CPRE

Certified Professional for Requirements Engineering

RE@Agile

Syllabus

Stefan Gärtner, Peter Hruschka,
Markus Meuten, Gareth Rogers,
Hans-Jörg Steffe

IREB

2.3.0 | 27 Febbraio 2026

Termini di utilizzo:

1. I soggetti individuali e i Training Provider possono utilizzare questo Syllabus come base per i seminari, a condizione che il copyright sia riconosciuto e incluso nei materiali del seminario. Chiunque utilizzi questo Syllabus a fini pubblicitari deve ottenere il consenso scritto di IREB e.V. a questo specifico utilizzo.
2. Un singolo individuo o gruppo di individui può utilizzare questo Syllabus come base per articoli, libri o altre pubblicazioni, a patto che venga riconosciuto in tali pubblicazioni il copyright degli autori e di IREB e V. come fonti e proprietari di questo documento.

© IREB e.V.

Tutti i diritti riservati. Nessuna parte di questa pubblicazione può essere riprodotta, memorizzata in un sistema accessibile o trasmessa in qualsiasi forma o con qualsiasi mezzo, elettronico, meccanico, fotocopia, registrazione o altro, senza la preventiva autorizzazione scritta degli autori o IREB e.V.

Crediti e ringraziamenti

La versione attuale di questo syllabus è stata redatta da: Stefan Gärtner, Peter Hruschka, Kim Lauenroth, Markus Meuten, Gareth Rogers e Hans-Jörg Steffe.

Revisione a cura di Rainer Grau. I commenti di revisione sono stati forniti da Jan Jaap Cannegieter, Andrea Herrmann, Uwe Valentini e Sven van der Zee. La traduzione in Italiano è di Raffaele Bozzini. La revisione della versione in lingua italiana è stata effettuata da Salvatore Reale.

Approvato per il rilascio il 25 ottobre 2024 dal Consiglio IREB su raccomandazione di Xavier Franch.

Si ringraziano tutti coloro che hanno collaborato.

Il copyright © 2017-2025 di questo Syllabus appartiene agli autori sopra elencati. I diritti sono stati trasferiti a IREB International Requirements Engineering Board e.V.

Scopo del documento

Questo syllabus definisce i certificati per il RE@Agile Practitioner e per il RE@Agile Specialist dell'International Requirements Engineering Board (IREB). Il Syllabus fornisce ai Training Provider la base per la creazione del proprio materiale per la formazione. Gli studenti possono utilizzare il syllabus per avere una panoramica dei contenuti previsti e degli obiettivi di apprendimento. I dettagli sui contenuti per la preparazione del materiale del corso e per l'esame sono riportati nel "Manuale RE@Agile, istruzione e formazione per l'IREB Certified Professional for Requirements Engineering RE@Agile Practitioner | Specialist".

Contenuto del Syllabus

Il modulo RE@Agile si rivolge a persone che si occupano di Requirements Engineering, Business Analysis, Business Engineering, Sviluppo di software e sistemi, nonché a figure già presenti nella comunità agile (Product Owner, Scrum Master, Developer) che desiderano approfondire le proprie conoscenze nella gestione dei requisiti.

Livello di dettaglio

Il livello di dettaglio in questo Syllabus garantisce la consistenza dei corsi e degli esami a livello internazionale. Per raggiungere tale obiettivo, il Syllabus contiene i seguenti:

- Obiettivi didattici generali,
- Contenuti con una descrizione degli obiettivi didattici, e
- Riferimenti a ulteriore documentazione (se necessari).

Obiettivi didattici / livelli di conoscenza cognitiva

A ogni modulo e obiettivo didattico di questo Syllabus viene assegnato un livello cognitivo. Vengono utilizzati i seguenti livelli cognitivi:

- **L1: Conoscere** (identificare, ricordare, recuperare, richiamare alla memoria, riconoscere) – Il candidato riconoscerà, ricorderà e richiamerà alla memoria un termine o un concetto.
- **L2: Comprendere** (riassumere, generalizzare, astrarre, classificare, confrontare, mappare, contrastare, esemplificare, interpretare, tradurre, rappresentare, inferire, concludere, categorizzare, costruire modelli) – Il candidato è in grado di selezionare le ragioni o le spiegazioni delle affermazioni relative all'argomento e può riassumere, confrontare, classificare, categorizzare e fornire esempi del concetto.
- **L3: Applicare** (implementare, eseguire, utilizzare, seguire una procedura, applicare una procedura) – Il candidato può selezionare la corretta applicazione di un concetto o di una tecnica e applicarla a un determinato contesto.
- **L4: Analizzare** (analizzare, organizzare, trovare coerenza, integrare, delineare, analizzare, strutturare, attribuire, decostruire, differenziare, discriminare, distinguere, focalizzare, selezionare) – Il candidato è in grado di separare le informazioni relative a una procedura o tecnica nelle sue parti costitutive per una migliore comprensione e può distinguere tra fatti e inferenze. Un'applicazione tipica è quella di analizzare un documento, un software o una situazione di progetto e proporre azioni appropriate per risolvere un problema o un compito.
- **L5: Valutare** (criticare, giudicare) – Il candidato è in grado di fornire una critica ben argomentata di un determinato artefatto e di esprimere un giudizio approfondito in un caso specifico.

Si noti che un obiettivo di apprendimento al livello di conoscenza cognitiva Ln contiene anche elementi di tutti i livelli cognitivi sottostanti (da L1 a Ln−1).

> Esempio: un obiettivo di apprendimento del tipo "Applicare la tecnica RE xyz" si colloca al livello di conoscenza cognitiva (L3). Tuttavia, la capacità di applicare richiede che gli studenti conoscano la tecnica RE xyz (L1) e che capiscano a cosa serve la tecnica (L2).

Tutti i termini usati in questo Syllabus e definiti nel Glossario IREB devono essere conosciuti (L1), anche se non sono esplicitamente menzionati negli obiettivi didattici.

Il glossario è disponibile per il download sul sito web di IREB all'indirizzo https://www.ireb.org/en/downloads/#cpre-glossary-2-0

Il Syllabus e il relativo manuale utilizzano l'abbreviazione "RE" per indicare il Requirements Engineering (Ingegneria dei Requisiti).

Struttura del syllabus

Il Syllabus è composto da sei capitoli principali. Ogni capitolo copre una unità didattica (EU, Educational Unit). I titoli dei capitoli contengono il livello cognitivo dei capitoli stessi, che è il livello più alto dei loro sottocapitoli. Il tempo di insegnamento suggerito è il minimo che un corso dovrebbe dedicare a quel capitolo. Le aziende di formazione sono libere di dedicare più tempo alle EU e agli esercizi. Tuttavia, dovrebbero assicurarsi che il tempo richiesto sia mantenuto in relazione alle altre EU. I termini importanti utilizzati in un capitolo sono elencati all'inizio del capitolo.

Esempio: EU2 Un inizio pulito di un progetto (L2)
Durata: 120 minuti + 60 minuti di esercitazione
Termini: Vision del prodotto, obiettivo del prodotto, Stakeholder, persona, ambito del prodotto, confini del sistema

Questo esempio mostra che il capitolo 2 contiene obiettivi didattici di livello L2 e che sono previsti 180 minuti per l'insegnamento del materiale contenuto in questo capitolo.

Ogni capitolo può contenere sotto-capitoli. Il loro titolo include anche il livello cognitivo del loro contenuto.

Vengono enumerati gli obiettivi didattici (EO). La numerazione mostra a quale sottocapitolo appartengono.

Esempio: EO 3.3.1 Essere in grado di applicare i criteri INVEST nella stesura dei requisiti (L3)

Questo esempio mostra che l'obiettivo didattico EO 3.3.1 è descritto nel sottocapitolo 3.3 e che è previsto il livello cognitivo L3.

L'Esame

Questo Syllabus copre le unità didattiche e gli obiettivi didattici per gli esami di certificazione di

- CPRE RE@Agile – Practitioner
- CPRE RE@Agile – Specialist

L'esame per ottenere la certificazione CPRE RE@Agile – Practitioner consiste in un **test a scelta multipla**.

L'esame per ottenere la certificazione CPRE RE@Agile – Specialist consiste in un **compito scritto**.

Entrambi gli esami includono domande su tutte le unità didattiche e tutti gli obiettivi didattici del Syllabus.

Ogni domanda d'esame può includere materiale tratto da più capitoli del Syllabus, nonché da più obiettivi didattici o anche da parti di un unico obiettivo didattico.

L'esame a scelta multipla per la certificazione Practitioner

- verifica tutti gli obiettivi didattici del Syllabus. Tuttavia, per gli obiettivi didattici relativi ai livelli cognitivi L4 e L5, le domande di esame sono limitate agli elementi relativi ai livelli cognitivi da L1 a L3.
- può essere sostenuto immediatamente dopo un corso di formazione, ma anche indipendentemente dai corsi (ad esempio, a distanza o in un centro d'esame).

