF

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 3.2: Cosa è RMF

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 3.3: L'Adozione Incrementale è Supportata e Raccomandata

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 3.4: RMF 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 3.5: RMF 2

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 3.6: RMF 3

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 4: È Agile?

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 4.1: “Individui e Interazioni”, “Software Funzionante”

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 4.2: Collaborazione con il Cliente

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 4.3: Rispondere al Cambiamento

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 4.4: Trasparenza

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 4.5: Si Adatta al Processo, Coerente con Agile

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Parte II: Creare Spazio per la Preparazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 5: La Prima Estensione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 5.1: Il Lavoro di Preparazione è *Lavoro*

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 5.2: Naturalizzare il Lavoro di Preparazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 5.3: Un Incidente Illustrativo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 5.4: Impatto Reciproco

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 5.5: La Funzione di RMF 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 6: Perché Le Persone Non Lo Fanno?

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.1: Il Lavoro di Preparazione Come Cittadino di Seconda Classe

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.2: Un'Allergia al Lavoro Non Produttivo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.3: Quindi È Stato Sepolto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.4: L'Influenza della Gestione dei Progetti

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.5: Il Pattern

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.6: Progetti e Stime

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.7: Come le Non-Stime Influenzano il Lavoro di Preparazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.8: Misurare la Velocità, Non la Direzione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.9: Misurazioni Sbagliate, Risultati Sbagliati

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 6.10: Dove Non Risiede la Colpa

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 7: Lavoro Esplicito di Preparazione (RMF 1)

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 7.1: Integrazione con il Synapse Framework™

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 7.2: Anatomia di RMF 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 7.3: Comportamento: Riservare Capacità per la Collaborazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 7.4: Artefatto: L'Elemento di Preparazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.



## **7.5: Attività: La Riunione di Collaborazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **7.6: Comportamento: Continuare a Collaborare Fino al Raggiungimento della Comprensione Condivisa**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **7.7: Comportamento: Confermare Sempre la Comprensione Condivisa**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **7.8: Come RMF 1 Cambia il Flusso di Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 8: Effetti del RMF 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 8.1: La Vita Prima di RMF 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 8.2: Prima: Tempo Speso nella Comprensione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 8.3: Dopo: Tempo Speso nella Comprensione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 8.4: La Vita Dopo l'Adozione di RMF 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 8.5: Fondamentale

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 9: Mettere in Pratica RMF

## 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

### 9.1: Formazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

### 9.2: Requisiti Minimi per Tipo di Team

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

### 9.3: Accordo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

### 9.4: Preparazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

### 9.5: Progetto Pilota

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 9.6: Implementazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 9.7: Follow Up

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 9.8: Dichiarare il Successo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 9.9: Rimanere Vigili

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 9.10: E il “Come”?

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 9.11: È Ora di Metterlo in Pratica!

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# **Parte III: Controllo del Completamento del Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 10: La Prossima Necessità

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 10.1: Margine di Interpretazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 10.2: Ridurre il Margine di Interpretazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 10.3: Una Terza Opzione: Nessun “Margine di Manovra”

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 10.4: Potenziale Impatto sul Completamento

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 10.5: Potenziale Impatto sull'Esecuzione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **10.6: Alternativa Proposta: Non Lasciare Spazio a Interpretazioni**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **10.7: Benefici**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **10.8: Sui Timori della Paralisi da Analisi**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **10.9: La Prossima Necessità: Definizioni di Done Su Misura**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 11: Cosa Fanno Solitamente le Persone

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 11.1: Se È Così Ottimo, Perché le Persone Non lo Fanno?

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 11.2: Dal College al Coaching

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 11.3: Sovraccarico di Coaching

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 11.4: Un Modo in Cui le Persone Gestiscono la DoD: Non Farlo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.



## 11.5: Solo Criteri di Accettazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 11.6: Definition of Done Globale

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 11.7: Senza Mordente

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 11.8: Sommario: Il Termine “DoD” È Più Frequente delle Effettive Definizioni di Completamento

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 12: Definire una Definizione di Fatto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.1: A proposito di un singolo elemento di lavoro

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.2: Completezza

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.3: Precisione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.4: Struttura di una Definizione di Fatto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.5: Specifiche

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.6: Criteri di Uscita dell'Engineering

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.7: Criteri di Ingresso del Prodotto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.8: Più Parti, Un Solo Gate

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.9: Esempio

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.10: Mappatura sul Vostro Processo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 12.11: Riepilogo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 13: Definition of Done Personalizzata (RMF 2)

