



Certified Professional for Requirements Engineering

Syllabus

Requirements Management

(Gestione dei Requisiti)

Practitioner | Specialist

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 come materiale pubblicitario deve chiedere un consenso scritto a IREB per l'uso a tale scopo.
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

Questo Syllabus è stato creato da (in ordine alfabetico): Stan Bühne, Frank Engel, Sven Eselgrimm, Günter Halmans, Andrea Herrmann, Frank Houdek, Patrick Mäder, Alexander Rachmann, Thomas Schölzl, Amin Soesanto, Frank Stöckel e Malik Tayeh.

Traduzione italiana a cura di Cristina Sobrero. Revisione della traduzione a cura di Salvatore Reale.

Si ringraziano tutti coloro che hanno collaborato.

Copyright © 2015 – 2024 per questo Syllabus IREB Certified Professional for Requirements Engineering attribuito agli autori elencati sopra. I diritti sono stati trasferiti a IREB International Requirements Engineering Board e.V.

Scopo del documento

Questo syllabus definisce gli obiettivi educativi e una sintesi dei contenuti educativi per le certificazioni Requirements Management Practitioner e Specialist, istituite da 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 prepararsi all'esame.

Contenuto del Syllabus

Il modulo Requirements Management Advanced Level si rivolge a professionisti nel campo del Requirements Engineering, della Business Analysis, del Business Engineering (ingegneria del business), della progettazione organizzativa, ecc. che desiderano approfondire le proprie conoscenze e competenze nell'area del Requirements Management.

Obiettivo del Syllabus

A livello Practitioner e Specialist, come il Foundation Level, fornisce i principi del Requirements Engineering che sono ugualmente validi per qualsiasi sistema, come sistemi embedded, sistemi safety-critical, sistemi informativi tradizionali. Questo non significa che l'adeguatezza degli approcci alle singole aree non possa essere trattata in un corso di formazione che tenga conto delle caratteristiche specifiche di queste aree. Il Syllabus non ha quindi l'obiettivo di presentare aspetti di Requirements Engineering specifici per un determinato dominio.

I contenuti trattati in questo Syllabus possono essere utilizzati allo stesso modo nei progetti (di sviluppo) e nel product management; nell'evoluzione dei sistemi esistenti o nella gestione dei requisiti in modo continuo e trasversale ai progetti. Per motivi di semplicità, spesso vengono menzionati solo i progetti, invece di questo elenco completo.

Non si assume un particolare insieme di procedure o un modello di processo associato, che prescriva la pianificazione, la direzione e la sequenza dell'applicazione pratica dei concetti appresi.

L'obiettivo non è quello di promuovere un processo particolare per il Requirements Engineering, per il Software Engineering o per il System Engineering in generale. Tuttavia, in capitolo 10 vengono discussi aspetti specifici del Requirements Management nei progetti Agile.

Il Requirements Management senza l'uso di strumenti è difficile nella pratica. Le possibilità e le limitazioni del supporto di strumenti sono discussi nei singoli capitoli, senza focalizzarsi su uno strumento particolare.

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 formativi generali
- Contenuti con una descrizione degli obiettivi formativi
- Riferimenti a ulteriore documentazione (se necessari)

Obiettivi formativi/ livelli cognitivi di conoscenza

A ogni modulo e obiettivo formativo di questo Syllabus viene assegnato un livello cognitivo. I livelli sono classificati come segue:

- **L1: Conoscere** (descrivere, enumerare, caratterizzare, riconoscere, nominare, ricordare, ...) – Il candidato è in grado di ricordare o recuperare materiale appreso in precedenza.
- **L2: Capire** (spiegare, interpretare, completare, riassumere, giustificare, classificare, confrontare, ...) – Il candidato è in grado di cogliere/costruire il significato da materiale o situazioni specifiche.
- **L3: Applicare** (specificare, scrivere, progettare, sviluppare, implementare, ...) – Il candidato è in grado di applicare la conoscenza e le competenze in determinate situazioni.

- **L4: Analizzare** (indagare, trarre conclusioni, fornire argomentazioni, ...) – Il candidato è in grado di analizzare un problema dato, argomentare ciò che dovrebbe/potrebbe essere fatto, scomporre il problema in parti, applicare il pensiero critico, argomentare su cause ed effetti.
- **L5: Valutare** (criticare, giudicare) – Il candidato è in grado di fornire una critica ben argomentata di un artefatto specifico; esprimere un giudizio approfondito su un dato caso.

Si noti che un obiettivo formativo al livello cognitivo L_n contiene anche elementi di tutti i livelli cognitivi inferiori (da L₁ a L_{n-1}).

Esempio: Un obiettivo formativo del tipo "Applicare la tecnica di RE xyz" è al livello cognitivo (L3). Tuttavia, la capacità di applicare richiede che gli studenti conoscano prima la tecnica di RE xyz (L1) e capiscano per cosa viene utilizzata la tecnica (L2).



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

Il glossario è disponibile per il download nella homepage IREB all'indirizzo: <https://www.ireb.org/en/downloads/#cpre-glossary-2-0>

Questo Syllabus e il relativo handbook utilizzano l'abbreviazione "RE" per Requirements Engineering.

Struttura del syllabus

Il Syllabus consiste di undici capitoli principali. Ogni capitolo copre un'unità educativa (EU, Educational Unit). I titoli di ogni capitolo principale contengono il livello cognitivo del capitolo stesso, che corrisponde al livello più alto dei relativi sottocapitoli (o paragrafi). Viene inoltre suggerito il tempo minimo di insegnamento che un corso dovrebbe pianificare per quel capitolo. I Training Provider sono liberi di dedicare più tempo alle EU e alle esercitazioni, ma devono assicurare che le proporzioni tra le EU vengano mantenute. I termini importanti utilizzati in un capitolo sono elencati all'inizio del capitolo.

Esempio:

Capitolo 3 Assegnazione di Attributi e Viste ai Requisiti (L5)

Durata: 2 ore

Termini: Attributo, Schema di Attributo, Vista

Questo esempio mostra che il Capitolo 3 contiene obiettivi formativi di livello L5, e che due ore e mezza sono ritenute necessarie per insegnare il materiale di questo capitolo.

Ogni capitolo contiene dei sottocapitoli. Il titolo di ogni sottocapitolo include anche il livello cognitivo del relativo contenuto.

Gli obiettivi formativi (EO, Educational Objectives) sono elencati prima del testo. La numerazione mostra a quale sottocapitolo appartengono.

Example: EO 3.3.1

L'esempio mostra che l'obiettivo formativo EO 3.3.1 è descritto nel sottocapitolo 3.3

L'esame

Questo Syllabus copre le unità educative e gli obiettivi formativi per gli esami di certificazione di

- Requirements Management – Practitioner
- Requirements Management – Specialist

L'esame per il conseguimento della certificazione Requirements Management – Practitioner consiste in **un esame a scelta multipla**.

L'esame per il conseguimento della certificazione Requirements Management – Specialist consiste in un **compito scritto**.

Entrambi gli esami includono domande che coprono tutte le unità educative e tutti gli obiettivi formativi del Syllabus.

Ogni domanda d'esame può includere materiale relativo a più capitoli del Syllabus e relativo a più obiettivi formativi o parti di un obiettivo formativo.

L'esame a scelta multipla per la certificazione **Practitioner**

- Verifica tutti gli obiettivi formativi del Syllabus. Tuttavia, per gli obiettivi formativi 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 svolto subito dopo un corso, ma anche indipendentemente da esso (ad esempio, a distanza o presso un test center).

Il compito scritto per la **certificazione Specialist**

- Verifica tutti gli obiettivi formativi del Syllabus relativi ai livelli di conoscenza cognitiva indicati per ciascun obiettivo formativo.
- segue la descrizione dei compiti per il Requirements Management – Specialist –, presente all'indirizzo <https://www.ireb.org/en/downloads/tag:advanced-level-written-assignment#top>.
- Viene svolto in autonomia e viene sottoposto a un Organismo di Certificazione autorizzato.

I seguenti obiettivi formativi generici si applicano anche al **compito scritto** per la certificazione **Specialist**:

EO G1: Analizzare e illustrare i problemi di Requirements Management in un contesto che è familiare al candidato, o che è simile a tale contesto (L4).

EO G2: Valutare e riflettere sull'uso di pratiche, metodi, processi e strumenti di Requirements Management 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

Data	Versione	Autore	Descrizione della modifica
20 Aprile 2023	2.0.0	SBe, FH, StS	Versione iniziale basata sulla versione inglese 2.0.0
05 Aprile 2024	2.1.0	StB	La definizione dei livelli cognitivi di conoscenza è stata modificata in base allo standard IREB. EO 3.7, EO 7.2.3 Parola chiave modificata per corrispondere al livello cognitivo L4. EO 7.2.2 Parola chiave modificata per corrispondere al livello cognitivo L5.

Contenuti

1	Che cosa è il Requirements Management? (L2)	11
1.1	Definizione di Requirements Management (L1)	11
1.2	Compiti nel Requirements Management (L1)	12
1.3	Obiettivi e Vantaggi del Requirements Management (L2)	13
1.4	Requirements Management Plan (L2)	14
1.5	Standard Rilevanti (L1)	14
2	Requirements Information Model (L3)	16
2.1	Fondamenti (L2)	16
2.2	Forme di presentazione (L2)	18
2.3	Creazione di un Requirements Information Model (L3)	20
3	Assegnazione di attributi e viste ai requisiti (L5)	23
3.1	Obiettivi dell'Assegnazione di Attributi (L1)	23
3.2	Utilizzo di uno schema di attributi (L2)	24
3.3	Progettazione di uno schema di attributi (L3)	25
3.4	Change Management degli Schemi di attributi (L5)	27
3.5	Obiettivi e tipi di viste (L2)	28
3.6	Definizione di vista e rischi delle viste (L1)	29
3.7	Ottimizzare il popolamento degli attributi e la creazione di viste (L4)	30
4	Valutazione e prioritizzazione dei requisiti (L3)	32
4.1	Principi di valutazione (L1)	32
4.2	Prioritizzazione dei requisiti (L2)	33
4.3	Tecniche di prioritizzazione ad-hoc (L3)	34

4.3.1	Classificazione a due criteri (L3)	34
4.3.2	Tecnica dei 100 dollari (L3)	35
4.4	Tecniche di prioritizzazione analitica (L3)	35
4.5	Combinazione di tecniche di prioritizzazione (L2)	36
5	Version management e change management (L3)	37
5.1	Versioning dei requisiti (L2)	37
5.1.1	Version control dei requisiti e dei documenti di requisiti (L1)	37
5.1.2	Configurazione dei requisiti (L2)	38
5.1.3	Baseline dei requisiti (L1)	38
5.1.4	Branching dei requisiti (L2)	39
5.2	Change management dei requisiti (L2)	39
5.2.1	Cause, sorgenti e tempistiche delle modifiche dei requisiti (L2)	39
5.2.2	Tipi di modifiche ai requisiti (L2)	40
5.2.3	Analisi e documentazione della stabilità dei requisiti (L2)	40
5.3	Processo di change management (L3)	41
6	Tracciabilità dei requisiti (L3)	43
6.1	I Motivi della tracciabilità dei requisiti (L1)	43
6.1.1	Cosa Significa Tracciabilità dei Requisiti? (L1)	43
6.1.2	Perché la tracciabilità dei requisiti e degli artefatti è importante (L1) ...	44
6.2	Diverse viste della tracciabilità (L2)	44
6.3	Tipi di relazione per le relazioni di tracciabilità (L1)	46
6.4	Forme di rappresentazione delle relazioni di tracciabilità (L3) ...	48
6.4.1	Documentazione implicita ed esplicita della tracciabilità (L3)	48
6.4.2	Relazioni di tracciabilità bidirezionali e unidirezionali (L2)	48
6.4.3	Forme di presentazione delle relazioni di tracciabilità (L3)	49
6.5	Sviluppo di una strategia per la tracciabilità specifica del progetto (L3)	53
6.6	Creare e utilizzare modelli di tracciabilità specifici (L3)	54
6.6.1	Un processo per sefinire uno appecifico modello di tracciabilità	54
6.6.2	Utilizzo di uno Specifico Modello di tracciabilità	55

6.7	Misure per la valutazione della tracciabilità implementata (L2) ...	55
6.8	Sfide nella tracciabilità di artefatti non-testuali (L1)	56
7	Gestione delle varianti per i requisiti (L5)	58
7.1	Uso di varianti dei requisiti (L1)	58
7.2	Forme di documentazione Esplicita delle varianti e loro valutazione (L5).....	60
7.3	Modellazione delle funzionalità (L3)	64
8	Reporting nel Requirements Management (L3)	68
8.1	Obiettivi e benefici del reporting nel Requirements Management (L1)	68
8.2	Stabilire un sistema di reporting nel Requirements Management (L2)	69
8.2.1	Interfacce (L2)	69
8.2.2	Contenuto di un report (L1)	69
8.2.3	Suggerimenti per lo sviluppo e l'applicazione del reporting (L2).....	70
8.2.4	Processo di definizione dei report (L2)	71
8.3	Figure chiave nel Requirements Engineering (L3)	71
8.3.1	Figure chiave nel Requirements Management (L1).....	71
8.3.2	Derivare le figure chiave utilizzando il metodo Goal-Question-Metric (Obiettivo-Domanda-Metrica) (L3)	72
8.4	Rischi e problemi nel reporting (L2)	72
9	Gestione dei processi di Requirements Engineering (L3) ..	74
9.1	Requirements Engineering come processo (L2)	74
9.2	Parametri del processo di Requirements Engineering (L3)	76
9.3	Documentare il processo di Requirements Engineering (L3)	78
9.4	Monitoraggio e controllo del processo di Requirements Engineering (L1).....	79
9.5	Process improvement nel processo di Requirements Engineering (L3) .	80
10	Requirements Management nei progetti agile (L2)	83

10.1	Background (L1).....	83
10.2	Requirements Management nei progetti agile (L1)	84
10.3	Mappare le attività di Requirements Management con le attività Scrum (L2).....	85
11	Uso degli strumenti nel Requirements Management (L1)	87
11.1	Ruolo degli strumenti nel Requirements Management (L1)	87
11.2	Procedura di base per la selezione dello strumento (L1)	88
11.3	Scambio dati tra strumenti di Requirements Management (L1)	88
12	Riferimenti Bibliografici	90

1 Che cosa è il Requirements Management? (L2)

Durata: 1 ¼ ora

Termini: Requirements Engineering (Ingegneria dei Requisiti), Requirements Management (Gestione dei Requisiti), Requirements Manager, Requirements Management Plan (Piano di Requirements Management)

Obiettivi Formativi

- EO 1.1 Conoscere la definizione di Requirements Engineering e Requirements Management (L1)
- EO 1.2 Conoscere le attività di Requirements Management (L1)
- EO 1.3 Conoscere gli obiettivi e i benefici del Requirements Management (L2)
- EO 1.4 Conoscere la necessità di un Requirements Management Plan (L2)
- EO 1.5 Conoscere gli standard rilevanti per il Requirements Management (L1)

1.1 Definizione di Requirements Management (L1)

Il Requirements Management (RM) può essere considerato da due prospettive:

1. Gestione dei requisiti e degli artefatti dei requisiti nel processo di sviluppo
2. Gestione delle attività di Requirements Engineering (cioè vedere il Requirements Management come process management o gestione dei processi), [POHL2010], capitolo 30.1]

Il Requirements Management può essere utilizzato nel contesto di un progetto di sviluppo, nell'evoluzione di un sistema esistente, nella gestione di un prodotto software o nella gestione continua e cross-progetto dei requisiti.

Il glossario IREB **Fehler! Verweisquelle konnte nicht gefunden werden.** definisce il Requirements Management come un processo di gestione dei requisiti esistenti e dei relativi artefatti. Questo include la memorizzazione, la modifica e il tracciamento dei requisiti e degli altri artefatti. Questo include, tra l'altro, la strutturazione, la preparazione, la modifica in modo consistente e l'implementazione dei requisiti [RuSo2009].

In letteratura si trovano spesso definizioni diverse di Requirements Engineering (RE) e Requirements Management (RM). A seconda della definizione

- Il Requirements Management è una parte del Requirements Engineering (p.e. in [SCHI2001]), oppure
- Il Requirements Management è una parte del Requirements Engineering (p.e. in [IREB2023]), oppure
- Il Requirements Engineering e il Requirements Management sono definiti come aspetti coesistenti (p.e. CMMI2010).

IREB definisce il Requirements Management **come una parte** del Requirements Engineering.

Nell'ambito di questo Syllabus, i termini sono quindi definiti come segue:

- Il termine **Requirements Engineering** copre le aree di elaborazione, documentazione, validazione/negoziazione e gestione dei requisiti.
- Il termine **Requirements Management** invece, si riferisce al compito di gestire i requisiti all'interno del Requirements Engineering.

1.2 Compiti nel Requirements Management (L1)

Il compito del Requirements Management è di fornire informazioni sui requisiti in modo tale che gli altri ruoli del progetto possano lavorare in modo efficiente. Inoltre, devono essere stabilite le regole e i metodi che consentano di fornire le informazioni [RuSo2009].

Durante l'identificazione dei requisiti, vengono applicate tre assunzioni fondamentali da cui possono essere derivati tutti i compiti del Requirements Management e con i quali possono essere giustificati compiti e metodi (si veda anche [RuSo2009]):

- I requisiti devono essere utilizzati da molte persone,
- I requisiti cambiano,
- I requisiti devono poter essere riutilizzati.

I seguenti compiti e concetti sono parte del Requirements Management [PoRu2011]:

- **Assegnazione di attributi:** gli attributi consentono di descrivere i requisiti e i loro metadati in modo più strutturato, di raggrupparli o di renderli confrontabili con altri requisiti. I requisiti che hanno attributi sono la base per la creazione di viste dei requisiti.
- **Valutazione e prioritizzazione:** le priorità permettono di distinguere tra requisiti importanti e meno importanti, complessi e meno complessi, con l'aiuto di criteri di valutazione e prioritizzazione. Questa distinzione serve a sua volta come base per le decisioni di Project Management e di pianificazione dei rilasci.
- **Tracciabilità:** la tracciabilità permette di tracciare un requisito durante l'intero ciclo di vita del sistema. Sulla base di queste informazioni, è possibile identificare i requisiti dipendenti e gli altri artefatti di sviluppo, ad esempio quando un requisito viene modificato.
- **Versioning:** il versioning consente di tracciare le modifiche ai requisiti durante il loro ciclo di vita.
- **Reporting:** il reporting è la raccolta, la valutazione e la presentazione di informazioni sui requisiti o sul processo di Requirements Engineering (in breve: processo di RE) sottoforma di reportistica. Le informazioni contenute nei report non servono solo come pura informazione, ma anche come base per le decisioni di progetto e per il controllo del processo di RE.
- **Gestione dei processi:** la gestione dei processi del Requirements Engineering consente di progettare in modo efficiente i processi del team.

Le attività di Requirements Management sono pianificate ed eseguite dal Requirements Engineer o dal Requirements Manager.

1.3 Obiettivi e Vantaggi del Requirements Management (L2)

L'obiettivo del Requirements Management è gestire i requisiti e gli altri artefatti relativi ai requisiti in modo tale che i requisiti possano essere sistematicamente analizzati, raggruppati, valutati, modificati e tracciati con un effort ragionevole. Tra le altre cose, fornisce risposte alle seguenti domande:

- Quali requisiti sono parte del sistema? (Assegnazione di attributi)
- Quali requisiti provengono da quale sorgente? (Assegnazione di attributi e tracciabilità)
- Quali requisiti sono urgenti e importanti? (Valutazione e prioritizzazione)
- Quale requisito genera costi troppo alti con pochi benefici? (Valutazione e prioritizzazione)
- Quali (sotto)requisiti di sistema appartengono a quali requisiti di stakeholder? (Tracciabilità)
- Quali requisiti sono parte dei miei sistemi/prodotti? (Tracciabilità)
- Quale versione del requisito è stata implementata nel mio sistema? (Versioning)
- Chi ha modificato l'ultima volta il requisito? (Versioning)
- Quali metriche chiave (key performance indicators) possono essere utilizzate per controllare il mio Requirements Engineering? (Reporting)
- Come lavora in modo efficiente il mio Requirements Engineering? (Gestione del processo di RE [process management]).

L'importanza del Requirements Management all'interno del processo di sviluppo è strettamente correlata alle condizioni al contorno del progetto [RuSo2009]. Il Requirements Management diventa ancora più importante...

- ... più è elevato il numero di requisiti,
- ... più è lunga la durata di vita stimata del prodotto,
- ... più sono previste modifiche,
- ... più è elevato il numero di partecipanti al processo di RE,
- ... più è difficile raggiungere o coinvolgere gli stakeholder,
- ... più sono elevate le richieste di qualità del sistema,
- ... più è richiesto che venga applicato il riutilizzo,
- ... più è complesso il processo di sviluppo,
- ... più sono disomogenee le opinioni degli stakeholder,
- ... più è necessario sviluppare release,
- ... più è importante l'uso degli standard.

Requirements Management ...[RuSo2009]

- ... aumenta la qualità dei requisiti, dei prodotti e dei processi,
- ... riduce i costi di progetto e la durata del progetto,
- ... facilita il monitoraggio di progetti complessi durante tutte le fasi,
- ... migliora la comunicazione all'interno del team e tra i team,
- ... aumenta la soddisfazione dei clienti,
- ... riduce il rischio di progetto.

1.4 Requirements Management Plan (L2)

Simile a una project management guide (guida del project management), il Requirements Management Plan (RMP, Piano di Requirements Management) descrive le specifiche per l'implementazione del Requirements Engineering. Include la pianificazione dei tipi di requisiti che devono essere documentati, come i requisiti sono gestiti e chi ha le responsabilità nel processo di RE.

Il Requirements Management Plan copre:

- Il Requirements Information Model (RIM), che fornisce un landscape (panoramica) dei requisiti, cioè i tipi di requisiti da gestire e il loro livello di dettaglio (si veda 2),
- Attributi e viste dei requisiti (si veda 3),
- Criteri e metodi di prioritizzazione (si veda 4),
- Linee guida per il processo delle modifiche e per la gestione delle versioni dei requisiti e degli artefatti dei requisiti (si veda 5),
- Linee guida per la gestione della tracciabilità dei requisiti (si veda 6),
- Linee guida per la descrizione delle varianti dei requisiti (si veda 7),
- Linee guida per il reporting dei requisiti (si veda 8),
- Processo di RE con attività, ruoli e responsabilità (si veda 9),
- Linee guida per gli strumenti da utilizzare (si veda 11).

In genere il Requirements Management Plan non è un documento indipendente, ma è parte del Project Management Guide, del Configuration Management Plan o di altri documenti di specifica per il processo di sviluppo.

1.5 Standard Rilevanti (L1)

Il Requirements Management viene indirizzato da diversi standard e linee guida in modi differenti. Di seguito sono elencati alcuni rappresentanti importanti.

- Il CMMI (versione 1.3 CMMI2010) considera, tra gli altri processi, i processi "Requirements Development" (Sviluppo dei Requisiti) e "Requirements Management", dove alcuni degli obiettivi assegnati differiscono in modo significativo dalle definizioni specificate da IREB.
- ISO 9000 / ISO 9001 [ISO 9000] è uno standard per il quality management nelle organizzazioni. ISO 9001 definisce i requisiti minimi per un quality management system (sistema di quality management) e descrive, ad esempio, i requisiti per la realizzazione del prodotto, per la misurazione e il miglioramento, indirizzando temi come l'identificabilità o la tracciabilità.
- ISO/IEC 12207:2008 e 15288 ("Systems and software engineering – Software life cycle processes" e "Systems engineering – Systems life cycle processes") sono standard che definiscono i processi più importanti per lo sviluppo di sistemi e per lo sviluppo software, e descrivono anche i compiti nell'area del Requirements Engineering e del Requirements Management.

- IEC 61508 ("Functional safety of safety-related electrical/electronic/programmable electronic systems") è uno standard che si occupa della definizione dei requisiti per la safety funzionale dei sistemi. Particolare attenzione viene data al tema della tracciabilità.
- ISO/IEC/IEEE 29148:2011 ("Systems and software engineering – Life cycle processes – Requirements engineering") è uno standard che descrive i processi specifici per il Requirements Management.
- SOX (Sarbanes–Oxley Act) [USCO2002], [NEMN2006] è una legge federale statunitense in risposta agli scandali contabili, intesa a migliorare l'affidabilità del reporting delle società quotate sul mercato pubblico dei capitali negli USA. In sostanza, la legge Sarbanes–Oxley Act ci impone di sapere chi ha fatto quali modifiche e quando, e quindi si riferisce anche ai compiti fondamentali del Requirements Management.

A causa di vincoli esterni (p.e. requisiti espliciti del cliente o normative), possono essere applicati uno o più di questi standard o linee guida, e l'implementazione del Requirements Management risulta essere obbligatoria.

Alcuni standard utilizzano definizioni differenti dei termini e sono quindi solo parzialmente compatibili tra loro.

2 Requirements Information Model (L3)

Durata: 2 ore

Termini: Landscape dei Requisiti (Panoramica dei Requisiti), Tipo di Requisito, Forme di Presentazione, Requirements Information Model (RIM, Modello Informativo dei Requisiti)

Obiettivi Formativi

- EO 2.1.1 Conoscere le classificazioni di base dei requisiti (L1)
- EO 2.1.2 Conoscere il modello Twin Peaks (L2)
- EO 2.2 Conoscere le forme tipiche di presentazione dei requisiti (L2)
- EO 2.3 Padroneggiare e utilizzare il Requirements Information Model (RIM)(L3)

2.1 Fondamenti (L2)

I requisiti possono essere classificati in base a diversi criteri:

- Tipo di requisito
- Indipendenza dalla soluzione (solution independence)
- Livello di dettaglio o livello di astrazione del requisito

La classificazione per tipo di requisito è conosciuta da IREB Foundation Level [IREB2023], capitolo 1, [POHL2010]:

- Requisiti funzionali
- Requisiti di qualità
- Vincoli

Le caratteristiche dei diversi tipi di requisiti e la loro ulteriore categorizzazione sono trattate in dettaglio in [POHL2010], [PoRu2011].

In specifici progetti, questi tipi di requisiti vengono spesso raffinati o vengono definiti con nomi differenti. Schemi dettagliati di classificazione dei requisiti possono essere trovati, ad esempio, in [YOUN2004], [RuSo2009], [WiBe2013], [RoRo2014].

Inoltre, i requisiti possono essere classificati ortogonalmente in base al loro grado di indipendenza dalla soluzione [POHL2010]:

Indipendenza alta dalla soluzione (p.e. gli obiettivi): Gli obiettivi descrivono l'intenzione del sistema senza descrivere l'implementazione, e quindi descrivono la forma più indipendente dalla soluzione di un requisito.

Indipendenza media dalla soluzione (p.e. gli scenari): Gli scenari descrivono in modo esemplare le possibili sequenze di interazione per raggiungere uno o più obiettivi. Sono descritti dal punto di vista dell'utente.

