



# Certified Professional for Requirements Engineering

Livello Foundation  
Syllabus

Stan Bühne  
Martin Glinz  
Hans van Loenhoud  
Stefan Staal

## Termini d'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 e.V. 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 di IREB e.V.

## Crediti e ringraziamenti

La prima versione di questo Syllabus è stata creata nel 2007 da Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl e Chris Rupp. Hanno collaborato anche Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyer e Joy Beatty.

La versione 3 è una revisione importante, creata da Stan Bühne, Martin Glinz, Hans van Loenhoud e Stefan Staal. Sono stati supportati da Karol Frühauf, Rainer Grau, Kim Lauenroth, Chris Rupp e Camille Salinesi.

Durante la revisione, sono stati forniti commenti da Xavier Franch, Karol Frühauf, Rainer Grau, Frank Houdek e Thorsten Weyer. Ulteriori commenti sono stati fatti da Wim Decoutere e Hans-Jörg Steffe.

Le revisioni del documento sono state eseguite da Christoph Ebert, Barbara Paech e Chris Rupp.

Il nuovo Syllabus è stato approvato per il rilascio il 22 luglio 2020 dal Consiglio IREB su raccomandazione di Xavier Franch e Frank Houdek.

La traduzione in Italiano è di Cristina Sobrero.

La revisione della versione in lingua italiana è stata effettuata da Luisa Mich e Salvatore Reale.

Si ringraziano tutti coloro che hanno collaborato.

Copyright © 2007 – 2022 per questo Syllabus attribuito agli autori elencati sopra. I diritti sono stati trasferiti a IREB International Requirements Engineering Board e.V. (IREB), Karlsruhe, Germania.

## Prefazione

Nell'estate del 2017 IREB ha condotto un'indagine per verificare l'importanza attribuita alla certificazione Certified Professional for Requirements Engineering (CPRE) Foundation Level (versione 2.2). L'obiettivo dell'indagine era quello di ottenere informazioni sull'importanza nel mercato reale della certificazione, dal punto di vista dei Training Provider e dei professionisti certificati CPRE che lavorano come Requirements Engineer [MFeA2019]. L'indagine ha evidenziato che il Syllabus CPRE Foundation Level v 2.2 in generale rispondeva alle esigenze più importanti del mercato, trasmettendo ai candidati le conoscenze fondamentali di RE (Requirements Engineering). Tuttavia, si era rilevato che alcune tecniche non sono praticamente più utilizzate, mentre mancano quelle relative a uno sviluppo più iterativo e adattivo. Questo risultato era in linea con la percezione di IREB rispetto ai cambiamenti nell'ambito del Requirements Engineering (RE). Si è quindi deciso di eseguire una revisione approfondita del Syllabus CPRE Foundation Level, eliminando i contenuti obsoleti e aggiungendo nuovi elementi. La Versione 3 di questo Syllabus riflette lo stato attuale del RE, coprendo sia l'approccio plan-driven che l'approccio Agile per la specifica e la gestione dei requisiti.

I candidati alla certificazione CPRE in accordo a questo Syllabus devono avere una conoscenza di base dello sviluppo dei sistemi con approcci Agile e plan-driven.

## Scopo del documento

Questo Syllabus definisce il Foundation Level (Livello Foundation) della certificazione "Certified Professional for Requirements Engineering" istituita 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 Foundation Level risponde alle esigenze di tutti coloro che sono coinvolti nella disciplina argomento di Requirements Engineering. Sono incluse quindi le persone che ricoprono ruoli come Requirements Engineer, Business Analyst, System Analyst, Product Owner o Product Manager, sviluppatore, Project Manager o IT Manager, o esperto di dominio.

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

## Obiettivo del Syllabus

CPRE Foundation Level fornisce i principi fondamentali che sono ugualmente validi per qualsiasi tipo di sistema (ad es. applicazioni mobile, sistemi informativi o sistemi cyber-fisici). Inoltre, CPRE Foundation Level non presuppone un processo di sviluppo specifico, e non è orientato a un dominio applicativo specifico. I Training Provider possono offrire una formazione che si focalizza su specifici tipi di sistemi, processi o domini applicativi, a condizione che gli obiettivi formativi di questo Syllabus siano pienamente coperti.

## 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)

## Formulazioni sensibili al genere

In questo documento abbiamo volutamente evitato di utilizzare formulazioni specifiche di genere.

Certamente in IREB siamo a favore di formulazioni sensibili al genere. Tuttavia, vediamo anche la necessità di formulare argomenti complessi in modo tale che possano essere facilmente compresi.

Testi che richiederebbero nella realtà una forma maschile e femminile potrebbero essere meno leggibili e quindi più difficili da capire. D'altra parte, lo scopo di questo documento è presentare e comunicare i contenuti con precisione e chiarezza. Dal momento che vogliamo aiutare i lettori a mantenere l'attenzione sul contenuto, in questo documento usiamo deliberatamente solo la forma personale maschile.

Questo non deve essere interpretato come un'espressione di mancanza di rispetto.

## Obiettivi Formativi / Livelli di Conoscenza

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

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

I livelli più alti includono quelli più bassi. Si noti che tutti i termini definiti nel glossario specificati come termini fondamentali 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/downloads/#cppe-glossary>

## Struttura del Syllabus

Il Syllabus consiste di sette capitoli principali. Ogni capitolo copre un'unità educativa (EU, Educational Unit). Il titolo di ogni capitolo principale contiene il livello cognitivo del capitolo stesso, che corrisponde al livello più alto dei relativi sottocapitoli. La durata indica il tempo

che un corso di formazione dovrebbe pianificare per quel capitolo. Le aziende di formazione sono libere di allocare un tempo superiore alla durata indicata, ma dovrebbero mantenere le proporzioni tra le diverse EU. I termini importanti utilizzati in un capitolo sono elencati all'inizio del capitolo.

### Esempio

EU 4 Pratiche per l'Elaborazione dei Requisiti (L3)

Durata: 4 ore e 30 minuti

Termini: Sorgente dei requisiti, confine di sistema, contesto del sistema, elicitazione dei requisiti, negoziazione dei requisiti, validazione dei requisiti, stakeholder, modello Kano, conflitto, risoluzione del conflitto

Questo esempio mostra che il Capitolo 4 contiene obiettivi formativi di livello L3 e che per insegnare il materiale di questo capitolo sono ritenute necessarie quattro ore e mezza.

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. Ad esempio, l'obiettivo formativo EO 4.2.1. è descritto nel sottocapitolo 4.2.

### Ordine degli argomenti nel Syllabus

L'ordine dei capitoli del Syllabus costituisce un ordine logico di argomenti. Tuttavia, gli argomenti non devono essere insegnati esattamente in tale ordine. I Training Provider sono liberi di insegnare il materiale in un qualsiasi ordine (compresa una combinazione di argomenti relativi a EU diverse) che ritengano appropriato nel contesto della loro formazione e che si adatti ai loro modelli didattici.

### L'Esame

Questo Syllabus è la base per gli esami della certificazione CPRE Foundation Level.



Una domanda d'esame può coprire il contenuto di più capitoli del Syllabus. Tutti i capitoli (da EU 1 a EU 7) del Syllabus possono essere oggetto d'esame.

Il formato delle domande d'esame è a scelta-multipla.

Gli esami possono essere svolti immediatamente dopo un corso di formazione, ma anche indipendentemente dai corsi (ad es. in un centro d'esame). Un elenco degli Organismi di Certificazione autorizzati da IREB è disponibile sul sito web <https://www.ireb.org>.

## Storia delle Versioni

Versione	Data	Commenti
2.2.0	Agosto 2017	Versione iniziale
2.2.1	25 Settembre 2017	<p>Completata la revisione della traduzione italiana</p> <p>Alcuni termini precedentemente tradotti in italiano sono stati riportati nella propria versione originale inglese in quanto maggiormente utilizzati in questa forma.</p> <p>Migliorata la leggibilità complessiva della versione italiana</p> <ul style="list-style-type: none"><li>▪ Apportate minori modifiche editoriali</li></ul>
2.2.2	2 Novembre 2017	EU 5.2: Verbi modali corrispondenti alle forme grammaticali italiane.
3.0	1 Aprile 2021	Importante aggiornamento per riportare lo stato attuale del RE, che copre sia gli approcci plan-driven, sia gli approcci agili per specificare e gestire i requisiti.
3.1.0	1 Ottobre 2022	<p>Tipi e riferimenti corretti in tutto il documento per migliorare la leggibilità.</p> <p>EU 1:</p> <p>Obiettivo di apprendimento da EO 1.3.2 a EO 1.1.2.</p> <p>EO 1.2.2 e 1.3.1 aggiornati</p> <p>EU 3</p> <p>Obiettivo formativo EO 3.1.3 aggiornato</p> <p>EU 3.1.2: Rielavorato il paragrafo sui livelli di astrazione</p> <p>EU 3.4: Tutti i tipi di modello che non richiedono di essere applicati nel Foundation Level sono stati spostati in un nuovo paragrafo 3.4.6</p> <p>EU 3.6: Aggiornata il titolo in "Struttura della Documentazione e dei Documenti dei Requisiti"</p> <p>EU 4</p> <p>EU 4.1: La descrizione degli stakeholder e dei ruoli degli stakeholder è stata resa più precisa.</p>

EU 4.2: L'introduzione e la motivazione per le tecniche di progettazione e generazione di idee sono state aggiornate per essere più precise.

EU 4.3: Aggiornato il titolo in "Risolvere i Conflitti relativi ai Requisiti"

EU 6

EO 6.3.1 e EO 6.5.2 sono stati leggermente modificati

3.1.1	1 Gennaio 2024	nuovo corporate design
3.2.0	26 Febbraio 2024	Correzioni del layout per il nuovo corporate design

# Contenuto

Contenuto .....	8
<b>1 Introduzione e Panoramica del Requirements Engineering (L2)</b> .....	<b>11</b>
1.1 Requirements Engineering: Cosa (L1) .....	11
1.2 Requirements Engineering: Perché (L2) .....	12
1.3 Requirements Engineering: Dove (L2) .....	12
1.4 Requirements Engineering: Come (L1) .....	12
1.5 Il Ruolo e i Compiti di un Requirements Engineer (L1) .....	12
1.6 Cosa Imparare sul Requirements Engineering (L1) .....	13
<b>2 Principi Fondamentali del Requirements Engineering (L2)</b> .	<b>14</b>
2.1 Panoramica dei Principi (L1) .....	14
2.2 Spiegazione dei Principi (L2) .....	14
<b>3 Prodotti di Lavoro e Best Practice per la Documentazione (L3)</b> .....	<b>19</b>
3.1 Prodotti di lavoro nel Requirements Engineering (L2) .....	20
3.1.1 Caratteristiche dei Prodotti di Lavoro (L1) .....	20
3.1.2 Livelli di astrazione (L2) .....	21
3.1.3 Livello di Dettaglio (L2) .....	21
3.1.4 Aspetti da Considerare nei Prodotti di Lavoro (L1) .....	22
3.1.5 Linee Guida Generali per la Documentazione (L1) .....	22
3.1.6 Pianificare i Prodotti di Lavoro da Utilizzare (L1) .....	22
3.2 Prodotti di Lavoro Basati sul Linguaggio Naturale (L2) .....	23
3.3 Prodotti di Lavoro Basati sui Template (L3) .....	24
3.4 Prodotti di Lavoro Model-Based (Basati sui Modelli) (L3) .....	24
3.4.1 Il Ruolo dei Modelli nel Requirements Engineering (L2) .....	25



3.4.2	Modellare il Contesto (L2) .....	26
3.4.3	Modellare Struttura e Dati (L3) .....	26
3.4.4	Modellare Funzione e Flusso (L3) .....	27
3.4.5	Modellare Stato e Comportamento (L2) .....	27
3.4.6	Altri Tipi di Modelli nel Requirements Engineering (L1) .....	27
3.5	Glossari (L2) .....	28
3.6	Strutture della Documentazione e dei Documenti dei Requisiti (L2) .	28
3.7	Prototipi nel Requirements Engineering (L1) .....	29
3.8	Criteri di Qualità per i Prodotti di Lavoro e per i Requisiti (L1)	30
<b>4</b>	<b>Pratiche per l'Elaborazione dei Requisiti (L3) .....</b>	<b>32</b>
4.1	Sorgenti dei Requisiti (L3) .....	32
4.2	Elicitazione dei Requisiti (L2) .....	34
4.3	Risolvere i Conflitti Relativi ai Requisiti (L2) .....	35
4.4	Validazione dei Requisiti (L2) .....	36
<b>5</b>	<b>Struttura del Processo e delle Attività (L3) .....</b>	<b>38</b>
5.1	Fattori di Influenza (L2) .....	38
5.2	Aspetti (Facet) del Processo di Requirements Engineering (L2) .....	39
5.3	Configurare un Processo di Requirements Engineering (L3) .....	41
<b>6</b>	<b>Pratiche per il Requirements Management (L2) .....</b>	<b>43</b>
6.1	Che cosa è il Requirements Management? (L1) .....	43
6.2	Gestione del Ciclo di Vita (L2) .....	43
6.3	Controllo della Versione (L2) .....	44
6.4	Configurazioni e Baseline (L1) .....	44
6.5	Attributi e Viste (L2) .....	44
6.6	Tracciabilità (L1) .....	45
6.7	Change Management (Gestione delle Modifiche) (L1) .....	45

6.8	Prioritizzazione (L1).....	46
7	Supporto degli Strumenti (L2).....	47
7.1	Strumenti nel Requirements Engineering (L1).....	47
7.2	Introduzione degli Strumenti (L2).....	48
	Riferimenti Bibliografici.....	49

# 1 Introduzione e Panoramica del Requirements Engineering (L2)

Obiettivo: Conoscere cosa è RE e comprendere il valore del RE

Durata: 1 ora

Termini: Requisito, specifica dei requisiti, Requirements Engineering (Ingegneria dei requisiti, RE), stakeholder, sistema, Requirements Engineer (Ingegnere dei requisiti)

## Obiettivi Formativi

- EO 1.1.1 Richiamare la terminologia principale (L1)
- EO 1.1.2 Comprendere i diversi tipi di requisiti (L2)
- EO 1.2.1 Spiegare il valore di RE (L2)
- EO 1.2.2 Enumerare i sintomi di un RE inadeguata (L1)
- EO 1.3.1 Sapere dove si può applicare RE e dove si verificano i requisiti (L1)
- EO 1.4.1 Conoscere le attività principali di RE e che un processo di RE deve essere adattato per eseguire tali attività (L1)
- EO 1.5.1 Caratterizzare il ruolo e le attività di un Requirements Engineer (L1)
- EO 1.6.1 Ricordare cosa necessita di apprendere un Requirements Engineer (L1)

## 1.1 Requirements Engineering: Cosa (L1)

Le persone e le organizzazioni hanno desideri e necessità di costruire nuove "cose" o di far evolvere "cose" esistenti. Tali necessità vengono specificate come *requisiti*.