Il compito scritto per la certificazione Specialist

- verifica tutti gli obiettivi didattici del Syllabus ai livelli di conoscenza cognitiva indicati per ciascun obiettivo di apprendimento.
- segue la descrizione del compito per RE@Agile – Specialist –, che si trova all'indirizzo https://cpre.ireb.org/en/downloads-and-resources/downloads#cpre-re-agile-specialist-written-assignment.
- Viene svolto in autonomia e viene sottoposto a un Organismo di Certificazione autorizzato.

I seguenti obiettivi di apprendimento generici si applicano anche al compito scritto per la certificazione Specialist:

EO G1: Analizzare e illustrare i problemi di RE@Agile utilizzando un contesto familiare al candidato o che gli assomiglia molto (L4).

EO G2: Valutare e riflettere sull'uso di pratiche, metodi, processi e strumenti RE@Agile nei progetti in cui il candidato è stato coinvolto (L5).

Un elenco degli Organismi di Certificazione autorizzati da IREB è disponibile sul sito web:

https://www.ireb.org

Storia delle Versioni

Versione	Data	Commenti
2.3.0	27 Febbraio 2026	Versione iniziale Italiana

Contenuto

Contenuto 7

1 Che cos'è RE@Agile? (L2) 9

2 Un avvio pulito del progetto (L3) 11

2.1 Vision e obiettivi (L3) 11

2.2 Specificare i confini del sistema (L3) 12

2.3 Identificazione e gestione degli stakeholder (L3) 12

2.4 Dipendenze tra vision/obiettivi, stakeholder e confini del sistema (L3) 13

3 Gestione dei requisiti funzionali (L4) 14

3.1 Diversi livelli di granularità dei requisiti 14

3.2 Comunicare e documentare a diversi livelli 15

3.3 Lavorare con le user story e gli elementi del Backlog 16

3.4 Tecniche di suddivisione e raggruppamento 17

3.5 Sapere quando fermarsi 18

3.6 Documentazione di progetto e di prodotto dei requisiti 18

4 Gestione dei requisiti e dei vincoli di qualità (L3) 20

4.1 Comprendere l'importanza dei requisiti di qualità e dei vincoli (L2) 20

4.2 Aggiungere precisione ai requisiti di qualità (L2) 20

4.3 Requisiti di qualità e Backlog (L3) 22

4.4 Rendere espliciti i vincoli (L2) 23

5 Priorità e stima dei requisiti (L3) 24

5.1 Determinazione del valore di business 24

5.2 Valore di business, rischi e dipendenze (L3) 25

5.3 Esprimere le priorità e ordinare il Backlog 25

5.4 Stima degli elementi del Backlog (L3) 26

5.5 Scegliere una strategia di sviluppo (L2) 27

6 Scalare RE@Agile (L2) 29

6.1 Requisiti di scalabilità e team (L2) 29

6.2 Criteri per la strutturazione dei requisiti e dei team su larga scala (L2)..... 30

6.3 Roadmap e pianificazione su larga scala (L2) 31

6.4 Validazione del prodotto (L2) 32

7 Definizioni di termini, glossario 33

8 Riferimenti Bibliografici 34

1 Che cos'è RE@Agile? (L2)

Durata: 45 minuti

Termini: Stakeholder, Product Owner, Cooperativo, Iterativo, Incrementale

Obiettivi didattici

EO 1.1 Conoscere la definizione di RE@Agile (L1)
EO 1.2 Comprendere gli obiettivi di RE@Agile (L2)
EO 1.3 Comprendere che la responsabilità di definire requisiti adeguati spetta al Product Owner (L2)

Contenuto

In base alle rispettive evoluzioni storiche, gli approcci RE e agili sono spesso considerati separatamente, piuttosto che insieme. Questo porta spesso al malinteso che esistano due modi di fare RE: RE classico e RE agile. Gli autori ritengono che esista solo un RE buono o cattivo – in un mondo non agile o agile. Pertanto, chiamiamo l'approccio RE@Agile.

Agile e RE sono due discipline con origini diverse e obiettivi distinti che possono tuttavia imparare molto l'una dall'altra. Nel RE@Agile Primer [Prim2017] abbiamo concluso che: "Il valore più importante è condiviso da RE e agile, ed è quello di rendere felice l'utente del prodotto perché la soluzione soddisfa le sue esigenze o cura i suoi principali malesseri."

RE@Agile è un approccio cooperativo, iterativo e incrementale con quattro obiettivi:

1. Conoscere i requisiti rilevanti con un livello di dettaglio adeguato (in qualsiasi momento durante lo sviluppo del sistema).
2. Raggiungere un accordo sufficiente sui requisiti tra gli stakeholder rilevanti.
3. Acquisire (e documentare) i requisiti in base ai vincoli dell'organizzazione.
4. Eseguire tutte le attività relative ai requisiti secondo i principi del Manifesto Agile.

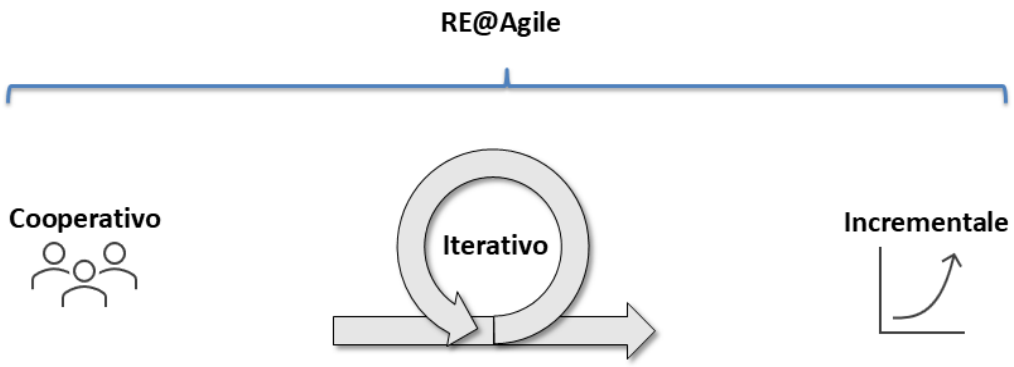

Figura 1: Approccio cooperativo, iterativo e incrementale

Negli approcci agili, i diversi ruoli forniscono i requisiti per il sistema desiderato. Indipendentemente da chi fornisce questi requisiti, da chi li struttura e da chi li dettaglia, la persona con il ruolo/responsabilità di Product Owner rimane responsabile del RE.

2 Un avvio pulito del progetto (L3)

Durata: 120 minuti + 60 minuti di esercitazione

Termini: vision, Obiettivo, Stakeholder, Perimetro del sistema, Context diagram, Use case diagram

Contenuto

Anche negli approcci agili è necessario stabilire alcuni prerequisiti importanti prima di poter avviare con successo lo sviluppo iterativo e incrementale del sistema.

- Definizione della vision e/o degli obiettivi del sistema
- Identificazione dell'ambito attualmente noto del sistema e dei confini del sistema
- Identificazione degli stakeholder rilevanti e di altre fonti di requisiti importanti

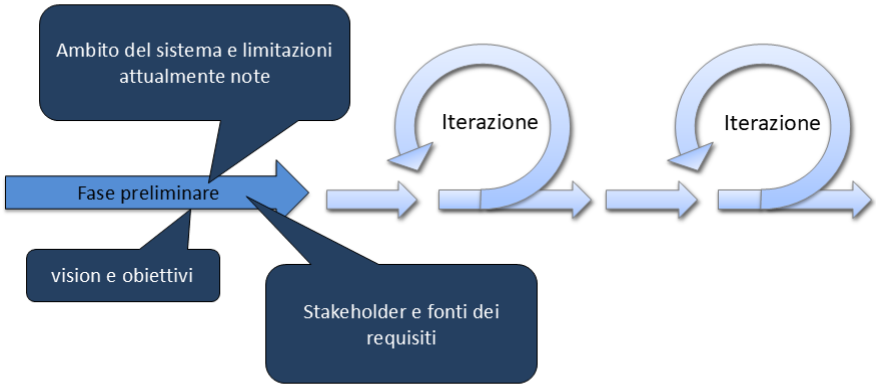

Figura 2: Un avvio pulito del progetto

2.1 Vision e obiettivi (L3)

Obiettivi didattici

EO 2.1.1 Applicare la specifica degli obiettivi e delle vision (L3)

Contenuto

La vision del sistema o del prodotto descrive l'obiettivo generale che deve essere raggiunto con il sistema/prodotto. La vision è di fondamentale importanza per ogni attività di sviluppo. Essa definisce il fondamento e funge da orientamento generale per tutte le attività di sviluppo. Ogni requisito dovrebbe supportare il raggiungimento della vision del sistema.