Indipendenza bassa dalla soluzione (p.e. i requisiti orientati alla soluzione o solution-oriented): I requisiti orientati alla soluzione descrivono i dati, le funzioni, il comportamento, gli stati e la qualità del, che sono richiesti per raggiungere gli obiettivi e per implementare gli scenari.

Nella forma documentata [POHL2010] queste tre classificazioni sono anche chiamate tipi di artefatti dei requisiti.

Un terzo tipo di classificazione utilizzata per differenziare i requisiti è il livello di dettaglio, chiamato anche livello di astrazione. I requisiti sono spesso descritti a diversi livelli di dettaglio: a un livello di dettaglio alto, con pochi dettagli, per fornire un overview, e poi a un livello di dettaglio basso, con maggiore dettagli, per specificare i requisiti in modo completo e preciso.

Modello Twin-Peaks

I requisiti costituiscono la base per la progettazione del sistema. La pratica ha dimostrato che un approccio waterfall (a cascata), in cui il sistema viene progettato in modo sequenziale dopo aver specificato i requisiti, spesso non corrisponde alla realtà [POHL2010], [CHSM2013]. Si tratta piuttosto di un processo iterativo in cui i requisiti modellano l'architettura di un sistema e viceversa. Figura 1 illustra questa relazione nel cosiddetto modello Twin-Peaks [NUSE2001]. L'asse verticale rappresenta il livello di dettaglio della descrizione del sistema, mentre l'asse orizzontale rappresenta l'orientamento crescente dalla descrizione del problema all'implementazione. La figura mostra che le descrizioni più dettagliate dei requisiti vengono sviluppate iterativamente, in parallelo con le architetture di sistema più dettagliate.

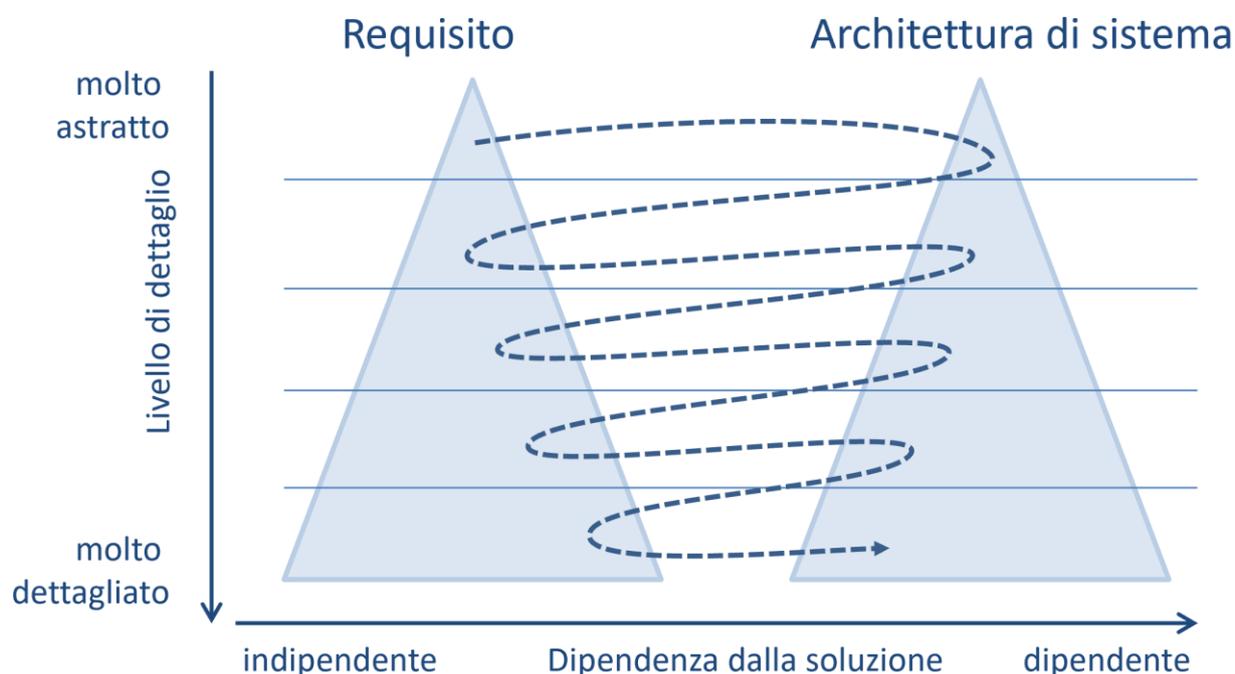


Figura 1: Modello Twin-Peaks

La definizione dei livelli di astrazione richiesti, cioè dei livelli in cui vengono descritti i diversi livelli di dettaglio dei requisiti, è specifica del progetto e non può essere descritta universalmente. Si raccomanda generalmente di dettagliare i requisiti fino a quando:

- Viene raggiunta una comprensione comune del requisito da tutti gli stakeholder, ed è chiaro a tutti che cosa è esattamente richiesto.
- Il grado di dettaglio rimanente per la soluzione è così piccolo che ulteriori chiarimenti genererebbero più costi che benefici (cioè il rischio residuo risultante dal grado di dettaglio rimanente è accettabile).
- I requisiti sono specificati nella misura in cui sono chiaramente verificabili (testabili) rispetto alla soluzione.

I livelli di astrazione più utilizzati sono, ad esempio: "sistema complessivo", "sottosistema", "componente tecnico", "requisiti del cliente e degli stakeholder", "requisiti di progettazione del sistema, dell'architettura e dei componenti", "requisiti di implementazione", oppure la suddivisione in "specifiche di business" e "specifiche tecniche".

I seguenti due aspetti della procedura sono cruciali per il Requirements Management:

- Per poter supportare prospettive differenti sui requisiti, i requisiti vengono specificati e memorizzati a diversi livelli di astrazione e a livelli di dettaglio espliciti.
- Per dare valore a questo processo incrementale e iterativo, e per garantire che la completezza e la consistenza siano mantenute tra i diversi livelli di dettaglio, è fondamentale collegare i requisiti ai diversi livelli di dettaglio, ma anche i requisiti ai loro elementi di implementazione nell'architettura. Questo viene ottenuto con le relazioni di tracciabilità (si veda 6).

2.2 Forme di presentazione (L2)

I tre tipi di requisiti, i tre gradi di indipendenza dalla soluzione e i livelli di dettaglio selezionati hanno come risultato una serie di potenziali "punti di intersezione" ("intersezioni" o "crossing point") (si veda Figura 2), di seguito indicati come classi di requisiti.

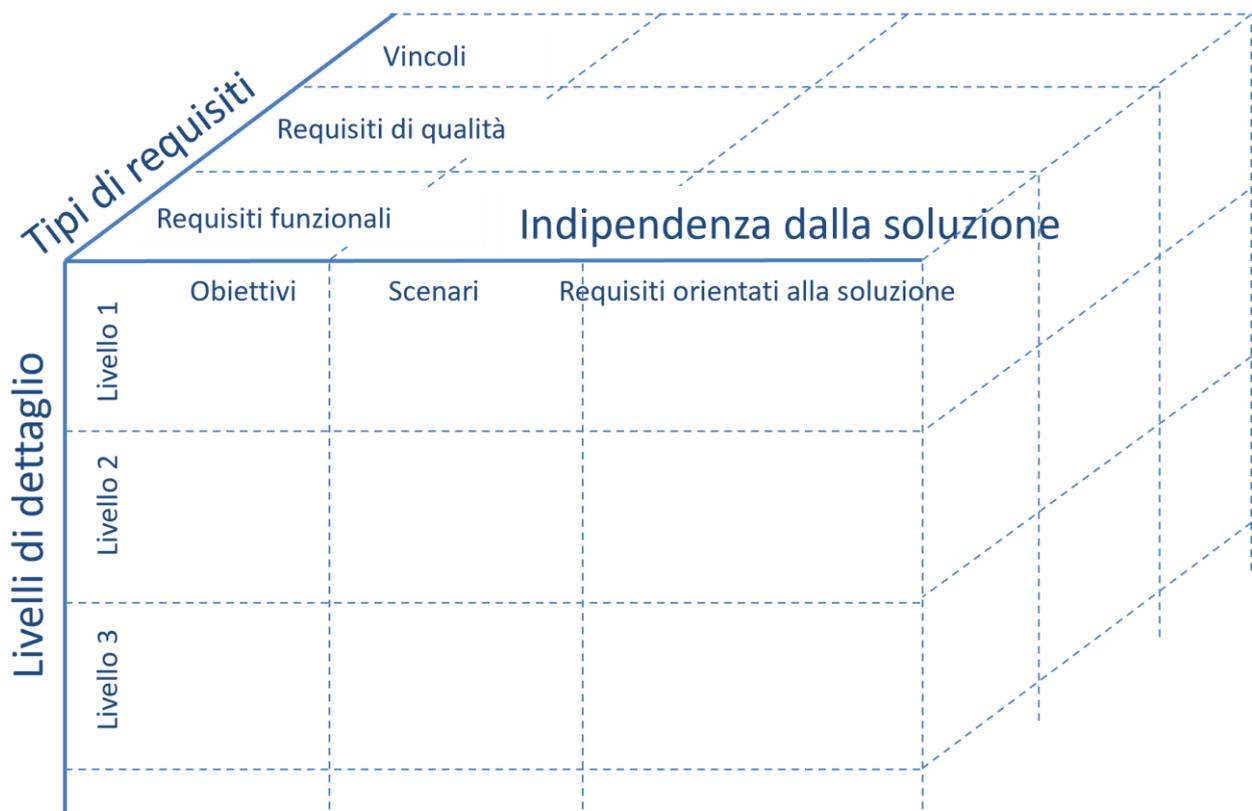


Figura 2: Numerosi "punti di intersezione" potenziali (classi di requisiti)

Deve essere determinato (1) quali "punti di intersezione" devono essere considerati e (2) come devono essere documentati. Un esame completo di tutti i "punti di intersezione" non è normalmente possibile dal punto di vista economico, e spesso non è nemmeno significativo in termini di contenuto.

Per la documentazione dei requisiti esistono forme differenti di presentazione, gradi di formalità e notazioni/linguaggi, ad esempio:

- Linguaggio naturale (p.e. descrizione in pura prosa)
- Testo strutturato (p.e. template, schemi)
- Forme di notazione grafica o model-based (p.e. UML, albero delle funzionalità)
- Descrizioni formali (p.e. funzioni matematiche)

Per ogni "punto di intersezione" richiesto, deve essere specificata quale forma di rappresentazione è richiesta. È possibile che siano richieste più rappresentazioni differenti per un "punto di intersezione", in base p.e. ai gruppi di stakeholder o ai componenti del sistema. Ad esempio, a livello di dettaglio "sottosistema", per il tipo "requisito funzionale" e con un'indipendenza dalla soluzione di "requisito orientato alla soluzione", può avere senso utilizzare requisiti in linguaggio naturale (per documentare i requisiti generici del cliente) oppure sequence diagram dei messaggi (per documentare i requisiti di interfaccia).

Il linguaggio in cui devono essere documentati i requisiti testuali e i requisiti model-based deve essere definito fin dall'inizio, per evitare inutile effort aggiuntivo di traduzioni successive.

2.3 Creazione di un Requirements Information Model (L3)

Nei progetti reali, verranno considerati solo alcuni dei potenziali "punti di intersezione" risultanti da tutte le combinazioni matematiche dei tipi di requisito, indipendenza dalla soluzione e livelli di dettaglio.

L'insieme dei "punti di intersezione" utilizzati in un progetto è chiamato landscape dei requisiti. (panoramica dei requisiti). Un landscape dei requisiti è una definizione (a) della classificazione da utilizzare in relazione ai tipi di requisiti, (b) della classificazione da utilizzare in relazione all'indipendenza dei requisiti dalle soluzioni, (c) dei livelli di dettaglio richiesti per ciascun tipo di artefatto dei requisiti e (d) delle forme di presentazione da utilizzare per ogni tipo di artefatto dei requisiti.

In alcuni casi, gli standard e le linee guida rilevanti forniscono già una prima indicazione dei "punti di intersezione" che devono essere sempre considerati. Ad esempio, [ISO 26262] richiede che i requisiti di safety funzionale e i conseguenti requisiti di safety tecnica siano derivati dagli obiettivi di safety (cioè una forma speciale di requisiti espressi come obiettivi).

[RuSo2009] fornisce un esempio di cinque livelli di dettaglio, chiamati livelli di specifica:

- **Livello di specifica 0:** descrive in modo approssimativo il progetto complessivo e i suoi obiettivi
- **Livello di specifica 1:** descrive gli use case e i processi di business delle aree di business che soddisfano gli obiettivi (specificazione funzionale)
- **Livello di specifica 2:** dettaglia i processi di business e i requisiti di business delle aree di business del livello di specifica 1 (specificazione funzionale)
- **Livello di specifica 3:** descrive i requisiti dell'utente dettagliati, suddivisi in sottosistemi, e una descrizione delle interfacce (specificazione funzionale)
- **Livello di specifica 4:** descrive la specifica tecnica con una separazione in hardware, software e altri componenti (specificazione tecnica)

Il livello di specifica 0 comprende quindi tutti i tipi di requisiti con "obiettivi" di indipendenza dalla soluzione e il livello di dettaglio più basso (più grossolano).

Questo esempio specifica anche che non tutti i livelli di dettaglio vengono utilizzati per documentare ogni requisito, ma dipende dal tipo di requisito e dall'indipendenza dalla soluzione.

Le definizioni di landscape dei requisiti (cioè i "punti di intersezione" richiesti e la forma/forme di rappresentazione scelte) devono essere documentate come Requirements Information Model (RIM, Modello Informativo dei Requisiti). Ad esempio, potrebbe essere creato un entity-relationship diagram (diagramma entità-relazione) o un class diagram. Un esempio può essere trovato in Figura 3. L'esempio si basa sui livelli di specifica di [RuSo2009], ma non li implementa completamente.

Oltre alle specifiche del tipo di requisiti, all'indipendenza dalle soluzioni, al livello di astrazione e alla forma di rappresentazione, il Requirements Information Model dovrebbe essere integrato da ulteriori aspetti:

- Quali sono gli attributi da utilizzare? (si veda 3)
- Quali viste saranno supportati? (si veda 3)
- Quali criteri di valutazione sono previsti per i requisiti? (si veda 4)
- Quali ruoli sono responsabili della manutenzione e della modifica di quali informazioni? (si veda 5)
- Quali relazioni di tracciabilità tra i requisiti e gli artefatti verso l'alto o verso il basso saranno documentate? (si veda 6)
- Come verranno documentate le varianti dei requisiti? (si veda 7)

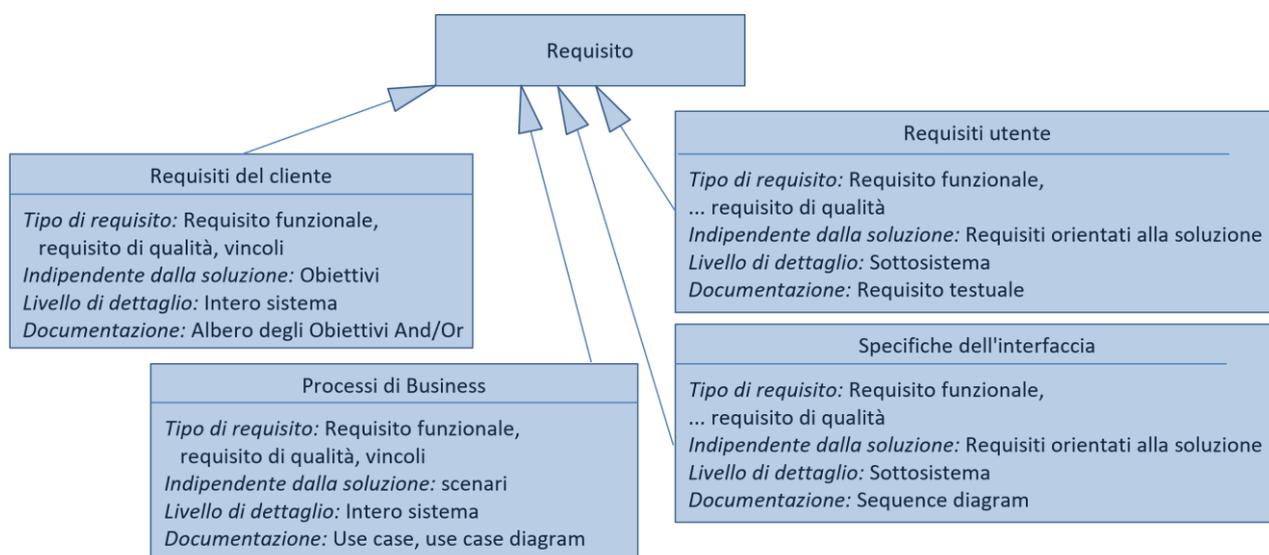


Figura 3: Esempio di uno specifico Requirements Information Model (RIM)

Il Requirements Information Model fornisce a tutti gli stakeholder di un progetto una vista centralizzata degli artefatti dei requisiti che vengono utilizzati, e del loro utilizzo nel Requirements Management, ed è quindi una parte essenziale del Requirements Management Plan (RMP).

Quando si definisce il Requirements Information Model, è sempre importante bilanciare i benefici ottenuti da una documentazione dei requisiti più completa rispetto ai costi che ne derivano [GLIN2008], [DAVI2005].

Quando si analizza un Requirements Information Model, si possono utilizzare le seguenti domande di verifica:

- **Completezza formale:** Per ogni classe di requisiti è chiaro quale è il tipo di requisito, quale è il suo grado di indipendenza dalla soluzione, a quale livello di dettaglio è assegnato e con quale forma di rappresentazione è documentato?
- **Relazioni del contenuto:** È chiaro quali sono livelli di dettaglio e come sono correlati? È chiaro come i requisiti nelle diverse classi di requisiti siano interconnessi?

- **Adeguatezza:** Tutte le classi di requisiti sono appropriate per fornire requisiti sufficientemente dettagliati e completi in modo da poter svolgere le attività di sviluppo relative?

In particolare, l'appropriatezza dipende fortemente anche dal modello di processo scelto. Gli approcci Agile (si veda 10), ad esempio, si focalizzano su "obiettivi" e "scenari" indipendenti dalla soluzione e generalmente omettono i requisiti (più complessi) orientati alla soluzione. Grazie allo stretto contatto con il cliente, alla prototipazione e alle brevi iterazioni, è possibile ottenere le informazioni necessarie in tempo utile.

3 Assegnazione di attributi e viste ai requisiti (L5)

Durata: 2 ore

Termini: Attributo, Schema di Attributo, Vista

Obiettivi Formativi

- EO 3.1 Conoscere l'obiettivo dell'assegnazione di attributi e i benefici di assegnare attributi nelle attività di Requirements Management (L1)
- EO 3.2 Conoscere i vantaggi degli schemi di attributi (L2)
- EO 3.3.1 Conoscere i passi per progettare uno schema di attributi (L1)
- EO 3.3.2 Padroneggiare e utilizzare i passi predefiniti per progettare uno schema di attributi (L3)
- EO 3.4.1 Conoscere i tipi di modifiche negli schemi di attributi (L2)
- EO 3.4.2 Valutare le modifiche agli schemi di attributi in base ai diversi tipi di modifiche (L5)
- EO 3.5 Conoscere gli obiettivi di creare viste dei requisiti e i tipi di viste (L2)
- EO 3.6 Conoscere i passi del processo per definire le viste (L1)
- EO 3.7 Analizzare il grado di popolamento degli attributi (L4)

3.1 Obiettivi dell'Assegnazione di Attributi (L1)

Gestire i requisiti di prodotti complessi durante il loro intero ciclo di vita richiede la gestione di una grande quantità di informazioni su questi requisiti. Ad esempio, per determinare chi ha specificato un requisito in conflitto, in caso di conflitto, è necessario documentare la sorgente di questo requisito.

L'obiettivo di assegnare attributi ai requisiti è quello di consentire agli stakeholder di documentare le informazioni sui requisiti in modo strutturato come parte del processo di RE [POHL2010], capitolo 18.2, [IREB2023], capitolo 8.1. Per questo motivo sono stati definiti diversi attributi in cui vengono registrate queste informazioni (p.e. sorgente, data di creazione, autore, ecc.).

L'assegnazione di attributi ai requisiti fornisce una base per una serie di compiti nel Requirements Management e per le altre attività di gestione, ad esempio:

- **Tracciabilità:** Gli attributi sono utilizzati nel Requirements Management per realizzare la tracciabilità. Un prerequisito, ad esempio, per la tracciabilità della sorgente di un requisito – per poter richiedere ulteriori dettagli – è che la sorgente sia stata documentata in un attributo appropriato. Inoltre, gli attributi facilitano anche i collegamenti con altri requisiti, in modo da poter stabilire, ad esempio, la tracciabilità tra obiettivi e requisiti orientati alla soluzione.
- **Viste:** L'implementazione delle viste dei requisiti si basa solitamente sugli attributi (cfr. EU 3.6).
- **Priorità:** La priorità dei requisiti viene documentata in uno o più attributi corrispondenti. È possibile definire diversi attributi per i differenti contenuti di prioritizzazione.

- **Gestione delle varianti:** Possono essere utilizzati attributi per gestire le varianti (si veda 6), per assegnare i requisiti a specifiche varianti.
- **Reporting:** Gli attributi forniscono una base per la generazione di report, come la valutazione del numero di requisiti in un determinato stato (p.e. "in progress" o "verificato").
- **Project Management:** Se l'effort dell'implementazione di ogni requisito è parte del Project Management, questo effort dell'implementazione deve essere memorizzato in un attributo appropriato. Inoltre, è necessario documentare lo stato di un requisito per supportare il project management.
- **Release Management:** L'interfaccia con il Release Management viene supportata da un attributo "Release" corrispondente. Questo permette di documentare quali requisiti sono stati implementati in quale release. In molti casi, viene fatta una distinzione tra la release desiderata e la release pianificata, per poter chiarire le differenze che spesso si verificano tra queste due release.

3.2 Utilizzo di uno schema di attributi (L2)

L'insieme di tutti gli attributi definiti per un tipo di requisito (p.e. requisiti funzionali, requisiti di qualità) è chiamato schema di attributi ([PoRu2011], paragrafo 8.1.2). Fornire uno schema di attributi per i requisiti porta ai seguenti vantaggi nel Requirements Management [POHL2010], capitolo 18.2:

- **Definizione accurata e consistente delle informazioni richieste:** Uno schema predefinito definisce quali informazioni o quali attributi per i requisiti devono essere inseriti, e quali valori sono ammessi per queste informazioni.
- **Identificazione delle lacune:** Se alcuni attributi non sono stati compilati, è possibile rilevare lacune nell'elaborazione dei requisiti.
- **Supporto alla formazione dei dipendenti:** I dipendenti che hanno già lavorato in un progetto precedente con gli stessi schemi di attributi o con schemi simili, possono trovare rapidamente le informazioni necessarie e valutare dove dovrebbero essere documentate particolari informazioni sul requisito.
- **Trovare le stesse informazioni nello stesso posto:** Poiché tutti i requisiti all'interno di un progetto sono documentati sulla base dello stesso schema di attributi, è specificato in modo chiaro dove si possono trovare le informazioni (ad esempio l'autore) per un requisito.

I tipici fattori di influenza per la definizione di uno schema di attributi sono [PoRu2011], paragrafo 8.1, [IREB2023], capitolo 8.1:

- Caratteristiche specifiche di un progetto, p.e. la dimensione del progetto
- Linee guida fornite dall'azienda, p.e. gli standard aziendali
- Proprietà e normative relative all'area applicativa, p.e. modelli di riferimento e standard
- Condizioni limite e restrizioni sul processo di sviluppo, p.e. standard di processo e convenzioni.

Quando si definisce uno schema di attributi, ad esempio per il Project Management, oltre ai vantaggi legati al processo di RE si deve tenere conto anche degli obiettivi e delle necessità dei processi correlati (si veda anche 3.1). che devono essere mappate in un tipo di attributo rilevante per il requisito.

3.3 Progettazione di uno schema di attributi (L3)

Uno schema di attributi dovrebbe essere definito prima di iniziare la documentazione dei requisiti, concordandolo con tutti gli stakeholder del processo di RE. In genere i miglioramenti e le modifiche successive sono fattibili solo con un grande effort [RuSo2009], capitolo 14.1.

I passi seguenti sono necessari per progettare uno schema di assegnazione degli attributi da utilizzare in uno specifico progetto:

1. Identificazione delle Sorgenti degli Attributi

Per selezionare gli attributi, è necessario identificare prima le sorgenti rilevanti per gli attributi. Le sorgenti che possono essere utilizzate per selezionare gli attributi includono:

- Uno schema di attributi da un progetto simile (ad esempio, simile nell'ambito, nel numero di dipendenti coinvolti, ecc.)
- Uno schema di riferimento dell'organizzazione o di un altro standard
- Regole dell'organizzazione che determinano, ad esempio, quali attributi devono essere utilizzati in tutti gli schemi di attributi di tutti i progetti
- Gli stakeholder del processo di RE

2. Selezione degli attributi

Per utilizzare con successo lo schema di attributi nel Requirements Management, è essenziale che gli attributi siano selezionati in modo mirato e appropriato per il progetto. Questo si applica sia per l'utilizzo di schemi di riferimento sia per la definizione di uno schema di attributi senza utilizzare uno schema di riferimento.

Di seguito sono elencati alcuni passi per selezionare gli attributi:

- Verificare gli attributi nello schema di riferimento; se necessario, modificare o integrare gli attributi.
- Per la selezione degli attributi per uno schema di attributi e per la valutazione della completezza di uno schema di riferimento nell'ambito di un progetto, possono essere di aiuto le sette categorie presentate in [POHL2010], paragrafo 18.3: "Identificabilità", "Relazioni con il contesto", "Aspetti di documentazione", "Aspetti di contenuto", "Aspetti di conformità", "Aspetti di validazione" e "Aspetti di gestione".
- Limitare gli attributi a un insieme pragmaticamente utilizzabile, focalizzandosi sugli obiettivi del Requirements Management nel progetto [RuSo2009].
- Definire il contesto in modo da soddisfare l'unicità del numero di identificazione (unico nel progetto, unico nell'azienda, ecc.).

3. Definizione dei valori di attributo consentiti e delle proprietà degli attributi

Dopo aver definito quali attributi vengono utilizzati in uno schema di attributi, devono essere definite le proprietà degli attributi, come:

- Qual è il tipo di attributo in esame (testo, numero, valore logico, enumerazione di date, ecc.)?
- Quali valori per i tipi enumerativi dovrebbero essere disponibili per la selezione in un attributo corrispondente?
- L'attributo è un campo obbligatorio?
- L'attributo può contenere più valori o un solo valore?
- Sono disponibili aiuti all'utente per la selezione, come "tutti i valori" o "nessun valore", e quali conseguenze ne derivano da questo?

4. Definizione delle dipendenze tra gli attributi e i loro valori

Gli attributi possono avere delle interdipendenze relative ai valori che possono assumere. Ad esempio, è possibile evitare che un requisito con il valore "volatile" nell'attributo "Stabilità" assuma contemporaneamente il valore "rilasciato" nell'attributo "Stato". Questo assicura che, ad esempio, solo i requisiti considerati stabili vengano approvati per lo sviluppo.