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 13.1: Principio: Ogni Elemento di Lavoro è Unico

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 13.2: Comportamento: Mantenere Uno o Più Template DoD

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 13.3: Attività: Definire il Template DoD

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 13.4: Mantenere e Migliorare il Template DoD

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.5: Molteplici Template DoD**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.6: Comportamento: Utilizzare i Template come Punti di Partenza per le Definizioni di Fatto**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.7: Comportamento: Concordare Definizioni di Fatto Personalizzate**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.8: Attività: Definire una Definition of Done per un Elemento di Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.9: Un'Altra Estensione del Flusso di Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.10: Comportamento: Maturare la DoD Prima di Iniziare l'Implementazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.11: Attività: Analisi Offline per Maturare la DoD di un PBI**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.12: Aggiungere la Maturazione al Flusso**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.13: Comportamento: Monitorare il Completamento negli Elementi di Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.14: Aggiungere il Monitoraggio dei Progressi**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.15: Comportamento: Controllare il Lavoro in base al Completamento**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **13.16: Attività: Utilizzare il DoD per Determinare il Completamento**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 13.17: Come Si Integra il Controllo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 13.18: In Sintesi

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 14: La Vita con RMF 1 e 2

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 14.1: Costo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 14.2: Tempistiche

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 14.3: Impatto sul Team di Implementazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 14.4: Impatto sul Product Owner

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 14.5: Impatto sulla Leadership

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.



## 14.6: Riepilogo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 15: Installazione di RMF 2

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 15.1: Coinvolgimento degli Stakeholder

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 15.2: Livello di Dettaglio

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 15.3: Accordo di Lavoro

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 15.4: Lavoro Iniziale

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 15.5: Implementazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 15.6: Riepilogo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Parte IV: Implementazione del Gating

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 16: Il Requisito Finale

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.1: Era Lì Tutto il Tempo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.2: Il Potere del Timing

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.3: Capovolgerlo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.4: Rischi e Costi

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.5: Il Valore dell'Attesa fino alla Prontezza

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.6: Un Beneficio Aggiuntivo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.7: La Dichiarazione del Problema

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.8: La Necessità

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 16.9: Riepilogo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 17: Background sulla Definizione di Prontezza

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 17.1: L'Autorizzazione al Lavoro del Lean

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 17.2: I Criteri di Ingresso delle Colonne Kanban

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 17.3: Scrum e altri Apocrifi dei Processi Agile

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 17.4: Fare Surf > Programmare

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 17.5: Soluzioni Puntuali

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 17.6: Siamo Pronti a Essere Pronti

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.



# Capitolo 18: Definire una Definizione di Pronto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 18.1: Scopo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 18.2: Su Misura

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 18.3: Anatomia

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 18.4: Esempio

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 18.5: Accordo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 18.6: Controllo degli Accessi

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 18.7: Riepilogo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 19: Definizione di Pronto Personalizzata (RMF 3)

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 19.1: Un Altro Cannello nel Processo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 19.2: La Struttura di una Definizione di Pronto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 19.3: Criteri di Uscita del Product

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 19.4: Criteri di Ingresso dell'Engineering

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 19.5: Nessun Danno nella Duplicazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.6: Comportamento: Mantenere Uno o Più Template DoR**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.7: Perché Avere un Template della Definizione di Pronto?**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.8: Attività: Definire il Template DoR**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.9: Mantenere i Template DoR nel Tempo**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.10: I Template Sono Solo un Punto di Partenza**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.11: Comportamento: Concordare Definizioni di Ready Personalizzate**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.12: Attività: Definire una Definition of Ready per un Elemento di Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.13: Comportamento: Rendere gli Elementi Pronti prima di Iniziare l'Implementazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.14: Condizioni Fuori dal Controllo del Team**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.15: Comportamento: Tracciare la Prontezza negli Elementi di Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.16: Comportamento: Controllare il Lavoro tramite la Prontezza**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **19.17: Attività: Utilizzare il DoR per Determinare la Prontezza**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 19.18: In Sintesi

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Parte V: Sintesi

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 20: La Maggior Parte delle Scadenze Non Conta

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.1: Scadenze Reali

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.2: Le Scadenze Arbitrarie Non Contano

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.3: Compagnie Aeree

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.4: L'Origine del Debito Tecnico

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.5: Scadenze di Marketing e Vendite

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.