Le "cose" da costruire o da far evolvere possono essere:

- *Prodotti* forniti ai clienti
- *Servizi* messi a disposizione dei clienti
- Qualsiasi altro *rilascio*, come dispositivi, procedure o strumenti che aiutano le persone e le organizzazioni a raggiungere un obiettivo specifico
- *Composizioni* o *componenti* di prodotti, servizi o altri rilasci.

Tutte queste "cose" possono essere considerate *sistemi*. In questo Syllabus, viene usato il termine *sistema* per indicare tutti i tipi di "cose" per le quali gli *stakeholder* hanno dei requisiti. Gli *stakeholder* sono persone o organizzazioni che influenzano i requisiti di un sistema o che sono 'impattati' da quel sistema.

L'obiettivo del RE è di specificare e gestire i requisiti dei sistemi in modo che i sistemi implementati e distribuiti soddisfino i desideri e le necessità dei loro stakeholder.

In RE, si distinguono tre tipi di requisiti [Glin2020]:

- *I requisiti funzionali* che riguardano un risultato o un comportamento che dovrà essere fornito da una funzione del sistema. Includono i requisiti per i dati o per l'interazione del sistema con il suo ambiente.
- *I requisiti di qualità* che si riferiscono ad argomenti di qualità che non sono coperti dai requisiti funzionali, come le prestazioni, la disponibilità, la sicurezza o l'affidabilità.

- I *vincoli*, requisiti che limitano lo spazio della soluzione oltre a quanto è necessario per soddisfare i requisiti funzionali e i requisiti di qualità identificati.

## 1.2 Requirements Engineering: Perché (L2)

Un adeguato RE aggiunge *valore* nel processo di sviluppo ed evoluzione di un sistema:

- Riduce il rischio di sviluppare il sistema sbagliato
- Migliora la comprensione del problema
- Costituisce una base per la stima dell'effort e dei costi dello sviluppo
- Costituisce un prerequisito per il testing del sistema

Tipici sintomi di un RE inadeguato sono requisiti mancanti, non chiari o non corretti. Questo è dovuto in particolare a:

- Pressioni sulle tempistiche per costruire il sistema
- Problemi di comunicazione tra le parti coinvolte
- Assunzioni che i requisiti stessi siano auto-esplicativi, evidenti e ovvi
- Inadeguata formazione e competenza su RE.

## 1.3 Requirements Engineering: Dove (L2)

RE può essere applicato ai requisiti di qualsiasi tipo di sistema. Tuttavia, attualmente il caso applicativo principale per RE è rappresentato da sistemi in cui il software gioca un ruolo importante. Tali sistemi sono tipicamente costituiti da componenti software, elementi fisici ed elementi organizzativi.

I requisiti possono presentarsi come:

- *Requisiti di sistema* – cosa dovrà fare un sistema
- *Requisiti degli stakeholder* – cosa vogliono gli stakeholder dalla loro prospettiva
- *Requisiti degli utenti* – cosa vogliono gli utenti secondo la loro prospettiva
- *Requisiti del dominio* – proprietà di dominio richieste
- *Requisiti di business* – obiettivi di business ed esigenze di un'organizzazione

## 1.4 Requirements Engineering: Come (L1)

Le attività principali nell'ambito di RE sono l'elicitazione (o identificazione) (4.2), la documentazione (3), la validazione (4.4) e la gestione (o requirements management) (6) dei requisiti. Il supporto degli strumenti (7) può aiutare a svolgere queste attività. L'analisi dei requisiti e la risoluzione dei conflitti di requisiti (4.3) sono considerati parte dell'elicitazione. Per poter svolgere le attività di RE in modo appropriato, un processo di RE adeguato deve essere adattato (tailoring) rispetto ad un'ampia gamma di possibilità (5).

## 1.5 Il Ruolo e i Compiti di un Requirements Engineer (L1)

Requirements Engineer in genere non è un job title, ma un *ruolo* che svolgono le persone che:

- Elicitano, documentano, validano e/o gestiscono i requisiti come parte dei propri compiti
- Hanno una conoscenza approfondita di RE
- Possono colmare il divario tra il problema e le potenziali soluzioni.

In pratica, i Business Analyst, gli specialisti applicativi, i Product Owner, i System Engineer e anche gli sviluppatori agiscono nel ruolo di Requirements Engineer.

## 1.6 Cosa Imparare sul Requirements Engineering (L1)

Questo Syllabus copre l'insieme di competenze principali che un Requirements Engineer deve apprendere. Comprende i principi fondamentali di RE (2), come documentare i requisiti in diverse forme (3), come elaborare i requisiti con differenti pratiche (4), come definire e lavorare con adeguati processi di RE (5), come gestire i requisiti esistenti (6), e come utilizzare strumenti di supporto (7).

# 2 Principi Fondamentali del Requirements Engineering (L2)

Obiettivo: Conoscere e comprendere i principi del RE

Durata: 1 ora e 30 minuti

Termini: Contesto, requisito, Requirements Engineering (Ingegneria dei Requisiti, RE), stakeholder, comprensione condivisa, validazione

## Obiettivi Formativi

EO 2.1.1 Enumerare i principi del RE (L1)

EO 2.2.1 Ricordare i termini associati ai principi (L1)

EO 2.2.2 Spiegare i principi e le ragioni del perchè sono importanti (L2)

## 2.1 Panoramica dei Principi (L1)

RE è governato da una serie di principi fondamentali che si applicano a tutti i compiti, le attività e le pratiche di RE. I seguenti nove principi costituiscono la base delle pratiche presentate nei capitoli successivi di questo Syllabus.

1. Orientamento al valore: I requisiti sono un mezzo per un fine, non sono essi stessi un fine
2. Stakeholder: RE consiste nel soddisfare i desideri e le necessità degli stakeholder
3. Comprensione condivisa: Lo sviluppo di sistemi di successo è impossibile senza una base comune
4. Contesto: I sistemi non possono essere compresi in modo isolato
5. Problema – Requisito – Soluzione: Una tripletta inevitabilmente intrecciata
6. Validazione: I requisiti non validati sono inutili
7. Evoluzione: Le modifiche ai requisiti non sono un caso, ma è la normalità
8. Innovazione: Più o meno lo stesso non è sufficiente
9. Lavoro sistematico e disciplinato: Non è possibile farne a meno in RE.

## 2.2 Spiegazione dei Principi (L2)

**Principio 1 - Orientamento al valore: I requisiti sono un mezzo per un fine, non sono essi stessi un fine**

Il valore di un requisito è uguale al suo beneficio meno il costo per l'elicitazione, la documentazione, la validazione e la gestione del requisito. Il beneficio di un requisito è il grado a cui esso contribuisce per:

- Costruire sistemi che soddisfino i desideri e le esigenze dei propri stakeholder.
- Ridurre il rischio di failure e di costosi rework durante lo sviluppo del sistema.

## Principio 2 - Stakeholder: RE consiste nel soddisfare i desideri e le necessità degli stakeholder

Poiché RE consiste nel comprendere i desideri e le esigenze degli stakeholder, collaborare in modo appropriato con gli stakeholder è un compito fondamentale di RE. Ogni stakeholder ha un ruolo nel contesto del sistema da costruire – per esempio, utente, cliente, operatore o controllore delle normative. Uno stakeholder può avere più di un ruolo contemporaneamente. Per i ruoli degli stakeholder con troppi individui o quando gli individui non sono conosciuti, possono essere definite, in sostituzione, delle descrizioni di archetipi fittizi, conosciute come *Persona*. Non è sufficiente considerare solo i requisiti degli utenti finali o dei clienti. Questo significherebbe che si potrebbero dimenticare requisiti critici da parte di altri stakeholder. Anche gli utenti che forniscono feedback su un sistema in uso dovrebbero essere considerati come stakeholder.

Gli stakeholder possono avere esigenze e punti di vista differenti, e questo può causare conflitto tra requisiti. È compito di RE identificare e risolvere tali conflitti.

Coinvolgere le persone giuste nei ruoli degli stakeholder importanti è cruciale per un RE di successo. Le pratiche per l'identificazione, la prioritizzazione e la collaborazione con gli stakeholder sono descritte in 4.

## Principio 3 - Comprensione condivisa: Lo sviluppo di sistemi di successo è impossibile senza una base comune

RE crea, promuove e assicura una comprensione condivisa tra le parti coinvolte: stakeholder, Requirements Engineer e sviluppatori. Esistono due forme di comprensione condivisa:

- *Comprensione condivisa esplicita* (raggiunta attraverso requisiti documentati e concordati)
- *Comprensione condivisa implicita* (basata sulla conoscenza condivisa delle necessità, della vision, del contesto, ecc.)

Una conoscenza del dominio, una precedente collaborazione di successo, una cultura e dei valori condivisi, e una fiducia reciproca sono fattori che favoriscono una comprensione condivisa; mentre la distanza geografica, l'outsourcing o grandi team con un elevato turnover sono degli ostacoli alla condivisione.

Le buone pratiche per raggiungere una comprensione condivisa includono la creazione di glossari (3.5), la creazione di prototipi (3.7), o l'utilizzo di un sistema esistente come punto di riferimento. Le pratiche per la valutazione della comprensione condivisa includono esempi di risultati attesi, l'esplorazione di prototipi o la stima del costo di implementazione di un requisito. La pratica più importante per ridurre l'impatto delle incomprensioni è l'utilizzo di un processo con brevi cicli di feedback (5).

#### Principio 4 - Contesto: I sistemi non possono essere compresi in modo isolato

I sistemi sono incorporati in un *contesto*. Senza la comprensione di tale contesto, è impossibile specificare un sistema in modo appropriato. In RE, il contesto di un sistema è la parte dell'ambiente del sistema rilevante per la comprensione del sistema e dei suoi requisiti. Il *confine del sistema* è il confine tra un sistema e il suo contesto circostante. Spesso il confine del sistema stesso non è chiaro all'inizio di RE, e può anche cambiare nel tempo.

Chiarire il confine del sistema e definire le interfacce esterne tra il sistema e gli elementi del contesto con cui interagisce sono compiti specifici di RE. Allo stesso tempo, deve essere determinato l'*ambito* del sistema – cioè l'insieme di elementi che possono essere modellati e progettati durante lo sviluppo del sistema. Deve essere considerato anche il cosiddetto *confine del contesto* che separa la parte dell'ambiente di un sistema che è rilevante per RE dal resto del mondo.

Quando si specifica un sistema, non è sufficiente considerare solo i requisiti all'interno dei confini del sistema. RE deve considerare anche i seguenti elementi:

- Modifiche al contesto che possono avere un impatto sui requisiti del sistema.
- Requisiti del mondo reale rilevanti per il sistema (e come mapparli in requisiti del sistema).
- Assunzioni del contesto che devono essere mantenute per far funzionare il sistema e soddisfare i requisiti del mondo reale.

#### Principio 5 - Problema - Requisito - Soluzione: Una terna inevitabilmente intrecciata

Un *problema* si verifica quando gli stakeholder non sono soddisfatti della situazione attuale (*as is*). I *requisiti* catturano ciò di cui gli stakeholder hanno necessità per eliminare il problema o per semplificarlo. Un sistema socio-tecnico che soddisfi questi requisiti costituisce una *soluzione*.