Quando si definisce uno schema di attributi, dovrebbe essere anche determinato se certe combinazioni non sono consentite per due attributi con, ad esempio, valori di attributi predefiniti. In questo caso, può avere senso combinare questi due attributi in un unico attributo e offrire solo le combinazioni consentite. Questo è particolarmente utile se lo strumento utilizzato non supporta l'analisi delle dipendenze tra i valori degli attributi.

Nella Gestione delle Varianti è possibile vietare l'assegnazione di requisiti a specifiche varianti. È anche possibile che certe combinazioni di varianti non siano consentite.

Le dipendenze tra gli attributi e i loro valori possono nascere anche dalla gerarchia dei requisiti. Ad esempio, se il requisito A è dettagliato dai requisiti A.1 e A.2, per gli attributi dello schema di attribuzione deve essere determinato se il valore di A dipende o meno dai valori di A.1 e A.2.

5. Supporto per la registrazione dei dati

La raccolta di informazioni aggiuntive sul requisito attuale rappresenta un effort aggiuntivo per il Requirements Engineer, che in alcuni casi può essere un beneficio per altri stakeholder (p.e. i project manager) piuttosto che per il Requirements Engineer stesso. Questa è una delle ragioni del perché è molto utile che il Requirements Engineer sia supportato il più possibile nella raccolta di tali informazioni.

Questo tipo di supporto è fornito principalmente (o addirittura esclusivamente) dall'utilizzo di uno strumento appropriato per la registrazione e la gestione dei requisiti. Un esempio semplice è la definizione di valori di default per gli attributi.

6. Documentazione dello schema di attributi

Gli schemi di attributi sono definiti in forma tabellare o in un modello informativo, a seconda del grado di complessità (ad esempio, per quanto riguarda il numero di attributi, le dipendenze tra gli attributi o i loro valori) (si veda 2).

La definizione in un modello informativo viene utilizzata da uno strumento che mappa il corrispondente schema di attributi e supporta il Requirements Engineer nella documentazione delle informazioni associate.

3.4 Change Management degli Schemi di attributi (L5)

Durante il progetto dovrebbero essere evitate modifiche a uno schema di attributi dopo la sua definizione [RuSo2009], capitolo 14.1.

Le conseguenze di una successiva modifica a uno schema di attributi dipendono dal tipo di modifica:

- **Aggiunta, modifica o cancellazione di un attributo:**

Quando si aggiunge un nuovo attributo, è necessario valutare quanto è time-consuming (costoso in termini di tempo) mantenere aggiornati i requisiti precedentemente documentati rispetto al nuovo attributo. Se un attributo deve essere cancellato, questo può comportare conseguenze negative se, ad esempio, in uno strumento alcuni moduli interrogano quell'attributo. Invece di cancellarlo, è possibile aggiungere "(non più utilizzato)" al suo nome.

- **Aggiunta, modifica o cancellazione di possibili valori di un attributo (intervallo di valori):**

Aggiungere valori di un attributo di solito non è un problema per lo strumento sottostante. Da un punto di vista tecnico, deve essere verificato se l'aggiunta comporta un nuovo significato per i valori esistenti dell'attributo, così che i requisiti precedentemente valutati rispetto a questo attributo debbano essere di nuovo analizzati definendo nuovi valori, se necessario. Quando si cancellano valori dell'attributo da un intervallo di valori, è importante assicurare che i requisiti non diventino inconsistenti a causa di entry vuote. I problemi sono causati principalmente dai campi obbligatori, poiché il requisito deve avere un valore nell'attributo considerato. In questo caso, una possibile soluzione è inserire nel campo un valore di default.

Quando vengono modificati i valori dell'attributo, è importante assicurare che le modifiche vengano apportate in tutti i requisiti che contengono il valore originale. In generale, quando si modificano o si cancellano valori di requisiti, è importante decidere se questa modifica impatterà i requisiti che sono già stati definiti o se si applicherà solo ai requisiti futuri.

Per gli attributi che dipendono dai valori dell'attributo modificato, deve essere assicurato che la cancellazione di un valore di attributo non generi dati inconsistenti.

- **Aggiunta o cancellazione di relazioni tra attributi:**

Aggiungere una relazione può portare a inconsistenze all'interno dell'insieme di requisiti già definiti. Ad esempio, se viene aggiunta una definizione per cui la selezione di un valore per l'attributo "Stabilità" dovrebbe sempre portare a popolare l'attributo "Rischio" (che quindi deve essere sempre popolato quando viene popolato "Stabilità"), allora i requisiti con un valore per "Stabilità" ma nessuno per "Rischio" devono essere di nuovo verificati.

Se viene cancellata la dipendenza tra gli attributi, deve essere assicurato che verranno cancellati i valori di default (cioè: se viene selezionato il valore A dell'attributo allora il valore B dell'attributo viene automaticamente predefinito da uno strumento).

- **Modifica dei valori di default per il tipo di attributo:**

Modificare i valori di default dovrebbe inizialmente influenzare solo l'inserimento di nuovi requisiti. Tuttavia, in questo contesto, dovrebbero essere analizzati i requisiti a cui è stato assegnato il precedente valore di default, per verificare se hanno ancora il valore corretto o se devono essere adeguati.

- **Modifica delle proprietà "Campi obbligatori" e "Campi opzionali":**

Una modifica da campo obbligatorio a campo opzionale di solito non comporta alcun effort aggiuntivo. Se viene pianificata una modifica da attributo opzionale a attributo obbligatorio, deve essere assicurato che l'attributo sia popolato con un valore appropriato per tutti i requisiti già documentati. Può essere necessario assegnare valori custom (personalizzati) piuttosto che utilizzare un valore di default.

In generale, quando si apportano modifiche agli schemi di attributi, deve essere analizzato l'impatto sugli strumenti sottostanti. Ad esempio, se nello strumento DOORS sono stati creati script che verificano o elaborano un particolare attributo, una modifica corrispondente all'attributo può rendere gli script non più eseguibili.

3.5 Obiettivi e tipi di viste (L2)

Un vantaggio nella definizione di schemi di attributi è la possibilità di definire viste per il modello informativo e di implementarle come parte del supporto degli strumenti (si veda anche [IREB2023], capitolo 8.2).

In un progetto, la complessità delle informazioni sui requisiti può essere molto elevata, soprattutto se deve essere documentato un ampio numero di requisiti. Molti stakeholder sono coinvolti nel progetto relativamente ai requisiti. Oltre al Requirements Engineer possono essere, ad esempio, sviluppatori, tester o project manager.

L'obiettivo di fornire specifiche viste è fornire ai rispettivi stakeholder informazioni sui requisiti che siano specifiche di un obiettivo o di un ruolo, riducendo così la complessità dal punto di vista dello specifico stakeholder. L'accesso ai requisiti può essere controllato in base a un ruolo specifico, creando delle viste (si veda anche [POHL2010], cap. 18.5).

Le viste possono essere classificate come segue:

- **Viste selettive:** Le viste selettive limitano il numero di requisiti da considerare utilizzando un filtro predefinito.
- **Viste proiettive:** Le viste proiettive nascondono gli attributi che non sono rilevanti per la vista considerata.
- **Viste condensate:** Le viste condensate consentono di valutare (ad esempio, creare dei totali) un insieme di requisiti rispetto a determinati attributi.

- **Combinazione di viste selettive, proiettive e aggregate:** Nelle viste combinate, alcuni attributi sono nascosti, l'insieme dei requisiti viene limitato rispetto a uno o più criteri e viene valutato separatamente utilizzando, ad esempio, un attributo.

Gli strumenti che forniscono viste dei requisiti di solito permettono anche di ordinare rispetto a diversi attributi [RuSo2009], capitolo 14.1. Questo consente agli stakeholder, nei rispettivi ruoli, di modificare il focus sui requisiti (ad esempio, visualizzare all'inizio di un elenco tutti i requisiti con un elevato effort di implementazione) senza nascondere i requisiti. Inoltre, gli strumenti spesso consentono di strutturare ulteriormente le viste raggruppando i requisiti in base a un attributo.

3.6 Definizione di vista e rischi delle viste (L1)

Il processo per definire le viste include i seguenti passi:

- **Identificazione degli stakeholder che necessitano di una o più viste:** Le viste possono fornire prospettive differenti sui requisiti. Per definire le viste necessarie, devono essere prima identificati tutti gli stakeholder che utilizzeranno le viste sui requisiti. Gli stakeholder sono una sorgente per definire una vista.

Riutilizzo: Le viste provenienti da altri progetti o da un progetto di riferimento possono essere utilizzate come sorgente di viste da definire.

Identificazione degli obiettivi di una vista in relazione ai differenti stakeholder: A seconda dei rispettivi stakeholder, deve essere determinato quale obiettivo sarà perseguito con una specifica vista. Questo può essere utilizzato, ad esempio, per determinare se è necessario un consolidamento o per definire quale ordinamento dovrebbe essere inizialmente impostato. In questo contesto, devono essere determinati anche i diritti e le viste degli stakeholder, cioè quale stakeholder dovrebbe essere in grado di attivare una specifica vista. In questo modo ha senso che una vista possa essere utilizzata da molti stakeholder.

Identificazione degli attributi necessari e confronto con lo schema di attributi: Per poter realizzare gli obiettivi di una vista, deve essere assicurato che sia possibile raccogliere le informazioni necessarie e che siano disponibili anche gli attributi corrispondenti. Tuttavia, il confronto con lo schema di attributi porta spesso a nuove viste da generare, perché solo quando lo schema di attributi viene esaminato in dettaglio, gli stakeholder si rendono conto, ad esempio, di quali valutazioni sarebbero ancora possibili. La definizione delle viste e degli schemi di attributi è quindi un processo altamente integrativo; entrambi i processi si influenzano a vicenda.

Implementazione della vista: Le viste predefinite devono essere implementate e testate nello strumento sottostante.

La creazione delle viste comporta alcuni rischi. In molti casi, gli utenti di un sistema di Requirements Management non sono consapevoli del fatto che la grande quantità di informazioni su un requisito può essere ristretta dalle viste richieste.

Spesso gli utenti lavorano con una vista globale e omnicomprensiva, e percepiscono lo strumento come troppo esteso e forse ostacolante nel loro lavoro.

Esiste anche il pericolo di definire viste in cui si perde il contesto. Ad esempio, se si crea una vista in cui i requisiti atomici sono riportati in una lista senza alcun contesto (p.e., gli use case), questa panoramica sarà significativa solo in misura limitata. Per evitare il più possibile queste viste inefficaci, è molto importante considerare l'obiettivo di fondo di una vista.

3.7 Ottimizzare il popolamento degli attributi e la creazione di viste (L4)

La pratica dimostra che in alcuni progetti gli attributi non vengono popolati nella misura prevista. Pertanto, si raccomanda di verificare regolarmente e in momenti specifici del progetto se e come un attributo deve essere utilizzato [RuSo2009], capitolo 14.1.

Un modo per garantire il grado di popolamento dell'attributo nel contesto di utilizzo dello strumento è quello di definire l'attributo come un "campo obbligatorio". Questo forza l'inserimento di input quando si crea un requisito. In questo contesto, è importante osservare che la definizione di troppi campi obbligatori può ostacolare notevolmente l'elaborazione entrare requisiti poiché, ad esempio, le informazioni possono non essere ancora disponibili nel momento in cui un requisito viene inserito per la prima volta. Per questo motivo, i campi obbligatori dovrebbero essere dichiarati come tali solo con parsimonia e attenzione.

Un prerequisito per verificare l'acquisizione degli attributi è una corrispondente valutazione, che di solito può essere effettuata attraverso l'uso di uno strumento.

Per gli attributi opzionali che non devono necessariamente essere inseriti, la loro valutazione può avere le seguenti conseguenze:

- **L'attributo non era stato utilizzato né in una vista né in un report:** Questo caso indica che l'attributo in questione non ha un obiettivo specifico, poiché non è visibile in nessuna vista o analisi.
- **L'attributo viene sempre popolato con lo stesso valore, ad esempio il valore di default:** In questo caso, non sembra esserci una distinzione reale nel contesto dell'attributo per i diversi requisiti, e questo suggerisce che l'elenco di valori proposto non è adeguato. Il Requirements Engineer dovrebbe verificare se l'attributo può essere omesso (perché non esiste un obiettivo reale per questo) o se l'elenco di selezione deve essere modificato. In questo ultimo caso, dovrebbero essere prese in considerazione le note di 3.4.
- **L'attributo non viene mai compilato:** Se l'attributo non è stato deliberatamente compilato, l'informazione può non essere importante. Se questa assunzione viene confermata, l'attributo deve essere eliminato. Tuttavia, la motivazione spesso risiede nel fatto che gli utenti non sono consapevoli dello scopo dell'attributo, o non vedono direttamente alcun beneficio poiché le informazioni vengono utilizzate da un altro stakeholder. Dovrebbe poi essere fatta formazione agli utenti per spiegare loro il valore aggiunto del particolare attributo. In questo caso, l'attributo può essere utilizzato ancora.
- **L'attributo viene compilato solo per pochi requisiti:** La domanda è se può essere raggiunto l'obiettivo associato all'attributo, o se l'attributo è ancora rilevante per tutti i requisiti. In caso contrario, l'attributo può essere eliminato.

Tuttavia, se si scopre che l'attributo è importante, può essere dichiarato come campo obbligatorio, e questo forza l'inserimento in futuro. In questo caso, i requisiti che non hanno alcun valore nell'attributo considerato devono essere aggiornati in base al risultato della domanda che ci siamo posti (ad esempio, popolamento automatico con un valore di default).

- L'**attributo non viene popolato in casi specifici**: Deve essere determinato se questo attributo è ancora rilevante per il progetto. In caso affermativo, il Requirements Engineer dovrebbe completare i requisiti rilevanti. Se l'attributo non è più considerato essenziale, può essere eliminato o le lacune possono essere tollerate.
- L'**attributo è sempre popolato**: In questo caso, non sono necessarie altre attività.

Se il feedback degli stakeholder del processo di RE porta a nuovi requisiti rispetto alle viste definite e implementate, questi dovrebbero essere esaminati, prioritizzati e, se necessario, implementati. Tali requisiti per le viste possono portare al Change Management dello schema di attributi se, ad esempio, è richiesto un attributo aggiuntivo o un valore aggiuntivo dell'attributo (si veda 3.4).

4 Valutazione e prioritizzazione dei requisiti (L3)

Durata: 1,75 ore

Termini: Prioritizzazione, Valutazione, Tecniche di Prioritizzazione, Tecniche di Prioritizzazione Ad-Hoc, Tecniche di Prioritizzazione Analitica

Obiettivi Formativi

- EO 4.1 Conoscere le caratteristiche di base di una valutazione (L1)
- EO 4.2.1 Conoscere la relazione tra valutazione e prioritizzazione (L2)
- EO 4.2.2 Conoscere la procedura di prioritizzazione di base (L1)
- EO 4.3 Padroneggiare e utilizzare le tecniche di prioritizzazione ad-hoc (L3)
- EO 4.4 Padroneggiare e utilizzare le tecniche di prioritizzazione analitiche (L3)
- EO 4.5 Conoscere la procedura per combinare le tecniche di prioritizzazione (L2)

4.1 Principi di valutazione (L1)

In tutte le attività di Requirements Engineering, i requisiti vengono valutati sulla base di vari criteri. Ad esempio, durante l'elicitazione i requisiti vengono categorizzati in base ai criteri di Kano (si veda [IREB2023]), mentre durante la documentazione i requisiti vengono differenziati in base alla loro natura rispetto ai loro vincoli legali, oppure vengono valutati in base alla relativa criticità di rispettare linee guida e standard (come [ISO 26262]).

Il compito del Requirements Management è documentare questi risultati della valutazioni in una forma appropriata (si veda 3) e di prevedere le conseguenze della valutazione per il processo di RE (si veda 9).

Ad esempio, la valutazione di un requisito può avere un effetto sull'ulteriore gestione del requisito valutato. Un requisito che è stato valutato come particolarmente safety-critical potrebbe, ad esempio, essere sottoposto a un controllo di qualità più dettagliata durante il testing rispetto a un requisito considerato non-critical.

I possibili criteri di valutazione possono essere:

- La natura di un requisito rispetto a vincoli legali,
- Effort/costi di implementazione,
- Criticità,
- Stabilità,
- Grado di innovazione.

Sorgenti per i criteri di valutazione possono essere:

- Standard di project management,
- Linee guida e standard,
- Lo schema di attributi dei requisiti,
- Le discipline del successivo sviluppo.

Quando si definiscono i criteri di valutazione richiesti è importante specificare le seguenti caratteristiche di base di ogni valutazione:

- Intervalli di valori della valutazione (p.e. "basso", "medio" e "alto" per il criterio di valutazione "Criticità"),
- Il gruppo di persone che possono effettuare la valutazione (p.e. la "rilevanza architettuale" dovrebbe essere valutata solo dagli architetti),
- Quando è possibile eseguire per la prima volta la valutazione (p.e. l'effort di implementazione può essere stimato solo quando è disponibile una progettazione dettagliata),
- Quando è possibile eseguire per l'ultima volta la valutazione (p.e. l'effort di implementazione deve essere stimato prima che un requisito possa essere assegnato a una release o a un'iterazione di sviluppo).

I risultati delle valutazioni che sono state già eseguite possono essere utilizzate nel corso del progetto per prendere decisioni su tutte le attività di Requirements Engineering [POHL2010]. Pertanto, le valutazioni possono servire come base per il consolidamento, utilizzando tecniche di consolidamento analitico (si veda [IREB2023]), o come base per la prioritizzazione dei requisiti.

4.2 Prioritizzazione dei requisiti (L2)

Non esiste un progetto dove le risorse disponibili siano illimitate. Per garantire che le risorse disponibili siano utilizzate in un modo orientato agli obiettivi, è essenziale prioritizzare i requisiti esistenti. In linea di principio, la prioritizzazione dovrebbe sempre seguire il seguente processo:

- Determinare gli obiettivi di prioritizzazione
- Determinare i criteri di prioritizzazione
- Determinazione degli stakeholder coinvolti nella definizione delle priorità
- Determinare i requisiti da prioritizzare
- Selezionare la tecnica di prioritizzazione
- Adattare lo schema di attributi, se necessario
- Eseguire la prioritizzazione.
- Eseguire una verifica regolare e, se necessario, riprioritizzare i requisiti

I criteri di prioritizzazione richiesti dipendono dall'obiettivo di prioritizzazione. Il primo passo è quindi determinare quali decisioni devono essere prese sulla base dei risultati della prioritizzazione. Quando si decide quali funzioni del sistema devono essere specificate in dettaglio per prime, devono essere considerati altri criteri di prioritizzazione rispetto a quando deve essere determinato l'ordine di implementazione. Ha quindi senso determinare all'inizio del progetto quali prioritizzazioni saranno richieste durante il progetto e quali valutazioni devono essere effettuate prima della prioritizzazione.

In base all'obiettivo perseguito, è necessario determinare quale criterio di valutazione o di prioritizzazione, o quale combinazione di criteri, viene utilizzato per determinare la priorità del requisito.

Quando si selezionano i requisiti da prioritizzare, i requisiti dovrebbero avere lo stesso livello di dettaglio, per evitare di distorcere il risultato della prioritizzazione [WiBe2013].

4.3 Tecniche di prioritizzazione ad-hoc (L3)

Esiste un insieme di tecniche per prioritizzare i requisiti, che a volte differiscono notevolmente in termini di effort richiesto per la prioritizzazione e l'adeguatezza a determinate situazioni di prioritizzazione. "Nei progetti, semplici tecniche di prioritizzazione ad-hoc [...] sono un approccio pragmatico [...] per una prioritizzazione efficace dei requisiti" [POHL2010]. Le seguenti tecniche di prioritizzazione ad-hoc funzionano bene nella pratica:

- Triage dei requisiti [DAVI2003]
- Classificazione [PoRu2011]
- Tecnica Top-Ten [PoRu2011]
- Classificazione in base ad un singolo criterio [PoRu2011]
- Planning poker
- Classificazione a due criteri
- Tecnica dei 100 dollari
- Classificazione Kano [PoRu2011]

4.3.1 Classificazione a due criteri (L3)

Se il risultato della classificazione a un singolo criterio non è sufficientemente differenziante, perché ad esempio devono essere considerati diversi criteri di prioritizzazione, o perché troppi requisiti rientrano nella stessa specifica del criterio selezionato, i valori di diversi criteri possono essere moltiplicati e i risultati prioritizzati in base a un ranking. Ogni requisito viene ora assegnato a una possibile combinazione di valori e riceve quindi la prioritizzazione assegnata a questa combinazione. Questa procedura viene normalmente presentata sotto forma di matrici. Un esempio di matrice con i criteri di prioritizzazione *Benefici per il cliente* e *Costi* è mostrato in Figura 4.

Anche il principio di Eisenhower, che prende il nome dall'ex presidente degli Stati Uniti, è molto popolare e propone una classificazione in base ai criteri di *Importanza* e *Urgenza* (si veda Figura 5).

		Costo		
		Alto	Medio	Basso
Vantaggi per il Cliente	Alto	Priorità 5	Priorità 2	Priorità 1
	Medio	Priorità 6	Priorità 4	Priorità 3
	Basso	Priorità 9	Priorità 8	Priorità 7

Figura 4: Esempio di classificazione a due criteri

		Urgenza	
		Urgente	Non urgente
Importanza	Importante	Priorità 1	Priorità 2
	Non importante	Priorità 3	Priorità 4

Figura 5: Classificazione a due criteri per Importanza e Urgenza

4.3.2 Tecnica dei 100 dollari (L3)

La tecnica dei 100 Dollari è molto adatta per prioritizzare pochi requisiti. L'uso di questa tecnica è consigliato per i requisiti più grossolani, con un livello di dettaglio più alto, o per cluster di requisiti.

Agli stakeholder vengono concesse 100 unità immaginarie (denaro, tempo, ecc.), che possono distribuire tra i requisiti. Gli stakeholder coinvolti hanno un periodo di tempo predefinito per riflettere sull'allocazione delle risorse a loro disposizione prima di allocarle ai requisiti. Se un requisito ha ricevuto alla fine più unità, la priorità di questo requisito è più alta.

In teoria, la tecnica funziona anche con quantità maggiore di unità (1000, 10000, ...), ma richiede molto più tempo ed effort. Si raccomanda di utilizzare uno strumento (p.e. un software) per verificare la somma delle unità allocate per ogni stakeholder.

Questa tecnica dovrebbe essere applicata solo una volta a un insieme specifico di requisiti, perché gli stakeholder potrebbero essere influenzati dalla distribuzione eseguita da altri stakeholder e quindi potrebbero potenzialmente distribuire le loro unità in modo differente la volta successiva.

4.4 Tecniche di prioritizzazione analitica (L3)

In alcuni casi, l'uso di tecniche di prioritizzazione ad-hoc sono fortemente influenzate dagli stakeholder coinvolti nel processo di prioritizzazione e sono quindi raccomandate solo in misura limitata per le decisioni molto critiche. Un risultato più neutro può essere ottenuto utilizzando le seguenti tecniche di prioritizzazione analitica.

- **Matrice di Prioritizzazione di Wieger:** La matrice proposta da Karl Wieger per la prioritizzazione dei requisiti compara il vantaggio relativo e lo svantaggio relativo di ogni requisito con i costi relativi e il rischio relativo di ogni requisito. Di conseguenza, ai

requisiti con alti benefici per il cliente e con bassi costi e rischi viene data maggiore priorità rispetto ai requisiti con gli stessi costi e rischi ma con benefici relativamente bassi per il cliente WiBe2013].

- **Processo di Gerarchia Analitica (AHP, Analytical Hierarchy Process):** L'idea di base di questa tecnica è il confronto a coppie di tutti i requisiti da prioritizzare. Una scala da 1 a 9 determina quanto sia più importante il requisito A rispetto al requisito B [KaRy1997].

4.5 Combinazione di tecniche di prioritizzazione (L2)

Le tecniche di prioritizzazione analitica hanno una scala molto ridotta rispetto alle tecniche di prioritizzazione ad-hoc. Pertanto, sia per il Processo di Gerarchia Analitica sia per la matrice di prioritizzazione di Wieger, si raccomanda un numero massimo di 25-30 requisiti per la prioritizzazione, in modo da mantenere la complessità e il tempo richiesto per i metodi all'interno di limiti gestibili WiBe2013], [MOIS2002].

Poiché molti progetti lavorano con un numero molto più elevato di requisiti, la prioritizzazione dettagliata in una scala di prioritizzazione con un range ristretto offre poco valore aggiunto, una combinazione di tecniche di prioritizzazione ad-hoc e analitiche si è dimostrata efficace.

Ad esempio, quando si pianifica una release, se è necessario decidere quali requisiti devono essere implementati in questa release, tutti i requisiti saranno prioritizzati ad-hoc in una prima fase. In questo caso sono ampiamente utilizzati il Triage dei Requisiti [DAVI2003], una classificazione a singolo criterio ispirata al settore medicale o il principio di Eisenhower.

Se la tecnica di prioritizzazione ad-hoc ha identificato un numero di requisiti da implementare superiore o inferiore rispetto a quanto può essere implementato nella release pianificata, i requisiti vicini al limite di accettazione vengono ri-prioritizzati utilizzando una tecnica di prioritizzazione analitica, e i requisiti con la priorità più alta vengono inseriti nella release mentre quelli con la priorità più bassa non vengono considerati.

5 Version management e change management (L3)

Durata: 2 ore

Termini: Version Control (Controllo della Versione), Configurazione dei Requisiti, Baseline dei Requisiti, Release (Rilascio), Modifica dei Requisiti, Processo di Change Management (Gestione delle Modifiche)

Obiettivi Formativi

- EO 5.1.1 Conoscere le attività di Version Control (L1)
- EO 5.1.2 Conoscere le caratteristiche della Configurazione dei Requisiti (L2)
- EO 5.1.3 Conoscere i compiti dello sviluppo supportati dalle Baseline dei Requisiti (L1)
- EO 5.1.4 Conoscere la necessità e gli svantaggi del branching dei requisiti (L2)
- EO 5.2.1 Conoscere le ragioni principali delle modifiche dei requisiti (L2)
- EO 5.2.2 Conoscere i tipi di modifica dei requisiti (L2)
- EO 5.2.3 Conoscere le euristiche per valutare la stabilità/volatilità dei requisiti (L2)
- EO 5.3.1 Conoscere l'obiettivo e i compiti di un Change Control Board (CCB) (L1)
- EO 5.3.2 Padroneggiare e utilizzare il processo di Change Management (L3)

5.1 Versioning dei requisiti (L2)

5.1.1 Version control dei requisiti e dei documenti di requisiti (L1)

Il versioning dei requisiti è una parte essenziale del Requirements Management [WiBe2013]. Il Version Control dei requisiti si riferisce al processo che permette di mantenere disponibili i passi dello sviluppo specifici dei requisiti e dei documenti dei requisiti durante il ciclo di vita di un sistema o di un prodotto [IREB2023]. Gli stakeholder possono così tracciare la storia dei requisiti e dei documenti di requisiti, e possono sempre riferirsi chiaramente a uno specifico stato di un requisito o di un documento di requisiti. Queste possibilità sono utili in tutti i progetti, ma supportano soprattutto i progetti collaborativi.