Gli approcci alternativi alla formulazione di obiettivi o vision sono:

- SMART [Dora1981]
- PAM [Robe2003]
- Product Vision Box [High2001]
- News from the future [HeHe2011]
- Vision Board

- Tecniche Canvas

2.2 Specificare i confini del sistema (L3)

Obiettivi didattici

EO 2.2.1 Applicare la specifica del confine del sistema (L3)

Contenuto

Una comprensione comune e condivisa dell'ambito e del contesto del sistema è un prerequisito per uno sforzo di sviluppo efficace ed efficiente [Glinz2024].

L'ambito e il confine del sistema possono essere documentati e chiariti con diverse tecniche, come ad esempio:

- Context diagram
- Linguaggio naturale
- Use case diagram
- Story map

2.3 Identificazione e gestione degli stakeholder (L3)

Obiettivi didattici

EO 2.3.1 Padroneggiare e utilizzare l'identificazione e la gestione degli stakeholder (L3)

Contenuto

Analogamente agli approcci tradizionali, anche all'inizio del processo agile è necessario identificare gli stakeholder più importanti, in modo da definire un quadro di riferimento per l'elicitazione dei requisiti. Nell'agilità, tuttavia, gli stakeholder possono cambiare e cambieranno costantemente. È quindi essenziale che l'identificazione e la definizione degli stakeholder siano intese come un processo cooperativo, incrementale e, soprattutto, iterativo (si veda anche il capitolo 1).

La mancata identificazione e il mancato coinvolgimento di uno stakeholder importante nel processo di sviluppo possono avere un impatto notevole. Se i requisiti di tale stakeholder vengono identificati troppo tardi (o non vengono identificati affatto), ciò può comportare modifiche costose o addirittura un sistema inutile [PoRu2021].

Il modello a cipolla di Ian Alexander [Alex2005] è uno strumento semplice per l'identificazione e la classificazione degli stakeholder. Il modello è composto da tre tipi di stakeholder (strati a cipolla) che possono essere ricercati sistematicamente:

- Gli stakeholder del sistema
- Stakeholder del contesto circostante
- Stakeholder dell'ambiente in generale

Come regola generale, l'identificazione degli stakeholder dovrebbe basarsi su un'ampia gamma di fonti.

A seconda del sistema e del dominio, anche la documentazione esistente, i sistemi vicini con interfacce al sistema sviluppato, i sistemi preesistenti o persino i sistemi concorrenti possono essere fonti importanti (oltre agli stakeholder) di requisiti.

2.4 Dipendenze tra vision/obiettivi, stakeholder e confini del sistema (L3)

Obiettivi didattici

EO 2.4.1 Applicare il cambiamento dinamico di vision e obiettivi, stakeholder e ambito del sistema (L3)

Contenuto

Le definizioni di vision e obiettivi, nonché di stakeholder e confini del sistema sono qui interdipendenti [PoRu2021]:

- Gli stakeholder rilevanti formulano la vision e gli obiettivi. Pertanto, l'identificazione di un nuovo stakeholder rilevante può avere un impatto sulla vision e sugli obiettivi.
- La vision e gli obiettivi possono essere utilizzati per guidare l'identificazione di nuovi stakeholder chiedendosi: quale stakeholder potrebbe essere interessato al raggiungimento della vision e degli obiettivi o potrebbe essere influenzato dal loro raggiungimento?
- La vision e gli obiettivi possono essere utilizzati per definire un ambito iniziale ponendo la seguente domanda: quali elementi sono necessari per realizzare la vision e gli obiettivi?
- La modifica dei confini del sistema (e quindi dell'ambito) può avere un impatto sulla vision e sugli obiettivi. Se un aspetto viene rimosso dall'ambito, è necessario verificare che il sistema disponga ancora di mezzi sufficienti per realizzare la vision e gli obiettivi.
- Gli stakeholder definiscono i confini del sistema. Pertanto, l'identificazione di un nuovo stakeholder rilevante può avere un impatto sull'ambito.
- Una modifica dell'ambito (ad esempio, per raggiungere un obiettivo) richiede l'accordo degli stakeholder rilevanti.

Queste interdipendenze devono essere utilizzate per bilanciare tutti e tre gli elementi (vision e obiettivi, stakeholder, ambito) per esaminare l'impatto della modifica di uno dei tre elementi sugli altri.

A causa di queste dipendenze tra vision e obiettivi, stakeholder e ambito, raccomandiamo di trattare tutti questi elementi insieme e in modo coerente.

3 Gestione dei requisiti funzionali (L4)

Durata: 195 minuti + 120 minuti di esercitazione

Termini: Requisito funzionale, Epic, Feature, Storia, User story, Definition of Ready (DoR), INVEST, Livelli di granularità

Contenuto

Questa EU centrale adotta una visione statica dei requisiti funzionali, ovvero struttura un ampio insieme di requisiti in gerarchie di astrazione.

Non appena si accetta l'idea che i requisiti esistono a diversi livelli di granularità, sorgono naturalmente alcune domande:

- Come gestire più livelli di granularità?
- Quali criteri si possono e si devono applicare per suddividere grandi argomenti astratti in parti più piccole?
- È talvolta necessario raggruppare molti piccoli requisiti in blocchi più grandi in modo da avere un "quadro più ampio" per orientarsi?
- Quanto dobbiamo essere precisi prima che gli sviluppatori possano iniziare l'implementazione?
- È necessario o consigliabile mantenere più livelli di requisiti, o possiamo eliminare le dichiarazioni astratte non appena disponiamo di requisiti più concreti?
- Preferiamo strutturare il Backlog in base alle relazioni funzionali/processi o in base ad altre relazioni come i contesti tecnici?
- Dobbiamo mettere tutto questo per iscritto o possiamo semplicemente parlarne?

3.1 Diversi livelli di granularità dei requisiti

Obiettivi didattici

EO 3.1.1 Conoscere l'esistenza di requisiti funzionali a diversi livelli di granularità (L1)

Contenuto

Gli stakeholder di solito comunicano i requisiti a diversi livelli di granularità. A volte richiedono grandi blocchi di funzionalità, altre volte chiedono che vengano aggiunti o modificati dettagli minori.

Gli epic (talvolta chiamati anche temi) o le storie di grandi dimensioni (che rappresentano processi di business potenzialmente complessi) sono un ottimo modo per ottenere una visione d'insieme di tutti i requisiti attualmente noti nel Backlog, ovvero una panoramica di tutto ciò che gli stakeholder si aspettano da un sistema o da un prodotto. Gli epic sono scomposti in elementi più dettagliati, in genere caratteristiche e storie. Le storie forniscono un livello di granularità più dettagliato e possono comunicare i requisiti da diverse prospettive, ad esempio quella dell'utente o quella tecnica.

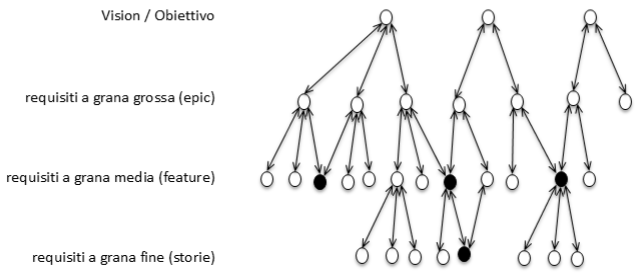

Figura 3: Granularità dei requisiti

3.2 Comunicare e documentare a diversi livelli

Obiettivi didattici

EO 3.2.1 Analizzare e applicare la scomposizione dei requisiti al livello più alto (L4)
EO 3.2.2 Analizzare e applicare diverse strategie di scomposizione su larga scala (L4)
EO 3.2.3 Analizzare e applicare l'identificazione, la documentazione e la comunicazione dei requisiti funzionali a diversi livelli di granularità (L4)

Contenuto

In base al principio "divide et impera", è necessario scomporre un sistema o un prodotto di grandi dimensioni in parti più piccole.

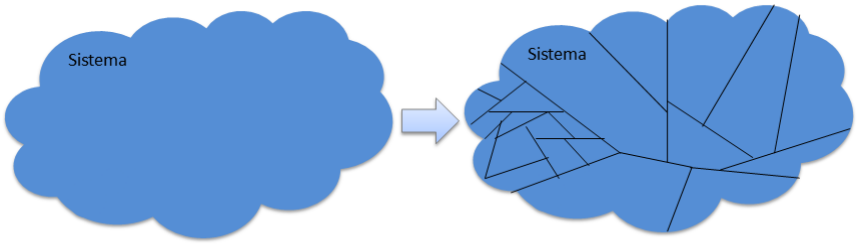

Figura 4: Decomposizione del sistema in parti più piccole

Esistono diversi approcci per raggiungere questo obiettivo (ad esempio, suddivisione in funzioni logiche, suddivisione in base ad aspetti organizzativi, suddivisione in base all'hardware, suddivisione in base ai dati, ...). In tal modo, esaminiamo l'approccio alla scomposizione dei requisiti in processi di creazione di valore avviati esternamente.