Problemi, requisiti e soluzioni non si verificano necessariamente in questo ordine. Le idee di soluzione possono creare esigenze dell'utente che devono essere elaborate come requisiti e implementate in una soluzione reale/effettiva. Questo è il tipico caso dell'innovazione.

- Problemi, requisiti e soluzioni sono strettamente intrecciati e correlati: non possono essere affrontati separatamente.
- Tuttavia, quando i Requirements Engineer pensano, comunicano e documentano, il loro obiettivo è di separare per quanto possibile problemi, requisiti e soluzioni. Questa separazione dei concetti rende le attività di RE più facili da gestire.



## Principio 6 - Validazione: I requisiti non validati sono inutili

Infine, è necessario validare che il sistema implementato soddisfi i desideri e le esigenze degli stakeholder. Per poter controllare fin dall'inizio il rischio di stakeholder insoddisfatti, la validazione dei requisiti deve iniziare già durante RE. Si deve verificare se:

- L'accordo sui requisiti è stato raggiunto tra gli stakeholder
- I desideri e le esigenze degli stakeholder sono adeguatamente coperti dai requisiti
- Le assunzioni sul contesto (si veda Principio 4 sopra) sono valide.

Le pratiche per la validazione dei requisiti sono descritte in 4.4.

## Principio 7 - Evoluzione: Le modifiche ai requisiti non sono un caso, ma è la normalità

I sistemi e i loro requisiti sono soggetti all'*evoluzione*. Questo significa che cambiano costantemente. Le richieste di modifica di un requisito o di un insieme di requisiti di un sistema possono essere causate, ad esempio, da:

- Processi di business modificati
- Aziende concorrenti (competitor) che lanciano nuovi prodotti o servizi
- Clienti che modificano le loro priorità o opinioni
- Cambiamenti nella tecnologia
- Modifiche di leggi o normative
- Feedback da parte di utenti del sistema che desiderano funzionalità nuove o modificate.

Inoltre, i requisiti possono essere modificati a causa di feedback degli stakeholder in fase di validazione dei requisiti, a causa della rilevazione di difetti in requisiti precedentemente elicitati, o a causa di esigenze cambiate degli stakeholder.

Di conseguenza, i Requirements Engineer devono perseguire due obiettivi apparentemente contraddittori:

- Permettere la modifica dei requisiti
- Mantenere stabili i requisiti

I dettagli su come riuscirci sono discussi in 6.7.

## Principio 8 - Innovazione: Più o meno lo stesso non è sufficiente

Fornire agli stakeholder esattamente quello che vogliono fa perdere l'opportunità di costruire sistemi che soddisfino le esigenze degli stakeholder meglio di quanto si aspettino.

Un buon RE si sforza non solo di soddisfare gli stakeholder, ma di renderli felici, eccitati o di farli sentire al sicuro. Questo è in definitiva l'innovazione.

RE modella i sistemi innovativi:

- Su piccola scala, cercando di ottenere nuove ed entusiasmanti caratteristiche e facilità d'uso.
- Su larga scala, cercando nuove idee di impatto.

In 4.2 vengono discusse diverse tecniche per promuovere l'innovazione in RE.

**Principio 9 - Lavoro sistematico e disciplinato: Non è possibile farne a meno in RE**

Esiste la necessità di applicare processi e pratiche adeguate per elicitarne, documentare, validare e gestire i requisiti in modo sistematico, indipendentemente dal processo di sviluppo attuale utilizzato. Anche quando un sistema viene sviluppato ad-hoc, un approccio sistematico e disciplinato di RE migliora la qualità del sistema finale.

Non esiste un singolo processo o pratica in RE che funzioni bene in qualsiasi situazione o almeno nella maggior parte delle situazioni. Lavoro sistematico e disciplinato significa che i Requirements Engineer:

- Adattano i loro processi e le loro pratiche al problema, al contesto e all'ambiente considerato
- Non utilizzano sempre lo stesso processo e lo stesso insieme di pratiche
- Non riutilizzano i processi e le pratiche di attività di RE svolte con successo nel passato, senza prima aver riflettuto.

Per ogni tentativo di RE, i processi, le pratiche e i prodotti di lavoro devono essere selezionati in modo da adattarsi al meglio alla situazione specifica. I dettagli sono elaborati in 3, 4, 5 ed 6.

## 3 Prodotti di Lavoro e Best Practice per la Documentazione (L3)

**Obiettivo:** Comprendere il ruolo fondamentale dei prodotti di lavoro in RE ed essere in grado di creare prodotti di lavoro

**Durata:** 7 ore

**Termini:** Prodotto di lavoro, prodotti di lavoro basati sul linguaggio naturale, prodotti di lavoro basati sui template, prodotti di lavoro model-based (basati sui modelli), glossario, criteri di qualità, specifica dei requisiti

### Obiettivi Formativi

- EO 3.1.1 Conoscere le caratteristiche dei prodotti di lavoro di RE ed enumerare i tipi di prodotti di lavoro utilizzati di frequente (L1)
- EO 3.1.2 Conoscere per cosa può essere utilizzato ogni prodotto di lavoro e conoscere la durata dei prodotti di lavoro (L1)
- EO 3.1.3 Spiegare i diversi livelli di astrazione dei requisiti, includendo come scegliere i livelli di astrazione e i livelli di dettaglio appropriati (L2)
- EO 3.1.4 Conoscere gli aspetti da considerare nei prodotti di lavoro e le interrelazioni tra questi aspetti (L1)
- EO 3.1.5 Nominare le linee guida generali per la documentazione (L1)
- EO 3.1.6 Descrivere perchè vale la pena pianificare i prodotti di lavoro da utilizzare (L1)
- EO 3.2.1 Conoscere i prodotti di lavoro basati sul linguaggio naturale, i loro vantaggi e svantaggi (L1)
- EO 3.2.2 Spiegare le regole per una buona scrittura dei requisiti basati sul linguaggio naturale (L2)
- EO 3.3.1 Conoscere le categorie di prodotti di lavoro basati sui template, i loro vantaggi e svantaggi (L1)
- EO 3.3.2 Specificare un singolo requisito e una user story utilizzando un template di frase (L3)
- EO 3.3.3 Specificare uno use case utilizzando un template di form (modulo) (L3)
- EO 3.4.1 Comprendere il ruolo, lo scopo e l'uso dei modelli in RE (L2)
- EO 3.4.2 Comprendere i vantaggi e i limiti della modellazione in RE (L2)
- EO 3.4.3 Conoscere i termini: modello, linguaggio di modellazione, modello delle attività (activity model), activity diagram, modello delle classi (class model), class diagram, modello del contesto (context model), context diagram, modello del dominio, modello degli obiettivi, modello di interazione, modello di processo, sequence diagram, statechart, state machine, state machine diagram, use case, use case diagram (L1)
- EO 3.4.4 Comprendere come selezionare un tipo di modello appropriato per specificare i requisiti in una data situazione (L2)
- EO 3.4.5 Comprendere e interpretare modelli semplici, scritti in UML dove applicabile, dei seguenti tipi: context model, use case e use case diagram, modello del dominio, class model, activity model, modello di processo e statechart (L2)
- EO 3.4.6 Specificare un semplice modello dei dati di un sistema o degli oggetti in un dominio utilizzando un class diagram UML (L3)
- EO 3.4.7 Specificare una semplice funzione di sistema o un processo di business attraverso un activity diagram UML (L3)
- EO 3.5.1 Spiegare lo scopo dei glossari e come crearne uno (L2)
- EO 3.6.1 Conoscere i documenti di specifica dei requisiti che vengono utilizzati di frequente (L1)

- EO 3.6.2 Spiegare quali strutture di documenti servono per uno specifico scopo e quali sono i criteri per strutturarli (L2)
- EO 3.7.1 Conoscere differenti tipi di prototipi e per cosa vengono utilizzati (L1)
- EO 3.8.1 Conoscere i criteri di qualità per i singoli requisiti (L1)
- EO 3.8.2 Conoscere i criteri di qualità per i prodotti di lavoro (L1)

## 3.1 Prodotti di lavoro nel Requirements Engineering (L2)

Un prodotto di lavoro è un risultato registrato, intermedio o finale generato in un processo di lavoro. Esistono una varietà di prodotti di lavoro in RE, che possono essere, ad esempio, semplici schizzi grafici di breve durata generati dall'evoluzione di un insieme di user story, oppure documenti di specifica dei requisiti contrattuali formalmente rilasciati e costituiti da centinaia di pagine.

### 3.1.1 Caratteristiche dei Prodotti di Lavoro (L1)

I prodotti di lavoro sono caratterizzati dal loro scopo, dalla loro rappresentazione, dimensione e durata. I prodotti di lavoro che vengono in pratica utilizzati di frequente per gli scopi indicati sono descritti di seguito. Si noti che un prodotto di lavoro può contenere altri prodotti di lavoro.

- I prodotti di lavoro per un singolo requisito includono i singoli requisiti e le user story.
- I prodotti di lavoro per un insieme consistente di requisiti includono use case, modelli grafici di un tipo definito (3.4), descrizioni dei compiti, descrizioni delle interfacce esterne ed epic.
- I prodotti di lavoro che costituiscono documenti o strutture di documentazione completi includono le specifiche dei requisiti di sistema, i product backlog e sprint backlog, e le story map.
- Altri prodotti di lavoro includono glossari, note testuali, schizzi grafici e prototipi.

I prodotti di lavoro possono essere *rappresentati* in diversi formati:

- Basati sul linguaggio naturale (3.2)
- Basati sui template (3.3)
- Model-based (basati sui modelli) (3.4)
- Altre rappresentazioni, come disegni o prototipi (3.7)

La maggior parte dei prodotti di lavoro possono essere *memorizzati* in formato elettronico, sotto forma di file, in data base o in strumenti di RE. I prodotti di lavoro informali e temporanei possono essere memorizzati anche su altri supporti – ad esempio, carta o post-it su una Kanban board.

Quando si guarda alla durata dei prodotti di lavoro, si distinguono tre categorie:

- *Prodotti di lavoro temporanei*: creati per supportare la comunicazione e per creare una comprensione condivisa.
- *Prodotti di lavoro in evoluzione*: normalmente si sviluppano nel tempo attraverso diverse iterazioni; necessitano di alcuni metadati (6.5); può essere applicato il controllo delle modifiche.

- *Prodotti di lavoro duraturi*: sono parte di una *baseline* o sono stati *rilasciati*; necessitano di un insieme completo di metadati; (6.5); deve essere applicato il processo di change management (6.3, 6.4).

Un prodotto di lavoro temporaneo può essere convertito in un prodotto di lavoro in evoluzione (memorizzandolo e aggiungendo metadati). Analogamente, un prodotto di lavoro temporaneo o in evoluzione può diventare un prodotto di lavoro duraturo se viene creata una sua baseline o se viene rilasciato.

### 3.1.2 Livelli di astrazione (L2)

I requisiti esistono di solito a diversi *livelli diversi di astrazione* – ad esempio, si va dai requisiti di alto livello per un nuovo processo di business fino ai requisiti di un livello molto dettagliato, come la reazione di un componente software specifico ad un evento di eccezione.

La scelta del livello di astrazione appropriato dipende dall'oggetto da specificare e dallo scopo della specifica. È importante, tuttavia, non mescolare requisiti che sono su differenti livelli di astrazione. Nei prodotti di lavoro di piccole e medie dimensioni, i requisiti dovrebbero essere più o meno allo stesso livello di astrazione.

Nei prodotti di grandi dimensioni, come una specifica dei requisiti di sistema, i requisiti su diversi livelli di astrazione dovrebbero essere tenuti separati, strutturando le specifiche di conseguenza (3.6). Un requisito a un alto livello di astrazione può essere raffinato in diversi requisiti dettagliati a livelli più bassi e più concreti.

Quando i requisiti di alto livello di business e i requisiti degli stakeholder vengono specificati nei prodotti di lavoro duraturi – come le specifiche dei requisiti di business, le specifiche dei requisiti degli stakeholder o i documenti di vision – questi anticipano la specifica dei requisiti di sistema. In altre situazioni, i requisiti di business, i requisiti degli stakeholder e i requisiti di sistema possono evolvere in parallelo.

### 3.1.3 Livello di Dettaglio (L2)

Il livello di *dettaglio* a cui dovrebbero essere specificati i requisiti dipende da diversi fattori, in particolare:

- Il problema e il contesto di sviluppo
- Il grado di comprensione condivisa del problema
- Il grado di libertà lasciato ai progettisti e ai programmatori
- La disponibilità di rapidi feedback degli stakeholder durante la progettazione e l'implementazione
- Il costo vs. valore di una specifica dettagliata
- Gli standard e vincoli normativi da rispettare.

Maggiore è il livello di dettaglio dei requisiti specificati, minore è il rischio di ottenere eventualmente qualcosa di inaspettato o non specificato. Tuttavia, il costo della specifica aumenta con l'aumentare del livello di dettaglio.

### 3.1.4 Aspetti da Considerare nei Prodotti di Lavoro (L1)

Quando si specificano i requisiti nei prodotti di lavoro, è necessario considerare diversi aspetti.