In base a [WiBe2013], il Version Control consiste di tre attività principali:

1. Definire uno schema per l'identificazione delle versioni
2. Identificare le versioni dei singoli requisiti
3. Identificare gli artefatti dei requisiti

Nello stesso progetto, i requisiti possono essere versionati sia a livello di singoli requisiti (si veda attività 2) sia a livello di artefatti dei requisiti (si veda attività 3). Il Version Control a livello di documento consente solo una prova approssimativa delle modifiche, mentre a livello di requisito ogni modifica a un requisito è tracciabile con precisione. Pertanto, il Version Control a livello di requisito è molto più complesso di quello a livello di documento.

Le versioni di un requisito o di un documento di requisiti devono essere identificate in modo univoco. Per questo scopo, deve essere definito un corrispondente schema di etichettatura da utilizzare durante ogni progetto.

Ogni partecipante del progetto deve avere accesso alla versione attuale dei requisiti. Le modifiche devono essere chiaramente documentate e comunicate alle persone impattate [WiBe2013].

Quando si crea una nuova versione di un requisito o di un documento di requisiti, è importante memorizzare il membro di progetto responsabile delle modifiche, le tempistiche richieste per la modifica e il motivo della modifica [RuSo2009].

5.1.2 Configurazione dei requisiti (L2)

Una configurazione dei requisiti riassume un insieme consistente di requisiti o artefatti di requisiti logicamente correlati **Fehler! Verweisquelle konnte nicht gefunden werden.**, dove ogni requisito e ogni artefatto è disponibile nella configurazione al più in una versione. In base a [POHL2010], una configurazione ha le seguenti proprietà:

- **Consistenza** – I requisiti e i documenti dei requisiti in una configurazione sono consistenti e logicamente correlati tra loro.¹
- **Unicità** – Una configurazione ha un identificativo che la identifica in modo univoco.
- **Immutabilità** – Le modifiche ai singoli requisiti o ai documenti dei requisiti di una configurazione portano a una nuova versione della configurazione.

5.1.3 Baseline dei requisiti (L1)

Una Baseline dei Requisiti è una configurazione dei requisiti consolidata che è stabile in termini di contenuto **Fehler! Verweisquelle konnte nicht gefunden werden.**, [POHL2010].

La baseline dovrebbe quindi contenere solo i requisiti pianificati per una particolare versione del prodotto (ad esempio, la release) e non quelli che sono proposti o ancora in corso di pianificazione [WiBe2013].

In questo modo si crea una base di comunicazione non ambigua per tutti i partecipanti del progetto. Il Version Control permette di correlare lo stato attuale dei requisiti selezionati con una baseline, e di ricostruirla successivamente senza ambiguità.

In base a [RuSo2009] e [POHL2010], le baseline dei requisiti supportano tre attività essenziali nel processo di sviluppo:

- Forniscono la base **per la pianificazione delle release (rilasci)**. Come configurazioni dei requisiti del prodotto che sono stabili e visibili per il cliente, servono come base di discussione per la definizione di una release.
- Vengono utilizzati per **stimare i costi di implementazione** di una particolare release.
- Permettono un **confronto con prodotti competitivi** presenti sul mercato.

¹ Spesso nella pratica vengono create configurazioni che non sono consistenti nel contenuto. Tali configurazioni sono create dall'esigenza di congelare lo stato attuale del lavoro per potervi accedere successivamente, se necessario. Ad esempio, si può creare una configurazione che documenti il punto di partenza delle attività di review.

Un momento adatto a creare una baseline dei requisiti potrebbe essere il raggiungimento di una milestone, il completamento di una specifica di sottosistema o il rilascio di un sistema [RuSo2009].

5.1.4 Branching dei requisiti (L2)

Un branch (ramo) dei requisiti descrive un insieme di requisiti che sono stati copiati dall'attuale configurazione dei requisiti, per essere modificati in modo indipendente dall'originale da quel momento in poi. A differenza delle versioni, i requisiti sono validi contemporaneamente in diversi branch [RuSo2009]. I branch dei requisiti vengono creati, ad esempio, se devono essere create varianti specifiche per il cliente di una specifica di requisiti per un prodotto.

La gestione dei requisiti è complicata e causa più danni che benefici quando viene eseguito il branching in modo indisciplinato [RuSo2009]. In particolare, si verificano i seguenti problemi:

- L'identificazione chiara dei requisiti diventerà più difficile.
- Oltre alle versioni e ai miglioramenti, il branching forma una terza dimensione dello sviluppo dei requisiti e quindi aumenta la complessità del Requirements Management.
- I branch generano informazioni ridondanti sui requisiti, che devono essere mantenute in parallelo e poi nuovamente unificate nel lungo termine.

5.2 Change management dei requisiti (L2)

5.2.1 Cause, sorgenti e tempistiche delle modifiche dei requisiti (L2)

I requisiti di un sistema (software) sono soggetti a modifiche durante il ciclo di vita di questo sistema (software). Queste modifiche possono essere necessarie per diversi motivi:

- Esigenze nuove o modificate degli stakeholder [POHL2010], capitolo 37.1, VANL2009], pag. 222
- Modifiche del contesto di sistema (ad esempio, modifiche legislative) [POHL2010], capitolo 37.1, VANL2009] pag. 222
- Errori nei requisiti esistenti
- Impatto della modifica di un requisito su altri requisiti dipendenti
- Decisioni architetturali e implementative che hanno ripercussioni sui requisiti

Le modifiche ai requisiti possono verificarsi durante l'intera ciclo di vita dello sviluppo e dell'operatività di un sistema.

La complessità derivante dai diversi momenti in cui possono verificarsi modifiche richiede che sia un processo ben definito, con metodi e strumenti adeguati.

5.2.2 Tipi di modifiche ai requisiti (L2)

Le modifiche ai requisiti vengono classificate come segue:

- Integrazione di un nuovo requisito
- Cancellazione di un requisito esistente
- Modifica di un requisito

Le modifiche comprendono non solo la descrizione diretta del requisito, ma anche altri attributi (ad esempio, la valutazione della stabilità) o le relazioni con altri artefatti (ad esempio, con gli use case o altri requisiti).

5.2.3 Analisi e documentazione della stabilità dei requisiti (L2)

I requisiti possono essere classificati in base alla loro stabilità, e quindi in base alla probabilità che la versione attuale del requisito venga modificata. I requisiti instabili saranno influenzati di più dalle modifiche rispetto ai requisiti stabili. Tale classificazione può essere eseguita attraverso la valutazione di un attributo fornito per questo scopo (si veda 3). Questa valutazione può essere utilizzata, ad esempio, per valutare il rischio di implementare uno specifico requisito. La valutazione può anche esprimere la stima dell'effort ancora necessario per creare una documentazione stabile del requisito. Inoltre, nel caso di modifiche importanti a un sistema, la valutazione può anche essere utilizzata per considerare separatamente i requisiti volatili, poiché tali requisiti si basano su decisioni considerate rispetto a diverse alternative.

Le seguenti regole euristiche servono per individuare modifiche probabili o requisiti instabili, e per mantenere un focus su questi (si veda anche VANL2009], pag. 224).

- I gruppi di requisiti che hanno lo stesso obiettivo e sono generalmente molto stabili (misurati in base alla frequenza delle modifiche) hanno minore probabilità di modifica rispetto ai singoli requisiti.
- Gli obiettivi e gli aspetti concettuali sono più stabili dei requisiti orientati alla soluzione.
- I requisiti funzionali che soddisfano gli obiettivi principali sono più stabili dei requisiti di qualità.
- I requisiti funzionali che compaiono ripetutamente nell'insieme dei requisiti (come accorpamenti, estensioni o varianti) sono solitamente considerati come requisiti stabili.
- I requisiti che descrivono scelte alternative dovrebbero essere gestiti con particolare cautela, e sono generalmente meno stabili di quelli descritti sopra, poiché le decisioni si basano spesso su conoscenze e assunzioni incomplete.
- I requisiti che sono assegnati a una variante o a un miglioramento del sistema sono più stabili di quei requisiti che non sono stati ancora assegnati.
- È improbabile che i requisiti modificati frequentemente in passato siano stabili.
- I requisiti che appartengono a un gruppo di requisiti prevalentemente stabili sono più stabili di quelli che appartengono a un gruppo di requisiti prevalentemente volatili.

5.3 Processo di change management (L3)

L'obiettivo del Change Management è quello di verificare preventivamente ogni modifica a uno o più requisiti, per poter controllare il rischio associato e per permettere di tracciare successivamente ogni modifica. Il processo di Change Management raggiunge questo obiettivo definendo le attività, le responsabilità e gli artefatti necessari per descrivere una procedura chiara per la gestione delle change request dei requisiti.

Il Change Control Board (CCB) svolge un ruolo centrale nel processo di Change Management. Per poter svolgere al meglio il proprio compito, ovvero la valutazione delle modifiche, ha senso che il CCB sia definito da ruoli provenienti da diverse aree, ad esempio [PoRu2011], capitolo 8.5:

- Cliente
- Architetto
- Requirements Engineer
- Sviluppatore
- Tester

È essenziale che sia coinvolto almeno un rappresentante di ciascuna parte, lato cliente e lato fornitore.

Le attività principali del processo di Change Management possono essere descritte come segue. L'input per un processo di Change Management è una change request, che viene definita attraverso un modulo predefinito (template) per il processo.

Fase 1: Preparazione della change request

Fase 2: Valutazione formale della change request

Fase 3: Durante la classificazione di una change request, viene determinato se è una modifica correttiva, adattiva o un'eccezione. Il Requirements Engineer è coinvolto nella valutazione, per determinare, ad esempio, la causa di una modifica. Questa valutazione è importante, ad esempio, per determinare se una modifica è stata richiesta dal fornitore del sistema (software) o dal cliente.

Fase 4: L'obiettivo dell'analisi degli impatti è stimare e documentare le conseguenze delle modifiche. Queste conseguenze devono essere valutate non solo per gli altri requisiti, ma anche per gli altri artefatti (architettura, codice sorgente, test case, materiale formativo).

Fase 5: I risultati dell'analisi degli impatti vengono utilizzati dal Change Control Board per determinare se approvare o rifiutare la change request. Non è sempre ragionevole accettare e implementare una change request. Le ragioni per un eventuale rifiuto di una change request sono, ad esempio:

- La modifica è troppo costosa e non è giustificata rispetto all'effort richiesto per la sua implementazione o al beneficio atteso.
- La modifica richiesta contraddice altri requisiti.
- L'implementazione della modifica comporterebbe un rischio troppo elevato per la stabilità del sistema (software) in considerazione.
- La modifica non è coperta da un contratto.

Per motivi di tracciabilità e per raggiungere un accordo tra gli stakeholder, è essenziale documentare le decisioni del Change Control Board.

Fase 6: Le change request accettate vengono priorizzate dal Change Control Board.

Fase 7: Le change request accettate vengono pianificate per l'implementazione e vengono implementate.

6 Tracciabilità dei requisiti (L3)

Durata: 2 ½ ore

Termini: Artefatto, Artefatto dei Requisiti, Tracciabilità, Modello di Tracciabilità

Obiettivi Formativi

- EO 6.1 Conoscere i motivi della tracciabilità dei requisiti (L1)
- EO 6.2 Conoscere le diverse viste della tracciabilità (L2)
- EO 6.3 Conoscere i tipi di relazione per le relazioni di tracciabilità (L1)
- EO 6.4 Conoscere le forme di presentazione delle relazioni di tracciabilità (L3)
- EO 6.5 Padroneggiare e utilizzare una specifica strategia di tracciabilità (L3)
- EO 6.6 Padroneggiare e utilizzare uno specifico modello di tracciabilità (L3)
- EO 6.7 Conoscere le misure per valutare la tracciabilità implementata (L2)
- EO 6.8 Conoscere le sfide della tracciabilità di artefatti non testuali (L1)

6.1 I Motivi della tracciabilità dei requisiti (L1)

La tracciabilità dei requisiti è essenziale per il Requirements Management. Nel contesto del Requirements Management, l'implementazione della tracciabilità si riferisce fondamentalmente alla manutenzione delle relazioni tra diversi requisiti e altri artefatti di sviluppo o di quality assurance.

La manutenzione orientata agli obiettivi delle relazioni di tracciabilità consente di conoscere le dipendenze esistenti tra gli artefatti, ad esempio per dimostrare l'implementazione dei requisiti o per identificare quali modifiche derivano dall'adattamento di un determinato requisito.

Nei paragrafi seguenti di questa unità didattica, viene di nuovo trattato il concetto di tracciabilità prima di spiegarne i vantaggi.

Come primo passo, vengono spiegati i diversi termini relativi alla tracciabilità dei requisiti. In letteratura esistono termini differenti che hanno essenzialmente lo stesso significato: tracciamento, verificabilità, tracciabilità, tracciabilità dei requisiti, ecc. In questa unità didattica utilizzeremo il termine **tracciabilità**, a meno che non venga fatto un riferimento specifico della letteratura considerata.

6.1.1 Cosa Significa Tracciabilità dei Requisiti? (L1)

Per tracciabilità dei requisiti si intende l'abilità di tracciare le dipendenze dai requisiti ad altri artefatti durante il loro ciclo di sviluppo o di vita. Le informazioni da documentare per la tracciabilità vengono definite dall'obiettivo che deve essere raggiunto attraverso la tracciabilità.

Ad esempio, se la tracciabilità viene utilizzata per garantire che tutti i requisiti di business in un progetto siano coperti dai requisiti di sistema, o viceversa che esista almeno un requisito di sistema per ogni requisito di business, può essere sufficiente un semplice riferimento bidirezionale tra questi artefatti.

6.1.2 Perché la tracciabilità dei requisiti e degli artefatti è importante (L1)

La tracciabilità dei requisiti e degli altri artefatti non è di solito un obiettivo di progetto, ma piuttosto un mezzo per raggiungere un obiettivo, ad esempio per dimostrare se e come un requisito è stato implementato e testato. In letteratura si possono trovare diverse ragioni che motivano la tracciabilità tra gli artefatti, si veda [HuJD2011], [PoRu2011], [WiBe2013].

- Dimostrare come gli obiettivi e i requisiti devono essere soddisfatti
- Verificare perché, se e come un requisito è stato implementato
- Identificare requisiti e proprietà del sistema non necessarie (soluzioni gold plated)
- Identificare artefatti mancanti (p.e. test case mancanti)
- Semplificazione dell'allocazione dei costi di sviluppo ai requisiti
- Supportare il riutilizzo degli artefatti
- Supportare la manutenzione, l'amministrazione e l'ulteriore sviluppo dei sistemi

La tracciabilità dei requisiti aiuta anche a rispondere a domande importanti, come l'impatto di una modifica o il motivo per cui un requisito esiste. In particolare, le seguenti analisi sono semplificate in modo significativo dalla presenza delle relazioni di tracciabilità (si veda [HuJD2011], pagina 11 e segg., [EBER2012], pagina 305 e segg., [PMI2021]):

- **Analisi degli impatti:** Analisi di quali artefatti sono impattati da una modifica (riduzione o estensione dell'ambito) (si veda 5.3 Processo di Change Management)
- **Analisi della sorgente:** Analisi del perché esiste un determinato artefatto (p.e. un requisito) per poter identificare ed evitare, ad esempio, requisiti non necessari
- **Analisi della copertura:** Analisi per verificare se tutti i requisiti e i successivi artefatti di sviluppo sono stati presi in considerazione, in modo che il prodotto desiderato possa essere completamente definito, sviluppato e testato
- **Earned Value Analysis:** Analisi per determinare lo stato di avanzamento del lavoro (valore delle prestazioni), per poterlo comparare con il piano di progetto originale e, se necessario, intraprendere azioni appropriate

Inoltre, la tracciabilità tra i requisiti e gli altri artefatti è necessaria per soddisfare determinati livelli di maturità per i modelli di riferimento (p.e. CMMI) o i vincoli legali (p.e. ISO 12207).

6.2 Diverse viste della tracciabilità (L2)

La tracciabilità dei requisiti può essere distinta essenzialmente nelle seguenti dimensioni:

- **Tracciabilità tra requisiti allo stesso livello di dettaglio.** Questo tipo di tracciabilità descrive, ad esempio, le dipendenze sul contenuto tra i requisiti funzionali.
- **Tracciabilità tra requisiti a livelli differenti di dettaglio.** Questo tipo di tracciabilità descrive, ad esempio, i requisiti legali dettagliati dai requisiti di sistema.
- **Tracciabilità tra versioni di requisiti:** Questo tipo di tracciabilità descrive l'evoluzione di un requisito nel tempo. Una particolarità di questa vista è che esiste una sola versione valida in un determinato momento.

- **Tracciabilità forward tra requisiti e artefatti di sviluppo ad essi correlati, che li realizzano.** Questo tipo di tracciabilità descrive, ad esempio, le dipendenze che documentano l'implementazione/realizzazione di un requisito come componente di sistema o test case.
- **Tracciabilità backward tra requisiti e artefatti che sono la sorgente/origine.** Questo tipo di tracciabilità descrive la giustificazione o la sorgente di un requisito.

Le ultime due dimensioni si trovano spesso nella letteratura specializzata sotto il termine di Tracciabilità Pre-Requirements Specification e Post-Requirements Specification [GoFi1994] o di Tracciabilità pre-RS e post-RS estesa [POHL2010], [PoRu2011].

Tracciabilità Pre- e Post-Requirements Specification

- La **tracciabilità Pre-Requirements Specification** è la tracciabilità dei requisiti dal contesto di sistema, come i documenti esistenti e gli stakeholder, verso la loro origine, ad esempio verso gli obiettivi, le vision o altre sorgenti dei requisiti.
- La **tracciabilità Post-Requirements Specification** è la tracciabilità dei requisiti verso i successivi artefatti di sviluppo, come la progettazione architettonica, l'implementazione e i test case.

Tracciabilità Pre- e Post-Requirements Specification Avanzata

Oltre a distinguere la tracciabilità Pre-Requirements Specification e Post-Requirements Specification, viene anche presa in considerazione la tracciabilità tra gli artefatti dei requisiti. Questa include i raffinamenti e la tracciabilità verso le funzionalità dipendenti, i requisiti di qualità, ecc.

Figura 6 illustra graficamente il concetto di tracciabilità Pre-Requirements Specification e Post-Requirements Specification estesa descritto sopra, dalla prospettiva dei requisiti verso gli artefatti che li realizzano e verso gli artefatti sorgente, e la tracciabilità tra i requisiti, allo stesso livello di astrazione o a livelli differenti.

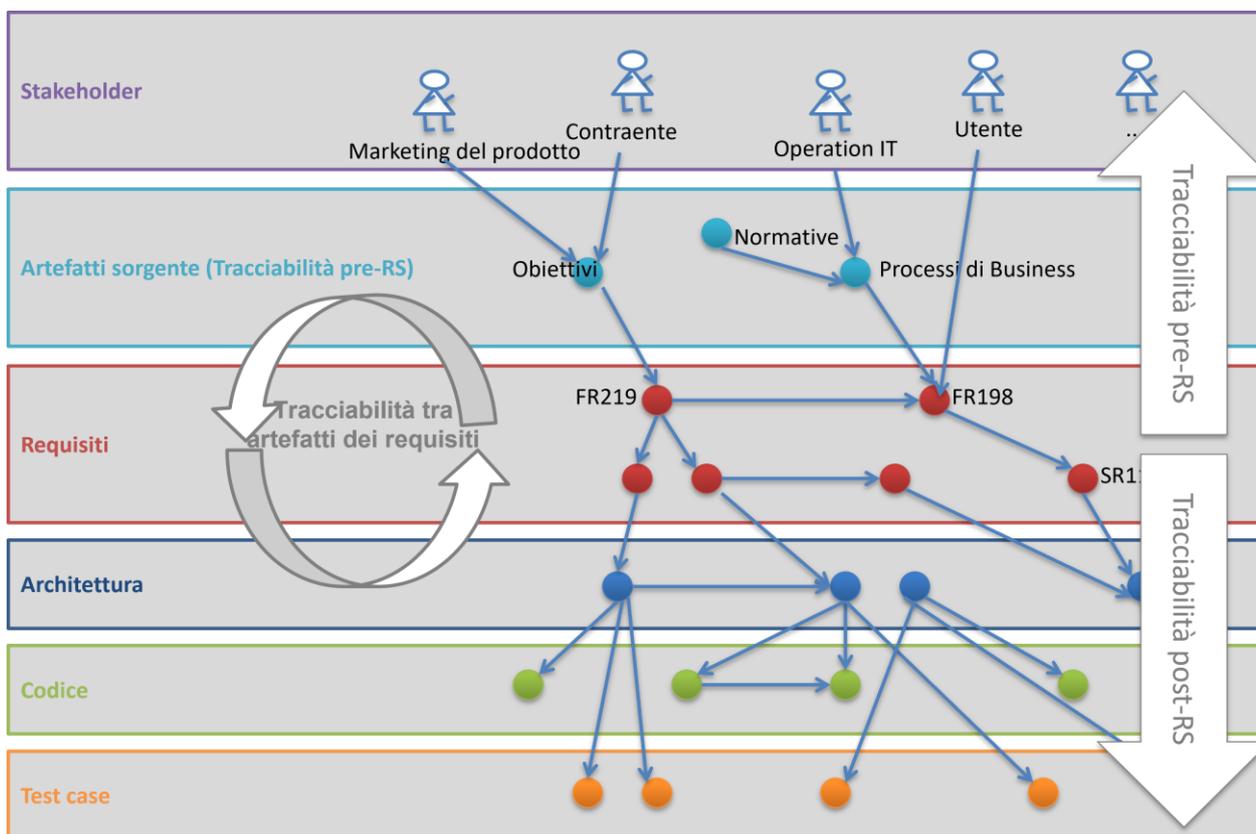


Figura 6: Distinzione delle viste di tracciabilità

È importante capire che la tracciabilità dei requisiti non inizia o finisce con i requisiti, ma dovrebbe essere "olistica" attraverso tutti i livelli di astrazione e le fasi, quindi dalla sorgente (tracciabilità pre-RS) all'implementazione o accettazione (tracciabilità post-RS).

La granularità della tracciabilità può essere a livello di singolo requisito, ma anche a livello di gruppi di requisiti o artefatti di requisiti.

6.3 Tipi di relazione per le relazioni di tracciabilità (L1)

Non esiste una definizione universalmente accettata dei tipi di relazioni di tracciabilità. È essenziale che le relazioni di tracciabilità siano utilizzate in base all'obiettivo della documentazione, e che esista un accordo nel team sui tipi di relazioni utilizzate, sul loro significato e su quali artefatti vengono utilizzati.

In teoria, le relazioni di tracciabilità potrebbero essere rappresentate da un unico tipo di relazione (p.e. "related to" o "relativo a"). Tuttavia, questa relazione non dice nulla su come gli artefatti sono relazionati tra loro. Un artefatto può essere un dettaglio, una realizzazione, un test case, una variante o una contraddizione di un altro artefatto.

Pertanto, in letteratura si trovano diversi tipi di relazioni di tracciabilità che possono essere utilizzate per una specifica documentazione di tracciabilità.

Pohl divide le relazioni di tracciabilità in classi differenti [POHL2010], alle quali possono essere assegnati diversi tipi di relazione:

- **Condizione:** La classe *condizione* include le relazioni di tracciabilità che descrivono le dipendenze relative al contenuto tra due artefatti (vincolo, precondizione, ecc.).

Esempio: **Requisito 1** (è_condizione_per) **Requisito 12**

- **Contenuto:** La classe *contenuto* include le relazioni di tracciabilità che confrontano il contenuto di due artefatti (uguaglianza, contraddizione, conflitto, ecc.).

Esempio: **Requisito 6** (è_in_contradizione_con) **Requisito 10**

- **Documentazione:** La classe *documentazione* include le relazioni di tracciabilità che forniscono ulteriori informazioni su un artefatto (motivo, esempio, commento, test case, ecc.).

Esempio: **Artefatto di test 99** (è_test_case_per) **Requisito 3**

- **Astrazione:** La classe *astrazione* include le relazioni di tracciabilità che descrivono le relazioni di astrazione tra due artefatti (classificazione, aggregazione, generalizzazione, ecc.).

Esempio: **Requisito 43** (generalizza) **Requisito 84**

- **Evoluzione:** La classe *evoluzione* include le relazioni di tracciabilità che descrivono il modo in cui un requisito viene successivamente sviluppato (soddisfatto, raffinato, sostituito, esteso, ecc.).

Esempio: **Requisito 73, versione 1.2** (sostituisce) **Requisito 73, versione 1.1**

Non è possibile fornire una risposta generale su quali tipi/classi di relazioni siano rilevanti per un particolare progetto. È importante pensare all'obiettivo e alle relazioni di tracciabilità da utilizzare prima di iniziare la documentazione (come una politica aziendale o una specifica di progetto) e definirli per tutti i partecipanti (si veda anche [MaGP2009], [MJZC2013]).

Esempio: Per garantire che tutti i requisiti all'interno dell'ambito di un progetto siano giustificati e che ogni requisito sia testato, possono essere utili i seguenti tipi di relazioni di tracciabilità:

- Relazione di tipo "soddisfatto", per garantire che non ci siano requisiti che non possono essere assegnati a nessun obiettivo di business
- Relazione di tipo "testato" per garantire che esista un test case per ogni requisito
- Relazione di tipo "giustificato" per garantire che le decisioni sulle modifiche dei requisiti siano state documentate

Per garantire che le relazioni di tracciabilità siano definite e utilizzate in modo consapevole, gli artefatti rilevanti e i tipi di relazione tra questi artefatti dovrebbero essere documentati in un modello di tracciabilità (si veda 6.6). Vedere anche 2.3.

6.4 Forme di rappresentazione delle relazioni di tracciabilità (L3)

6.4.1 Documentazione implicita ed esplicita della tracciabilità (L3)

Una relazione di tracciabilità può essere documentata in modo implicito o esplicito. Anche se questa unità di apprendimento si concentra principalmente sulla documentazione esplicita della tracciabilità, questo paragrafo approfondirà questa distinzione.

Documentazione esplicita della tracciabilità: La tracciabilità esplicita si ottiene attraverso relazioni definite e deliberatamente stabilite tra gli artefatti (si veda capitolo 6.4.3).

Documentazione implicita della tracciabilità: La tracciabilità implicita può essere ottenuta, ad esempio, attraverso le naming-convention o la struttura della documentazione.

6.4.2 Relazioni di tracciabilità bidirezionali e unidirezionali (L2)

Le relazioni di tracciabilità possono essere documentate come unidirezionali (dirette) o bidirezionali (non dirette), in base all'obiettivo da raggiungere con la tracciabilità.

- **Tracciabilità unidirezionale:** consente la tracciabilità da un artefatto a un altro artefatto, ma non viceversa. Ad esempio, il riferimento da un requisito di test a un requisito di sistema permette di verificare perché il requisito di test esiste o da cosa dipende. Tuttavia, il requisito di sistema non sarà in grado di rilevare un riferimento univoco di un requisito di test. Questo tipo di relazione è spesso presente nelle tecniche basate sui documenti (document-based), dove le relazioni sono mantenute manualmente, ad esempio tramite riferimenti testuali, e si riferiscono all'artefatto predecessore o successore. Si noti che la documentazione della direzione della tracciabilità è definita specificando il riferimento verso l'artefatto per cui esiste una dipendenza.