Oltre a scomporre i requisiti in parti più piccole, dobbiamo anche occuparci di comunicare e documentare i requisiti funzionali. La scelta di base è tra disegno (ad esempio, con modelli) e scrittura (ad esempio, in linguaggio naturale). Soprattutto, spetta all'intero team decidere il formato preferito.

Ad esempio, la panoramica potrebbe essere fornita utilizzando uno use case diagram, mentre le fasi del processo dei casi d'uso sono descritte con parole semplici (ad esempio, come una user story).

3.3 Lavorare con le user story e gli elementi del Backlog

Obiettivi didattici

EO 3.3.1 Applicare l'idea di User Centered Thinking ai requisiti con le user story (L3)
EO 3.3.2 Creare requisiti basati sull'idea della user story e utilizzando il modello di user story (L3)
EO 3.3.3 Applicare i criteri INVEST quando si scrivono i requisiti (L3)

Contenuto

Le user story sono diventate molto importanti nell'ambiente agile perché sottolineano l'importanza di strutturare e sviluppare i requisiti e i prodotti risultanti in modo centrato sull'utente. In altre parole, mettendo l'utente al primo posto. Questa prospettiva dell'utente è importante perché sviluppiamo principalmente in modo agile per ricevere regolarmente il feedback degli utenti.

È importante notare che nell'ambiente agile il termine "user story" viene spesso utilizzato con tre scopi diversi:

- Come metodo orientato all'utente per discutere e formulare i requisiti.
- Come livello più basso (più dettagliato) di strutturazione dei requisiti (Epic -> Features -> (User-) Story). In effetti, nella struttura dei requisiti si usa spesso il termine "user story" al posto di "story".
- Come template per descrivere i requisiti a diversi livelli gerarchici dal punto di vista dell'utente.

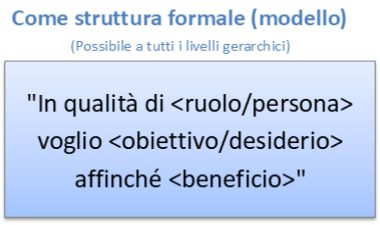

Figura 5: Modello di user story

In questo Syllabus, al fine di mitigare questa ambiguità, utilizziamo il termine "storia" per gli elementi più dettagliati nella struttura dei requisiti; pertanto, una "user story" è un tipo particolare di storia, ovvero una storia incentrata sull'utente. Quando ci riferiamo al modello presentato nella Fig. 5, useremo il termine "modello di user story". Si noti che il modello di user story può essere utilizzato per descrivere non solo storie, ma anche epic e feature, se lo si ritiene opportuno.

In generale, va notato che le user story non sono requisiti completi di per sé; sono più che altro una promessa di comunicazione. Per creare requisiti completi e corrispondenti elementi di Backlog, sarà necessario fornire ulteriori dettagli (documentandoli e/o discutendoli).

Quando ha introdotto le user story [Cohn2010], Bill Wake [Wake2003] ha definito che le user story devono rispettare i principi INVEST (Independent, Negotiable, Valuable, Estimated, Small, Testable).

Gli elementi del Backlog vengono spesso annotati su schede o post-it, oltre che su strumenti IT (ad esempio Jira MS-Devops), e disposti su pareti o tavoli per facilitare la pianificazione e la discussione. Utilizzando il modello 3C (card, conversation, confirmation), questi elementi del Backlog possono quindi essere discussi in modo più dettagliato e integrati con altri artefatti relativi ai requisiti (ad esempio modelli, documentazione).

3.4 Tecniche di suddivisione e raggruppamento

Obiettivi didattici

EO 3.4.1 Analizzare e applicare tecniche di suddivisione per requisiti funzionali di alto livello (L4)

EO 3.4.2 Analizzare e applicare il raggruppamento e l'astrazione dei requisiti funzionali dettagliati in requisiti più generici (L4)

Contenuto

Al fine di generare elementi di Backlog sufficientemente piccoli da poter essere inseriti in una singola iterazione, gli elementi di Backlog più grandi possono essere suddivisi in elementi più dettagliati (storie). Numerosi autori hanno suggerito modelli che possono essere applicati a questo scopo, che vanno dalla riduzione dell'elenco delle caratteristiche alla restrizione delle variazioni del business o dei canali di input [Leffl2010]. Si noti che anche le storie a grana fine dovrebbero essere definite in modo tale da fornire un valore aggiunto ad almeno uno stakeholder.

Gli elementi di Backlog più piccoli possono essere raggruppati in blocchi o processi più grandi (ad esempio, casi d'uso) per maggiore chiarezza e visualizzati graficamente utilizzando una story map. Questo aiuta a mantenere una visione d'insieme dei requisiti e, allo stesso tempo, a presentare la strategia a medio termine assegnando i requisiti agli sprint e ai rilasci.

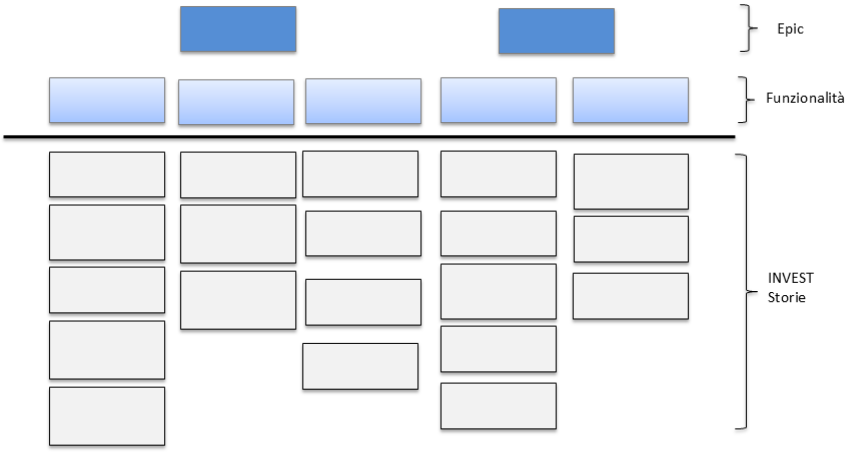

Figura 6: Suddivisione e raggruppamento

3.5 Sapere quando fermarsi

Obiettivi didattici

EO 3.5.1 Applicare l'affinamento dei requisiti (L4)
EO 3.5.2 Applicare Quality Assurance delle storie in un contesto agile (L4)

Contenuto

Il Product Owner è responsabile di continuare le discussioni con gli sviluppatori fino a quando entrambe le parti non hanno una comprensione comune dei requisiti Meye2014].

Per questo livello di comprensione comune, viene definita la Definition of Ready (DoR), che può essere utilizzata per la quality assurance degli elementi del Backlog per garantire che sia stata raggiunta una comprensione comune e un livello di dettaglio sufficiente.

3.6 Documentazione di progetto e di prodotto dei requisiti

Obiettivi didattici

EO 3.6.1 Comprendere come distinguere tra informazioni/documentazione relative al progetto e al prodotto (L2)
EO 3.6.2 Conoscere metodi e tecniche per conservare le informazioni per un uso futuro (L1)

Contenuto

Un Product Backlog può essere considerato come un sostituto del documento dei requisiti di un progetto tradizionale. Tuttavia, è importante ricordare che la parte scritta di un elemento del Backlog (ad esempio, utilizzando il modello di user story: "Come utente, voglio...") è incompleta fino a quando non si è discusso di tale elemento del Backlog.

Spesso è meglio considerare la parte scritta come un riferimento a una rappresentazione più precisa di questo requisito. Gli elementi del Backlog (Epics, Features, Stories, ...) potrebbero rimandare a un diagramma che descrive un flusso di lavoro, a un foglio di calcolo che illustra come eseguire un calcolo o a qualsiasi altro artefatto desiderato dal Product Owner o dal team.

Oltre al Product Backlog, possono esserci diverse ragioni per una documentazione dettagliata dei requisiti, ad esempio per scopi di comunicazione, di riflessione, legali o di archiviazione.

La definizione di un livello adeguato di documentazione dipende da numerosi fattori, quali l'ambito del progetto, il numero di stakeholder coinvolti, i vincoli legali e/o gli aspetti critici per la sicurezza del progetto. Sulla base di questi fattori, i team in un ambiente agile cercano di evitare un eccesso di documentazione e di trovare un minimo comune denominatore per la documentazione utile.

Lavorare con un Product Backlog "vivente" è un modo efficiente di gestire la documentazione, ma non sempre è sufficiente. Una documentazione strutturata e aggiornata di tutti i requisiti implementati in un prodotto non solo è obbligatoria per legge in alcuni progetti, ma costituisce anche un punto di partenza ideale per identificare più rapidamente le richieste di modifica sulla base della documentazione esistente.