## 20.6: Scadenze di Progetto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.7: Esiste Un Altro Modo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.8: Le Scadenze Arbitrarie Non Sono Necessarie

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.9: Le Scadenze Arbitrarie Devono Essere Abolite

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.10: Le Scadenze Reali Sono Ancora un Fattore

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.11: *Pis Aller*<sup>1</sup>

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 20.12: Conclusione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

<sup>1</sup>Una mossa di ultima risorsa, originariamente dalla lingua francese.

# Capitolo 21: Competenza 1: Flusso di Maturazione dei Requisiti

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 21.1: Come Sotto, Così Sopra

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 21.2: Principio: Trasparenza in Tutto il Lavoro Necessario

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 21.3: Comportamento: La Responsabilità Viaggia con il Lavoro

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **21.4: Comportamento: Tracciare Visibilmente lo Stato dei Requisiti**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **21.5: Comportamento: Rivelare Tutto il Lavoro Associato alla Preparazione e all'Implementazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **21.6: Attività: Lavoro di Preparazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **21.7: Comportamento: Rivelare Tutto il Lavoro Necessario per Completare un Elemento di Lavoro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **21.8: Comportamento: Preferire la Preparazione alle Scadenze**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **21.9: Conclusione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# **Capitolo 22: Come Fluiscono il Lavoro e le Informazioni in Scrum con RMF**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **22.1: Note Preliminari**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **22.2: Acquisizione e Preparazione del Requisito Iniziale**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **22.3: Avvio del Lavoro di Preparazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **22.4: Pianificazione ed Esecuzione del Lavoro di Preparazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **22.5: Revisione dei Risultati della Preparazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **22.6: Pianificazione e Completamento dell'Implementazione**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 23: L'Impatto dell'RMF

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 23.1: Il Ciclo di Vita di un Requisito

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 23.2: Esempio di Flusso di Informazioni e Lavoro

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 23.3: Prima

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 23.4: Dopo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 23.5: Benefici

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Capitolo 24: Transizione verso RMF

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 24.1: Schema di Adozione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 24.2: Installazione in un Flusso di Lavoro Scrum

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 24.3: Altri Framework

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 24.4: Principale Resistenza: Elementi di Lavoro di Preparazione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 24.5: Il Grande Cambiamento: Mentalità

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 24.6: Consigli per il Cambiamento

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 24.7: Conclusione

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.



# Capitolo 25: Sta a Te

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 25.1: Riepilogo

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## 25.2: Ora Tocca a Te

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Parte VI: Risorse

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Appendice A: Scrum Non È il Problema

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Cos'è Scrum?

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Framework

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Scrum Gestisce il Progetto e il Lavoro

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Le Follie di Trattare Scrum Come un Framework di Product Management

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Nessun Meccanismo Prescritto per la Maturazione dei Requisiti**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Nessun Investimento di Competenze Ingegneristiche nello Sviluppo dei Requisiti**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Antipattern**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Estensione Necessaria**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Aggiungere elementi a Scrum**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# **Appendice B: The Synapse Framework™**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Cosa copre Synapse**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Le Tre Maestrie**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Come viene adottato Synapse**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **L'Unione di Due Framework**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **La Struttura del Framework Synapse**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Maestrie Organizzative**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Competenze Organizzative**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Abitudini Organizzative**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Struttura del Framework Synapse**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Anatomia di una Prassi**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **L'Importanza dell'Ordine**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **L'Impatto di Synapse su questo Libro**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Appendice C: Obiezioni e Ostacoli Comuni all'RMF 1

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Obiezioni Comuni

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Ostacoli Comuni

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# **Appendice D: Liste di Criteri Iniziali DoD**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Lista Iniziale dei Criteri di Uscita Engineering**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## **Lista Base dei Criteri di Ingresso del Prodotto**

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.



# Appendice E: Liste di Criteri Iniziali per la DoR

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Criteri di Uscita del Prodotto

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Criteri di Ingresso Tecnici

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

## Criteri di Ingresso dello Sprint

Questo contenuto non è disponibile nel libro di esempio. Il libro può essere acquistato su Leanpub all'indirizzo <https://leanpub.com/ready-it>.