- **Tracciabilità bidirezionale:** consente la tracciabilità da un artefatto a un altro artefatto, e viceversa. A differenza della relazione unidirezionale, è possibile navigare tra gli artefatti, ad esempio da un requisito a un test case (p.e. attraverso un riferimento testuale verso un test case) e da un test case al requisito corrispondente che deve essere verificato con questo test case. Questo tipo di relazione permette di identificare gli artefatti predecessori e successori (tracciabilità Pre-Requirements Specification e Post-Requirements Specification). Negli strumenti di Requirements Management, queste relazioni sono di solito generate automaticamente, in modo che lo strumento supporti la navigazione o l'analisi degli impatti in entrambe le direzioni. Per i riferimenti puramente testuali, è richiesta una manutenzione esplicita per ogni artefatto impattato.

6.4.3 Forme di presentazione delle relazioni di tracciabilità (L3)

Per la documentazione esplicita della tracciabilità si possono scegliere diverse forme di rappresentazione. Vedi [POHL2010], [PoRu2011]:

- **Riferimenti testuali:** La documentazione e la rappresentazione tramite riferimenti testuali è il modo più semplice per implementare le relazioni di tracciabilità tra gli artefatti (si veda Figura 7). La relazione descrive il tipo di relazione e un identificativo (ID) univoco dell'artefatto a cui la relazione si riferisce (p.e. [TEST_CASE_FOR → ReqID 1189]). Questo tipo di presentazione ha il vantaggio di poter essere utilizzato indipendentemente da uno strumento di Requirements Management, e di essere facilmente comprensibile. Di solito viene documentato direttamente in un artefatto, ad esempio in un test case esiste un riferimento a un requisito.

Test case	Riferimento al requisito	Descrizione del test case	Priorità
TC_0021	FR_0012 FR_0013 FR_0016	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea ..	Alto
...			
TC_0150	FR_0020	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed ..	Alto

Figura 7: Riferimenti testuali

- **Hyperlink (collegamenti ipertestuali):** A differenza dei riferimenti testuali, gli hyperlink consentono la navigazione diretta verso l'artefatto destinatario (target). Gli hyperlink saranno sempre creati dall'artefatto sorgente verso l'artefatto destinatario. Le relazioni bidirezionali possono essere create tramite riferimenti incrociati (cross-reference). Rispetto ai semplici riferimenti testuali, l'uso di hyperlink ha il vantaggio di

poter "saltare" direttamente agli artefatti referenziati. Tuttavia, questo è generalmente possibile solo all'interno di uno strumento.

- **Matrici di tracciabilità:** In una matrice di tracciabilità, le relazioni di tracciabilità sono rappresentate da riferimenti nelle celle di una matrice (si veda Figura 8). La matrice risultante documenta la relazione dall'artefatto sorgente all'artefatto destinatario. Questo tipo di presentazione permette una presentazione astratta delle dipendenze tra gli artefatti, p.e. BR_0010 viene dettagliato da UC_10; BR_0020 è dettagliato da UC_30 e UC_40; UC_40 dettaglia (direzione inversa di lettura) BR_0020 e BR_0030.
- Le matrici di tracciabilità possono essere utilizzate per rappresentare esattamente un solo tipo di relazione tra due artefatti. Tuttavia, è anche possibile la presentazione di differenti tipi di relazione. Gli strumenti di Requirements Management come DOORS creano automaticamente queste presentazioni in base agli artefatti e alle relazioni, fornendo un overview delle relazioni tra gli artefatti. Nella pratica, tali matrici diventano rapidamente molto grandi e di difficile comprensione.

		Artefatto destinatario			
		UC_10	UC_20	UC_30	UC_40
Artefatto sorgente	BR_0010	è dettagliato da			
	BR_0011		è dettagliato da		
	BR_0020			è dettagliato da	è dettagliato da
	BR_0030				è dettagliato da

Figura 8: Matrice di tracciabilità (BR = Requisito di Business, UC = Use Case)

- **Tabelle di tracciabilità:** A differenza delle matrici di tracciabilità, le tabelle di tracciabilità consentono di descrivere le relazioni di tracciabilità tra tutti gli artefatti a differenti livelli di dettaglio (si veda Figura 9). Offrono quindi un potente strumento per documentare la tracciabilità dagli obiettivi, attraverso gli use case e i requisiti funzionali, ai test case. Questo strumento può essere utilizzato indipendentemente da uno strumento specifico di Requirements Management, per documentare la tracciabilità tra artefatti documentati in strumenti differenti (Word, Excel, Rational Rose, Visual Paradigm, Quality Center, ecc.).

Requisito di business	Use Case	Requisito funzionale	Requisito di sistema	Elemento GUI	Test case
BR_0010	UC_10	FR_0012 FR_0013 FR_0016	CRM_0011 CRM_0020 DWH_0010 Billing_0020	GUI_0081	TC_0021 TC_0022 TC_0025
BR_0011	UC_20	FR_0020	CRM_0011 CRM_0020		TC_0060 TC_0150

Figura 9: Tabella di tracciabilità (BR = Requisito di Business, UC = Use Case, FR = Requisito Funzionale, CRM = Customer Relationship Management, DWH = Data Warehouse, GUI = Graphical User Interface, TC = test case)

- Grafi di tracciabilità:** In un grafo di tracciabilità, i nodi rappresentano gli artefatti e le linee rappresentano le relazioni tra gli artefatti (si veda Figura 10). Per poter distinguere in modo immediato artefatti di sviluppo differenti (p.e. scenario, requisito, test case) e relazioni (p.e. raffina, implementa, è_testcase_per), si raccomanda di definire una forma di notazione appropriata. Il loro uso è comunque consigliato solo se questi grafi possono essere creati automaticamente in base agli artefatti e alle relazioni. Una riproduzione manuale di tali grafi e la loro manutenzione risulteranno troppo complesse nella pratica. In linea di principio, questi grafi forniscono un modo facile da comprendere per verificare le dipendenze e navigare tra i diversi artefatti.

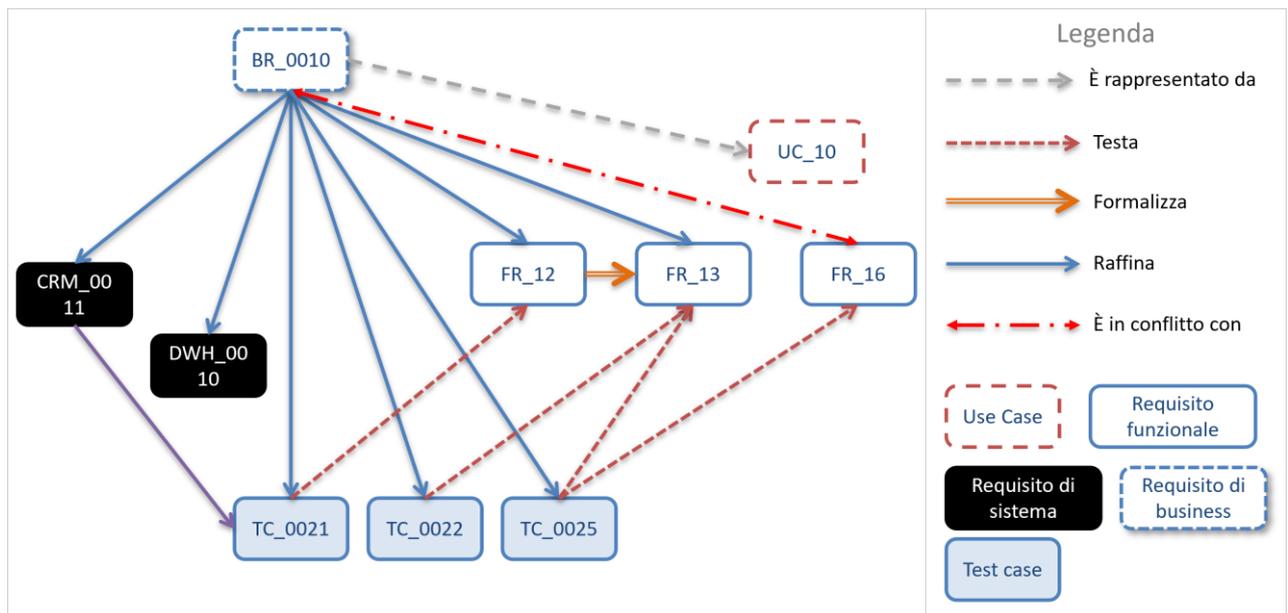


Figura 10: Grafo di rintracciabilità

La Tabella 1 riassume e valuta le seguenti relazioni di tracciabilità.

Forma di presentazione	Positiva	Negativa	Adatta per
Documentazione lineare della tracciabilità			
Riferimenti testuali	<p>Può essere implementato indipendentemente dagli strumenti e in modo completo</p> <p>La relazione è visibile nell'artefatto come testo normale.</p>	Le analisi di tracciabilità sono molto complesse.	Per rappresentare la tracciabilità in documenti di specifica scritti in linguaggio naturale e in formato cartaceo.
Hyperlink	<p>La relazione è visibile nell'artefatto come testo normale.</p> <p>Navigazione semplice tra gli artefatti per individuare le dipendenze dirette.</p>	La tracciabilità tra diversi strumenti non è facilmente realizzabile.	Per rappresentare la tracciabilità nei documenti di specifica in formato elettronico.
Documentazione ortogonale di tracciabilità			
Matrici di tracciabilità	La dipendenza tra due artefatti è visibile in modo rapido e semplice.	La creazione manuale delle matrici di tracciabilità è costosa in termini di tempo (time-consuming) e genera matrici di grandi dimensioni e poco popolate.	Rappresentazione di un singolo tipo di relazione tra due tipi di artefatti specifici (p.e. use case e requisiti)
Tabelle di tracciabilità	<p>Permette di visualizzare in modo chiaro la tracciabilità Pre-RS e Post-RS estesa.</p> <p>Permette una serie di analisi sulla tracciabilità.</p>	Elevata complessità della creazione delle tabelle.	Rappresentazione della tracciabilità tra artefatti text-based e artefatti model-based in documenti/strumenti differenti
Grafi di tracciabilità	Presentazione grafica della tracciabilità; consente una presentazione "astratta" delle relazioni di tracciabilità tra gli artefatti.	L'utilizzo è possibile solo con il supporto di uno strumento appropriato.	Rappresentazione della tracciabilità complessa tra artefatti in uno strumento di Requirements Management.

Tabella 1: Forme di presentazione delle relazioni di tracciabilità

6.5 Sviluppo di una strategia per la tracciabilità specifica del progetto (L3)

Come già detto, deve essere pianificata la creazione e l'utilizzo della tracciabilità in un progetto. Non è generalmente appropriato documentare ogni relazione tra gli artefatti. All'inizio del progetto, si dovrebbe infatti pensare al perché la tracciabilità è necessaria in questo progetto, in quale punto sarà necessaria e quale tipo di tracciabilità sarà utilizzata.

Oltre a definire gli artefatti e i tipi di relazione rilevanti, per una specifica strategia di tracciabilità è necessario definire un sistema per registrare e utilizzare queste informazioni.

Una specifica strategia di tracciabilità considera i seguenti aspetti:

- **Obiettivo della tracciabilità:** Determinare per quale ragione (perché) la tracciabilità è necessaria nel contesto del progetto o cosa si intende ottenere (per cosa) (cfr. 6.1.2).
- **Strategia di utilizzo:** Definizione delle strategie per l'utilizzo delle informazioni di tracciabilità da parte del team di sviluppo. Ad esempio, una strategia di utilizzo potrebbe essere l'analisi degli impatti delle modifiche, dove le relazioni di tracciabilità vengono utilizzate per determinare quali artefatti sono impattati da una modifica.
- **Strategia di registrazione:** Definizione delle strategie per la registrazione delle informazioni di tracciabilità da parte del team. Per poter essere eseguita, la responsabilità della documentazione delle relazioni di tracciabilità deve essere assegnata in modo esplicito. La responsabilità dovrebbe essere definita per ogni tipo di relazione (p.e. il Business Analyst per la relazione dallo use case al requisito funzionale; il Quality Manager per la relazione dal requisito funzionale al test case). Una delle strategie di registrazione potrebbe essere la documentazione cronologica delle relazioni di tracciabilità proposta da [HuJD2011] o WiBe2013]. La relazione tra due artefatti viene creata non appena viene creato il nuovo artefatto (p.e. il raffinamento del requisito o il test case). Il vantaggio è che esiste una chiara responsabilità sulla definizione delle relazioni di tracciabilità.
- **Modello di tracciabilità specifico del progetto:** Definizione delle informazioni di tracciabilità da registrare e della forma di presentazione. Un modello di tracciabilità descrive quali tipi di relazioni (p.e. è_raffinato_da; è_testato_da) tra quali artefatti (p.e. come riferimento testuale dell'altro artefatto per entrambi gli artefatti) dovrebbero essere documentati (si veda 6.6).

Quando si definisce una strategia di tracciabilità, deve essere prestata attenzione che:

- Tutto il team sia consapevole della necessità di tracciare i requisiti,
- Il modello di tracciabilità sia comprensibile e accettato da tutte le parti coinvolte,
- I membri del team conoscano e accettino le responsabilità loro assegnate per la documentazione delle relazioni di tracciabilità,
- Siano stabiliti i prerequisiti necessari per documentare correttamente la tracciabilità.

6.6 Creare e utilizzare modelli di tracciabilità specifici (L3)

Per poter creare un modello di tracciabilità specifico per il progetto, si dovrebbe prima considerare tra quali artefatti dovrebbe essere stabilita la tracciabilità e quali relazioni di tracciabilità tra questi artefatti sono necessarie o consentite. Queste specifiche dovrebbero essere descritte da un modello di tracciabilità specifico per il progetto (si veda [MJZC2013] et [POHL1994] e comunicate all'interno del progetto.

In uno specifico modello di tracciabilità tutti i partecipanti del progetto possono vedere chiaramente quali artefatti esistono, quali tipi di relazioni devono essere mantenute, chi deve mantenerle e come (si veda [POHL1994], [POHL2010], [MaGP2009]).

6.6.1 Un processo per definire un specifico modello di tracciabilità

Di seguito viene descritto un esempio di processo per definire uno specifico modello di tracciabilità.

1. Selezione di uno schema di riferimento
Il primo passo dovrebbe essere di verificare se un modello di tracciabilità esistente può essere riutilizzato e adattato. Un modo efficace per definire uno specifico modello di tracciabilità è riutilizzare un modello di tracciabilità esistente di un progetto simile oppure usato a livello aziendale. Questo modello di tracciabilità può servire come base per definire lo specifico modello di tracciabilità e di solito conterrà già un gran numero di artefatti e di dipendenze che devono essere definite.
2. Selezione degli artefatti
Durante questa fase viene definito tra quali artefatti dovrebbe essere assicurata la tracciabilità, per poter supportare l'obiettivo fissato nella strategia di tracciabilità e negli scenari di utilizzo, ad esempio la tracciabilità tra use case e requisito funzionale e tra requisito e test case.
3. Definizione dei tipi di relazione consentiti tra i tipi di artefatti
In questa fase dovrebbero essere specificati quali tipi di relazione sono consentiti per rappresentare la tracciabilità (si veda 6.3) tra due tipi di artefatti, p.e. una relazione valida tra requisito e test case è: "validato da".
4. Identificazione del numero di relazioni di tracciabilità (a livello di istanza)
In questa fase viene specificato il numero minimo di relazioni tra gli artefatti reali (a livello di istanza del modello di tracciabilità), p.e. ogni requisito richiede una relazione di tracciabilità con un test case.
5. Definizione della dipendenza tra artefatti
In questa fase viene specificato quale artefatto dipende da un altro artefatto, p.e. un test case dipende dal contenuto di un requisito. Quando si utilizzano relazioni unidirezionali, dovrebbe essere posta attenzione al riferimento (si veda 6.4.2)

6.6.2 Utilizzo di uno Specifico Modello di tracciabilità

Oltre alla definizione degli artefatti e delle relazioni di tracciabilità, p.e. documentate in un modello informativo, è necessario considerare ulteriori aspetti per l'implementazione e l'utilizzo di uno specifico modello di tracciabilità:

1. Definizione della forma di presentazione
Dopo aver definito quali sono le relazioni che dovrebbero essere documentate e tra quali artefatti, deve essere chiarito in quale tipo e forma di presentazione dovrebbero essere documentate le relazioni di tracciabilità. La selezione della forma di presentazione delle relazioni di tracciabilità è normalmente influenzata dalla forma di rappresentazione degli artefatti (si veda 6.4.3).
2. Fornire supporto per la registrazione dei dati
La registrazione delle relazioni di tracciabilità tra gli artefatti rappresenta un effort aggiuntivo (si veda **Tabella 1**), che di solito serve agli altri stakeholder (p.e. i project manager). E' quindi molto utile che venga supportata il più possibile la documentazione delle relazioni di tracciabilità, attraverso strumenti di Requirements Management o soluzioni auto-implementate, ad esempio con macro Word.
3. Allineare uno strumento agli artefatti di progetto
Quando si utilizza uno strumento di Requirements Management, è generalmente richiesta una traduzione nella terminologia esistente dello strumento. In questa fase, gli identificativi degli artefatti e i tipi di relazione definiti nel modello vengono correlati agli identificativi offerti dallo strumento e vengono referenziati in modo univoco. Ad esempio, se lo strumento offre un solo tipo di artefatto "Requisito", ma il modello di tracciabilità distingue tra "Requisito utente" e "Requisito di sistema", è necessario un mapping appropriato e, se necessario, l'assegnazione di un attributo aggiuntivo che consenta una differenziazione successiva.

6.7 Misure per la valutazione della tracciabilità implementata (L2)

La tracciabilità porta inevitabilmente a valutare se le informazioni di tracciabilità tra gli artefatti (requisiti, decisioni, codice, test case, ecc.) siano state documentate in modo corretto e completo e se la tracciabilità implementata ha raggiunto l'obiettivo effettivo (si veda 6.1).

La verifica delle informazioni di tracciabilità fornisce una vista sulla qualità dell'attuale documentazione. Questi risultati sono utili anche per identificare processi bloccati o modelli di tracciabilità "inadeguati".

I seguenti esempi di misure possono aiutare a verificare la completezza e la qualità delle relazioni di tracciabilità:

- Rapporto tra il numero di relazioni di tracciabilità corrette e il numero totale di relazioni di tracciabilità (correttezza)
- Rapporto tra il numero di relazioni di tracciabilità esistenti e il numero totale di relazioni di tracciabilità richieste (completezza)

- Rapporto tra il numero di requisiti con relazioni di tracciabilità e il numero totale di requisiti (densità)

Una misura di completezza bassa indica che le relazioni di tracciabilità non sono state mantenute in modo consistente. Una misura di correttezza bassa suggerisce che le relazioni non sono state mantenute in modo adeguato o che le modifiche non sono state applicate in modo consistente a tutti gli artefatti interessati. Qualsiasi deviazione dall'obiettivo dichiarato può avere diverse motivazioni che devono essere analizzate. I valori di soglia corrispondenti, per i quali devono essere intraprese azioni, dovrebbero essere definiti in modo specifico.

Oltre a riconoscere che la strategia di tracciabilità definita non è stata implementata, si valuta perché la tracciabilità non è stata implementata o è stata implementata in modo non corretto.

Le possibili ragioni di documentazione della tracciabilità mancante o errata sono le seguenti:

- Non è stata riconosciuta la necessità della tracciabilità
- Una strategia di tracciabilità (chi deve documentare cosa e perché) è mancante
- I vincoli di tempo non consentono di documentare la tracciabilità
- Non esiste un modello di tracciabilità concordato
- Supporto insufficiente dello strumento per la registrazione delle relazioni di tracciabilità

6.8 Sfide nella tracciabilità di artefatti non-testuali (L1)

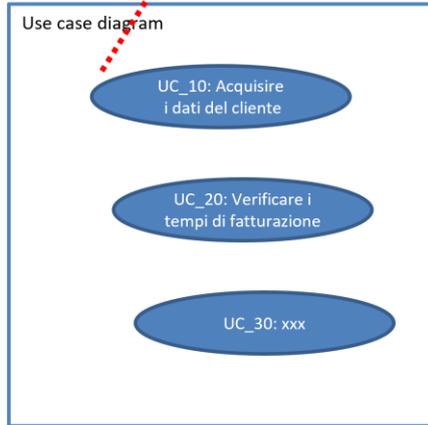
La tracciabilità tra artefatti testuali (p.e. requisiti funzionali) e artefatti model-based (p.e. le attività in un activity diagram UML), o tra gli stessi artefatti model-based, può essere ottenuta solo con un elevato effort.

Indipendentemente da come sono state create le relazioni di tracciabilità, l'utilizzo delle informazioni di tracciabilità per valutazioni e analisi (p.e. quale effetto fa modificare un requisito di business sul software esistente) è ancora complesso, poiché di solito non è possibile aprire direttamente l'artefatto corrispondente (come è possibile con un hyperlink). E' proprio questa manutenzione che rende comunque possibile un'analisi degli impatti.

Una delle opzioni, che è indipendente dallo strumento, per documentare i riferimenti tra gli artefatti testuali e gli elementi del modello sono le tabelle di tracciabilità (si veda 6.4.3), che possono permettere di referenziare in modo chiaro sia gli artefatti testuali sia gli elementi del modello. Per fare questo è necessario assegnare identificativi (ID) univoci (in modo manuale o automatico) agli elementi del modello. Utilizzando un riferimento testuale o una tabella di tracciabilità è possibile referenziare in modo chiaro gli elementi del modello, come gli use case ai requisiti testuali corrispondenti.

Figura 11 mostra un semplice esempio di riferimento ai requisiti testuali e agli use case in uno use case diagram utilizzando una tabella di tracciabilità.

Requisito di business	Use Case	Requisito funzionale	Requisito di sistema	Elemento GUI	Test case
BR_0010	UC_10	FR_0012 FR_0013 FR_0016	CRM_0011 CRM_0020 DWH_0010 Billing_0020	GUI_0081	TC_0021 TC_0022 TC_0025
BR_0011	UC_20	FR_0020	CRM_0011 CRM_0020		TC_0060 TC_0150



FR_ID	Requisito funzionale	Priorità
FR_0012	Il nome del cliente deve essere verificati rispetto ai dati del cliente nell'archivio dati prima dell'inserimento.	alto
FR_0013	Il conto bancario del cliente deve essere validato rispetto a HUB SEPA	alto
...

Figura 11: Tracciabilità tra artefatti testuali e artefatti model-based

7 Gestione delle varianti per i requisiti (L5)

Durata: 3 ore

Termini: Variabilità, Riutilizzo, Famiglia di Prodotti, Linea di Prodotti, Funzionalità, Modello di Funzionalità, Punto di Variazione, Variante, Tempo di Vincolo

Obiettivi Formativi

- EO 7.1.1 Conoscere le ragioni dell'utilizzo di varianti dei requisiti (L1)
- EO 7.1.2 Conoscere i termini chiave nel campo della Gestione delle Varianti per i Requisiti (L1)
- EO 7.1.3 Conoscere i benefici della documentazione esplicita della variabilità (L2)
- EO 7.2.1 Conoscere le forme comuni di presentazione per creare le varianti dei requisiti (L3)
- EO 7.2.2 Valutare una determinata forma di rappresentazione della variabilità in base a specifici criteri (L5)
- EO 7.2.3 Analizzare una forma di rappresentazione della variabilità rispetto a una situazione operativa in uno specifico contesto (L4)
- EO 7.3.1 Conoscere il concetto di modellazione della funzionalità (L1)
- EO 7.3.2 Padroneggiare e utilizzare un modello di funzionalità (L3)

7.1 Uso di varianti dei requisiti (L1)

Spesso lo sviluppo del prodotto non crea solo un singolo prodotto ma, in parallelo o in modo sequenziale, anche una serie di prodotti simili. In questo caso, ha senso non solo eseguire gli sviluppi dei prodotti in modo indipendente l'uno dall'altro, ma riutilizzare in modo appropriato gli artefatti di sviluppo, come i requisiti, le architetture, il codice o i test case [ISO 26550].

Se prodotti simili vengono sviluppati attraverso il riutilizzo sistematico e pianificato degli artefatti di sviluppo, si parla di linea di prodotti o famiglia di prodotti [CINo2007].

La definizione del termine linea di prodotti è la seguente: "Una linea di prodotti software è un insieme di sistemi software-intensivi che condividono un insieme comune e gestito di funzionalità, che soddisfano le richieste specifiche di un particolare segmento di mercato o di una mission, e che sono sviluppati in modo prescrittivo a partire da un insieme comune di asset core". (da [C1No2007]).

Gli asset core includono generalmente i requisiti. I requisiti per un insieme di prodotti non vengono gestiti in modo indipendente l'uno dall'altro, ma sono opportunamente inseriti in un "pool di requisiti" comune e vengono assegnati ai singoli prodotti.

Il concetto di linea di prodotti può anche essere limitato solo a livello di requisito. Anche in questo caso, i requisiti per i diversi prodotti vengono gestiti in un "pool di requisiti", e i requisiti per specifici prodotti vengono derivati da questo pool, anche se lo sviluppo dei diversi prodotti procede in modo indipendente. In questo caso si parla di linea di prodotti dei requisiti.

In entrambi i casi, un pool di requisiti è un insieme di requisiti che contiene più dell'insieme di requisiti per un prodotto specifico. Un pool di requisiti può anche contenere requisiti che non sono attualmente inclusi in alcun prodotto.

Nello sviluppo di una linea di prodotti è comune distinguere due processi [WeLa1999], [ISO 26550]:

- Ingegneria di dominio (domain engineering): nell'ingegneria di dominio viene sviluppato un "prodotto di base"
- Ingegneria applicativa (application engineering): in questo caso il "prodotto base" viene adattato allo specifico prodotto

Un concetto centrale per il riutilizzo dei requisiti è la **variabilità** [POHL2010], [PoBL2005]. La variabilità permette di definire e realizzare prodotti differenti selezionando varianti concrete da un insieme specifico di possibili varianti.

Ogni variante si riferisce sempre a un punto di variazione. I punti di variazione descrivono dove ci sono variazioni all'interno di una linea di prodotti, le varianti descrivono le specifiche (dei requisiti) possibili (accettabili) in un punto di variazione. In genere un documento di requisiti contiene un gran numero di punti di variazione con diverse varianti. Ma non tutte le combinazioni di singole varianti sono ammissibili. Esistono delle tipiche dipendenze delle varianti. Le dipendenze delle varianti possono essere facoltative oppure obbligatorie. Opzionale significa che una variante può (ma non deve) essere selezionata. Obbligatorio significa che una variante deve essere selezionata (p.e. tra diverse alternative).

Selezionando varianti specifiche per ogni punto di variazione, viene "configurato" uno specifico prodotto. Se vengono considerate tutte le dipendenze delle varianti, il risultato è un prodotto valido e specifico.