4 Gestione dei requisiti e dei vincoli di qualità (L3)

Durata: 90 minuti + 30 minuti di esercitazione

Termini: Requisiti di qualità, Vincoli, Albero della qualità, Definition of Done (DoD), Criteri di accettazione

Contenuto

Questa EU esamina i requisiti di qualità e i vincoli nei progetti agili. Anche se il termine "requisiti non funzionali" è ancora spesso usato nella pratica come termine generico, noi utilizziamo le categorie più concrete e precise di "requisiti di qualità" e "vincoli", secondo [Glinz2024].

4.1 Comprendere l'importanza dei requisiti di qualità e dei vincoli (L2)

Obiettivi didattici

EO 4.1.1 Comprendere l'importanza dei requisiti di qualità in un contesto agile (L2)
EO 4.1.2 Comprendere gli schemi di categorizzazione dei requisiti e dei vincoli di qualità (L2)

Contenuto

Molti metodi agili si concentrano esclusivamente sui requisiti funzionali e non danno sufficiente importanza alle qualità e ai vincoli Meye2014].

I vincoli e le qualità chiave previsti per il sistema dovrebbero essere resi espliciti nelle prime fasi del ciclo di vita di un prodotto, poiché determinano le scelte architetturali fondamentali (infrastruttura, architettura software e progettazione software). Ignorarli o scoprirli troppo tardi nel corso del progetto può mettere a rischio l'intero sforzo di sviluppo. Altre qualità possono essere acquisite in modo iterativo, just in time, come nel caso dei requisiti funzionali Meye2014].

Gli schemi di categorizzazione per i requisiti e i vincoli di qualità (ad esempio [RoRo2013], [ISO25010]) possono essere utilizzati come liste di controllo per non dimenticare categorie importanti.

4.2 Aggiungere precisione ai requisiti di qualità (L2)

Obiettivi didattici

EO 4.2.1 Comprendere la suddivisione o la scomposizione dei requisiti e dei vincoli di qualità (L2)
EO 4.2.2 Comprendere la derivazione dei requisiti funzionali dai requisiti di qualità (L2)
EO 4.2.3 Comprendere la specifica dei criteri di accettazione per i requisiti di qualità (L2)
EO 4.2.4 Comprendere il valore aggiunto degli alberi di qualità (L2)

Contenuto

Inizialmente i requisiti di qualità sono spesso volutamente vaghi. Devono essere acquisiti nel loro formato vago come punto di partenza. I requisiti e i vincoli di qualità vaghi possono essere perfezionati in requisiti più precisi. A volte da essi derivano requisiti funzionali concreti.

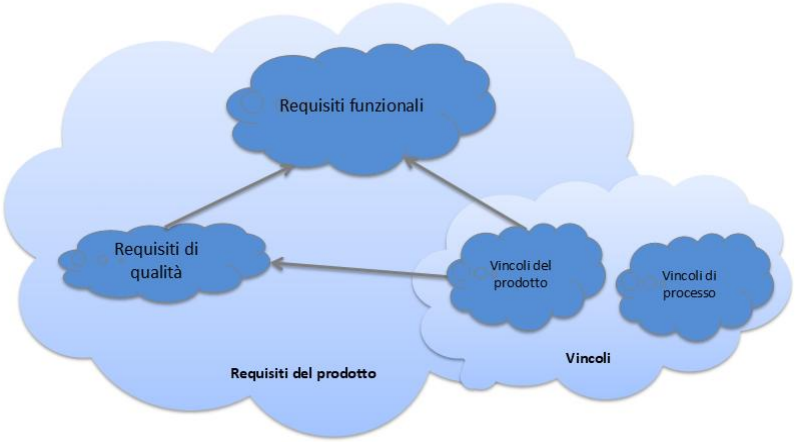

Figura 7: Scomposizione dei requisiti e dei vincoli di qualità

Scomporre (o dettagliare) un requisito di qualità significa specificare i requisiti di qualità a un livello di dettaglio inferiore, ad esempio utilizzando le generalizzazioni negli schemi di categorizzazione come "usabilità" e rendendoli più precisi attraverso la scoperta dei requisiti di "facilità d'uso" e "facilità di apprendimento".

Derivare significa che i requisiti di qualità possono essere raggiunti definendo requisiti funzionali, cioè suggerendo funzioni che raggiungano la qualità o i vincoli desiderati. Un esempio di affinamento di un requisito di sicurezza è l'introduzione di un concetto di ruolo e di password.

Anche gli alberi della qualità ([BOSS2022], [CleA2001]) sono un modo collaudato per strutturare i requisiti di qualità.

È inoltre necessario definire criteri di accettazione per i requisiti di qualità al fine di renderli testabili in una fase successiva, proprio come altri tipi di requisiti [PoRu2021]. Il tipo di criteri di accettazione utilizzati dipenderà dalla categoria della qualità.

4.3 Requisiti di qualità e Backlog (L3)

Obiettivi didattici

EO 4.3.1 Applicare l'integrazione dei requisiti di qualità nei requisiti funzionali (L3)
EO 4.3.2 Applicare la creazione di elementi separati nel Backlog per i requisiti di qualità (L3)
EO 4.3.3 Comprendere i requisiti di qualità come parte del DoD (L2)
EO 4.3.4 Comprendere la differenza tra requisiti di qualità e criteri di accettazione (L2)

Contenuto

I requisiti di qualità generalizzati devono essere collegati a requisiti funzionali più specifici [PoRu2021], ad esempio, un throughput quantificabile associato a una storia o caratteristiche di sicurezza specifiche associate a un epic.

Altre qualità, ad esempio scalabilità, manutenibilità o robustezza, dovrebbero essere comunicate allo sviluppo e verificate in ogni iterazione. Un modo comune per raggiungere questo obiettivo è includerle nella Definition of Done. Questo è spesso supportato dal testing automatizzato [Leffl2010].

Un altro approccio consiste nel registrare separatamente (al di fuori del Product Backlog) tali qualità per mantenerle visibili ai team, ad esempio sotto forma di elenco comune o di checklist. Questi requisiti hanno tutti la stessa importanza (poiché devono essere tutti soddisfatti) [Leffl2010].

È inoltre buona prassi rendere visibili le relazioni tra i requisiti funzionali e i requisiti di qualità che ne risultano impattati, predisponendo una matrice a parete e indicando la relazione «impattato da» con delle marcature nelle celle pertinenti.

Quando strutturano un Backlog, i Product Owner si trovano spesso di fronte al dubbio se un requisito di qualità identificato/emerso sia davvero un requisito di qualità, un criterio di accettazione oppure, eventualmente, un criterio di accettazione relativo a un requisito di qualità.

- I requisiti di qualità si riferiscono a aspetti qualitativi che non sono coperti dai requisiti funzionali. Ad esempio prestazioni, disponibilità, manutenibilità, sicurezza o affidabilità.
- I criteri di accettazione sono criteri che un requisito (che può essere un requisito funzionale o un requisito di qualità) deve soddisfare per essere accettato dagli stakeholder.

Possiamo vedere che sia i requisiti funzionali che quelli di qualità possono e devono avere criteri di accettazione.

4.4 Rendere espliciti i vincoli (L2)

Obiettivi didattici

EO 4.4.1 Comprendere i diversi tipi di vincoli in un contesto Agile (L2)
EO 4.4.2 Saper caratterizzare i vincoli (L1)

Contenuto

I vincoli sono un tipo importante di requisiti che limitano le scelte progettuali degli sviluppatori [Glinz2024]. I vincoli possono essere classificati come vincoli di prodotto o di processo. I vincoli del prodotto includono l'uso richiesto di tecnologie, il riutilizzo di componenti esistenti, le decisioni di acquisto o di vendita o le risorse sotto forma di materiali, conoscenze e competenze. I vincoli di processo, invece, sono definiti dai processi organizzativi o di sviluppo. Questi includono politiche e regolamenti organizzativi, limiti finanziari, norme e standard, regolamenti di conformità e audit, vincoli legali e governativi.

È importante rendere espliciti tali vincoli in modo che tutti i membri del team ne siano consapevoli. Quelli più limitanti dovrebbero essere noti fin dalle prime fasi del progetto. Gli altri dovrebbero emergere non appena vengono scoperti. In generale, i vincoli di solito riguardano diversi requisiti funzionali. Ciò solleva la questione di come i vincoli (così come i requisiti di qualità) debbano essere documentati. Abbiamo già spiegato le possibilità in merito nel capitolo 0. Lo stesso vale per i vincoli: come minimo, nella Definition of Done dovrebbe essere incluso un controllo per assicurare che siano stati rispettati.

Si noti che non è necessario scrivere tutti i vincoli come elementi del Backlog. Può essere sufficiente informare il team che, ad esempio, C# e ORACLE sono vincoli non negoziabili.

Tali vincoli sono normalmente applicabili a una gamma più ampia di progetti. Una volta registrati, possono essere facilmente riutilizzati in altri progetti o prodotti.