1. I requisiti sono classificati in base alla loro natura (1.1) in:
  - a) Requisiti funzionali
  - b) Requisiti di qualità
  - c) Vincoli
2. I requisiti funzionali si focalizzano su diversi aspetti della funzionalità di un sistema:
  - a) Struttura e dati
  - b) Funzione e flusso
  - c) Stato e comportamento.
3. Infine, i requisiti possono essere compresi solo nel contesto (Principio 4 in 2):
  - a) Il *contesto del sistema*, compresi gli attori esterni
  - b) Confine di sistema e *interfacce esterne*

Esistono molte interrelazioni e dipendenze tra gli aspetti menzionati. Ad esempio, la richiesta di un utente (contesto) può attivare una transizione di stato (stato e comportamento), che inizia un'azione seguita da un'altra azione (funzione e flusso), che richiede dati (struttura e dati) per fornire un risultato all'utente (contesto) entro un dato intervallo di tempo (qualità).

Alcuni prodotti di lavoro si focalizzano su un aspetto specifico, astraendo dagli altri aspetti. Questo vale in particolare per i modelli dei requisiti (3.4). Altri prodotti di lavoro, come le specifiche dei requisiti di sistema, coprono tutti questi aspetti. Quando aspetti differenti sono documentati in prodotti di lavoro separati o in capitoli separati dello stesso prodotto di lavoro, questi prodotti di lavoro o capitoli devono essere mantenuti consistenti tra loro.

### 3.1.5 Linee Guida Generali per la Documentazione (L1)

Indipendentemente dalle tecniche utilizzate, per la creazione di prodotti di lavoro si applicano le seguenti linee guida:

- Selezionare un tipo di prodotto di lavoro che si adatti allo *scopo previsto*.
- *Evitare ridondanze* facendo riferimento al contenuto invece di ripetere lo stesso contenuto.
- *Assicurarsi che non esistano inconsistenze* tra prodotti di lavoro, in particolare quando coprono aspetti differenti.
- *Utilizzare i termini in modo consistente*, come definiti nel glossario.
- *Strutturare i prodotti di lavoro in modo appropriato*.

### 3.1.6 Pianificare i Prodotti di Lavoro da Utilizzare (L1)

Ogni progetto e ogni dominio è differente, quindi l'insieme dei prodotti di lavoro risultanti deve essere definito per ognuno di essi. Pertanto, devono essere concordate le risposte ai seguenti punti:

- In quali prodotti di lavoro dovranno essere memorizzati i requisiti e per quale scopo?

- Quali livelli di astrazione devono essere considerati?
- Fino a quale livello di dettaglio devono essere documentati i requisiti per ogni livello di astrazione?
- Come dovranno essere rappresentati i requisiti in questi prodotti di lavoro?

I prodotti di lavoro da utilizzare dovrebbero essere definiti nelle fasi iniziali di un progetto.

Questo ha diversi vantaggi:

- Aiuta a pianificare l'effort e le risorse.
- Assicura l'utilizzo di notazioni appropriate.
- Assicura che tutti i risultati siano memorizzati nei giusti prodotti di lavoro.
- Assicura che non sia necessario un grande rimescolamento delle informazioni e un "editing finale".
- Aiuta ad evitare ridondanze, che risulta in minor lavoro e una manutenibilità più semplice.

## 3.2 Prodotti di Lavoro Basati sul Linguaggio Naturale (L2)

I requisiti in linguaggio naturale sono sempre stati un mezzo fondamentale per specificare i requisiti.

I prodotti di lavoro basati sul linguaggio naturale presentano grandi vantaggi:

- Il linguaggio naturale non formale è estremamente espressivo e flessibile.
- Quasi tutti i requisiti in qualsiasi aspetto possono essere espressi in linguaggio naturale.
- Il linguaggio naturale viene utilizzato nella vita quotidiana e insegnato a scuola, quindi non è necessaria una formazione specifica per leggere e comprendere i testi in linguaggio naturale.

Questi vantaggi vanno a scapito del fatto che i testi scritti in linguaggio naturale possono essere frequentemente interpretati in modi differenti, e questo costituisce un problema quando si specificano i requisiti. Inoltre, l'identificazione di ambiguità, omissioni e inconsistenze in tali testi è difficile e costosa.

Una buona descrizione dei requisiti basata sul linguaggio naturale può essere ottenuta seguendo le seguenti best practice:

- Scrivere frasi brevi e ben strutturate.
- Definire e utilizzare in modo consistente una terminologia uniforme (3.5).
- Evitare frasi e termini che siano vaghi o ambigui.
- Conoscere le insidie della scrittura tecnica, elencate di seguito.

Quando si scrivono documenti tecnici in linguaggio naturale, esistono alcune note insidie che dovrebbero essere evitate o utilizzate con cura [GoRu2003].

Da evitare:

- Descrizioni incomplete
- Sostantivi non specifici

- Condizioni incomplete
- Confronti incompleti.

Da usare con cura:

- Voci passive
- Quantificatori universali (come "tutti" o "mai")
- Nominalizzazioni (cioè sostantivi derivati da un verbo, ad esempio "autenticazione").

### 3.3 Prodotti di Lavoro Basati sui Template (L3)

I prodotti di lavoro basati sui template vengono utilizzati per superare alcune carenze dei prodotti di lavoro basati sul linguaggio naturale, fornendo strutture predefinite per i requisiti.

- I *template di frase* forniscono una struttura sintattica predefinita per una frase che esprime un requisito, in particolare un singolo requisito o una user story.
- I *template di form* (modulo) forniscono un insieme di campi predefiniti in un modulo da compilare, ad esempio, per scrivere uno use case o un requisito di qualità misurabile.
- I *template di documento* forniscono una struttura predefinita per un documento dei requisiti.

In letteratura sono stati descritti vari template. [ISO29148], [MWHN2009], e [Rupp2014] forniscono template di frase per i singoli requisiti. [Cohn2004] definisce un template di frase ampiamente utilizzato per le user story e [Cock2001] descrive i template di form per gli use case. [Laue2002] ha proposto un template per la descrizione delle attività. [ISO29148] e [RoRo2012] forniscono template di documento per specifiche complete. Inoltre, un cliente potrebbe prescrivere per un progetto l'uso di template specifici del cliente stesso.

*Vantaggi* dei prodotti di lavoro basati sui template:

- Fornire una struttura chiara e riutilizzabile
- Aiutare a catturare le informazioni più importanti
- Rendere uniforme il contenuto dei requisiti e delle specifiche dei requisiti
- Migliorare la qualità complessiva dei requisiti e delle specifiche dei requisiti

*Svantaggi* e insidie dei prodotti di lavoro basati sui template:

- Spesso le persone si focalizzano sulla completezza formale del template piuttosto che sul contenuto.
- Gli aspetti che non sono inclusi nel template hanno maggiore probabilità di essere omessi.

### 3.4 Prodotti di Lavoro Model-Based (Basati sui Modelli) (L3)

I requisiti rappresentati in linguaggio naturale hanno dei limiti [Davi1993], in particolare ad ottenere una panoramica di un insieme di requisiti e a comprendere le relazioni tra i requisiti. La modellazione dei requisiti indirizza queste limitazioni.



### 3.4.1 Il Ruolo dei Modelli nel Requirements Engineering (L2)

Un *modello* è una rappresentazione astratta di una parte di realtà esistente o di una parte di realtà da creare. La nozione di realtà comprende qualsiasi insieme concepibile di elementi, fenomeni o concetti, inclusi altri modelli. Rispetto a un modello, la parte modellata della realtà è chiamata *originale*. Esempi di modelli al di fuori dell'ambito dell'ingegneria del software sono i building information model (BIM) [ISO19650], che modellano gli elementi necessari per pianificare, costruire e gestire edifici e altri elementi di costruzione.

In RE, i modelli aiutano a comprendere le relazioni e le interconnessioni tra i requisiti, e forniscono una panoramica di un insieme di requisiti. Questo risultato si ottiene principalmente focalizzandosi su alcuni aspetti – per esempio, il comportamento – e astruendo da tutti gli altri aspetti. E' possibile ottenere una visione d'insieme anche grazie all'utilizzo di una notazione grafica per un modello. Tuttavia, i modelli possono essere rappresentati anche non in modo grafico, ad esempio utilizzando tabelle.

I modelli dei requisiti hanno dei vantaggi rispetto ai requisiti rappresentati in linguaggio naturale:

- Le relazioni e le interconnessioni tra requisiti sono più facilmente comprensibili con i modelli grafici rispetto a quando sono specificati in linguaggio naturale.
- Focalizzarsi su un singolo aspetto riduce lo sforzo cognitivo necessario a comprendere i requisiti modellati.
- I linguaggi di modellazione dei requisiti hanno una sintassi ristretta che riduce le possibili ambiguità e omissioni.

I modelli hanno anche delle limitazioni:

- Mantenere consistenti tra loro dei modelli che si focalizzano su aspetti diversi è spesso una sfida.
- Le informazioni provenienti da diversi modelli richiedono di essere integrate per una comprensione causale.
- I modelli si focalizzano principalmente sui requisiti funzionali; la maggior parte dei requisiti di qualità e dei vincoli non possono essere espressi attraverso modelli con un effort ragionevole.
- La sintassi ristretta di un linguaggio di modellazione grafica implica che non tutti gli elementi dell'informazione rilevanti possono essere espresse in un modello.

Pertanto, i modelli dei requisiti e i requisiti in linguaggio naturale sono spesso combinati tra loro.

In RE, i modelli possono essere utilizzati per:

- *Specificare* i requisiti (principalmente funzionali) in modo parziale o completo, come mezzo per sostituire i requisiti rappresentati testualmente.
- *Decomporre* una realtà complessa in aspetti ben definiti e complementari; ogni aspetto è rappresentato da un modello specifico.
- *Parafrasare* requisiti rappresentati testualmente per migliorare la loro comprensibilità, in particolare rispetto alle relazioni tra requisiti.

- *Validare* i requisiti rappresentati testualmente con l'obiettivo di scoprire omissioni, ambiguità e inconsistenze.

I *linguaggi di modellazione* sono utilizzati per esprimere modelli. Diversi linguaggi di modellazione, ad esempio UML [OMG2017] o BPMN [OMG2013], sono stati standardizzati. Quando i requisiti vengono specificati in un linguaggio di modellazione non standard, è richiesta una legenda che spieghi la sintassi e la semantica del linguaggio di modellazione utilizzato.

Esistono molti tipi di modelli che possono essere utilizzati in RE. Un Requirements Engineer deve comprendere quale tipo di modello è il più adatto a specificare i requisiti in una data situazione.

In una fase iniziale, il Requirements Engineer inizia spesso con la modellazione del contesto (3.4.2) o degli obiettivi del sistema previsto.

### 3.4.2 Modellare il Contesto (L2)

I modelli che si focalizzano sull'aspetto del contesto specificano come la struttura di un sistema è incorporata nel suo ambiente, e definiscono l'interazione tra un sistema e gli attori nel contesto di sistema.

I *modelli del contesto* (context model) specificano un sistema e gli attori nel contesto di sistema che interagiscono con il sistema stesso. Un modello del contesto descrive anche le interfacce tra un sistema e il suo contesto (ad esempio, in termini di quali informazioni vengono scambiate).

I *context diagram* vengono utilizzati come linguaggio di modellazione grafica per esprimere i modelli del contesto. Non esiste una notazione standardizzata per i context diagram. Per esprimere i modelli del contesto possono essere utilizzati context diagram dall'analisi strutturata [DeMa1978] o diagrammi [Glin2019] box-and-line personalizzati.

Nel linguaggio di modellazione UML [OMG2017], *gli use case diagram* forniscono un mezzo per modellare un sistema e il suo contesto in termini di use case di sistema e di attori nel contesto di sistema che interagiscono con il sistema attraverso questi use case.

Gli *use case* modellano l'interazione dinamica tra un attore nel contesto del sistema e un sistema dalla prospettiva dell'attore. Gli use case sono generalmente scritti utilizzando template di form in linguaggio naturale (3.3) o utilizzando activity diagram UML (3.4.4).

### 3.4.3 Modellare Struttura e Dati (L3)

I modelli che si focalizzano sugli aspetti della struttura e dei dati specificano requisiti relativi alle proprietà strutturali statiche di un sistema o di un dominio.

I modelli di dominio statico specificano gli oggetti (di business) e le loro relazioni in un dominio di interesse. Possono essere espressi con i class diagram UML [OMG2017].

I *modelli delle classi* specificano principalmente le classi di un sistema con i relativi attributi e relazioni. Le classi rappresentano entità tangibili e intangibili del mondo reale che il sistema

deve conoscere per poter svolgere i propri compiti. I class diagram UML sono tipicamente utilizzati come linguaggio di modellazione per i modelli delle classi.

### 3.4.4 Modellare Funzione e Flusso (L3)

I modelli che si focalizzano sugli aspetti relativi alla funzione e al flusso specificano i requisiti per la sequenza di azioni richieste per generare i risultati attesi da specifici input, o specificano le azioni richieste per eseguire un processo (di business). I modelli includono il control flow (flusso di controllo) e il data flow (flusso dei dati) tra le azioni, e chi è responsabile di specifiche azioni.