In pratica, durante l'identificazione dei requisiti non vengono assegnati varianti concrete a tutti i punti di variazione: viene selezionata una delle diverse possibili varianti. Alcuni punti di variazione rimangono aperti, cioè non vengono ancora "vincolati" a una variante specifica. Il tempo di vincolo (cfr. [CoHW1998]) può durare fino al rilascio del sistema o alla sua operatività in produzione.

I possibili tempi di vincolo sono "prima dello sviluppo", "durante lo sviluppo" (tempo di implementazione), "durante la preparazione della build" (tempo di build), "durante il commissioning" (tempo di installazione), "durante il lancio di sistema" (tempo di start-up) e "durante l'esecuzione" (runtime) [ATKI2002].

La variabilità può essere documentata in modo implicito o esplicito. Nel caso di documentazione implicita, il lettore deve capire già dalla formulazione di un requisito che sono possibili diverse varianti di prodotto (specifiche di prodotto).

Nella documentazione **implicita**, la parola "o" ("or") può indicare che sono possibili versioni differenti del prodotto. Tuttavia, la parola "o" non è un indicatore affidabile di un punto di variazione, poiché viene utilizzata molto spesso anche nelle condizioni logiche. Sono possibili anche altre parole chiave, che indicano specifiche di prodotto differenti (ad esempio, "sia... sia"), e anche questi non sono sufficientemente chiari.

La documentazione **esplicita** della variabilità può essere integrata nella documentazione dei requisiti oppure nella documentazione ortogonale, cioè in un modello separato. Nel caso di requisiti testuali, i punti di variazione e le possibili varianti vengono entrambi mostrati in modo esplicito nel testo del requisito in una documentazione integrata.

Nella documentazione ortogonale, il requisito testuale rimane inalterato. La documentazione dei punti di variazione e delle varianti viene fatta in un modello separato.

Nello sviluppo della linea di prodotti e in particolare nel Requirements Management, la documentazione esplicita della variabilità presenta i seguenti vantaggi [POHL2010]:

- **Comunicazione:** La documentazione esplicita dei punti di variazione e delle possibili varianti supporta la comunicazione con gli stakeholder impattati, poiché è facile comprendere quali varianti possono essere selezionate e in quali punti.
- **Supporto alle decisioni:** La documentazione esplicita porta a decisioni più consapevoli: (1) in quali punti viene fornita la variabilità e (2) quale variante concreta è stata selezionata per un determinato prodotto.
- **Tracciabilità:** Grazie alla documentazione esplicita delle dipendenze delle varianti, queste possono essere analizzate e, in caso di modifiche ai requisiti, essere utilizzate per analizzare le successive modifiche.

7.2 Forme di documentazione Esplicita delle varianti e loro valutazione (L5)

Nei documenti dei requisiti è possibile trovare molte forme diverse di documentazione delle varianti, che utilizzano i concetti introdotti nel paragrafo 7.1: punto di variazione, variante, assegnazione del prodotto e documentazione dei tempi di vincolo in modi molto diversi.

Le forme più comuni di rappresentazione [BOUT2011] verranno presentate in forma schematica e illustrate con un breve esempio dall'industria automotive. Verranno poi analizzate in base ai concetti introdotti in 7.1, e successivamente verranno valutate, dopo aver identificato i criteri per valutare i punti di forza e i punti di debolezza delle forme di rappresentazione.

I modelli delle funzionalità rappresentano un'altra forma di rappresentazione, che sono discussi in dettaglio nel paragrafo 7.3.

Forma di rappresentazione 1: Assegnazione testuale dei requisiti ai prodotti concreti

In questo caso, i prodotti impattati vengono esplicitamente riportati nei singoli requisiti (Figura 12).

ID	Requisito
R32	L'aletta parasole della Classe-A dovrebbe essere rivestita in plastica.
R33	L'aletta parasole della Classe-E dovrebbe essere rivestita in pelle.
R34	L'aletta parasole di tutti i prodotti dovrebbe contenere uno specchio illuminato per il trucco.

Figura 12: Assegnazione testuale dei requisiti ai prodotti concreti

Forma di rappresentazione 2: Assegnazione esplicita dei requisiti a specifici prodotti

In questo caso, i singoli requisiti vengono assegnati direttamente ai prodotti (varianti di prodotto) interessati (Figura 13).

ID	Requisito	Classe-A	Classe-E
R32	L'aletta parasole dovrebbe essere rivestita in plastica.	X	
R33	L'aletta parasole dovrebbe essere rivestita in pelle.		X
R34	L'aletta parasole dovrebbe contenere uno specchio illuminato per il trucco.	X	X

Figura 13: Assegnazione esplicita dei requisiti a specifici prodotti

Esistono diverse sotto-varianti relative alla forma concreta dell'assegnazione diretta:

- Assegnazione esplicita a colonne di prodotto separate
- Selezione multipla in una colonna di prodotto (si veda anche la prima colonna di prodotto in Figura 14)

Forma di rappresentazione 3: Assegnazione esplicita dei requisiti a specifiche funzionalità di prodotto

In questo caso, i singoli requisiti vengono assegnati direttamente a diverse funzionalità del prodotto (Figura 14). Un prodotto specifico viene definito da diverse funzionalità di prodotto, che possono avere caratteristiche differenti. Un requisito appartiene a un prodotto se il requisito viene assegnato alle funzionalità appartenenti al prodotto. Ad esempio, i requisiti R33 e R34 appartengono al prodotto "Classe-E negli USA".

ID	Requisito	Serie	Mercato
R32	L'aletta parasole dovrebbe essere rivestita in plastica.	Classe-A	USA Europa
R33	L'aletta parasole dovrebbe essere rivestita in pelle.	Classe-E	USA
R34	L'aletta parasole dovrebbe contenere uno specchio illuminato per il trucco.	Classe-A Classe-E	USA Europa

Figura 14: Assegnazione esplicita dei requisiti a specifiche funzionalità di prodotto

Nota: a questa forma di rappresentazione è associato il rischio che configurazioni invalide diventino improvvisamente valide a causa di specifiche combinazioni. Esempio: In caso di R34, il requisito dovrà essere applicato solo alla Classe-A USA, Classe-E USA e Classe-E Europa. L'esclusione della classe-A Europa non può essere rappresentata dalla metodologia di classificazione selezionata.

Forma di rappresentazione 4: Assegnazione indiretta dei requisiti ai prodotti attraverso le funzionalità

In questo caso, i singoli requisiti vengono assegnati alle funzionalità di prodotto. Una configurazione di prodotto separata determina quindi quali funzionalità di prodotto sono contenute in uno specifico prodotto (cioè una specifica variante di prodotto), che determina indirettamente i requisiti rilevanti (Figura 15).

ID	Requisito	Funzionalità
R32	L'aletta parasole dovrebbe essere rivestita in plastica.	Superficie in plastica
R33	L'aletta parasole dovrebbe essere rivestita in pelle.	Superficie in pelle
R34	L'aletta parasole dovrebbe contenere uno specchio illuminato per il trucco.	

Prodotto	Funzionalità
Classe-A	Superficie in plastica e ...
Classe-E	Superficie in pelle e ...

Figura 15: Assegnazione indiretta dei requisiti alle funzionalità e alla configurazione di prodotto

I requisiti non assegnati ad alcuna funzionalità si applicano a tutti i prodotti.

Analisi delle forme di rappresentazione

Nell'analisi delle forme di rappresentazione, devono essere considerati i seguenti aspetti secondo 7.1:

- **Descrizione dei tempi di vincolo**
Tutte le forme di rappresentazione descritte sono limitate a un unico tempo di vincolo. In questo caso, si considera solo il tempo di vincolo "durante lo sviluppo" (tempo di implementazione).
- **Punti di Variazione e Varianti**
I punti di variazione possono essere identificati in queste forme di rappresentazione solo indirettamente, in quanto esistono diversi requisiti che sono palesemente contraddittori, e la contraddizione può essere risolta solo assegnando varianti a prodotti specifici (si veda requisiti R32 e R33). Le specifiche varianti sono descritte direttamente nel testo del requisito.
- **Dipendenze e Verifica delle Varianti**
Nei casi presentati, le dipendenze dalle varianti non vengono documentate (Forme di rappresentazione 1, 2, 3) o sono documentate solo indirettamente (Forma di rappresentazione 4) e quindi non possono essere verificate. Nel caso migliore, gli errori nella configurazione possono essere identificati da un ispettore umano se, ad esempio, vengono selezionati requisiti per lo stesso prodotto che sono in conflitto (p.e. se fossero stati selezionati R32 e R33 per la Classe-A). Nella forma di rappresentazione 4, la dipendenza da una variante può essere dedotta indirettamente se esistono diversi requisiti che si riferiscono alle stesse funzionalità.

Analisi dei Punti di Forza e di Debolezza

Quando si valuta una forma specifica di rappresentazione utilizzata per la variabilità, i seguenti criteri sono rilevanti per l'applicazione pratica [BOUT2011]:

- **Facilità di insegnamento:** Quanto facilmente la forma di rappresentazione scelta può essere insegnata a personale non tecnico?
- **Scalabilità:** Quanto facilmente la forma di rappresentazione scelta può essere utilizzata per un gran numero di prodotti?
- **Espandibilità:** Quanto effort è necessario per configurare un nuovo prodotto?

- Capacità di migrazione: In che misura la documentazione dei requisiti esistente può essere ulteriormente sviluppata secondo la forma di rappresentazione scelta, senza informazioni esplicite sulla variabilità?
- Verificabilità: In che misura è possibile identificare automaticamente le configurazioni errate nella forma di rappresentazione selezionata?
- Comparabilità: In che misura i requisiti di prodotti differenti possono essere facilmente confrontati?
- Modificabilità: In che misura i requisiti esistenti possono essere modificati per un singolo prodotto senza impattare gli altri prodotti nella famiglia di prodotti?

7.3 Modellazione delle funzionalità (L3)

La modellazione delle funzionalità è una tecnica comune per documentare la variabilità. Il più noto rappresentante della modellazione delle funzionalità è FODA – Feature-Oriented Domain Analysis [KCeA1990]. La modellazione delle funzionalità si è molto diffusa, soprattutto nello sviluppo di linee di prodotti, ed esistono molte estensioni e modifiche all'approccio originale FODA[ScHT2006].

Una funzionalità è quindi definita come un "*aspetto, qualità o caratteristica rilevante o distintiva visibile all'utente di un sistema software o di un sistema*" [KCeA1990]. Un modello di funzionalità descrive le funzionalità e le loro interdipendenze. Una variante di prodotto consiste in un insieme di funzionalità che descrivono il prodotto.

Una configurazione di prodotto (valida) e le funzionalità che contiene sono definite dalle condizioni al contorno specificate dal modello di funzionalità. I modelli di funzionalità sono spesso rappresentati graficamente sotto forma di feature diagram (diagramma di funzionalità).

Gli elementi descrittivi di un modello di funzionalità possono essere suddivisi nelle seguenti tre categorie:

- Elementi di base
- Elementi avanzati
- Elementi basati sulla cardinalità

Gli elementi di base di un modello di funzionalità descrivono le funzionalità parent (genitore) e le relative funzionalità child (figlio), ed esprimono le relazioni tra loro. Le funzionalità child possono avere le seguenti relazioni con le funzionalità parent:

- Obbligatorio – La funzionalità child è obbligatoria
- Opzionale – La funzionalità child può essere utilizzata
- OR – Deve essere selezionata almeno una delle funzionalità child
- Alternativo – Deve essere selezionata esattamente una delle funzionalità child

Gli elementi avanzati possono essere utilizzati per definire dipendenze aggiuntive tra le funzionalità. Le dipendenze più note sono

- A richiede B – La selezione della funzionalità A implica la selezione della funzionalità B.

- A esclude B – Le funzionalità A e B non possono essere contenute nello stesso prodotto.

Gli elementi basati sulla cardinalità possono essere utilizzati per specificare ulteriormente le relazioni consentite tra gli elementi di base, ad esempio aggiungendo notazioni come [min, max] alla relazione parent-child.

La notazione utilizzata di seguito è tratta da [CzEi2000].

Figura 16 mostra un modello di funzionalità di esempio. La funzionalità F contiene due funzionalità obbligatorie f1 e f4, dove f1 contiene due funzionalità obbligatorie f2 e f3, mentre f4 contiene solo una funzionalità opzionale f5.

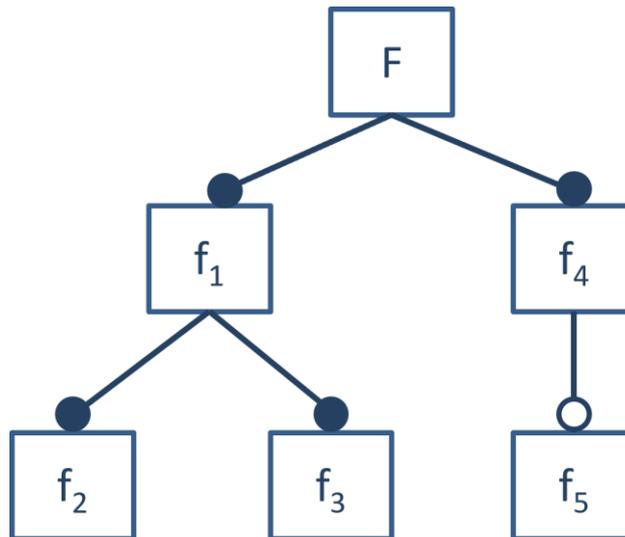
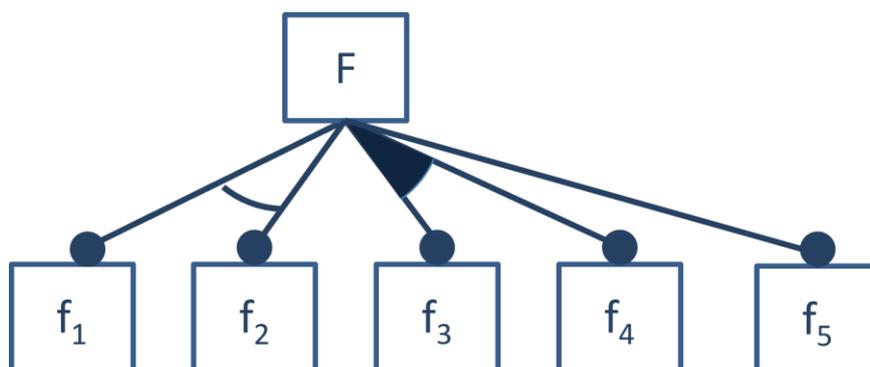


Figura 16: Esempio di un semplice modello di funzionalità

Figura 17 mostra la rappresentazione della relazione alternativa (arco vuoto) e della relazione OR (arco pieno) nel modello di funzionalità. A destra sono elencate tutte le possibili configurazioni di prodotto risultanti da questo modello.



Possibili configurazioni del prodotto:

{F; f1; f3; f5}, {F; f1; f3; f4; f5} {F; f1; f4; f5}

{F; f2; f3; f5}, {F; f2; f3; f4; f5} {F; f2; f4; f5}

Figura 17: Rappresentazione della relazione alternativa e della relazione OR (a sinistra), e delle possibili configurazioni di prodotto risultanti (a destra)

Un ulteriore esempio di dipendenze tra funzionalità è descritto in Figura 18.

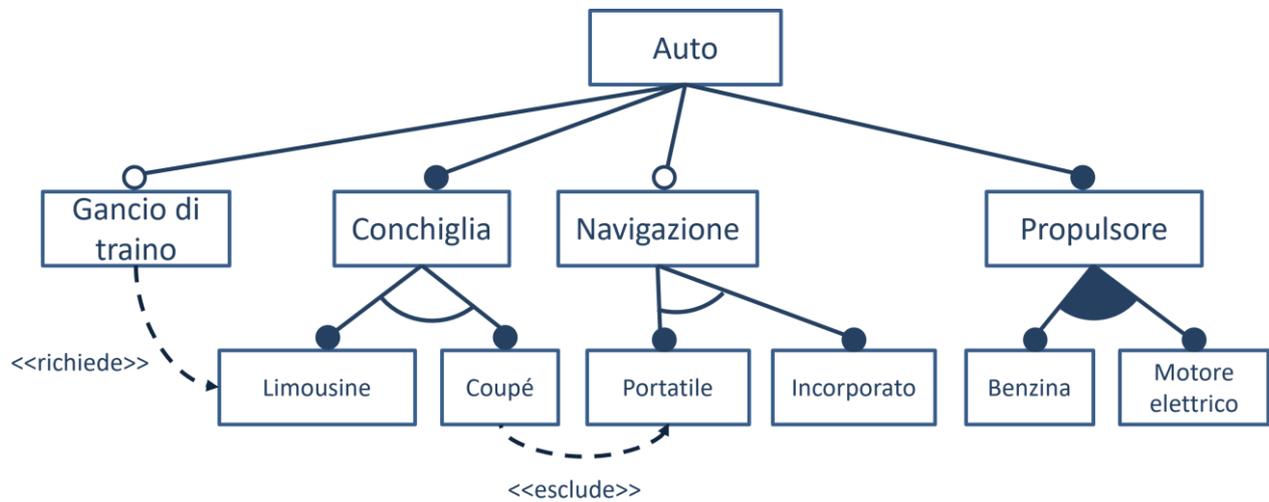


Figura 18: Modello di funzionalità per un'automobile con dipendenze

In questo esempio, il gancio di traino può essere montato solo se è una limousine.

Un'auto coupé esclude l'installazione di un navigatore portatile. Il motore può essere solo a combustione interna (a benzina), solo elettrico (motore elettrico) o una combinazione di questi (motore ibrido).

Nella modellazione delle funzionalità, le varianti sono rappresentate da elementi foglia nel modello di funzionalità. I punti di variazione non sono elementi foglia.

Le relazioni tra elementi "OR" e "Alternativo" hanno un valore più forte rispetto alle relazioni relative a un singolo elemento "Obbligatorio" e "Facoltativo".

Quando si utilizzano "OR" e "Alternativo", si può omettere l'utilizzo delle etichette "Opzionale" o "Obbligatorio" per le funzionalità child impattate.

Un esempio di elementi basati sulla cardinalità si trova in Figura 19. Il prodotto F deve avere due o tre delle funzionalità da f1 a f4.

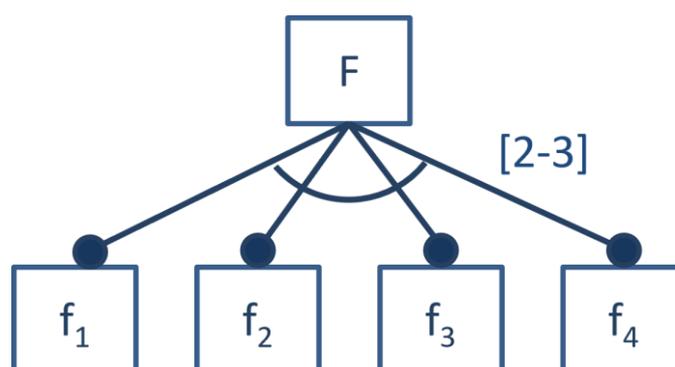


Figura 19: Modello di funzionalità con elementi basati sulla cardinalità

Identificazione delle Funzionalità

Se si vuole introdurre una gestione delle varianti basata sulle funzionalità, si pone la questione di quali funzionalità saranno utilizzate in futuro. Di norma, ha senso non inventare funzionalità completamente nuove, ma utilizzare documenti dei requisiti esistenti per definire le funzionalità [BoHo2011].

Un buon punto di partenza per l'identificazione delle funzionalità è considerare i sostantivi presenti nel testo di un requisito come potenziali funzionalità. È importante ignorare i sostantivi generici che non hanno nulla a che fare con il prodotto in quanto tale (p.e. nei paragrafi che trattano gli aspetti contrattuali o il processo di sviluppo, tali sostantivi sono chiamati anche "stop word"). Sulla base di queste liste di sostantivi, un esperto è in grado di identificare rapidamente le funzionalità potenziali. Uno svantaggio di questa procedura è che non possono essere identificati i punti di variazione che non sono esplicitamente menzionati nel testo.

I punti di variazione possono essere identificati quando il Requirements Analyst chiede in modo persistente il "perché" delle differenti varianti.

Supporto degli Strumenti

Se la variabilità deve essere esplicitamente documentata, di solito non è possibile farlo senza l'utilizzo di strumenti speciali. Sono disponibili sul mercato strumenti che consentono all'utente di

- Creare modelli di funzionalità,
- Creare configurazioni di prodotto,
- Verificare l'ammissibilità delle configurazioni di prodotto.

In genere, questi strumenti hanno interfacce distinte verso altri strumenti in cui sono presenti gli artefatti di sviluppo effettivi (p.e. i requisiti o i test case) e rendono quindi possibile mettere in relazione gli artefatti di sviluppo con le funzionalità.

8 Reporting nel Requirements Management (L3)

Durata: 1,75 ore

Termini: Reporting, Report, Figura Chiave [Misura], Metodo Goal-Question-Metric (Obiettivo-Domanda-Metrica)

Obiettivi Formativi

- EO 8.1 Conoscere gli obiettivi e i benefici del reporting nel Requirements Management (L1)
- EO 8.2 Conoscere le interfacce, i contenuti e il processo di definizione di un report (L2)
- EO 8.3.1 Conoscere le figure chiave tipiche del Requirements Management (L1)
- EO 8.3.2 Padroneggiare le figure chiave del Requirements Management utilizzando il metodo Goal-Question-Metric (L3)
- EO 8.4 Conoscere i rischi e i problemi nell'utilizzo del reporting (L2)

8.1 Obiettivi e benefici del reporting nel Requirements Management (L1)

I report sono parte del controllo dei progetti e dell'organizzazione. Servono a collezionare informazioni sui progetti o sulle unità organizzative, e a prepararle in modo appropriato per determinati gruppi target, per poter soddisfare le loro richieste di informazioni.

[ZIEG1998] definisce il reporting come "la creazione e la diffusione di report cross-funzionali nel senso di una compilazione organizzata di messaggi destinati esclusivamente al management". Un'altra definizione enfatizza la preparazione e gli obiettivi del sistema di reporting: "Può essere inteso come l'insieme di persone, facility, norme, dati e processi utilizzati per creare e distribuire report". In questo modo i report rappresentano le informazioni sommarizzate nell'ambito di un obiettivo generale, per scopo informativo." [KÜPP2005].

Il reporting nel Requirements Management (RM) è la raccolta, valutazione e presentazione delle informazioni sui requisiti o sul processo di Requirements Management, e l'approvvigionamento di queste informazioni.

Un report è un documento che combina una o più viste per uno specifico stakeholder e scopo.

Quanto riportato in questo paragrafo sul reporting nel Requirements Management si focalizza principalmente sullo sviluppo di nuovi sistemi. Tuttavia, queste procedure e questi principi possono essere applicati anche in altri contesti, come il miglioramento continuo dei sistemi e la gestione cross-progetto dei requisiti.

8.2 Stabilire un sistema di reporting nel Requirements Management (L2)

8.2.1 Interfacce (L2)

Il Requirements Management è strettamente integrato con il Project Management, il Product Management e il Quality Management. Di conseguenza, queste interfacce esistono anche all'interno del sistema di reporting, e ha senso coordinare il reporting di queste tre aree e dei relativi dati. Il Project Management e il Quality Management sono i principali destinatari del reporting del Requirements Management.

I report del Requirements Management e del Quality Management sono spesso creati insieme.

8.2.2 Contenuto di un report (L1)

In linea di principio, i report possono essere creati in qualsiasi formato (p.e. in modo informale nel testo di un email). Esistono molti modelli che forniscono a ogni report la stessa struttura. In questo modo i report sono semplici ed efficienti da leggere e creare. Le stesse informazioni si trovano sempre nello stesso punto del report. Per l'autore, è particolarmente pratico se il report può essere generato automaticamente dallo strumento in cui sono gestite le informazioni necessarie.

I contenuti standard dei report sono i seguenti:

- **Nome del progetto:** Il report deve specificare a quale progetto si riferisce. (Se il report è relativo a un'unità organizzativa, ad esempio un dipartimento, viene visualizzato il nome del dipartimento invece del nome del progetto.)
- **Data di creazione del report:** I contenuti presentati nel report cambiano ogni giorno o addirittura ogni ora. È quindi importante specificare quando i dati sono stati estratti, cioè lo stato dell'informazione su cui si basa il report.
- **Version number:** Se esistono più versioni di un report, ad esempio perché qualcuno ha fatto una modifica, la nuova versione deve avere un nuovo version number per garantire l'unicità del report e una migliore tracciabilità delle modifiche.
- **Periodo di reporting:** I report possono riferirsi a giorni, settimane, mesi, anni o qualsiasi altro intervallo di tempo. I report settimanali e mensili sono i più comuni, ma nelle fasi critiche di progetto possono essere generati anche su base giornaliera o di mezza giornata. Quando si interpretano i contenuti del report, l'interpretazione è differente se si riferisce a ciò che è stato realizzato in una settimana piuttosto che in un mese.
- **Creatore e destinatario:** Un report ha un creatore (autore) e un destinatario (lista di distribuzione). I destinatari possono anche essere distinti tra coloro che lo ricevono a titolo informativo e coloro che devono approvarlo. I nomi di queste persone sono di solito menzionati nel report e sono quindi documentati.
- **Stato del rilascio:** Se il report necessita di un rilascio, dovrebbe essere riportato in questo campo. Il report può avere contenuti differenti nei diversi stati del rilascio.

- **Stato globale:** Nelle fasi iniziali del report, il lettore vorrebbe avere una panoramica veloce di quanto è critico il progetto. I manager impegnati leggono il report solo se il progetto è critico. I report sui progetti che evolvono secondo il piano non contengono alcun valore informativo per il supervisore, poiché non è richiesto il suo supporto. Le scale basate sui semafori con i noti colori verde, giallo e rosso sono molto diffuse.
- **Contenuto tecnico:** È la parte fondamentale del report.

Il contenuto tecnico di un report di Requirements Management è costituito da una o più delle seguenti classi di contenuto:

- **Viste** selettive o aggregate **dei requisiti** (vedi 3.5). Possono essere generate automaticamente da uno strumento di Requirements Management.
- **Figure chiave** sui requisiti o sul processo di Requirements Engineering. Queste figure chiave sono generalmente ottenute grazie a valutazioni automatizzate. Questa valutazione può essere effettuata con uno strumento di Requirements Management.
- **Descrizioni** descrittive o valutative.