# Indice

- Agile, 24
- AI, 20
- Altri Framework, 72
- ambiente di sviluppo, 10
- API, 22
- assunzione, 24
- auto-antonimi, 15
- banche, 10
- Barreras, Tom, vii
- base di codice, 25
- base di conoscenza, 8
- Behavior-Driven Development, 7
- Behavior: Visibly Track the State of Requirements, 68
- caricare, 8
- Central Oregon, vii
- cicli di feedback, 13
- codice esistente, 11
- Come sai quando è finito?, 22
- competenze del team, i
- completamento elemento di lavoro, 19
- Comportamento: Rivelare Tutto il Lavoro
  - Associato alla Preparazione e all'Implementazione, 68
- comprensione condivisa, 34
- comprensione condivisa confermata, 16
- consulenti, 8
- contare i pezzi, 21
- Costi Cumulativi, 24
- Creare Spazio per la Preparazione, 29
- Criteri di Ingresso dell'Engineering, 60
- cultura, 10
- Data loss or corruption, 20
- debito tecnico, 65
- defining 'done', 19
- Definition of Done, 22, 46
- Definition of Done personalizzate, 45
- Definizione di Fatto dell'Elemento del Product Backlog, 47
- Definizioni di Pronto Personalizzate, 60
- Definizioni di Ready Personalizzate, iii
- Dichiarare il Successo, 37
- difetto, 24
- dominio, 10
- dottrine di programmazione di metà ventesimo secolo, 7
- elementi del backlog, 24
- elemento di lavoro, 24
- Elemento di Preparazione, 33
- Engineering executives, 4
- esperto di dominio, 11
- esperto in materia, 6
- fidelizzazione dei talenti, 25
- fiducia, 25
- finanza, 10
- flusso, 13
- flusso di lavoro, 34, 46
- Flusso di Maturazione dei Requisiti, i, iii, iv, 67
- Fungible people, 23
- Garbage-in, garbage-out, 15
- Gate in the Process, 60
- gold plating, 20
- gruppo dirigente, 6
- Implementazione del Gating, 53
- individual contributors, 4

- infrastruttura finanziaria nazionale degli Stati Uniti, 2
- Initial Work, 51
- Installazione in un Flusso di Lavoro Scrum, 72
- Istituto di Credito Federato, 2
- iterazione, 22
- Kanban, i
- La Maggior Parte delle Scadenze Non Conta, 65
- Lavoro Esplicito di Preparazione, 33
- lavoro extra, 19
- leadership, 10
- Level of Detail, 51
- lingue naturali, 16
- market share, 20
- Marketing e Vendite, 65
- Maturare la DoD, 46
- mid-Sprint changes, 6
- Monitoraggio del Completamento, 47
- organizzazione Product, 4
- OutSystems, 3
- parti interessate, 17
- pattern di progettazione, iv
- PKB-Driven Development, iv
- portale di rimborso prestiti, 3
- pratiche ingegneristiche, 10, 15
- Principale Resistenza, 72
- Producere, iii
- Product Backlog Item, 7, 10, 12, 14, 21
- Product e Engineering, 14
- Product Exit Criteria, 60
- Product Manager, 5
- Product Owner, 5, 10, 19
- Product, role of, 4
- Project Management Institute, 2
- pronti*, 12
- puzzle building, 21
- Qualcosa Manca, 1
- qualità del codice, 25
- Ready, i, ii
- Regulatory violations, 20
- Requirements Maturation Flow, 26, 69
- requisiti errati, 17
- responsabilità e tracciabilità, 14
- rilavorazione, 17
- rischi, 18
- Risorse, 75
- risorse del server, 23
- ritardo, 25
- RMF 1, 30, 80
- RMF 2, 51
- Rollout, 51
- scadenze, 22
- Scadenze di Progetto, 66
- Schema di Adozione, 72
- Scrum, 5, 16, 22, 69
- Security vulnerabilities, 20
- servizi dipendenti, 10
- servizio a monte, 12
- sistema, 9
- sistema legacy, 6
- software developers, 19
- soluzione C#/.NET, 3
- soluzione low-code, 3
- spreco, 24
- Sprint, 11, 15, 19, 22, 24
- Stakeholder Involvement, 51
- stesura dei requisiti, 15
- strappare la sconfitta dalle fauci della vittoria, 18
- Structure of a Definition of Ready, 60
- sviluppo a grandi lotti, 16
- Sviluppo Guidato dal Comportamento, iv
- sviluppo software, 15, 18, 21
- System downtime, 20
- team di engineering, 9
- Team di Sviluppo, 19
- team di sviluppo, 10

- tecnica di stesura dei requisiti, 10
- tecniche di sviluppo software, 7
- Terminator-style revolt, 20
- test-development, vii
- Tester, 12
- The Bank, 13
- The Big Turnaround, 13
- Transizione verso Requirements Maturation
  - Flow, 72
  - trasformazioni Agile, iv
  - vero requisito, 19
  - wasted time, 20
  - Waterfall, 16
  - Working Agreement, 51