I *modelli delle attività* (activity model) vengono utilizzati per specificare le funzioni di sistema. Nel linguaggio di modellazione UML [OMG2017], gli *activity diagram* vengono utilizzati per descrivere i modelli delle attività. Forniscono elementi per modellare le azioni e il control flow tra le azioni. Gli activity diagram possono anche descrivere chi è responsabile di specifiche azioni. Elementi di modellazione avanzati (non coperti dal Syllabus CPRE Foundation Level) forniscono i mezzi per modellare il data flow.

I *modelli di processo* sono utilizzati per descrivere i processi di business o i processi tecnici. Possono essere espressi con activity diagram UML o con specifici linguaggi di modellazione dei processi come BPMN [OMG2013]. Nel Syllabus CPRE Foundation Level, per la modellazione dei processi vengono utilizzati solo gli activity diagram UML.

### 3.4.5 Modellare Stato e Comportamento (L2)

I modelli che si focalizzano sullo stato e sul comportamento specificano i requisiti per il comportamento di un sistema o di un componente del dominio in termini di reazioni dipendenti dallo stato rispetto ad eventi, oppure in termini di dinamiche di interazione dei componenti.

Le *state machine* (*macchine a stati*) modellano gli eventi che attivano la transizione da uno stato ad un altro, e le azioni che devono essere eseguite quando avviene una transizione di stato. Gli *statechart* [Hare1988] sono delle state machine, con stati decomposti gerarchicamente e/o ortogonalmente. Le state machine, compresi gli statechart, possono essere espresse nel linguaggio di modellazione UML [OMG2017] con gli *state machine diagram* (detti anche *state diagram*).

### 3.4.6 Altri Tipi di Modelli nel Requirements Engineering (L1)

Nel Syllabus CPRE Foundation Level, la comprensione e l'applicazione dei modelli è limitata ai tipi di modelli importanti, selezionati. Esistono altri tipi di modelli utilizzati nel Requirements Engineering. Nel Syllabus CPRE Foundation Level, è sufficiente conoscerli e sapere perchè vengono utilizzati.

*Modelli degli obiettivi* rappresentano un insieme di obiettivi, sotto-obiettivi e relazioni tra questi. I modelli degli obiettivi possono anche includere le attività e le risorse necessarie per

raggiungere un obiettivo, gli attori che vogliono raggiungere un obiettivo e gli ostacoli che impediscono il raggiungimento di un obiettivo.

In SysML [OMG2019], i *block definition diagram* possono essere adattati per creare context diagram, utilizzando blocchi stereotipati per il sistema e gli attori. I block definition diagram possono essere anche utilizzati per modellare la struttura di un sistema in termini di entità concettuali del sistema e di relazioni tra queste.

I *modelli delle storie del dominio (domain story model)* possono essere utilizzati per modellare funzioni e flussi, specificando visual story su come gli attori interagiscono con i dispositivi, gli artefatti e altri elementi di un dominio, utilizzando generalmente simboli specifici del dominio [HoSch2020]. Sono un mezzo per comprendere il dominio applicativo in cui opererà un sistema.

I *modelli di interazione (interaction model)* modellano le interazioni dinamiche tra oggetti o attori. I *sequence diagram* UML sono un mezzo comune per specificare l'interazione tra oggetti.

### 3.5 Glossari (L2)

In ogni attività di RE che coinvolge più di una persona, esiste il rischio di una mancanza di comprensione condivisa della terminologia – cioè che alcune persone interpretino gli stessi termini in modi differenti. Per mitigare questo problema, la comprensione condivisa dei termini rilevanti viene memorizzata in un glossario.

Un *glossario* è una raccolta centralizzata di definizioni per: termini specifici del contesto, termini quotidiani con un significato speciale nel contesto specificato, abbreviazioni e acronimi. I sinonimi (termini diversi che indicano la stessa cosa) dovrebbero essere contrassegnati come tali. Gli omonimi (utilizzo dello stesso termine per cose differenti) dovrebbero essere evitati o contrassegnati come tali.

Per i glossari si applicano le seguenti regole:

- Gestire il glossario in modo centralizzato.
- Mantenere il glossario per tutto il ciclo di sviluppo del sistema.
- Definire una persona o un piccolo gruppo come responsabile del glossario.
- Utilizzare uno stile e una struttura uniformi per il glossario.
- Coinvolgere gli stakeholder e cercare un accordo sulla terminologia.
- Rendere il glossario accessibile a tutte le persone coinvolte.
- Rendere obbligatorio l'uso del glossario.
- Verificare l'uso appropriato del glossario nei prodotti di lavoro.

### 3.6 Strutture della Documentazione e dei Documenti dei Requisiti (L2)

I documenti di specifica dei requisiti (3.1.1) comprendono diversi prodotti di lavoro di RE. È quindi importante organizzare tali documenti con una struttura ben definita, per creare una

raccolta di requisiti consistente e manutenibile. Oltre ai requisiti, un documento dei requisiti può anche contenere informazioni e spiegazioni aggiuntive – ad esempio, un glossario, condizioni di accettazione, informazioni di progetto o caratteristiche dell'implementazione tecnica.

I requisiti possono essere organizzati anche in strutture di documentazione diverse dai documenti classici.

I documenti utilizzati di frequente sono:

- Specifica dei Requisiti degli Stakeholder (Stakeholder Requirements Specification)
- Specifica dei Requisiti degli Utenti (User Requirements Specification, un sottoinsieme di una specifica dei requisiti degli stakeholder che copre solo i requisiti degli utenti)
- Specifica dei Requisiti di Sistema (System Requirements Specification)
- Specifica dei Requisiti di Business (Business Requirements Specification)
- Documento della Vision.

Le strutture di documentazione alternative frequentemente utilizzate sono:

- Product backlog
- Sprint backlog
- Story map.

Sia la selezione di una struttura della documentazione sia l'organizzazione interna della struttura scelta dipendono da:

- Processo di sviluppo selezionato (5)
- Dominio e tipo di sviluppo
- Contratto (un cliente può prescrivere l'utilizzo di una determinata struttura della documentazione)
- Dimensione del documento.

I template di documento possono aiutare a strutturare una specifica dei requisiti. I template sono disponibili in letteratura [Vole2020], [RoRo2012] e negli standard [ISO29148]. I template possono anche essere riutilizzati da precedenti progetti simili, o possono essere imposti da un cliente. Un'organizzazione può anche decidere di creare un template come standard interno.

### 3.7 Prototipi nel Requirements Engineering (L1)

In RE, i *prototipi* sono un mezzo per specificare i requisiti tramite esempi e per validare i requisiti. In particolare, i prototipi possono essere utilizzati se gli stakeholder coinvolti non vogliono scrivere e svolgere la review di prodotti di lavoro basati sul linguaggio naturale, basati sui template o model-based.

I [LiSZ1994] *prototipi esplorativi* vengono utilizzati per creare una comprensione condivisa, per chiarire i requisiti o per validare i requisiti a diversi livelli di fedeltà. Questi vengono scartati dopo l'uso.

- I *wireframe* sono prototipi low-fidelity (di bassa fedeltà) costruiti con materiali semplici o con strumenti per gli schizzi; servono principalmente per discutere e validare le idee di progettazione e i concetti della user interface.
- I *mock-up* sono prototipi medium-fidelity (di media fedeltà). Quando si specificano sistemi digitali, questi utilizzano schermate reali e click-flow, ma senza una reale funzionalità. Servono principalmente per specificare e validare le user interface.
- I *prototipi nativi* sono prototipi high-fidelity (di alta fedeltà) che implementano specifiche parti critiche di un sistema in modo tale da permettere agli stakeholder di utilizzare il prototipo per verificare che la parte prototipata del sistema funzionerà e si comporterà come previsto.

In base al grado di fedeltà, i prototipi esplorativi possono essere un prodotto di lavoro costoso, quindi esiste sempre un compromesso tra costo e valore ottenuto.

I *prototipi evolutivi* [LiSZ1994] sono sistemi pilota che costituiscono il nucleo di un sistema da sviluppare. Il sistema finale evolve estendendo e migliorando in modo incrementale il sistema pilota attraverso diverse iterazioni.

### 3.8 Criteri di Qualità per i Prodotti di Lavoro e per i Requisiti (L1)

Un requisito deve soddisfare specifici criteri di qualità per essere considerato un buon requisito. Nel RE moderno con approcci orientati al valore (Principio 1 in 2), il grado di soddisfazione di un criterio di qualità deve corrispondere al valore creato da questo requisito. Questo significa che i requisiti non devono aderire completamente a tutti i criteri di qualità – ma più alto è il valore di un requisito, tanto più rilevanti sono i criteri di qualità, per ridurre il rischio di failure.

L'adeguatezza e la comprensibilità sono i criteri di qualità più importanti per i singoli requisiti. Senza questi criteri, un requisito è inutile o addirittura dannoso, indipendentemente dalla soddisfazione di tutti gli altri criteri.

Criteri di qualità per i *singoli requisiti*:

- Adeguato (descrive le esigenze degli stakeholder, reali e concordate)
- Necessario
- Non ambiguo
- Completo (auto-esplicativo)
- Comprensibile
- Verificabile.

Come descritto in 3.1.1, i requisiti vengono di solito memorizzati in diversi prodotti di lavoro che coprono uno o più requisiti. I criteri di qualità di cui sopra dovranno essere utilizzati per creare singoli requisiti di alta qualità in un prodotto di lavoro. Per i prodotti da lavoro che coprono più requisiti, si dovranno considerare i seguenti criteri di qualità aggiuntivi.

Criteria di qualità per prodotti di lavoro che coprono *più requisiti*:

- Consistente
- Non ridondante
- Completo (nessun requisito conosciuto e rilevante è stato dimenticato)
- Modificabile
- Tracciabile
- Conforme.

# 4 Pratiche per l'Elaborazione dei Requisiti

## (L3)

**Obiettivo:** Comprendere l'uso di pratiche per identificare le sorgenti dei requisiti, per elicitare i requisiti, per identificare e risolvere i conflitti e per validare i requisiti

**Durata:** 4 ore e 30 minuti

**Termini:** Sorgente dei requisiti, confine di sistema, contesto del sistema, elicitazione dei requisiti, validazione dei requisiti, stakeholder, modello Kano, risoluzione del conflitto

### Obiettivi Formativi

- EO 4.1.1 Determinare i confini del sistema per focalizzarsi sui requisiti rilevanti (L3)
- EO 4.1.2 Ricordare le sorgenti rilevanti per il sistema da creare (L1)
- EO 4.1.3 Identificare gli stakeholder e scrivere una lista degli stakeholder (L3)
- EO 4.1.4 Comprendere i benefici della gestione degli stakeholder (L2)
- EO 4.2.1 Comprendere come il modello Kano può aiutare a elicitare i giusti requisiti (L2)
- EO 4.2.2 Comprendere la differenza tra le tecniche di raccolta e le tecniche di progettazione e generazione di idee (L2)
- EO 4.2.3 Comprendere come scegliere una tecnica di elicitazione appropriata per una specifica situazione (L2)
- EO 4.3.1 Richiamare i differenti tipi di conflitto (L1)
- EO 4.3.2 Comprendere quali attività sono necessarie per risolvere i conflitti (L2)
- EO 4.3.3 Comprendere come applicare tecniche appropriate di risoluzione dei conflitti (L2)
- EO 4.4.1 Comprendere perché i documenti dei requisiti devono essere validati (L2)
- EO 4.4.2 Richiamare i quattro aspetti importanti per la validazione dei requisiti (L1)
- EO 4.4.3 Comprendere come applicare tecniche appropriate per la validazione dei requisiti (L2)

## 4.1 Sorgenti dei Requisiti (L3)

La qualità e la completezza dei requisiti dipendono in larga misura dalle sorgenti dei requisiti coinvolte. Dimenticare una sorgente importante porterà a una comprensione incompleta dei requisiti o a requisiti incompleti. L'identificazione delle sorgenti dei requisiti è un processo iterativo e ricorsivo che richiede una rivalutazione costante.

Una comprensione condivisa (Principio 3 in 2) del contesto del sistema da sviluppare è un prerequisito per poter identificare le sorgenti dei requisiti rilevanti. L'area tra il confine del sistema e il confine del contesto è chiamata il contesto (del sistema) (Principio 4 in 2). Il contesto (del sistema) è necessario per comprendere la natura dei requisiti da sviluppare e quindi per identificare le sorgenti originali dei requisiti.

Le sorgenti dei requisiti sono classificate in tre tipologie:

- Stakeholder
- Documenti
- Sistemi



Gli stakeholder di un sistema (vedi [Glin2020] per una definizione; vedi anche Principio 2in 2) sono la sorgente principale dei requisiti. I ruoli tipici degli stakeholder includono [BiSp2003]:

- Utenti (chiamati anche utenti finali)
- Sponsor
- Manager
- Sviluppatori
- Autorità
- Clienti.