5 Priorità e stima dei requisiti (L3)

Durata: 120 minuti + 90 minuti di esercitazione

Termini: Valore di business, MVP, MMP, Planning Poker, Cono di incertezza, Velocity, Dimensionamento, Storie di riferimento, T-Shirt size, sequenza di Fibonacci

Contenuto

Anche in un mondo agile perfetto sono necessarie delle previsioni per determinare quanto lavoro può essere "svolto" entro un'iterazione (timebox) precedentemente specificata. Inoltre, le organizzazioni di sviluppo con più di un team di solito hanno bisogno di previsioni per assegnare le priorità e pianificare correttamente il lavoro.

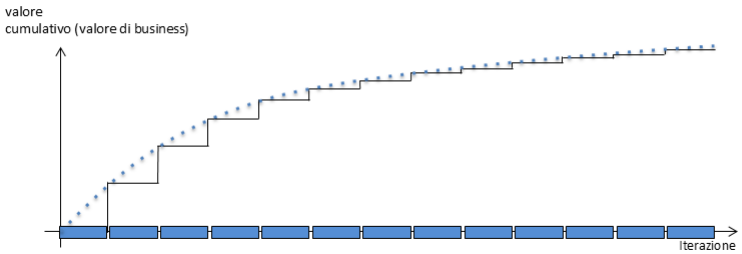

Figura 8: Crescita del valore di business

5.1 Determinazione del valore di business

Obiettivi didattici

EO 5.1.1 Comprendere la determinazione del valore di business (L2)
EO 5.1.2 Comprendere come utilizzare il valore di business per ordinare le voci nel Backlog (L2)
EO 5.1.3 Applicare metodi alternativi di calcolo del valore di business (L3)
EO 5.1.4 Comprendere come allineare la misurazione del valore di business agli obiettivi strategici dell'organizzazione (L2)

Contenuto

Gli approcci Agile mirano a massimizzare il valore di business complessivo e a ottimizzare in modo permanente l'intero processo di creazione del valore di business [Leffl2010]. In un Product Backlog, tutti i requisiti (sia grossolani che dettagliati) devono essere ordinati principalmente in base al valore che forniscono all'organizzazione. Un prerequisito per farlo è una definizione concordata di cosa sia il valore di business per questo prodotto/azienda.

Il valore di business non è definito solo dal profitto: calcoli alternativi includono il ritorno sull'investimento, il periodo di recupero dell'investimento, il valore attuale netto, il Weighted Shortest Job First (WSJF), il costo del ritardo e la Balanced Scorecard. Il valore di mercato, il time to market e la riduzione dei rischi potenziali rappresentano tutti potenzialmente tipi di valore di business, così come l'eccellenza operativa e organizzativa [Rein2009].

In effetti, la definizione di valore di business può essere diversa in ogni organizzazione, in ogni progetto e dal punto di vista dei diversi stakeholder. I professionisti dovrebbero comprendere come allineare la misurazione del valore di business agli obiettivi strategici dell'organizzazione ed essere in grado di adattare questo allineamento al mutare di tali obiettivi.

5.2 Valore di business, rischi e dipendenze (L3)

Obiettivi didattici

EO 5.2.1 Comprendere le dipendenze tra il potenziale valore di business e i rischi correlati (L2)

Contenuto

Molto spesso il valore di business potenziale e i rischi sono interdipendenti. Concentrarsi su un valore di business specifico potrebbe far sorgere rischi specifici, mentre cambiare il focus del valore di business potrebbe cambiare anche i rischi [Rein2009].

In ogni caso, l'ordine dei requisiti dovrebbe essere modificato in linea con la strategia selezionata, tenendo conto delle dipendenze tra i requisiti.

5.3 Esprimere le priorità e ordinare il Backlog

Obiettivi didattici

EO 5.3.1 Assegnare la priorità agli elementi del Backlog (L3)
EO 5.3.2 Applicare diverse strategie di base per la definizione delle priorità (L3)
EO 5.3.3 Applicare la determinazione delle dipendenze tra i requisiti (L3)
EO 5.3.4 Comprendere la sequenza degli elementi del Backlog in base alle loro dipendenze (L2)

Contenuto

Una volta determinato il significato di valore per voi, dovete esprimere queste priorità e ordinare il Backlog in base alle priorità assegnate alle voci del Backlog. Esistono molti metodi diversi per assegnare un valore alle voci del Backlog. Alcune sono molto semplici, altre molto complesse. Esempi di strategie di prioritizzazione includono:

- MoSCoW
- Alta/Media/Bassa
- Sulla base di una serie di numeri che esprimono il valore di business
- Basato sulla valutazione del business in base a più criteri

Voci del Backlog	Fatturato nel Q2		Ridurre al minimo il rischio tecnico		Aumentare l'usabilità		Risultato complessivo
	Peso = 5	Valore	Peso= 4	Valore di rischio	Peso= 2	Valore di usabilità	
Requisito 1	3	15	0	0	0	0	**15**
Requisito 2	0	0	3	12	1	2	**14**
Requisito 3	4	20	2	8	2	4	**32**
Requisito 4	2	10	2	8	3	6	**24**
Requisito 5	0	0	2	8	5	10	**18**
...							
...							

Figura 9: Priorità e ordine del Backlog

Quando si assegna una priorità agli elementi del Backlog, in genere si dovrebbe considerare che più si avvicina il momento dell'implementazione pianificata di un elemento del Backlog, più chiare dovrebbero essere le priorità degli elementi selezionati. Una prioritizzazione con i criteri "alto", "medio", "basso" spesso porta a un numero eccessivo di elementi del Backlog che ricevono il valore "alto". Ad esempio, se al 30% di tutte le voci del Backlog viene assegnata un'alta priorità, il risultato è che gli sviluppatori non sanno cosa sia più importante per il Product Owner. Ciò indica anche che il Product Owner non ha una strategia chiara per l'implementazione a breve e medio termine. L'obiettivo di una prioritizzazione dovrebbe sempre essere quello di fornire una chiara indicazione di ciò che gli stakeholder possono aspettarsi come valore del prodotto nel prossimo futuro.

5.4 Stima degli elementi del Backlog (L3)

Obiettivi didattici

EO 5.4.1 Applicare previsioni e stime (L3)
EO 5.4.2 Comprendere come ricavare una previsione a medio termine (L2)
EO 5.4.3 Comprendere il vantaggio delle stime relative, categorizzanti e di gruppo (L2)
EO 5.4.4 Comprendere le tecniche di stima (L2)

Contenuto

Le stime iniziali di un progetto sono spesso imprecise. Diventano sempre più precise man mano che l'attività viene ripetuta (un principio noto come Cono dell'incertezza). Analizzando ciò che è stato consegnato nelle iterazioni precedenti, è possibile calcolare la velocity del team. In questo modo è possibile stimare meglio la capacità per le iterazioni future.

Per una migliore stima a medio termine, i requisiti di grandi dimensioni, come epic, vengono suddivisi in features. In questo modo è possibile effettuare stime a livello di feature e poi sommarle nell'epic. Sebbene queste stime non siano ancora molto precise, sono comunque utili per la stima dell'epic. Inoltre, servono come una sorta di test della conoscenza dell'epic (ipotesi, ecc.).

I metodi Agile definiscono regole che aiutano a fare stime migliori e più accurate:

- Tutti coloro che partecipano alla stima devono avere la stessa comprensione del lavoro che deve essere "fatto" (done).
- Le stime devono essere eseguite da coloro che svolgono il lavoro, ovvero il team interfunzionale (i developer in Scrum). Ciò contribuisce ad allineare tutte le persone coinvolte sullo stesso livello di conoscenza, attraverso lo scambio di informazioni e ipotesi sul lavoro da svolgere.
- Le stime dovrebbero essere fatte in modo relativo / rispetto al lavoro già svolto (stima per analogia), poiché queste stime hanno maggiori probabilità di essere accurate rispetto alle stime assolute.
- Le stime dovrebbero essere effettuate in un'unità artificiale che rappresenti l'effort, la complessità e il rischio in un unico valore.

Diverse tecniche supportano la stima relativa. Le più note sono il Planning Poker [Cohn2006] e la Magic Stimulation.

5.5 Scegliere una strategia di sviluppo (L2)

Obiettivi didattici

EO 5.5.1 Comprendere il concetto di MVP (minimum viable product) (L2)
EO 5.5.2 Comprendere il concetto di MMP (minimum marketable product) (L2)

Contenuto

È possibile applicare diverse strategie nella selezione degli elementi da includere nelle prime versioni, in base al valore noto, al rischio e allo sforzo necessario per sviluppare un elemento del Backlog. Due concetti sono tipici dello sviluppo agile: lo sviluppo di un prodotto minimo funzionante (MVP) e lo sviluppo di un prodotto minimo commerciabile (MMP).