8.2.3 Suggerimenti per lo sviluppo e l'applicazione del reporting (L2)

Nello sviluppo e nell'applicazione del reporting (basato sui requisiti), esistono alcuni suggerimenti pratici che devono essere presi in considerazione:

- **Focus sull'essenziale:** Anche quando si conoscono gli stakeholder e i benefici del reporting, l'arte è di focalizzarsi sull'essenziale. Il Metodo Goal-Question-Metric (GQM), descritto nel paragrafo 8.3.2, è di grande aiuto.
- **Riconciliazione:** Le informazioni richieste per il reporting devono essere fornite in un modello informativo dei requisiti e in uno schema di attributi. Poiché è difficile modificare il modello informativo e lo schema di attributi in un secondo momento, e l'introduzione di un nuovo attributo richiede una manutenzione estesa dei contenuti, i modelli di dati del Requirements Management dovrebbero essere chiariti in anticipo, anche prima di un potenziale progetto di sviluppo. È utile usare modelli di riferimento che siano già stati allineati tra di loro.
- **Raccolta dei dati:** Coloro che devono raccogliere i dati (collettori dei dati) non sono gli stessi che hanno bisogno delle informazioni e che creano o leggono il report. I collettori dei dati non hanno alcuna motivazione intrinseca per inserire i dati. È quindi ancora più importante che la raccolta dei dati sia ben integrata nei processi di lavoro quotidiani e che sia chiaro chi deve inserire quali dati e quando.
- **Qualità dei dati:** La semplice presenza di attributi non significa necessariamente che tutti i contenuti siano corretti, mantenuti e tenuti aggiornati. Sebbene non abbia senso per un processo di lavoro efficiente introdurre troppi campi obbligatori, perché alcune informazioni non sono ancora disponibili al momento della creazione di un requisito, per il reporting sarebbe importante mantenere gli attributi. I contenuti mancanti portano a informazioni incomplete nei report.

8.2.4 Processo di definizione dei report (L2)

Il Requirements Manager è responsabile della definizione dei report basati sui requisiti e del loro coordinamento con il Modello Informativo dei Requisiti, oppure delega questo compito a una persona appropriata.

Secondo lo standard ISO 15288 ([ISO 15288], 6.3.7.3 a) da 1) a 4), un sistema di misurazione e reporting viene definito da questi passi:

1. Descrizione delle caratteristiche dell'organizzazione che sono rilevanti per la misurazione.
2. Identificazione e prioritizzazione delle esigenze informative.
3. Selezione e documentazione delle figure chiave che soddisfano queste esigenze informative.
4. Definizione delle procedure per la raccolta, l'analisi e il reporting dei dati.

8.3 Figure chiave nel Requirements Engineering (L3)

Le figure chiave o le misure chiave (spesso erroneamente chiamate metriche) sono una parte importante dei report. [EBER2012], pagina 436] definisce una misura come:

"(1) Un'attribuzione formale, precisa, riproducibile e oggettiva di un numero o di un simbolo a un oggetto per caratterizzare una specifica caratteristica.

(2) Matematico: Figura M di un sistema empirico C e le sue relazioni R in un sistema numerico M .

(3) L'utilizzo (raccolta, analisi, valutazione) di una misura. Esempi: Misura di un prodotto (ad esempio difetti, durata, deviazione dal piano) o di un processo (ad esempio costi dei difetti, efficienza, efficacia)".

8.3.1 Figure chiave nel Requirements Management (L1)

Le figure chiave nel Requirements Management possono essere suddivise in due classi principali:

- **Figure chiave del prodotto** (cioè figure chiave sui requisiti):
Questi includono figure chiave sul numero di requisiti e sulle caratteristiche dei requisiti.

Esempi: Numero di requisiti critici per la safety, lunghezza media di un requisito, numero di requisiti difettosi

- **Figure chiave del processo** (cioè figure chiave del processo di RE):

Esempi: Modifiche medie mensili dei requisiti, effort medio per verificare un requisito, frequenza media delle modifiche ai requisiti negli ultimi n mesi

8.3.2 Derivare le figure chiave utilizzando il metodo Goal-Question-Metric (Obiettivo-Domanda-Metrica) (L3)

Il Metodo Goal-Question-Metric (GQM) [BaCR1984] è un metodo potenziale per garantire che soltanto le figure chiave orientate agli obiettivi siano definite per i report o i contenuti dei report. GQM è una procedura sistematica per identificare queste figure chiave. Una figura chiave adeguata viene identificata rispondendo alle seguenti domande:

- Quale obiettivo deve essere raggiunto con la misurazione? (*Goal, Obiettivo*)
- Cosa dovrebbe essere misurato e a quali domande dovrebbe rispondere la misurazione? (*Question, Domanda*)
- Quali figure chiave possono descrivere le caratteristiche necessarie? (*Metric, Metrica*)

8.4 Rischi e problemi nel reporting (L2)

Esistono difficoltà pratiche nella raccolta e nella valutazione dei dati che fanno sì che i report non riflettano adeguatamente la realtà. Poiché i report hanno l'obiettivo di guidare a decisioni importanti da parte del management, un report incompleto o addirittura deliberatamente abbellito può avere conseguenze di vasta portata.

Valutazione dei Dati: Rappresentazione Condensata della Realtà

Un report è sempre un modello altamente condensato della realtà, in cui le informazioni simili vengono raggruppate in categorie e i dettagli insignificanti vengono omessi. È molto difficile creare un report che riesca a rispondere bene a qualsiasi domanda futura in qualsiasi momento.

Per questo motivo deve sempre essere presa in considerazione la superficialità di un report. E' importante evitare di trarre false conclusioni dai dati disponibili. Ad esempio, un report che mostra il 99% della tracciabilità dei requisiti non permette ancora di fare dichiarazioni sull'avanzamento del progetto o sulla qualità delle relazioni di tracciabilità. I requisiti che non sono ancora stati sottoposti a tracciabilità potrebbero essere i più importanti o quelli che più time-consuming (che richiedono più tempo), cioè quelli che contribuiscono in modo significativo al successo di un progetto. Quando si riduce la complessità delle figure chiave, si dovrebbe essere sempre consapevoli di questo problema. Spesso sono possibili solo affermazioni e conclusioni molto approssimative.

Scarsa qualità dei dati

I dati mancanti sono normalmente facili da rilevare. Non è così facile valutare la qualità dei dati: I dati corrispondono alla realtà? Sono aggiornati? Misurano esattamente ciò che dovrebbero, ad esempio, l'attributo "effort" misura solo l'effort di implementazione, anche se dovrebbe essere preso in considerazione anche l'effort del testing? La criticità è realmente il risultato di un'indagine di esperti, o è stata definita temporaneamente?

Le carenze nella qualità dei dati, conosciute oppure non rilevate, fanno sì che il report non rifletta correttamente la realtà di nuovi e ulteriori sviluppi. A causa di dati errati, è difficile prendere le giuste decisioni di management. E anche se la mancanza della qualità dei dati è nota, è difficile comunque prendere decisioni.

La scarsa qualità dei dati è spesso dovuta al fatto che le parti coinvolte trascurano la manutenzione dei dati perché ne traggono pochi benefici. Al contrario, possono essere interessati a rendere i dati troppo belli, o almeno a risparmiare tempo nella manutenzione dei dati, non eseguendo analisi accurate, ma inserendo dati che sembrano plausibili se analizzati in modo frettoloso.

La scarsa qualità dei dati può anche derivare dal fatto che non tutte le parti coinvolte hanno la stessa visione. Nello sviluppo agile (cfr. 10) la "*definition of done*" è un importante argomento di discussione. Deve essere definito in modo chiaro quando un requisito deve essere considerato completato. Possibili criteri per l'implementazione di un requisito sono, ad esempio: il codice è stato creato, gli unit test sono stati creati ed eseguiti con successo, la documentazione è stata adeguata e le regole di codifica sono state seguite.

Normative sulla protezione dei dati

Quando si definisce e si implementa un sistema di reporting, devono essere seguite delle normative generali applicative e delle normative specifiche dell'azienda sulla protezione dei dati. Se i dati personali vengono forniti dai partecipanti e successivamente comunicati in azienda a loro insaputa, sotto forma di report, possono generare problemi. E' importante concordare in modo chiaro con i creatori dei dati chi riceverà quali dati, nell'ambito delle decisioni da prendere. I dati personali e relativi alla persona dovrebbero essere utilizzati con parsimonia o non essere usati previo accordo. Quando si definiscono le viste, si dovrebbe assicurare che nessuna dichiarazione sugli individui venga fatta, in modo da non violare involontariamente le normative sulla protezione dei dati.

Reporting inflazionato

Se il volume di informazioni nei report aumenta in modo costante, questo potrebbe portare a una situazione in cui i destinatari dei report non sono in grado di elaborare questi dati a causa dei vincoli di tempo, e le decisioni importanti non possono più essere fatte su una solida base.

Quindi meno è meglio! È consigliato focalizzarsi sulle informazioni realmente necessarie. Questo può anche significare che gruppi target differenti ricevano report differenti, dove vengono rappresentati solo alcuni aspetti, a diversi livelli di dettaglio.

9 Gestione dei processi di Requirements Engineering (L3)

Durata: 2 ½ ore

Termini: Requirements Engineering (RE) Iterativo, Requirements Engineering (RE) Upfront, Requisiti Leggeri (lightweight), Processo di Requirements Engineering (Processo di RE), Ciclo PDCA (Plan Do Check Act), Continuous Process Improvement (CPI, Miglioramento Continuo del Processo)

Obiettivi Formativi

- EO 9.1 Conoscere il Requirements Engineering come processo (L2)
- EO 9.2.1 Conoscere i parametri selezionabili del processo di RE (L1)
- EO 9.2.2 Valutare l'adeguatezza di un processo di RE rispetto ai parametri di processo (L3)
- EO 9.3 Conoscere e applicare diversi metodi per documentare il processo di RE (L3)
- EO 9.4 Conoscere la necessità di monitorare e controllare il processo di RE (L1)
- EO 9.5.1 Avere padronanza e utilizzare i metodi per migliorare il processo di RE (L3)
- EO 1.4 Conoscere la necessità di un Requirements Management Plan (L2)

9.1 Requirements Engineering come processo (L2)

Un processo consiste di attività inter-dipendenti eseguite per raggiungere un obiettivo specifico. Per ogni attività, gli input (informazioni, materiali, energia, risorse) vengono trasformati in output (risultati) [ISO 9000]. Ogni attività è assegnata in modo univoco a un'entità organizzativa responsabile, ad esempio un ruolo. Quindi, Requirements Engineering e Requirements Management sono processi.

Il processo di RE (processo di Requirements Engineering) è un processo di sviluppo e gestione dei requisiti: "Un processo sistematico di sviluppo dei requisiti attraverso un processo co-operativo e iterativo di analisi del problema, di documentazione delle osservazioni risultanti in una varietà di forme di rappresentazione, e di verifica dell'accuratezza della comprensione ottenuta." [LoKa1995] sezione 13.

Questo processo di RE include le seguenti attività principali [IREB2023], capitolo 1:

- Elicitare i requisiti
- Documentazione dei requisiti
- Validazione e negoziazione dei requisiti
- Requirements Management

In ogni specifico progetto esistono diverse attività di investigative, come workshop e meeting con gli stakeholder, analisi dei documenti e così via. Lo stesso si applica anche alle altre attività principali.

Il processo di RE utilizza le esigenze e le idee degli stakeholder come informazioni di input. Anche lo status quo prima dell'inizio del progetto (p.e. sistema legacy) e i prodotti competitivi giocano un ruolo importante. Il risultato del processo di RE è una specifica di requisiti validata, senza conflitti, consistente, prioritizzata e di qualità garantita, che può servire come base affidabile per il lavoro di progetti successivi.

In generale, le quattro attività principali hanno le seguenti informazioni di input e i seguenti risultati (output), che possono essere diversi, soprattutto se devono essere soddisfatti requisiti o standard specifici dell'azienda (si veda Tabella 2):

Attività principale	Input	Risultato
Elicitazione dei requisiti	Gli stakeholder e le loro esigenze e idee; Se applicabile: un sistema legacy esistente e la relativa documentazione; i prodotti dei competitor	Requisiti orali e scritti, inclusa la vision di sistema
Documentazione dei requisiti	Requisiti orali e scritti	Specifica scritta dei requisiti (testuale, model-based o entrambi)
Validazione e Negoziazione dei Requisiti	Specifica dei requisiti scritta	Specifiche dei requisiti validate, prive di conflitti, coerenti, priorizzate e con garanzia di qualità
Requirements Management (Gestione dei Requisiti)	Specifiche scritte dei requisiti e richieste di modifica	Specifiche dei requisiti mantenute sempre aggiornate, validate, priorizzate e di qualità garantita Preparazione dei requisiti per i singoli gruppi di stakeholder

Tabella 2: Quattro principali attività del Requirements Engineering, i relativi input e risultati

Queste attività principali devono sempre essere eseguite, sia che siano documentate in modo esplicito o implicito. Standard differenti richiedono differenti implementazioni di queste attività e stabiliscono linee guida diverse sugli artefatti (si veda anche 1.4).

I risultati del processo di RE devono soddisfare i criteri di qualità nelle tre dimensioni indipendenti: specifica, rappresentazione e accordo [POHL1994]. I requisiti dovrebbero diventare più maturi nel tempo all'interno di queste dimensioni.

- **Specifica:** Questa dimensione descrive la completezza della specifica. All'inizio del processo di RE, i requisiti sono vaghi e poco chiari (opachi). Con l'avanzare del processo, i requisiti diventano più completi, nel senso di una copertura completa del problema da risolvere e di una descrizione sufficientemente dettagliata da poter essere compresa in modo appropriato. Diversi standard forniscono linee guida su quali condizioni devono essere soddisfatte dai requisiti per poter essere considerati completi. Non è comunque possibile dimostrare la completezza dei requisiti.

- **Rappresentazione:** La scala varia da informale a formale. La rappresentazione informale include schizzi, testo libero e prototipi. La rappresentazione semi-formale comprende modelli grafici come class diagram, state machine, use case diagram o data flow diagram. Anche gli use case rappresentati sottoforma di template in forma tabellare, che seguono rigorosamente una determinata struttura sintattica, sono semi-formali. Le specifiche formali descrivono chiaramente i requisiti utilizzando linguaggi logici e semantiche formali. La preparazione di una specifica formale inizia normalmente con forme di rappresentazione informali.
- **Accordo:** Stabilire un accordo (approvazione) è un altro obiettivo del processo di RE. La dimensione dell'accordo passa da una vista personale a una vista comune dei requisiti.

La specifica dei requisiti dovrebbe essere ottimizzata in tutte e tre le dimensioni. Le attività di elicitazione contribuiscono al miglioramento della dimensione della specifica, le attività di documentazione contribuiscono al miglioramento della dimensione della rappresentazione, le attività di validazione e negoziazione contribuiscono al miglioramento della dimensione del consenso. Il Requirements Management ha lo scopo di mantenere il livello di qualità in tutte e tre le dimensioni.

9.2 Parametri del processo di Requirements Engineering (L3)

Il processo di RE può essere molto vario e deve in particolare adattarsi ai vincoli dati. In tutti i diversi processi di RE esistenti nei differenti approcci, solo un numero limitato di parametri di processo possono essere modificati quando si seleziona o si adatta il processo di RE:

- Tempistica dell'elicitazione,
- Livello di dettaglio della documentazione, in particolare specifiche pesanti (heavyweight) rispetto a specifiche leggere (lightweight),
- Inserimento delle modifiche, in particolare: Change Request rispetto a Product Backlog,
- Assegnazione delle responsabilità.

Questi parametri devono essere selezionati in base ai vincoli. Questi vincoli sono:

- Le dimensioni del progetto.
- E' una nuova implementazione o un piccolo miglioramento, un'aggiunta o una variazione di un sistema o prodotto esistente e maturo?
- È stato concordato un prezzo prefissato?
- Esiste un team stabile che sta lavorando insieme da anni?
- Disponibilità di persone e loro qualifiche.

Tempistica dell'Elicitazione (Upfront o Iterativa)

I requisiti possono essere determinati completamente all'inizio del progetto (attività di Requirements Engineering upfront) o in modo iterativo (attività di Requirements Engineering iterativa): Nel primo caso (upfront), all'inizio del progetto viene creata una specifica dei requisiti (p.e. uno foglio di calcolo delle specifiche) che descrive completamente l'ambito del progetto pianificato.

Un Requirements Engineering iterativo non ha l'obiettivo di definire in modo completo i requisiti, o anche solo l'ambito del progetto, all'inizio del progetto, ma considera la documentazione dei requisiti (p.e. il Product Backlog) come una lista preliminare. I requisiti possono essere aggiunti o modificati in qualsiasi momento, anche durante l'implementazione.

Nota: Esiste una differenza tra il Requirements Engineering iterativo e lo sviluppo iterativo. È quindi plausibile creare prima una specifica completa dei requisiti (upfront) e poi implementare i requisiti attraverso uno sviluppo iterativo.

Se il progetto è un piccolo miglioramento, un'aggiunta o variazione di un sistema o prodotto esistente e maturo, ci si aspetta che possano essere definiti requisiti stabili per l'intero progetto, con poche sorprese. In questo caso è possibile e utile determinare i requisiti nelle fasi iniziali (upfront).

Se il progetto è fortemente innovativo, con molte incertezze, in un ambiente volatile, con stakeholder indecisi o in conflitto tra loro, o se esistono altri fattori di rischio che rendono impossibile una specifica iniziale (upfront) affidabile, il Requirements Engineering iterativo serve a ridurre i rischi.

Livello di dettaglio della documentazione dei requisiti

Il livello di dettaglio della documentazione o della specifica può variare tra requisiti pesanti (heavyweight) e requisiti leggeri (lightweight): le specifiche pesanti descrivono tutti i requisiti in dettaglio, includendo attributi e relazioni di tracciabilità, rendendo la specifica molto completa.

Le specifiche leggere descrivono i requisiti solo per il livello di completezza necessario e non più di quanto sia necessario. In base al modello di processo viene definito quando sono richieste determinate informazioni. Quello che è richiesto dipende dagli stakeholder, dalle loro esigenze e dal loro background. Un'analisi degli stakeholder specifici del progetto aiuta a definire il livello di dettaglio delle specifiche dei requisiti.

Lo scopo di una specifica deve essere anche di consentire allo sviluppatore di comprendere cosa vogliono gli stakeholder. I dettagli dell'implementazione sono lasciati allo sviluppatore (soprattutto se ha molta familiarità con il dominio), sono discussi verbalmente senza essere documentati o sono raffinati attraverso l'uso di un prototipo. Le specifiche dei requisiti leggere descrivono requisiti come, ad esempio, user story. I requisiti vengono specificati in dettaglio solo quando la loro implementazione sta per iniziare.

Le specifiche upfront sono di solito pesanti (p.e. nel modello waterfall e V-Modell) e le specifiche iterative sono di solito leggere (p.e. nei modelli Agile, si veda 10), i due parametri tempistica e livello di dettaglio sono indipendenti l'uno dall'altro. È possibile creare entrambi i tipi di specifica, definendo una specifica leggera upfront e una specifica pesante in modo iterativo (p.e. nel modello Rational Unified Process).

Change management: inserimento delle modifiche (change request rispetto al product backlog)

I requisiti si modificano durante il progetto. Alcuni processi di RE integrano i requisiti nuovi o modificati come change request nella specifica dei requisiti e nel processo di sviluppo. Si tratta in genere di progetti fixed-price con una specifica dei requisiti upfront, cioè l'elicitazione dei requisiti viene completata in un momento temporale predefinito dal punto di vista organizzativo e legale. Da un punto di vista legale, le modifiche successive sono modifiche contrattuali. Una change request è un nuovo contratto a livello legale. Il contratto specifica come devono essere gestite le modifiche. Si segue una procedura di approvazione semplificata con i seguenti passi: analisi (dei requisiti e dei loro benefici), analisi degli impatti (analisi delle modifiche al sistema, i costi e i rischi relativi), decisione da parte del Change Control Board e implementazione. Una change request viene spesso descritta utilizzando un template di change request, che assegna un numero univoco e un titolo, descrive il problema da risolvere e la soluzione proposta, quantifica i costi, benefici e rischi, gestisce lo stato (aperta, accettata, rifiutata, posticipata, implementata).

Nel Requirements Engineering iterativo i requisiti vengono raccolti nel Product Backlog e tutti i requisiti - vecchi e nuovi - vengono trattati allo stesso modo. Questo è possibile perché non ci si basa mai su un ambito di sistema definito. Tuttavia, non è obbligatorio che una specifica dei requisiti upfront consideri i requisiti successivi come change request.

Dovrebbe essere possibile adeguare successivamente gli artefatti dei requisiti upfront senza registrare e approvare le modifiche come change request. Le modifiche agli artefatti dei requisiti devono comunque essere documentate e tracciabili.

Assegnazione delle responsabilità

Un singolo ruolo (ad esempio, il Requirements Manager) può essere responsabile del processo di RE per la pianificazione, il controllo e il miglioramento del processo di RE.

Può anche eseguire tutte le attività del processo di RE. Tuttavia, può esistere anche un team completo o diversi ruoli responsabili del Requirements Engineering, sia per attività diverse sia per contenuti diversi (p.e. requisiti funzionali rispetto a requisiti di usabilità o di accessibilità). Il Requirements Engineering può anche essere strettamente integrato al processo di sviluppo senza un processo di RE separato o senza il ruolo di Requirements Analyst. In questo caso, il team di sviluppo esegue le attività di Requirements Engineering, cioè i membri del team raccolgono, documentano, verificano e gestiscono i requisiti.

9.3 Documentare il processo di Requirements Engineering (L3)

Il processo di RE consiste di numerose attività dei quattro tipi specificati sopra, come i workshop di elicitazione, le review delle specifiche, ecc., come descritto nel Syllabus Foundation Level [IREB2023]. Molte di queste attività sono pianificate come meeting, poiché vengono coinvolte molte persone. L'ordine di queste attività deriva dalla scelta dei parametri di processo (si veda 9.2), che possono essere definiti a livello di progetto o a livello aziendale. Le attività e la loro sequenza possono essere presentate come un activity diagram UML. L'activity diagram può anche rappresentare l'assegnazione delle attività ai ruoli.

L'assegnazione delle responsabilità delle attività ai ruoli può anche essere presentata in modo più dettagliato utilizzando una matrice RACI come la seguente. RACI significa:

- R = responsible = responsabile dell'esecuzione dell'attività
- A = accountable = responsabile per l'autorizzazione, ad esempio dell'attività e del budget
- C = consulted = (sarà) consultato, soprattutto in termini di responsabilità tecnica e di contenuto
- I = informed = da tenere informato, cioè la persona che deve essere informata

La Tabella 3 mostra un esempio di un estratto da una matrice RACI.

Attività	Requirements Engineer	Responsabile di progetto	Utente chiave
Analisi del documento: Handbook dei sistemi legacy	R, A	I	
Workshop di creatività con gli utenti chiave	R	A	C
...			

Tabella 3: Esempio di matrice RACI per il Requirements Engineering

Per gestire date e budget in modo quantitativo, il processo di RE può anche essere presentato come project plan (piano di progetto).

Altri documenti che possono rappresentare e supportare il processo di RE sono: Project plan, checklist, template, documenti campione e linee guida.

Se molte persone sono coinvolte nel processo di RE, ha senso supportare questo processo con uno strumento. Tutti i Workflow Management System sono adatti a questo scopo.

9.4 Monitoraggio e controllo del processo di Requirements Engineering (L1)

Monitorare il processo di RE significa assicurare che tutte le attività siano svolte, i risultati definiti siano rilasciati in tempo e le attività rimangano all'interno del budget. A questo scopo sono utili i report che registrano regolarmente le date, il budget utilizzato, lo stato e la percentuale di completamento del processo di RE e delle singole attività, e che confrontano i valori effettivi con i valori target della pianificazione (cfr. 8).

Controllare il processo di RE significa eseguirlo secondo il piano o, se il processo si discosta dal piano, intraprendere azioni correttive. Ad esempio, se risulta evidente che la deadline o il budget non possono essere rispettati, devono essere determinate le conseguenze per tutto il progetto e, se appropriato, devono essere prese delle contromisure. Per adeguare il processo in corso al piano, è possibile che le attività pianificate debbano essere omesse, anticipate o eseguite con un effort minore.

Devono essere accettati dei compromessi che causino il minor danno possibile, p.e. non intervistare gruppi di stakeholder, non chiarire dei punti aperti, non specificare dettagli, rifiutare change request non importanti, e così via. È importante considerare il rischio: Il beneficio del risparmio è maggiore del possibile danno?

9.5 Process improvement nel processo di Requirements Engineering (L3)

Un processo può essere ulteriormente migliorato. La base di ogni process improvement (miglioramento del processo) è l'analisi e la documentazione dello stato attuale (si veda 9.3). Lo stato attuale del Requirements Management può essere documentato in un Requirements Management Plan (RMP). Descrive il Requirements Management di un'azienda o di un progetto: il processo di RE, il Requirements Information Model, gli attributi e le viste e tutte le altre specifiche descritte in questo Syllabus. Il Requirements Management Plan serve come documentazione dello stato attuale, per descrivere l'approccio attuale e per strutturare il process improvement pianificato.

Un process improvement può essere eseguito in modo rapido – una riorganizzazione del processo – o in modo continuativo. Una riorganizzazione del processo modifica contemporaneamente molte attività e molti parametri del processo. Questo ha il vantaggio di poter ottenere un aumento significativo dell'efficienza, che di solito si verifica solo dopo che tutti i partecipanti si sono abituati al nuovo processo. Esiste anche il rischio che il nuovo processo non si dimostri valido e riduca l'efficienza. Il reset comporterà nuovamente un alto effort.

Il continuous process improvement (miglioramento del processo continuo) evita questo rischio e porta a miglioramenti a breve termine con poco effort. Secondo il principio del Continuous Process Improvement (CPI), i processi vengono gradualmente ottimizzati ripetendo iterativamente le seguenti quattro attività (PDCA):

- **Plan:** Il processo attuale e, in particolare, la necessità di miglioramento vengono analizzati. Basandosi su questa analisi, il processo target (desiderato) viene pianificato e documentato.
- **Do:** Le azioni di miglioramento vengono implementate e testate in un progetto pilota, accompagnate da misure.
- **Check:** Viene verificato che le azioni abbiano portato al miglioramento desiderato. I valori effettivi vengono confrontati con i valori pianificati.
- **Act:** Sulla base dei risultati dell'analisi del piano attuale, vengono introdotte continuamente azioni di miglioramento o, se necessario, vengono pianificate nuove azioni. L'implementazione delle azioni viene monitorata e accompagnata da misure.

Il processo attuale e il processo target vengono classificati in base alle misure ottenute (si veda 8). Queste misure possono essere:

- Percentuale del budget di progetto investito nel Requirements Engineering. Sia il troppo che il troppo poco possono essere discutibili. Di solito si tratta del 10–30% del budget di progetto.

- Numero di requisiti ancora da implementare (pesato in base all'effort atteso). Misura il lavoro ancora da eseguire prima della fine del progetto.
- Burndown rate o velocità di burndown, cioè il numero di requisiti che sono stati implementati per unità di tempo, pesato in base all'effort. Oltre alla conoscenza dei requisiti ancora da implementare, è possibile fare previsioni (forecast) sulla durata residua del progetto.
- Frequenza di modifiche dei requisiti. Una frequenza di 1–5% dei requisiti al mese (misurato in effort) e di 30–50% nell'arco della durata del progetto è considerata normale [EBER2012]. Poche modifiche possono significare che nessuno è realmente interessato ai requisiti e che gli stakeholder non sono sufficientemente coinvolti. Anche troppe modifiche sono un segnale di allarme: i requisiti non sono ancora stabili, i gruppi di stakeholder possono essere troppo eterogenei o in conflitto tra loro, ed è ancora troppo presto implementare i requisiti.
- Il tempo di throughput delle change request dalla richiesta all'implementazione.