Inoltre, le persone o le organizzazioni che sono *impattate* da un sistema dovrebbero essere considerate come stakeholder (indiretti).

L'identificazione sistematica degli stakeholder dovrebbe avvenire nella fase iniziale del progetto di sviluppo, e i risultati dovrebbero essere gestiti durante tutto lo sviluppo come un'attività continuativa. Questa include l'identificazione dei ruoli degli stakeholder e delle persone che ricoprono tali ruoli.

Per tutti i sistemi con una user interface, gli *utenti finali* del sistema costituiscono un gruppo di stakeholder di particolare interesse per il Requirements Engineer. Gli utenti finali devono essere aggregati in gruppi (ad esempio, per ruoli, compiti o responsabilità simili).

Quando gli utenti finali possono essere identificati individualmente, è necessario scegliere i rappresentanti di ciascun gruppo. Altrimenti, possono essere definite Personas per rappresentare i gruppi di utenti finali rilevanti [Coop2004].

Potenziati sorgenti per l'identificazione degli stakeholder rilevanti sono:

- Checklist di gruppi e ruoli dei tipici stakeholder
- Strutture organizzative
- Documentazione dei processi di business
- Report di mercato
- Stakeholder iniziali per l'identificazione di stakeholder *aggiuntivi*.

Gli stakeholder dovrebbero essere documentati in una lista di stakeholder mantenuta aggiornata, con (almeno) le seguenti informazioni:

- Nome
- Funzione (ruolo)
- Dati aggiuntivi personali e di contatto
- Disponibilità temporale e spaziale durante l'avanzamento del progetto
- Rilevanza
- Area e ampiezza della competenza
- Obiettivi e interessi in termini di progetto.

Possono nascere problemi con gli stakeholder se i diritti e gli obblighi di uno stakeholder non sono chiari, o se le esigenze dello stakeholder non sono sufficientemente indirizzate. La gestione delle relazioni degli stakeholder [Bour2009] è un modo efficace per identificare i problemi con gli stakeholder.

In molti contesti del sistema, sono disponibili più sorgenti. Gli stakeholder devono essere considerati anche per un nuovo sistema di successo, in quanto la maggior parte degli stakeholder non esplicita quello che viene ritenuto ovvio: i loro requisiti "subconsci" (4.2).

Sorgenti aggiuntive dei requisiti includono:

- Sistemi esistenti e legacy
- Documenti di processo
- Documenti legali o normativi
- Normative specifiche dell'azienda
- Informazioni (di marketing) sui potenziali futuri utenti

Un'altra sorgente dei requisiti può essere identificata esaminando situazioni simili in domini completamente differenti.

## 4.2 Elicitazione dei Requisiti (L2)

Durante l'elicitatione, è compito del Requirements Engineer comprendere i desideri e le esigenze degli stakeholder, assicurando che i requisiti siano stati raccolti da tutte le sorgenti rilevanti dei requisiti, attraverso l'applicazione di tecniche appropriate per la loro elicitatione. Un punto importante nell'elicitatione è la trasformazione delle richieste implicite, dei desideri e delle aspettative in requisiti espliciti.

Per elicitare i requisiti, è cruciale conoscere la natura e l'importanza di ogni requisito. Questi possono cambiare da progetto a progetto e anche nel tempo. Il modello Kano [KaeA1984] classifica i requisiti in tre categorie importanti:

- Delighters (sinonimi: fattori di Excitement, requisiti inconsci)
- Satisfiers (sinonimi: fattori di Performance, requisiti consci)
- Dissatisfiers (sinonimi: fattori Basic, requisiti subconsci)

Esistono molte tecniche diverse per elicitare queste categorie di requisiti. Sono state differenziate come segue:

- Tecniche di raccolta
- Tecniche di progettazione e generazione di idee.

Le *tecniche di raccolta* sono tecniche stabilite per l'elicitatione dei requisiti [BaCC2015] che aiutano a identificare i requisiti satisfiers e dissatisfiers attraverso l'analisi di sorgenti differenti.

Si possono distinguere quattro categorie principali:

- Tecniche di indagine
- Tecniche di collaborazione
- Tecniche di osservazione
- Tecniche basate sugli artefatti.

Le *tecniche di progettazione e di generazione di idee* hanno lo scopo di stimolare la creatività durante l'elicitatione dei requisiti. Il loro scopo è quello di creare idee per risolvere un problema e di esplorare idee di progettazione [Kuma2013]. Questo può portare a requisiti

nuovi o innovativi, che sono spesso *delighter*. Esempi popolari di queste tecniche sono il *brainstorming* [Osbo1979], le analogie, la prototipazione (ad esempio, i *mock-up*), gli scenari e le *storyboard*.

Un concetto più ampio legato alla progettazione e alla generazione di idee è il *design thinking*. Esistono diversi approcci, come *d.school* [Sdsc2012] e *Designing for Growth* [LiOg2011], che offrono una vasta gamma di tecniche da utilizzare per l'elicitazione dei requisiti.

Le tecniche di elicitazione dovrebbero essere in grado di rilevare tutti i tipi di requisiti – requisiti funzionali, requisiti di qualità e vincoli. Nella pratica, viene data meno attenzione ai requisiti di qualità e ai vincoli.

Per elicitare i *requisiti di qualità*, un modello di qualità come lo standard ISO 25010 [ISO25010] dovrebbe essere usato come *checklist*. Questo modello può essere utile anche per quantificare i requisiti.

I *vincoli* possono essere trovati considerando possibili restrizioni dello spazio potenziale della soluzione – ad esempio, problematiche tecniche, legali, organizzative, culturali o ambientali.

Scegliere le giuste tecniche di elicitazione è una competenza chiave critica che dipende da molti fattori differenti, come:

- Tipo di sistema
- Modello del ciclo di vita dello sviluppo software
- Persone coinvolte
- Setup organizzativo

I migliori risultati sono generalmente raggiunti con una combinazione di diverse tecniche di elicitazione. [CaDJ2014] rappresentano un approccio sistematico per la selezione delle tecniche.

### 4.3 Risolvere i Conflitti Relativi ai Requisiti (L2)

Le tecniche di elicitazione da sole non assicurano che l'insieme risultante dei requisiti sia consistente, completo, conforme, ecc. (3.8). Tuttavia, per l'insieme finale dei requisiti tutti gli stakeholder devono comprendere e concordare su tutti i requisiti che sono per loro importanti. Se alcuni stakeholder non sono d'accordo, la situazione deve essere riconosciuta come un conflitto, che dovrebbe essere risolto di conseguenza. Le tecniche più adatte alla risoluzione del conflitto dovrebbero essere selezionate in base al tipo di conflitto e alle informazioni contestuali. Questo richiede una comprensione approfondita della natura del conflitto dei requisiti e degli atteggiamenti degli stakeholder coinvolti.

Le attività per identificare e risolvere i conflitti sono:

- Identificazione del conflitto
- Analisi del conflitto
- Risoluzione del conflitto
- Documentazione della risoluzione del conflitto (decisioni prese).

È utile distinguere tra diversi tipi di conflitto [Moor2014]. I seguenti tipi di conflitto richiedono spesso l'attenzione del Requirements Engineer:

- Conflitto sull'argomento
- Conflitto sui dati (contenuto)
- Conflitto di interessi
- Conflitto di valori
- Conflitto di relazione
- Conflitto strutturale

Per risolvere i conflitti con successo, possono essere applicate tecniche comuni:

- Agreement (accordo)
- Compromesso
- Votazione
- Imposizione
- Definizione di varianti

Esistono ulteriori tecniche ausiliarie, ad esempio:

- Valutazione di tutti i fatti
- Interesse più-meno
- Matrice decisionale

## 4.4 Validazione dei Requisiti (L2)

La validazione dei requisiti è un passo importante verso un sistema di successo (Principio 6 in 2). Garantire la qualità dei requisiti in anticipo ridurrà l'effort speso inutilmente nelle fasi successive. Validare i requisiti significa verificare manualmente la qualità dei prodotti di lavoro e dei singoli requisiti in essi contenuti (per i dettagli, si veda 3 . 8).

Gli aspetti importanti da considerare nella validazione dei requisiti sono:

- Coinvolgimento dei giusti stakeholder
- Separazione dell'identificazione dei difetti dalla relativa correzione
- Validazione da diversi punti di vista
- Esecuzione di validazioni ripetute.

Esistono diverse tecniche per la validazione (ad esempio [GiGr1993], [OleA2018]). Queste tecniche di validazione sono spesso classificate in:

- *Tecniche di review*, che includono:
  - Walkthrough
  - Ispezione
- *Tecniche esplorative*, come ad esempio:
  - Prototipizzazione
  - Alpha Testing e Beta Testing

- Testing A/B [KoTh2017]
- Costruzione di un prodotto minimo funzionante (MVP, Minimum Viable Product)
- *Sviluppo di un campione (sample development).*

Queste tecniche differiscono tra loro per il livello di formalizzazione e per l'effort richiesto. La scelta della tecnica dipende da fattori quali il modello del ciclo di vita dello sviluppo software, la maturità del processo di sviluppo, la complessità e il livello di rischio del sistema, eventuali requisiti legali o normativi, e/o la necessità di un audit trail.

# 5 Struttura del Processo e delle Attività

## (L3)

Obiettivo: Spiegare i concetti di un processo di RE e applicare le configurazioni di processo appropriate

Durata: 1 ora e 15 minuti

Termini: Processo, processo di RE

### Obiettivi Formativi

- EO 5.1.1 Conoscere i fattori importanti che influenzano un processo di RE (L1)
- EO 5.1.2 Comprendere come e perché questi fattori influenzano il processo di RE (L2)
- EO 5.2.1 Comprendere i differenti aspetti (facet) da considerare per la configurazione di un processo di RE (L2)
- EO 5.3.1 Conoscere le tipiche configurazioni del processo di RE (L1)
- EO 5.3.2 Comprendere i passi per la configurazione di un processo di RE (L2)
- EO 5.3.3 Per semplici impostazioni di sistema e di sviluppo, selezionare e applicare una configurazione del processo di RE appropriata (L3)

Un processo di RE è necessario per modellare e strutturare le attività di RE da svolgere in un determinato contesto. Poiché non esiste un unico processo di RE valido per tutti i sistemi/progetti (1.4), deve essere configurato un tailoring (adattamento) del processo di RE che si adatti al contesto di sviluppo e di sistema considerato.

Il processo di RE modella il flusso informativo e il modello di comunicazione tra i vari partecipanti (ad es. clienti, utenti, Requirements Engineer, sviluppatori, tester) e definisce anche i prodotti di lavoro da utilizzare o produrre. Pertanto, il processo di RE fornisce il framework per l'elicitazione, la documentazione, la validazione e la gestione dei requisiti.

## 5.1 Fattori di Influenza (L2)

Molti fattori influenzano la configurazione di un processo di RE. I fattori principali sono:

- Adattamento del processo generale: il processo di RE deve adattarsi all'intero processo di sviluppo del sistema
- Contesto di sviluppo
- Capacità e disponibilità degli stakeholder
- Comprensione condivisa
- Complessità e criticità del sistema da sviluppare
- Vincoli
- Tempo e budget disponibile
- Volatilità dei requisiti
- Esperienza dei Requirements Engineer.

Un'analisi dei fattori di influenza fornisce informazioni su come configurare il processo di RE. I fattori di influenza limitano anche lo spazio delle possibili configurazioni del processo. Ad

esempio, quando gli stakeholder sono disponibili solo all'inizio del progetto, non è possibile scegliere un processo che si basi sul feedback continuo degli stakeholder.

## 5.2 Aspetti (Facet) del Processo di Requirements Engineering (L2)

Esistono tre aspetti (facet) decisivi che devono essere considerati quando si configura un processo di RE [Glin2019].

### Aspetto relativo al Tempo: Lineare vs Iterativo

In un processo lineare, i requisiti vengono specificati in anticipo, durante una singola fase del processo. In un processo iterativo, i requisiti vengono specificati in modo incrementale, a partire da obiettivi generali e da alcuni requisiti iniziali, per poi aggiungere o modificare i requisiti in ogni iterazione.

Criteri per la scelta di un processo di RE *lineare*:

- Il processo di sviluppo del sistema è plan-driven (guidato da una pianificazione) ed è per lo più lineare.
- Gli stakeholder conoscono i propri requisiti e sono in grado di specificarli in anticipo.
- E' richiesta una specifica dei requisiti completa come base contrattuale per l'outsourcing della progettazione e dell'implementazione del sistema.
- Le autorità legali richiedono una specifica dei requisiti completa e formalmente rilasciata nelle fasi iniziali dello sviluppo.

Criteri per la scelta di un processo di RE *iterativo*:

- Il processo di sviluppo del sistema è iterativo e Agile.
- Molti requisiti non sono noti in anticipo, ma emergeranno ed evolveranno durante lo sviluppo del sistema.
- Gli stakeholder sono disponibili per definire brevi cicli di feedback come mezzo per mitigare il rischio di sviluppare il sistema sbagliato.
- La durata dello sviluppo permette più di una o due iterazioni.
- La capacità di modificare facilmente i requisiti è importante.