Un prodotto minimo funzionante è la versione di un nuovo prodotto che consente a un team di raccogliere la massima quantità di apprendimento convalidato dei clienti con il minimo sforzo. L'MVP è l'idea centrale della metodologia Lean Startup sviluppata da Eric Ries, che si basa sul ciclo Build-Measure-Learn. L'MVP non è necessariamente un prodotto software distribuibile.

L'MMP descrive il prodotto con il minor numero possibile di funzionalità che soddisfa le esigenze dei primi utenti (innovatori e early adopters) e può quindi essere commercializzato.

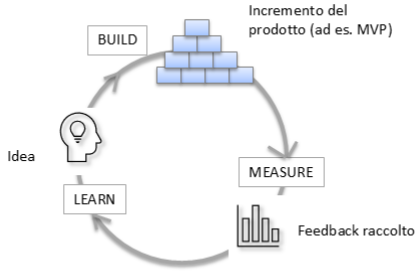

Figura 10: Il ciclo "Build-Measure-Learn" del lean development

6 Scalare RE@Agile (L2)

Durata: 105 minuti

Termini: Scalabilità dei framework, requisiti e team di scalabilità, strutturazione dei requisiti e dei team su larga scala, roadmap e pianificazione su larga scala, validazione del prodotto

Contenuto

Il RE è più facile per i prodotti che sono abbastanza piccoli da poter essere gestiti da un unico team in un'unica sede.

In questa unità didattica discutiamo perché lo sviluppo dei prodotti a volte deve essere scalato e perché i prodotti devono essere sviluppati da più di un team, sia nella stessa sede che distribuiti geograficamente. Durante la scalabilità, il Product Owner del prodotto complessivo (in quanto responsabile della gestione dei requisiti) è probabilmente più impegnato dagli aspetti gestionali che da quelli relativi ai requisiti. Discuteremo del fatto che i due fattori time to market e complessità (sia complessità funzionale che requisiti di qualità impegnativi) giustificano e guidano il processo di scalabilità. Tuttavia, anche i vincoli organizzativi e tecnici influenzano il modo in cui avviene la scalabilità.

Le seguenti domande sono importanti:

- Cosa significa scalare e come influisce sui requisiti e sui team?
- Come (ri)organizziamo i requisiti e i team su larga scala?
- Come vengono definiti e utilizzati i rilasci e le roadmap nella pianificazione a lungo termine?
- Come vengono eseguite le operazioni di validazione dei requisiti in ambienti scalati?

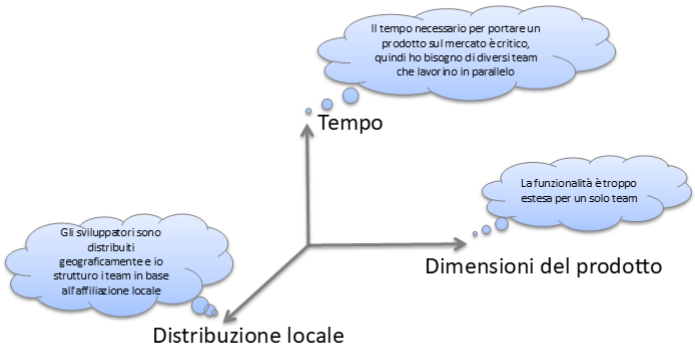

Figura 11: Motivi per scalare

6.1 Requisiti di scalabilità e team (L2)

Obiettivi didattici

EO 6.1.1 Conoscere esempi comuni di framework di scalabilità (L1)
EO 6.1.2 Comprendere le sfide e il meccanismo di scalabilità dei requisiti in un contesto agile (L2)

Contenuto

Utilizziamo il termine "scalabilità" per descrivere un cambiamento nelle dimensioni, sia del sistema che del prodotto, o del numero di persone coinvolte.

A partire dal 2010 circa, sono stati sviluppati diversi framework di scalabilità agile per affrontare queste problematiche. Tra questi, Nexus [Nexu2021], SAFe [SAFe2021a] [SAFe2021b], LeSS [LeSS], Scrum@Scale [S@SG2021], BOSSA Nova [BOSS2022], Scrum of Scrums [KnIvS2012] e Spotify [Spot2012], anche se ne esistono altri. I framework di scalabilità variano in termini di livello di maturità, numero di buone pratiche, linee guida e regole, nonché grado di adattabilità alle esigenze specifiche di un'organizzazione.

Quando si scala, due cose saranno sempre vere: si sarà costretti ad aggiungere una gerarchia ai requisiti e una gerarchia all'organizzazione. I requisiti a grana grossa sono necessari quando si discute del prodotto nel suo complesso; i requisiti a grana fine saranno necessari nei team che implementano alcuni aspetti del prodotto. E i team stessi dovranno organizzare la loro cooperazione per funzionare con successo all'interno di un team più grande.

6.2 Criteri per la strutturazione dei requisiti e dei team su larga scala (L2)

Obiettivi didattici

EO 6.2.1 Conoscere le sfide legate all'organizzazione di un Backlog e alla comunicazione dei requisiti all'interno di una rete di team (L1)

EO 6.2.2 Comprendere la suddivisione dei requisiti in diversi contesti (di progetto) del contesto agile (L2)

Contenuto

Nello sviluppo di prodotti su larga scala, nella maggior parte dei casi più team devono lavorare insieme sullo stesso prodotto. In pratica, ogni team sviluppa una parte specifica del prodotto che deve essere integrata con altre parti nel prodotto complessivo per creare una soluzione funzionante. Solo il prodotto integrato ha valore per gli stakeholder.

Quando si scala lo sviluppo del prodotto su più team, non è sufficiente che tutti i Product Owner si incontrino e discutano in qualche modo su quali team dovrebbero sviluppare quale parte del prodotto, per poi sperare per il meglio! Sono necessarie strutture e pratiche sofisticate per supportare la collaborazione tra i team, gestire le modifiche dei requisiti e consentire una rapida consegna del prodotto. In caso contrario, gli sviluppatori potrebbero sprecare energie nel coordinamento con team che non sono rilevanti per il loro lavoro.

Dal punto di vista dei requisiti, dobbiamo chiudere il cerchio: dai requisiti (di business) iniziali richiesti dagli stakeholder, passando per la suddivisione dei requisiti complessi in parti più piccole gestibili dagli sviluppatori, fino ad arrivare alla garanzia che i risultati assemblati si combinino per formare una soluzione che possa essere rilasciata sul mercato.

Per poter lavorare sui requisiti in modo collaborativo e prendere decisioni ragionevoli in modo autonomo, i team devono avere una comprensione generale dei requisiti degli altri team con cui devono collaborare, senza però essere sopraffatti da tutti i dettagli.

I Product Owner dovrebbero quindi trovare un livello di dettaglio adeguato, sufficiente affinché i team possano comprendere l'impatto delle loro decisioni sugli altri team.

Per fornire incrementi di prodotto distribuibili con dipendenze minime da altri team, i team in un ambiente agile dovrebbero lavorare su funzionalità end-to-end debolmente accoppiate. Nel nostro contesto, il termine "funzionalità end-to-end" si riferisce a un insieme di funzioni coerenti che svolgono un compito specifico che fornisce valore di business agli stakeholder. I casi d'uso sono un approccio alla strutturazione dei requisiti, non sempre tipicamente associato ad Agile, ma comunque raccomandato da diversi autori (ad esempio [Jaco2011], Cockburn, [Leffl2010]).

Purtroppo, in molti casi non è così semplice scomporre i requisiti basati su unità di funzionalità end-to-end a basso accoppiamento. A causa della progettazione architetturale (ad esempio tecnologia, infrastruttura, componenti di sistema, piattaforma comune, livelli architetturali come front-end e back-end) e di considerazioni organizzative (competenze specialistiche, ubicazione del team, subappaltatori), le unità di funzionalità possono sovrapporsi. Ciò significa che team diversi in un ambiente agile devono lavorare insieme per implementare caratteristiche specifiche e che i rispettivi Product Owner devono collaborare più strettamente sui requisiti. In alternativa, è possibile creare un team dedicato che si occupi specificamente della sovrapposizione e collabori con ciascuno dei team originali focalizzati su un'unità di funzionalità.

6.3 Roadmap e pianificazione su larga scala (L2)

Obiettivi didattici

EO 6.3.1 Conoscere la differenza tra una roadmap e un Backlog in un contesto agile (L1)
EO 6.3.2 Comprendere la creazione e la gestione di una roadmap in un contesto agile (L2)

Contenuto

Nello sviluppo di prodotti su larga scala, i Product Owner gestiscono i requisiti nel Backlog incentrato sul prodotto. A differenza del Backlog, una roadmap viene utilizzata per pianificare lo sviluppo del prodotto in modo incrementale. Una roadmap è una previsione di come il prodotto crescerà [Pich2016]. Le roadmap non modificano il contenuto degli elementi del Backlog, ma li organizzano su una linea temporale. Rispondono alla domanda su quando possiamo aspettarci approssimativamente quali funzionalità.