Con l'aiuto del benchmarking è possibile scoprire quali valori sono significativi e raggiungibili come valori target.

Le azioni di improvement possono riferirsi sia ai parametri di processo descritti in 9.2, sia a come le singole attività vengono eseguite in dettaglio, ad esempio con l'aiuto di quali metodi.

Un'altra possibilità di process improvement consiste nell'analizzare gli errori commessi nel Requirements Engineering, p.e. i difetti rilevati durante l'ispezione delle specifiche o i difetti rilasciati in produzione che possono essere ricondotti al Requirements Engineering. Ci si interroga sulle loro cause e sulle cause delle cause. Questo fornisce delle idee per azioni di miglioramento.

Modelli di maturità come CMMI [CMMI2010] o ITIL [BEIM2012], [EBEL2014] offrono un aiuto più concreto per il process improvement nel Requirements Engineering (e non solo). Descrivono le attività o le pratiche che devono essere eseguite per raggiungere un certo livello di maturità. L'introduzione di nuove attività e pratiche non ancora implementate rappresenta quindi un process improvement. Si possono utilizzare anche tutti gli altri metodi di process improvement, come TQM (Total Quality Management) e Six Sigma.

In particolare, il process improvement del Requirements Engineering supporta la raccolta di best practice di Sommerville e Sawyer [SoSa1997].

L'Action Plan (Piano di Azione) [WIEG2005], pag. 66, supporta la pianificazione concreta del process improvement e contiene quanto segue:

- Nome del progetto di improvement,
- Data,
- Obiettivi (di improvement, espressi come obiettivi di business),
- Indicatori di successo (cioè il raggiungimento degli obiettivi),
- Influenza organizzativa sulla modifica,
- Partecipanti (dipendenti, loro ruoli e budget di tempo),
- Processo di misurazione e reporting (quando saranno monitorati gli avanzamenti delle azioni previste dal piano, da chi e come),

- Dipendenze, rischi e condizioni al contorno,
- Data prevista del completamento di tutte le azioni previste dal piano,
- Azioni (3-10 per piano) con la persona responsabile, la data target, lo scopo, la descrizione, elementi da rilasciare e requisiti delle risorse.

Quando si esegue l'improvement del processo di RE, dovrebbe essere considerato che non può essere ottimizzato da solo, ma in collaborazione con altre attività di progetto come il Project Management, lo sviluppo e il testing. Le modifiche nel processo di RE influenzeranno anche queste attività di progetto.

10 Requirements Management nei progetti agile (L2)

Durata: 1 ½ ora

Termini: User Story, Sprint, Product Backlog, Burndown Chart

Obiettivi Formativi

- EO 10.1 Conoscere i principi fondamentali dello Sviluppo Software Agile (L1)
- EO 10.2 Conoscere le attività e gli artefatti chiave del Requirements Management nei progetti Agile (L1)
- EO 10.3 Sapere come l'attività di Requirements Management viene mappata nelle attività Scrum (L2)

10.1 Background (L1)

Gli approcci classici allo sviluppo software, e quindi anche al Requirements Engineering, enfatizzano un piano che si proietta nel futuro. Alla fine degli anni Novanta e all'inizio degli anni Duemila è emerso un movimento contrario, passando dai piani long-term (a lungo termine) ai piani short-term (a breve termine) con molti cicli di feedback. La classificazione di [BoTu2003] si basa su questa differenza: i due gruppi sono chiamati "Agile" (come proposto nel Manifesto Agile) e "plan-driven" (guidato dalla pianificazione). In generale, il Manifesto Agile è il fondamento comune a tutti gli approcci Agile. Dal punto di vista degli sviluppatori software, il Manifesto Agile afferma [AgiMa2001]:

Stiamo scoprendo modi migliori di creare software, sviluppandolo e aiutando gli altri a fare lo stesso. Grazie a questa attività siamo arrivati a considerare importanti:

Gli individui e le interazioni più che i processi e gli strumenti

Il software funzionante più che la documentazione esaustiva

La collaborazione col cliente più che la negoziazione dei contratti

Rispondere al cambiamento più che seguire un piano

Ovvero, fermo restando il valore delle voci a destra, consideriamo più importanti le voci a sinistra.

Questi valori dimostrano che l'approccio Agile si basa più sulla collaborazione del team, sulla produttività e sui punti di forza individuali piuttosto che sui contratti e sulla documentazione. Questo distingue i metodi Agile dagli approcci plan-driven che richiedono chiari elementi contrattuali (p.e. l'ambito del progetto, i piani di rilascio o un processo di change management definito).

10.2 Requirements Management nei progetti agile (L1)

Entrambi gli approcci Agile e plan-driven sono intrinsecamente eterogenei (si veda [KoBE2013]). Non si possono fare affermazioni generali su come vengono gestiti i requisiti nei progetti Agile. Le seguenti affermazioni non devono quindi essere considerate complete o esaustive. Il prossimo paragrafo considera Scrum [SuSc2013] come rappresentativo degli approcci Agile.

Scrum è un "framework all'interno del quale le persone possono indirizzare problemi adattivi complessi, rilasciando in modo produttivo e creativo prodotti con il massimo valore possibile" [SuSc2013].

Un requisito nei progetti Agile è la user story. Una user story descrive il ruolo del richiedente, il requisito stesso e il beneficio. Il product owner (PO) è responsabile della user story. Dalla user story, il team di sviluppo (abbreviato come TS) deriva istruzioni di lavoro concrete (task) per un ciclo di sviluppo (Sprint).

Una user story ha di solito il seguente formato:

```
AS <role> I want <TARGET/WISH>, so that <BENEFIT>.
Come <ruolo> voglio <obiettivo/desiderio>, in modo che <beneficio>.
```

L'indicazione del beneficio è facoltativo. I criteri di accettazione e i test case vengono specificati per fornire maggiori dettagli alla user story, e hanno il seguente formato:

```
Under the condition that <PRECONDITION>, if <TRIGGER> then <RESULT>
A condizione che <PRECONDIZIONE>, se <TRIGGER> allora < RISULTATO>.
```

Alle user story vengono spesso aggiunte la stima dell'implementazione ("Quanto è complessa l'implementazione della user story?") e il suo contributo di valore al progetto/vision. Le user story sono generalmente ordinate in base all'ordine di implementazione pianificato. Le user story possono essere raggruppate, p.e. in epic (una user story che è troppo grande per essere implementata da sola e quindi deve essere suddivisa) o in temi (user story con contenuti simili).

La maggior parte degli approcci Agile non specifica in modo rigoroso con quali artefatti lavorare. Lo Use Case, che è una forma di espressione ben consolidata nello sviluppo software tradizionale, può essere utilizzato come artefatto anche in un ambiente Agile (si veda [COCK2001] o [JaSB2011]).

Gli artefatti (p.e. user story o use case) sono spesso valutati e prioritizzati in un Product Backlog. In Scrum questo è di responsabilità del Product Owner. Il Product Owner è la persona che gestisce gli artefatti da implementare. Il Product Owner prioritizza gli artefatti nel Product Backlog, con una priorità basata sul contributo al valore da parte dell'artefatto e sulle esigenze degli stakeholder. L'ordine di implementazione degli artefatti deriva da un lato dalla loro priorità e dall'altro dalle dipendenze tecnologiche.

Negli approcci Agile, il reporting scritto è ridotto al minimo grazie a un lavoro altamente interattivo: la comunicazione di tutte le parti coinvolte dovrebbe ridurre l'effort richiesto per i report scritti. A questo scopo vengono utilizzati meeting periodici ("eventi"),

che variano in base ai diversi approcci. In Scrum si utilizzano meeting quotidiani del team di sviluppo ("Daily Standup"), Review e Retrospective. Durante la review viene presentato il risultato di un ciclo di sviluppo (in Scrum: "Sprint") per essere approvato dal Product Owner. In una Retrospectiva, l'interazione di un team viene valutata retrospettivamente. I report scritti includono i Burndown Chart ("Quanto lavoro deve essere ancora fatto in questo sprint?") o un Impediment Backlog ("Cosa ostacola il team nel suo lavoro?"). Lo Scrum Master (SM) assicura che questi meeting si svolgano secondo le regole Scrum.

Anche se il termine processo viene usato raramente negli approcci Agile, le linee guida degli approcci implicano un workflow concreto. Il miglioramento di questi processi sarà ottenuto attraverso retrospective regolari.

Gli approcci Agile che scalano su team ampi e distribuiti sono in fase iniziale e si stanno sviluppando alcuni framework. Alcuni approcci si trovano in [ECKS2004], [ECKS2010], [LEFF2011] e [KoBE2013].

10.3 Mappare le attività di Requirements Management con le attività Scrum (L2)

La Scrum Guide (Guida Scrum) specifica solo i processi generali di lavoro. Nella Tabella 4 le attività di Requirements Management vengono assegnate alle attività o agli artefatti Scrum. In Scrum, viene specificato anche il ruolo che esegue le attività. Non tutte le attività di Requirements Management sono coperte dalla Scrum Guide. Oltre alla Scrum Guide, esiste una quantità non trascurabile di letteratura che descrive modifiche a Scrum più o meno di successo. Dipende dal team Scrum se e come l'attività di Requirements Management viene eseguita in un progetto Scrum.

Attività di RM	Attività o artefatto Scrum	Ruolo Scrum
Attributo	Lista delle user story nel backlog: Descrizione, ordine, stima, stato e valore. Opzionale: Raggruppamento	PO, TS
Valutazione e prioritizzazione	<ul style="list-style-type: none"> ▪ Stima dei benefici e dei costi attraverso il Planning Poker (TS) ▪ Organizzazione delle User Story nel Product Backlog (PO) ▪ Selezione delle User Story per uno Sprint (PO e TS) ▪ Prioritizzazione all'interno di uno Sprint (TS) 	<ul style="list-style-type: none"> ▪ TS ▪ PO ▪ PO e TS ▪ TS

Attività di RM	Attività o artefatto Scrum	Ruolo Scrum
Tracciabilità	<p>Esiste una tracciabilità implicita delle user story verso i corrispondenti test case di accettazione e, con un'adeguata assegnazione di attributi, verso le sorgenti delle user story.</p> <p>La tracciabilità è possibile all'interno del Product Backlog (dipendenze) e dalle User Story verso il codice sorgente.</p> <p>Scrum Guide non fornisce indicazioni sul collegamento delle user Story all'interno del Product Backlog. Sarebbe plausibile la tracciabilità attraverso le epic (user story raggruppate).</p> <p>La tracciabilità sarà documentata solo se necessaria.</p>	Nessuna
Versioning dei requisiti	Il versioning delle user story non è necessario. La versione attuale di una user story è sempre rilevante.	Nessuno o PO
Modifiche	Le modifiche possono essere proposte in qualsiasi momento. Nuovi requisiti generano nuove user story, modifiche ai requisiti generano modifiche alla user story o sostituzioni di una user story con una nuova.	PO
Gestione delle Varianti	I metodi Agile non supporta in modo esplicito la gestione delle varianti. Tuttavia, è possibile utilizzare i metodi standard di gestione delle varianti.	PO
Reporting	<p>I report sono soprattutto verbali. Gli artefatti utilizzati per tracciare lo stato di completamento possono anche servire come report:</p> <ul style="list-style-type: none"> ▪ Daily Standup ▪ Review dello Sprint ▪ Retrospettiva dello Sprint ▪ Product backlog ▪ Sprint backlog ▪ Burndown chart 	TS
Process Management (Gestione del Processo)	Retrospettiva dello Sprint e Impediment Backlog	SM, TS

Tabella 4: Mapping delle attività di Requirement Management con le attività Scrum.

11 Uso degli strumenti nel Requirements Management (L1)

Durata: 0,75 ore

Termini: Strumento, Selezione dello Strumento, Scambio Dati

Obiettivi Formativi

- EO 11.1 Conoscere il ruolo degli strumenti nel Requirements Management e le relative funzioni principali (L1)
- EO 11.2 Conoscere la procedura di base per la selezione degli strumenti di Requirements Management (L1)
- EO 11.3 Conoscere la necessità di scambio dati tra strumenti (L1)

11.1 Ruolo degli strumenti nel Requirements Management (L1)

L'uso di strumenti ha lo scopo di facilitare il Requirements Manager nella documentazione e nella gestione dei requisiti. Grazie alle loro speciali funzionalità, gli strumenti di Requirements Management consentono una vista olistica dei requisiti, in quanto possono essere rappresentate le relazioni tra requisiti differenti e il ciclo di vita dei singoli requisiti.

Uno strumento di Requirements Management è un'applicazione software il cui obiettivo principale è supportare le attività del Requirements Management.

Molti strumenti sono utilizzati nello sviluppo di software e sistemi. Altri coprono alcuni aspetti del Requirements Engineering e/o del Requirements Management. La distinzione tra questi strumenti e quelli per il Requirements Management non è quindi sempre netta.

Gli strumenti per il Requirements Management si basano spesso su assunzioni sul metodo di sviluppo. Alcuni strumenti sono progettati per specifici:

- Modelli di processo (p.e. sviluppo Agile, prototipazione o product management)
- Domini applicativi (p.e. industria automotive, ingegneria biomedica o sistemi di difesa)
- Ambienti di lavoro (p.e. ufficio virtuale, collaborazione locale o globale)

Questo si traduce in una varietà di strumenti di Requirements Management che spesso supportano solo alcuni dei compiti di un Requirements Manager.

Le funzioni centrali di uno strumento di Requirements Management sono [SoSa1997]:

- Editor per i requisiti e i relativi attributi
- Import nello strumento dei requisiti a partire da documenti esistenti e export dei requisiti gestiti dallo strumento in altri formati
- Tracciamento dei requisiti
- Versioning dei requisiti e creazione delle configurazioni
- Creazione di viste dei requisiti

Come descritto in [RuSo2009], è importante assicurarsi che lo strumento di Requirements Management si adatti alle procedure e ai processi stabiliti nell'azienda.

Una volta che il processo e il modello dati sono stati creati per l'azienda, è possibile effettuare una valutazione dello strumento.

11.2 Procedura di base per la selezione dello strumento (L1)

In base a [RuSo2009], per la scelta di uno strumento di Requirements Management sono necessari i seguenti passi:

- Avvio di un progetto di selezione dello strumento.
- Definizione di una bozza dei criteri di selezione attraverso requisiti di base.
- Esecuzione di una selezione iniziale (lunga lista) per identificare i primi sistemi potenziali.
- Raffinamento del catalogo dei criteri di selezione sulla base di requisiti nuovi e raffinati dello strumento.
- Esecuzione di una selezione accurata (lista breve), fino a identificare un candidato software favorito.
- Se nessuno strumento soddisfa esattamente i requisiti, è necessario adattare (customizzare) l'applicazione software.
- Per rafforzare l'accettazione in azienda e per eliminare eventuali preoccupazioni, verrà alla fine lanciato un progetto pilota.

Nel caso in cui il progetto pilota riveli che lo strumento selezionato non fornisce il supporto desiderato, la selezione dello strumento deve essere ripetuta. Se nessuno strumento soddisfa esattamente i requisiti, è possibile eseguire un adeguamento del processo invece di un adeguamento dello strumento.

11.3 Scambio dati tra strumenti di Requirements Management (L1)

Quando diverse aziende collaborano in un progetto, spesso è necessario scambiare i requisiti tra diversi strumenti di Requirements Management.

Questo è il caso, ad esempio, di un fornitore che vuole raffinare ulteriormente i requisiti del cliente nell'ambito dello sviluppo di un sistema, o che vuole implementare la tracciabilità degli artefatti di sviluppo successivi, come la progettazione o i test case.

Lo scambio di informazioni attraverso formati standard di documentazione (che possono essere tipicamente esportati e importati) è insufficiente perchè le informazioni rilevanti, come un identificativo univoco, l'autore, le informazioni sulla struttura o sul versioning dei requisiti, vengono generalmente perse. Ma soprattutto, questo approccio non supporta lo scambio degli aggiornamenti (p.e., dopo che le modifiche sono state incorporate).

Per consentire uno scambio dei requisiti indipendente dallo strumento, pur mantenendo le caratteristiche dei requisiti in modo centralizzato, è stato creato e standardizzato lo standard industriale Requirements Interchange Format (ReqIF) dall'Object Management Group.

I vantaggi che derivano dall'uso di ReqIF sono i seguenti:

- ReqIF permette di migliorare la collaborazione tra le aziende applicando i metodi di Requirements Management in tutte le aziende.
- I partner non devono necessariamente lavorare con lo stesso strumento. I fornitori devono possedere uno strumento di Requirements Management per ogni cliente.
- I requisiti possono essere trasferiti all'interno di un'organizzazione, anche attraverso gli strumenti.
- Con ReqIF, i requisiti con tutti gli attributi e i metadati possono essere scambiati senza alcuna perdita, a differenza dell'export dei documenti in formati Word, PDF, ecc.

Diversi strumenti supportano il formato ReqIF.

12 Riferimenti Bibliografici

- [AgiMa2001] Manifesto per lo Sviluppo Software Agile. Article available at <http://agilemanifesto.org/>. Ultima revisione febbraio 2024.
- [ATKI2002] C. Atkinson: Component-based Product Line Engineering with UML. Addison-Wesley, 2002.
- [BaCR1984] V. Basili, G. Caldiera, and H. D. Rombach: The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. John Wiley & Sons, 1994, S. 528–532.
- [BEIM2012] Martin Beims: IT-Service Management mit ITIL®: ITIL® Edition 2011, ISO 20000:2011 und PRINCE2® in der Praxis, Carl Hanser Verlag GmbH & Co. KG, 3., aktualisierte Auflage, 2012.
- [BoTu2003] B. Boehm und R. Turner: Using Risk to Balance Agile and Plan-Driven Methods. In: IEEE Computer, Volume 36, Issue 6, June 2003, pp. 57–66.
- [BOUT2011] E. Boutkova: Experience with Variability Management in Requirement Specifications. In: D.E. Almeida, T. Kishi, C. Schwanninger, I. John and K. Schmid (eds): Software Product Lines – 15th International Conference (SPLC), München, 2013, pp. 303–312.
- [BoHo2011] E. Boutkova and F. Houdek: Semi-automatic identification of features in requirement specifications. In: Proceedings of the 19th International Requirements Engineering Conference, Trento, Italy, September 2011, pp. 313–318.
- [COCK2001] A. Cockburn: Writing Effective Use Cases. Addison-Wesley, 2001.
- [CzEi2000] K. Czarnecki and U.W. Eisenecker: Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000.
- [CHSM2013] J. Cleland-Huang, R. S. Hanmer, S. Supakkul, and M. Mirakhorli: The Twin Peaks of Requirements and Architecture. IEEE Software, vol. 30, n. 2, pp. 24–29, March–April, 2013
- [CINo2007] P. Clements and L. Northrop: Software Product Lines: Practices and Patterns. Addison-Wesley, Boston, 6. Auflage, 2007.
- CMMI2010] CMMI® for Development, version 1.3. CMU/SEI-2010-TR-033. Article available at https://insights.sei.cmu.edu/documents/853/2010_005_001_15287.pdf.
- [CoHW1998] J. Coplien, D. Hoffmann, and D. Weiss: Commonality and Variability in Software Engineering. In: IEEE Software, Volume 15, Issue 6, 1998, pp. 37–45.
- [DAVI2003] A. Davis: The Art of Requirements Triage. In: IEEE Computer, Volume 36, Issue 3, 2003, pp. 42–49.
- [DAVI2005] A. Davis: Just Enough Requirements Management, Dorset House, 2005.
- [EBEL2014] N. Ebel: ITIL®(R) 2011 Edition: Grundlagen und Know-how für das IT Service Management und die ITIL®(R)-Foundation-Prüfung, dpunkt.verlag GmbH, 1. Auflage, 2014

- [EBER2012] C. Ebert: Systematisches Requirements Engineering. Dpunkt, 4. Auflage, 2012.
- [ECKS2010] J. Eckstein: Agile Software Development with Distributed Teams. Dorset House Publishing, 2010.
- [ECKS2004] J. Eckstein: Agile Software Development in the Large. Dorset House Publishing, 2004.
- [GABLI2014] Springer Gabler Verlag (publisher), Gabler Wirtschaftslexikon, keyword: product line, online on the Internet: , available in german only.-43488
- [GLIN2008] M. Glinz: A Risk-Based, Value-Oriented Approach to Quality Requirements. IEEE Software, Nr. 2, pp. 34-41, 2008.
- [GoFi1994] O.C.Z. Gotel and A.C.W Finkelstein: An Analysis of the Requirements Traceability Problem. In Proceedings of IEEE International Conference on Requirements Engineering, 1994, pp. 94-101.
- [HuJD2011] E. Hull, K. Jackson and J. Dick: Requirements Engineering. Springer, 3. Auflage, 2011.
- [DIN 61508] IEC DIN EN 61508-2 Functional safety of safety-related electrical/electronic/programmable electronic systems. VDE Verlag, 2002.
- [IREB2023] Syllabus IREB Certified Professional for Requirements Engineering – Foundation Level . Ultima revisione febbraio 2024.
- [IREB Glossary] M. Glinz: A Glossary of Requirements Engineering Terminology. <https://www.ireb.org/en/downloads/tag:glossary#top>. Ultima revisione febbraio 2024.
- [ISO 12207] International Organization for Standardization (ISO): ISO/IEC 12207: 2008, Systems and software engineering — Software life cycle processes. 2008.
- [ISO 15288] International Organization for Standardization (ISO): ISO/IEC 15288: 2015, Systems and software engineering — Software life cycle processes. 15288.
- [ISO 26262] International Organization for Standardization (ISO): ISO 26262, Road vehicles -- Functional safety. 2011.
- [ISO 26550] International Organization for Standardization (ISO): ISO/IEC 26550:2013: Software and systems engineering -- Reference model for product line engineering and management. 2013.
- [ISO 9000] International Organization for Standardization (ISO): ISO 9000:2005: Quality management systems — Fundamentals and vocabulary. 2005.
- [JaSB2011] I. Jacobson, I. Spence, and K. Bittner: Use Cases 2.0. Ivar Jacobson International, 2011.
- [KCeA1990] C. Kang, S. Cohen, J. Hess, W. Novak, and A. Peterson: Feature-Oriented Domain Analysis (FODA) – Feasibility Study. Software Engineering Institute, 1990.

- [KaRy1997] J. Karlsson and K. Ryan: A Cost-Value Approach for Prioritizing Requirements. IEEE Software 14, Nr. 5, pag. 67-74, 1997.
- [KoBE2013] H.-P. Korn and J.P. Berchez (eds.): Agiles IT-Management in großen Unternehmen. Symposium, 2013. Disponibile solo in tedesco.
- [KÜPP2005] H.-U. Küpper: Controlling: Konzeption, Aufgaben, Instrumente. Schäffer-Poeschel, 4. Auflage, 2005. Disponibile solo in tedesco.
- [LEFF2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley Professional, 2011.
- [LoKa1995] P. Loucopoulos and V. Karakostas: System Requirements Engineering. McGraw-Hill, 1995.
- [MaGP2009] P. Mäder, O. Gotel and I. Philippow: Getting Back to Basics: Promoting the Use of a Traceability Information Model in Practice. In: Proceedings of 5th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE2009), Vancouver, Canada, May 2009, pp. 21-25.
- [MJZC2013] P. Mäder, P.L. Jones, Y. Zhang, and J. Cleland-Huang: Strategic Traceability for Safety-Critical Projects. In: IEEE Software, Volume 30, Issue 3, May / June 2013, pp. 58-66.
- [MOIS2002] F. Moisiadis: The fundamentals of prioritizing requirements. In: Systems Engineering, Test & Evaluation Conference, Sydney, October 2002.
- [NEMN2006] B. Nemnich: Der Sarbanes-Oxley Act und die Wirkung auf die IT. Diplomarbeit, Grin Verlag, 2006. Disponibile solo in tedesco.
- [NUSE2001] B. Nuseibeh: Weaving the Software Development Process between Requirements and Architecture. In: Proc. of ICSE2001 Workshop STRAW-01, Toronto, May 2001.
- [OMG2013] Requirements Interchange Format (ReqIF). Object Management Group, Version 1.1., 2013, available at <http://www.omg.org/spec/ReqIF/1.1/PDF/> Ultima revisione febbraio 2024.
- [PMI2021] PMI: Project Management Book of Knowledge (PMBOK). Project Management Institute, 7th Ed., 2021.
- [POHL1994] K. Pohl: The Three Dimension of Requirements Engineering: A Framework and its Application. Information Systems 3, 19 (June 1994), pp. 243-258.
- [POHL1994] K. Pohl: Process-Centered Requirements Engineering. John Wiley Research Science Press, 1996.
- [POHL2010] K. Pohl: Requirements Engineering - Fundamentals, Principles, Techniques. Springer, 2010.
- [PoBL2005] K. Pohl, G. Böckle e F. van der Linden: Software Product Line Engineering - Foundations, Principles, and Techniques. Springer, 2005.
- [PoRu2011] K. Pohl and Chris Rupp: Requirements Engineering Fundamentals. Rocky Nook, 2011.

- [RoRo2014] S. Robertson, J. Robertson: Mastering the Requirements Process – Getting Requirements Right. Addison–Wesley, 3rd Edition, 2014.
- [RuSo2009] C. Rupp & die SOPHISTen: Requirements–Engineering und –Management, Hanser, 5. aktualisierte und erweiterte Auflage, 2009.
- [SEI1999] Carnegie Mellon SEI (1999): The Capability Maturity Model, Guidelines for Improving the Software Process. Addison–Wesley, 1999.
- [SEI2010] Carnegie Mellon SEI (2010): CMMI for Services, Version 1.3, Improving processes for providing better services, 2010.
- [SCHI2001] B. Schienmann: Kontinuierliches Anforderungsmanagement Prozesse – Techniken – Werkzeuge. Addison–Wesley, 2001.
- [ScHT2006] P.–Y. Schobbens, P. Heymans, and J.C. Trigaux: Feature Diagrams: A Survey and a Formal Semantics. In: Proceedings of the 14th International Requirements Engineering Conference (RE'06), September 2006, pp. 139–148.
- [SoSa1997] I. Sommerville and P. Sawyer: Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997.
- [SuSc2013] J. Sutherland and K. Schwaber: Scrum Guide, July 2013, available at <http://www.scrum.org>. Ultima revisione febbraio 2024.
- [USCO2002] US Congress: Sarbanes–Oxley Act. Washington, USA, 107th Congress of the United States of America, 23.01.2002.
- VANL2009] A. van Lamsweerde: Requirements Engineering – from System Goals to UML Models to Software Specifications. John Wiley and Sons, 2009.
- [WeLa1999] D. Weiss and C. Lai: Software product–line engineering: a family–based software development process. Addison–Wesley, 1999.
- [WIEG2005] Karl Wiegers: Software Requirements, Microsoft Press Deutschland, 1. Auflage, 2005.
- WiBe2013] K. Wiegers e J. Beatty: Software Requirements, 3rd Edition. Microsoft Press, 2013.
- [YOUN2004] R. Young: The Requirements Engineering Handbook, Artech House, Boston, 2004.
- [ZIEG1998] K. Ziegbein: Controlling. Kiehl Friedrich Verlag, 6. Auflage, 1998.