### Aspetto relativo allo Scopo: Prescrittivo vs Esplorativo

In un processo di RE prescrittivo, la specifica dei requisiti costituisce un contratto: tutti i requisiti sono vincolanti e devono essere implementati. In un processo di RE esplorativo, solo gli obiettivi sono noti a priori, mentre i requisiti concreti devono essere esplorati.

Criteri per la scelta di un processo di RE *prescrittivo*:

- Il cliente richiede un contratto fissato per lo sviluppo del sistema.
- La funzionalità e l'ambito hanno la precedenza sui costi e sulle scadenze.
- Lo sviluppo del sistema specificato può essere dato in appalto oppure in outsourcing (esternalizzato).

Criteri per la scelta di un processo di RE *esplorativo*:

- Gli stakeholder inizialmente hanno solo una vaga idea dei loro requisiti.
- Gli stakeholder sono fortemente coinvolti e forniscono feedback continui.
- Scadenze e costi hanno la precedenza sulla funzionalità e sull'ambito.
- Non è chiaro a priori quali requisiti dovranno essere effettivamente implementati e in quale ordine saranno implementati.

### Aspetti legati all'Obiettivo (Target): Customer-Specific (Specifico per il Cliente) vs Market-Oriented (Orientato al Mercato)

In un processo di RE Customer-Specific, il sistema viene commissionato da un cliente e sviluppato da un fornitore. In un processo di RE Market-Oriented, il sistema viene sviluppato come prodotto o servizio per un mercato, rivolgendosi a specifici segmenti di utenza.

Criteri per la scelta di un processo di RE *Customer-Specific*:

- Il sistema sarà utilizzato principalmente dall'organizzazione che ha ordinato il sistema e pagherà per la relativa implementazione.
- Gli stakeholder importanti sono soprattutto quelli relativi all'organizzazione del cliente.
- Si possono identificare singoli individui per i ruoli degli stakeholder considerati.
- Il cliente vuole una specifica dei requisiti che possa servire come contratto.

Criteri per la scelta di un processo di RE *Market-Oriented*:

- L'organizzazione che sta sviluppando il sistema intende vendere il sistema in qualche segmento di mercato come prodotto o servizio.
- I potenziali utenti non sono identificabili singolarmente.
- I Requirements Engineer devono progettare i requisiti in modo che corrispondano alle esigenze previste per gli utenti target.
- I Product Owner, gli addetti al marketing, i progettisti digitali e gli architetti di sistema sono gli stakeholder principali.

### Suggerimenti e Avvertenze

- I criteri presentati sopra sono da considerare *euristici*, piuttosto che regole fisse. Ad esempio, l'outsourcing dello sviluppo del sistema viene svolto preferibilmente con un processo di RE prescrittivo piuttosto che con un processo esplorativo, perché il contratto tra il cliente e il fornitore è tipicamente basato su una specifica dei requisiti completa. Tuttavia, è anche possibile negoziare un contratto di outsourcing basato su un processo di RE esplorativo.
- I processi di RE lineari funzionano solo se è in atto un processo sofisticato per le modifiche dei requisiti.
- I processi di RE lineari implicano cicli di feedback lunghi: Per mitigare il rischio di sviluppare un sistema sbagliato, i requisiti devono essere validati in maniera approfondita.
- Quando si definisce un processo di RE, *lineare* e *prescrittivo* vengono spesso scelti insieme.
- I processi di RE esplorativi sono tipicamente anche processi iterativi (e viceversa).
- In un processo Market-Oriented, il feedback degli utenti è l'unico mezzo per validare se il prodotto soddisferà effettivamente le esigenze del segmento di utenza target.



- Un approccio Market-Oriented non si combina bene con gli aspetti lineare e prescrittivo.

## 5.3 Configurare un Processo di Requirements Engineering (L3)

In un contesto concreto di sviluppo di un sistema, le persone responsabili di RE devono configurare il processo di RE da applicare. Sulla base di un'analisi dei fattori di influenza in 5.1, può essere utilizzata una combinazione adeguata degli aspetti del processo descritti in 5.2 [Glin2019]. Di seguito vengono descritte tre combinazioni tipiche.

### Processo di RE Partecipativo: Iterativo & Esplorativo & Customer-Specific

Caso applicativo principale:	Fornitore e cliente collaborano in modo stretto e continuo; gli stakeholder sono fortemente coinvolti sia nel processo di RE che nel processo di sviluppo
Tipici prodotti di lavoro:	Product Backlog con user story e/o descrizioni di attività, prototipi
Tipico flusso informativo:	Interazione continua tra gli stakeholder, i Product Owner, i Requirements Engineer e gli sviluppatori; può includere il feedback degli utenti

### Processo di RE Contrattuale: Tipicamente Lineare (a volte Iterativo) & Prescrittivo & Customer-Specific

Caso applicativo principale:	La specifica dei requisiti costituisce la base contrattuale per lo sviluppo di un sistema da parte di persone non coinvolte nella specifica e con limitata interazione con gli stakeholder dopo la fase dei requisiti
Tipici prodotti di lavoro:	Specifica classica dei requisiti del sistema, che consiste in modelli e requisiti basati sul linguaggio naturale
Tipico flusso informativo:	Principalmente dagli stakeholder verso i Requirements Engineer

## Processo di RE Product-Oriented: Iterativo & Esplorativo & Market-Oriented

Caso applicativo principale: Un'organizzazione specifica e sviluppa software per venderlo o distribuirlo come prodotto o servizio

Tipici prodotti di lavoro: Product Backlog, prototipi

Tipico flusso informativo: Interazione tra Product Owner, marketing, Requirements Engineer, progettisti digitali, sviluppatori e (eventualmente) feedback veloci da parte di clienti/utenti.

Si noti che possono esistere contesti di sistema e di sviluppo in cui non è adatta nessuna delle suddette configurazioni. Ad esempio, i vincoli normativi possono imporre l'uso di un processo conforme a determinati standard come ISO/IEC/IEEE 29148 [ISO29148].

Quando si configura un processo di RE, si raccomanda di utilizzare una procedura definita da cinque passi:

1. Analizzare i fattori di influenza (5.1)
2. Valutare i criteri degli aspetti (5.2)
3. Configurare il processo (5.3)
4. Determinare i prodotti di lavoro (3)
5. Selezionare le pratiche appropriate.

# 6 Pratiche per il Requirements Management

## (L2)

Obiettivo: Comprendere la necessità e il beneficio del requirements management

Durata: 2 ore

Termini: Requirements management (gestione dei requisiti), change management (gestione delle modifiche), tracciabilità, attributi dei requisiti, ciclo di vita dei requisiti, prioritizzazione

### Obiettivi Formativi

- EO 6.1.1 Conoscere che cosa è il requirements management e perché è necessario (L1)
- EO 6.2.1 Spiegare perché i prodotti di lavoro dei requisiti richiedono un modello di ciclo di vita/stati (L2)
- EO 6.3.1 Spiegare come si presenta un concetto di versioning dei requisiti in una data situazione di progetto (L2)
- EO 6.4.1 Conoscere l'utilizzo delle configurazioni e delle baseline dei requisiti (L1)
- EO 6.5.1 Conoscere lo scopo degli attributi per i requisiti (L1)
- EO 6.5.2 Spiegare come si presenta un insieme appropriato di attributi per i requisiti in una data situazione di progetto (L2)
- EO 6.5.3 Spiegare lo scopo delle viste ed elencare le diverse viste dei requisiti (L2)
- EO 6.6.1 Elencare le ragioni per una tracciabilità dei requisiti (L1)
- EO 6.6.2 Riassumere le differenze tra tracciabilità implicita ed esplicita (L1)
- EO 6.6.3 Conoscere come può essere documentata la tracciabilità esplicita (L1)
- EO 6.7.1 Conoscere come gestire le modifiche negli approcci lineari (plan-driven) e Agile (L1)
- EO 6.8.1 Conoscere le ragioni per una prioritizzazione dei requisiti e conoscere i criteri di valutazione significativi (L1)
- EO 6.8.2 Elencare i passi per prioritizzare i requisiti (L1)
- EO 6.8.3 Elencare differenti categorie di tecniche di prioritizzazione (L1)

## 6.1 Che cosa è il Requirements Management? (L1)

Requirements management è il processo di gestione dei requisiti esistenti, memorizzati in diversi prodotti di lavoro. In particolare, include la memorizzazione, la modifica e il tracciamento dei requisiti [Glin2020]. Il requirements management può avvenire in diversi modi e a diversi livelli, in base al processo di sviluppo selezionato e al contesto – si veda, ad esempio [Leff2011], [Rupp2014], [WiBe2013]. Indipendentemente dalle circostanze, il compito del requirements management è di mantenere i requisiti in modo tale che tutti i ruoli coinvolti in un progetto possano lavorare in modo efficace ed efficiente.

## 6.2 Gestione del Ciclo di Vita (L2)

La gestione del ciclo di vita si riferisce al processo in cui si tiene traccia di tutti i prodotti di lavoro rispetto allo stato del loro ciclo di vita. Ogni requisito documentato e ogni prodotto di lavoro ha un proprio ciclo di vita: viene creato, poi valutato e raffinato prima di essere sottoposto a review, rielaborato, consolidato, concordato e così via. Per consentire

l'identificazione di quale prodotto di lavoro è in quale stato, è necessario un modello di ciclo di vita che definisca tutti gli stati e transizioni di stato possibili del ciclo di vita. Lo stato attuale di un prodotto di lavoro dovrebbe essere sempre chiaro, includendo (di solito) la storia delle sue transizioni.

### 6.3 Controllo della Versione (L2)

Il controllo della versione dei requisiti si riferisce al processo in cui si tiene traccia di tutti i prodotti di lavoro durante la loro evoluzione. Qualsiasi modifica in un prodotto di lavoro dovrebbe essere associato a una nuova versione. Il versioning permette di far risalire la storia di un prodotto di lavoro alla sua origine e consente il ripristino di un prodotto di lavoro a qualsiasi versione precedente. A tal fine, il controllo delle versioni richiede l'adozione di tre misure:

- Un version number (numero di versione) per identificare in modo univoco la versione di un prodotto di lavoro.
- Una storia di ciò che è stato modificato.
- Un concetto per la memorizzazione del prodotto da lavoro.

Il versioning deve essere considerato per tutti i prodotti di lavoro [WiBe2013]. Un version number consiste normalmente di almeno due parti: la versione e l'incremento.

### 6.4 Configurazioni e Baseline (L1)

Una *configurazione* dei requisiti è un insieme consistente di prodotti di lavoro che contengono requisiti. Ogni configurazione è definita per uno scopo specifico e contiene al più una versione di ogni prodotto di lavoro [Glin2020]. Lo scopo delle configurazioni è, ad esempio, di eseguire la review di un insieme di prodotti di lavoro o di facilitare una stima dell'effort per lo sviluppo.

Una *baseline* è una *configurazione* di prodotti di lavoro stabile e controllata rispetto alle modifiche, che viene utilizzata per la pianificazione del rilascio in produzione o di altre milestone di rilascio in un progetto [Glin2020].

Le configurazioni hanno le seguenti proprietà:

1. Connessione logica
2. Consistenza
3. Unicità
4. Immutabilità
5. Base per il rollback (o reset)

### 6.5 Attributi e Viste (L2)

*Gli attributi* sono richiesti per documentare i metadati importanti per un prodotto di lavoro, e vengono normalmente utilizzati per rispondere a una serie di domande importanti durante il ciclo di vita del progetto o del prodotto.

L'obiettivo di utilizzare gli attributi per caratterizzare i requisiti è quello di consentire ai membri del team e agli altri stakeholder di ottenere le informazioni sui requisiti di cui hanno bisogno, in qualsiasi momento durante il progetto.

La definizione dell'insieme rilevante di attributi dipende dalle esigenze di informazione dei diversi stakeholder di progetto. Gli standard esistenti, ad esempio [ISO29148], forniscono una panoramica degli attributi più importanti.

Le *viste* sono un estratto dell'insieme complessivo dei requisiti, che includono solo il contenuto che è attualmente di interesse. Da un punto di vista tecnico, una vista è una combinazione di impostazioni di filtro e di ordinamento che può essere resa disponibile, o riutilizzabile, da altri partecipanti, attraverso la memorizzazione della combinazione selezionata.

Distinguiamo tre tipi di viste:

- *Viste selettive*
- *Viste proiettive*
- *Viste aggregate*

In molti casi, per la creazione dei report, le viste dei requisiti sono combinazioni delle tre viste, selettive, proiettive e aggregate

## 6.6 Tracciabilità (L1)

La tracciabilità [GoFi1994] è la capacità di *tracciare un requisito rispetto alla sua sorgente* (ad es. stakeholder, documenti, giustificazioni, ecc.) e *rispetto ai prodotti di lavoro successivi che lo realizzano* (ad es. test case), nonché *ad altri requisiti* da cui tale requisito dipende.

La tracciabilità è un prerequisito per il requirements management ed è spesso esplicitamente richiesta da standard, normative e leggi. Implementare la tracciabilità significa fondamentalmente mantenere le dipendenze tra differenti prodotti di lavoro (3.1) a diversi livelli di astrazione (3.1.2), livelli di dettaglio (3.1.3) e verso tutti i relativi predecessori e successori, per motivi di analisi, conformità e informazione.