Una roadmap è un mezzo utile per comunicare obiettivi e decisioni (strategiche) agli sviluppatori e agli altri stakeholder. Suddivide un obiettivo a lungo termine in iterazioni gestibili, rappresenta le dipendenze tra i team e fornisce indicazioni e trasparenza agli stakeholder.

Una roadmap mostra gli obiettivi strategici, le tappe fondamentali e i requisiti generali. Le milestone importanti possono essere interne o determinate da eventi esterni, come una fiera o l'introduzione di una nuova normativa sul mercato. La rappresentazione di una roadmap dipende dallo scopo, dal gruppo target e dall'orizzonte di pianificazione.

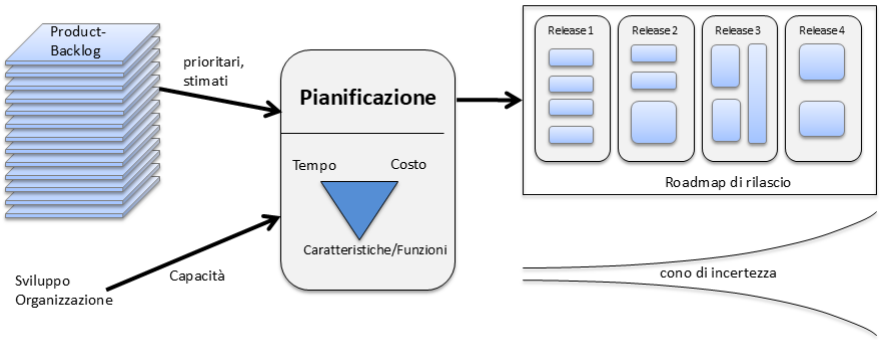

Figura 12: Roadmap e pianificazione su larga scala

Per sviluppare una roadmap di prodotto a lungo termine, un Product Owner deve prima definire una vision e una strategia di prodotto. Successivamente, i Product Owner devono individuare i requisiti generali coinvolgendo gli stakeholder necessari. In questa fase non è necessario investire tempo sui requisiti dettagliati. Sebbene i requisiti siano soggetti a un alto livello di incertezza in questa fase, la roadmap di prodotto come piano di iterazioni approssimativo e preliminare è sufficiente per supportare la pianificazione e la sincronizzazione.

6.4 Validazione del prodotto (L2)

Obiettivi didattici

EO 6.4.1 Comprendere metodi concreti per la validazione dei requisiti di prodotto in un contesto agile (L2)

Contenuto

Un'idea chiave dello sviluppo agile è quella di sviluppare una piccola parte del prodotto, generare feedback coinvolgendo gli stakeholder e adattare lo sviluppo del prodotto in base alle scoperte e alle intuizioni acquisite. Pertanto, seguendo il principio del ciclo Build-Measure-Learn (Costruire-Misurare-Imparare) [Ries2011], la validazione del prodotto diventa una fase importante per ottenere un rapido feedback. Ogni volta che viene rilasciato un nuovo incremento di prodotto, i Product Owner lo utilizzano per verificare il suo valore commerciale e per esaminare se i requisiti del prodotto sono stati compresi correttamente.

La validazione a livello di prodotto (ad esempio, la review dello sprint in Scrum) è un metodo importante nello sviluppo di prodotti su larga scala, poiché garantisce che i proprietari dei prodotti condividano la piena responsabilità, dai requisiti di business all'integrazione dei prodotti. È l'intero prodotto ad avere valore per gli stakeholder, non solo sue piccole parti.

Un altro approccio per la validazione nello sviluppo di prodotti su larga scala è quello basato sull'analisi dei dati [MaeA2016]. L'incremento del prodotto integrato viene consegnato agli utenti e, in base al loro comportamento, si misura se le caratteristiche del prodotto hanno un impatto positivo, neutro o negativo.

7 Definizioni di termini, glossario

Per le definizioni dei termini, rimandiamo il lettore al glossario IREB CPRE Requirements Engineering [Glinz2024], che non solo è un glossario completo della terminologia dell'ingegneria dei requisiti, ma definisce anche molti termini del campo dell'agilità. Per i termini specifici dell'agilità, il lettore può consultare l'attuale Guida Scrum [ScSu2020].

8 Riferimenti Bibliografici

[Alex2005] Alexander, I. F.: A Taxonomy of Stakeholders – Human Roles in System Development. International Journal of Technology and Human Interaction, Vol. 1, 1, 2005, pagg.23-59.

[BOSS2022] https://www.agilebossanova.com/#bossanova. Ultima visita marzo 2025.

[CleA2001] P. Clements et al.: Evaluating Software Architectures, SEI Series in Software Engineering, 2001

[Cohn2010] Cohn, M.: User Storys für die agile Software-Entwicklung mit Scrum, XP u.a., mitp, 2010

[Cohn2006] Cohn, M.: Agile Estimation and Planning, Addison Wesley, 2006

[Dora1981] Doran, G. T.: There's a S.M.A.R.T. way to write management's goals and objectives. AMA FORUM. 70 (11): 35–36 1981.

[Glinz2024] Glinz, M.: A Glossary of Requirements Engineering Terminology. Standard Glossary for the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version 2.1.1, 2024. https://cpre.ireb.org/en/knowledge-and-resources/downloads#cpre-glossary. Ultima visita marzo 2025.

[HeHe2011] Heath, C., Heath, D.: Switch: Veränderungen wagen und dadurch gewinnen. Crown Business, 2010

[High2001] Highsmith, J.: Design the Box. Agile Project Management E-Mail Advisor 2001, http://www.joelonsoftware.com/articles/JimHighsmithonProductVisi.html. Ultima visita marzo 2025.

[Hrus2017] https://b-agile.de/downloads/articles/story_splitting.pdf. Ultima visita marzo 2025.

[ISO25010] ISO/IEC 25010:2023: Systems and software engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and software quality models: https://www.iso.org/standard/78176.html. Ultima visita marzo 2025.

[Jaco2011] https://www.ivarjacobson.com/publications/white-papers/use-case-ebook. Ultima visita marzo 2025.

[Leffl2010] Leffingwell, D.: Agile Software Requirements – Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison Wesley, 2010.

[LeSS] Large-Scale Scrum: https://less.works/. Ultima visita marzo 2025.

[MaeA2016] Maalej, W., Nayebi, M., Johann T., Ruhe, G.: Toward Data-Driven Requirements Engineering. IEEE Software (Volume 33, Issue 1), 2016.

Meye2014] Meyer, B.: Agile! The Good, the Hype and the Ugly, Springer, 2014.

[Nexu2021] https://www.Scrum.org/resources/nexus-guide. Ultima visita marzo 2025.

[Pich2016] Pichler, R.: Strategize – Product Strategy and Product Roadmap Practices for the Digital Age, Pichler Consulting 2016.

[PoRu2021] Pohl, K., Rupp, C.: Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt.verlag, 5. Auflage, 2021.

[Prim2017] CPRE RE@Agile Primer https://cpre.ireb.org/en/downloads-and-resources/downloads#cpre-agile-primer-syllabus-and-study-guide. Ultima visita marzo 2025.

[Rein2009] Reinertsen, D.G.: The Principles of Product Development Flow – Second Generation Lean Product Development. Celeritas Publishing, 2009.

[Ries2011] Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Crown Business, New York, NY 2011.

[RoRo2013] Robertson S. Robertson J.: Mastering the Requirements Process – Getting Requirements Right, 3a edizione, Addison Wesley, 2013.

[Robe2003] Robertson, S.: Stakeholders, Goals, Scope: The Foundation for Requirements and Business Models, 2003, https://www.volere.org/wp-content/uploads/2018/12/StkGoalsScope.pdf. Ultima visita marzo 2025.

[SAFe2021a] https://www.scaledagileframework.com/roadmap/. Ultima visita marzo 2025.

[SAFe2021b] https://www.scaledagileframework.com/pi-planning/. Ultima visita marzo 2025.

[SAFe2021] https://www.scaledagileframework.com/safe-requirements-model/. Ultima visita marzo 2025.

[S@SG2021] Sutherland, J. e Scrum, Inc: Scrum@Scale Guide: https://www.Scrumatscale.com/Scrum-at-scale-guide/. Ultima visita marzo 2025.

[KnIvS2012] Kniberg, H.; Ivarsson, A.: Scaling Agile @ Spotify with Tribes, Sqads, Chapters & Guilds. https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf. Ultima visita marzo 2025.

[Spot2012] https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf. Ultima visita marzo 2025.

[ScSu2020] Schwaber, K., Sutherland, J.: The Scrum Guide. 2020 https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf. Ultima visita marzo 2025.

[Wake2003] Wake, B.: INVEST in Good Stories, and SMART Tasks, 2003, http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/. Ultima visita marzo 2025.