La tracciabilità può essere documentata *in modo implicito*, strutturando e standardizzando i prodotti di lavoro, o *in modo esplicito*, mettendo in relazione i prodotti di lavoro tra loro in base ai loro identificativi univoci nelle diverse forme di rappresentazione [HuJD2011]. Le forme di rappresentazione comuni sono hyperlink, riferimenti, matrici, tabelle o grafi.

## 6.7 Change Management (Gestione delle Modifiche) (L1)

I requisiti non sono statici. Le modifiche ai requisiti avvengono a causa di molte ragioni differenti e devono essere gestite in modo appropriato (Principio 7 in 2), ad esempio creando una *change request* formale o aggiungendo un nuovo elemento al *product backlog*.

Il processo decisionale, la pianificazione e il controllo dell'implementazione di una change request dipendono dall'approccio adottato per l'implementazione e dal momento in cui si verifica la modifica.

In un approccio *lineare*, la decisione di una modifica viene spesso presa da un Change Control Board (nei progetti) o da un Change Advisory Board (in produzione). In un approccio più *iterativo*, il Product Owner include la modifica nel product backlog e prioritizza il nuovo elemento di conseguenza.

## 6.8 Prioritizzazione (L1)

Non tutti i requisiti sono ugualmente importanti [Davi2005]. La valutazione e la prioritizzazione vengono utilizzate per determinare i requisiti più importanti per il successivo rilascio o incremento del prodotto.

La *valutazione* dei requisiti è la base per la loro prioritizzazione, spesso determinata utilizzando criteri di valutazione multipli, come il valore di business, l'urgenza, l'effort, le dipendenze e altro.

La *priorità di* un requisito descrive l'importanza di un singolo requisito rispetto ad altri requisiti secondo determinati criteri [Glin2020]. La *prioritizzazione* stessa viene eseguita sulla base di un unico criterio o di più criteri; questo dipende principalmente dalla tecnica di prioritizzazione scelta.

Passi per la prioritizzazione:

- Definire gli obiettivi e i vincoli principali per la prioritizzazione
- Definire i criteri di valutazione desiderati
- Definire gli stakeholder che devono essere coinvolti
- Definire i requisiti che devono essere prioritizzati
- Selezionare la tecnica di prioritizzazione
- Eseguire la prioritizzazione

Le *tecniche di prioritizzazione* possono essere classificate in:

- Tecniche di prioritizzazione *ad hoc*
- Tecniche di prioritizzazione *analitiche*

## 7 Supporto degli Strumenti (L2)

Obiettivo: Fornire una panoramica del ruolo degli strumenti di RE e degli aspetti per l'implementazione

Durata: 30 minuti

Termini: Strumento (tool), strumento di RE

### Obiettivi Formativi

EO 7.1.1 Conoscere i diversi tipi di strumenti di RE (L1)

EO 7.2.1 Spiegare cosa considerare quando si introducono strumenti di RE (L2)

### 7.1 Strumenti nel Requirements Engineering (L1)

Il processo di RE può essere supportato da strumenti (tool) che permettono di svolgere compiti e attività specifiche. Poiché i processi di RE sono completamente individuali (5), gli strumenti di RE esistenti spesso si focalizzano solo su alcuni aspetti di RE e raramente supportano tutte le attività. Prima di selezionare uno strumento, i Requirements Engineer dovrebbero decidere quali compiti e attività dovrebbero essere supportati durante il processo di RE, e come. Distinguiamo tra gli strumenti che supportano:

- La gestione dei requisiti:
  - La definizione e la memorizzazione degli attributi dei requisiti
  - La prioritizzazione dei requisiti
  - La gestione delle versioni e delle configurazioni
  - Il tracciamento e la tracciabilità dei requisiti
  - La gestione delle modifiche ai requisiti
- La gestione del processo di RE:
  - La misurazione e il reporting del processo di RE
  - La misurazione e il reporting della qualità del prodotto
  - La gestione del workflow di RE
- La documentazione della conoscenza sui requisiti:
  - La condivisione dei requisiti
  - La creazione di una comprensione condivisa dei requisiti
- La modellazione dei requisiti
- La collaborazione in RE
- Il testing/simulazione dei requisiti

Spesso sono disponibili strumenti per supportare più attività tra quelle sopra menzionate. Per garantire che tutti i compiti di RE siano coperti in modo appropriato, potrebbero essere combinati strumenti differenti.

A volte, per documentare o gestire i requisiti, vengono utilizzati altri tipi di strumenti, ad esempio strumenti office e di defect management (issue tracking, gestione dei difetti).

Questi strumenti hanno delle limitazioni e dovrebbero essere utilizzati solo quando il processo di RE è sotto controllo e i requisiti sono allineati e abbastanza stabili.

## 7.2 Introduzione degli Strumenti (L2)

La selezione di uno strumento di RE non è differente dalla selezione di uno strumento per qualsiasi altro scopo. L'obiettivo, il contesto e i requisiti devono essere descritti prima che la selezione venga completata con successo [Fugg1993].

Uno strumento appropriato può essere selezionato solo dopo aver introdotto appropriate procedure e tecniche di RE. L'introduzione di uno strumento richiede chiare responsabilità e procedure di RE. Nel processo di introduzione di uno strumento di RE sono importanti i seguenti aspetti:

- Tenere conto di tutti i costi del ciclo di vita, oltre ai costi delle licenze
- Considerare le risorse necessarie
- Utilizzare progetti pilota per evitare rischi
- Valutare lo strumento secondo criteri definiti
- Istruire i dipendenti su come utilizzare lo strumento.



## Riferimenti Bibliografici

- [AnPC1994] Annie I. Antón, W. Michael McCracken, Colin Potts: Goal Decomposition and Scenario Analysis in Business Process Reengineering. CAiSE (Conference on Advanced Information Systems Engineering), 1994, 94–104.
- [BaCC2015] K. Baxter, C. Courage, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.
- [BiSp2003] K. Bittner, I. Spence: Use Case Modelling. Pearson Education, Boston, 2003.
- [Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organisational Implementation. Gower Publishing Ltd, Burlington, 2009.
- [CaDJ2014] D. Carrizo, O. Dieste, N. Juristo: Systematizing requirements elicitation technique selection. Information and Software Technology 2014, 56(6): 644–669.
- [Cock2001] A. Cockburn: Writing Effective Use Cases. Addison–Wesley, Boston 2001.
- [Cohn2004] M. Cohn: User Stories Applied – For Agile Software Development. Addison–Wesley, Boston, 2004.
- [Coop2004] A. Cooper: The Inmates Are Running the Asylum: Why High–Tech Products Drive Us Crazy and How to Restore the Sanity. Que, Indianapolis, 2004.
- [Davi2005] A. M. Davis: Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, New York, 2005.
- [Davi1993] A. M. Davis: Software Requirements – Objects, Functions, & States, 2nd edition, Prentice Hall, New Jersey, 1993.
- [DeMa1978] T. DeMarco: Structured Analysis and System Specification. Yourdon Press, New York, 1978.
- [Fugg1993] A. Fuggetta: A classification of CASE technology. IEEE Computer 1993, 26 (12): 25–38.
- [GiGr1993] T. Gilb, D. Graham: Software Inspection. Addison Wesley, Boston, 1993.
- [Glin2019] M. Glinz: Requirements Engineering I. Course Notes, University of Zurich, 2019. <https://www.ifi.uzh.ch/en/rerg/courses/archives/hs19/re-i.html#resources>. Ultima visita Luglio 2020.
- [Glin2020] M. Glinz: A Glossary of Requirements Engineering Terminology. Version 2.0. <https://www.ireb.org/en/downloads/#cpre-glossary>. Last visited July 2020.
- [GoFi1994] O. Gotel, A. Finkelstein: An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering, Colorado Springs, 1994. 94–101.
- [GoRu2003] R. Goetz, C. Rupp: Psychotherapy for System Requirements. 2nd IEEE International Conference on Cognitive Informatics (ICCI'03), London, 2003. 75–80.

- [GRL2020] Goal oriented Requirement Language. University of Toronto, Canada  
<https://www.cs.toronto.edu/km/GRL>. Ultima visita Maggio 2020.
- [Hare1988] D. Harel: On Visual Formalisms. Communications of the ACM 1988, 31 (5): 514–530.
- [HoSch2020] S. Hofer, H. Schwentner: Domain Storytelling — A Collaborative Modeling Method. Available from Leanpub, <http://leanpub.com/domainstorytelling>. Ultima visita Luglio 2020.
- [HuJD2011] E. Hull, K. Jackson, and J. Dick: Requirements Engineering. Springer, 3rd Ed, 2011.
- [ISO29148] ISO/IEC/IEEE 29148: Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, International Organization for Standardization, 2018.
- [ISO19650] ISO 19650: Organization and Digitization of Information about Buildings and Civil Engineering Works, including Building Information Modelling (BIM)– Information Management Using Building Information Modelling – Part 1 and 2, International Organization for Standardization, 2018.
- [ISO25010] ISO/IEC/IEEE 25010:2011: Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models. International Organization for Standardization, Geneva, 2011.
- [Jack1995] M. A. Jackson: Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices. Addison–Wesley, New York, 1995.
- [Jack1995b] M. Jackson: The World and the Machine. 17th International Conference on Software Engineering 1995 (ICSE 1995). 287–292.
- [KaeA1984] N. Kano et al.: Attractive quality and must–be quality. Journal of the Japanese Society for Quality Control 1984, 14(2): 39–48. (in Japanese)
- [KoTh2017] R. Kohavi, S. Thomke: The Surprising Power of Online Experiments – Getting the most out of A/B and other controlled tests. Harvard Business Review, Sept–Oct 2017: 74–82.
- [Kuma2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. John Wiley & Sons, Hoboken, 2013.
- [Laue2002] S. Lauesen: Software Requirements. Styles and Techniques. Addison–Wesley, Harlow, 2002.
- [Leff2011] D. Leffingwell: Agile Software Requirements, Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison–Wesley, Boston, 2011.
- [LiOg2011] J. Liedtka, T. Ogilvie: Designing for Growth: A Design Thinking Tool Kit For Managers. Columbia University Press, 2011.
- [LiSZ1994] H. Lichter, M. Schneider–Hufschmidt, H. Zullighoven: Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. IEEE Transactions on Software Engineering 1994, 20 (11): 825–832.

- [MFeA2019] D. Méndez Fernández, X. Franch, N. Seyff, M. Felderer, M. Glinz, M. Kalinowski, A. Volgelsang, S. Wagner, S. Bühne, K. Lauenroth: Do We Preach What We Practice? Investigating the Practical Relevance of Requirements Engineering Syllabi – The IREB Case. *CibSE 2019*: 476–487.
- [Moor2014] C. W. Moore: *The Mediation Process – Practical Strategies for Resolving Conflicts*, 4th edition. John Wiley & Sons, Hoboken, 2014.
- [MWHN2009] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak: Easy Approach to Requirements Syntax (EARS). 17th IEEE International Requirements Engineering Conference (RE'09), Atlanta, Georgia, 2009. 317–322.
- [OleA2018] K. Olsen et al.: *Certified Tester, Foundation Level Syllabus – Version 2018*. International Software Testing Qualifications Board, 2018.
- [OMG2013] Object Management Group: *Business Process Model and Notation (BPMN)*, version 2.0.2. OMG document formal/2013–12–09 <http://www.omg.org/spec/BPMN>. Ultima visita Luglio 2020.
- [OMG2017] Object Management Group: *OMG Unified Modeling Language (OMG UML)*, version 2.5.1. OMG document formal/2017–12–05. <https://www.omg.org/spec/UML/About-UML/>. Ultima visita Luglio 2020.
- [OMG2019] Object Management Group: *OMG Systems Modeling Language (OMG SysML™)*, Version 1.6. OMG Document formal/19–11–01. <https://www.omg.org/spec/SysML/>. Ultima visita Gennaio 2022
- [Osbo1979] A. F. Osborn: *Applied Imagination*, 3rd revised edition. Charles Scribner's Sons, New York, 1979.
- [RoRo2012] S. Robertson and J. Robertson: *Mastering the Requirements Process*, 3rd edition. Addison–Wesley, Boston, 2012.
- [Rupp2014] C. Rupp: *Requirements–Engineering und Management*, 6. Auflage. Hanser, München, 2014. (in German).
- [Sdsc2012] Stanford d.school: *An Introduction to Design Thinking*. Hasso Plattner Institute of Design, Stanford, 2012. <https://dschool-old.stanford.edu/groups/designresources/wiki/36873>. Ultima visita Luglio 2020.
- [vLam2009] Axel van Lamsweerde: *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Chichester: John Wiley & Sons, 2009.
- [Vole2020] Volere: *Requirements Resources*. <https://www.volere.org>. Ultima visita Luglio 2020.
- [WiBe2013] K. Wiegers and J. Beatty: *Software Requirements*, 3rd edition. Microsoft Press, Redmond, 2013.