



# Certified Professional for Requirements Engineering

Syllabus

Elicitazione dei Requisiti

Practitioner | Specialist

Anja Brand, Dominik Häußer  
Kim Lauenroth, Hans van Loenhoud,  
Patrick Steiger



## Termini di utilizzo:

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

© IREB e.V.

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

## Crediti e ringraziamenti

La versione 1.0 di questo Syllabus è stata creata da (in ordine alfabetico) Oliver Creighton, Dominik Häußler, Kim Lauenroth, Henriette Katharina Lingg, Thomas Mödl, Michael Richter, Chris Rupp, Dirk Schüpferling, Patrick Steiger e Malik Tayeh.

La versione 2.0 è una revisione importante, creata da (in ordine alfabetico) Dominik Häußler, Kim Lauenroth, Hans van Loenhoud, Anja Brand e Patrick Steiger.

Durante la revisione, il feedback è stato fornito da (in ordine alfabetico) Juliane Blechinger, Nikolaos Kaintantzis, Kostas Kolovos, Michael Richter, Stefan Sturm e Roger Wouterse.

Le revisioni sono state effettuate da Birgit Penzenstadler (contenuti) e Gareth Rogers (lingua).

Approvato per la pubblicazione in data 11 luglio 2018 dal Consiglio IREB su raccomandazione di Thorsten Weyer.

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

Su suggerimento della nostra comunità, la precedente certificazione di livello avanzato è stata suddivisa in due livelli, Practitioner e Specialist. Questa suddivisione è stata implementata nella versione 3.0.

Si ringraziano tutti coloro che hanno collaborato.

Copyright © 2012 – 2024 per questo Syllabus IREB Certified Professional for Requirements Engineering attribuito agli autori elencati sopra. I diritti sono stati trasferiti a IREB International Requirements Engineering Board e.V.

## Scopo del documento

Questo syllabus definisce gli obiettivi educativi e una sintesi dei contenuti educativi per le certificazioni Elicitazione dei requisiti Practitioner e Specialist, istituite da International Requirements Engineering Board (IREB).

Il Syllabus fornisce ai Training Provider la base per la creazione del proprio materiale per la formazione. Gli studenti possono utilizzare il Syllabus per prepararsi all'esame.

### Contenuto del Syllabus

Il modulo Elicitazione dei Requisiti si rivolge a professionisti con profili professionali nel campo del *Requirements Engineering (Ingegneria dei Requisiti)*, della *Business Analysis*, del *Business Engineering (Ingegneria del Business)*, della *progettazione organizzativa*, ecc. che desiderano approfondire le proprie conoscenze e competenze nell'area dell'elicitazione dei requisiti.

### Obiettivo del Syllabus

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

Questo Syllabus non si basa su uno specifico approccio allo sviluppo del software e su un particolare modello di processo associato, che prescriva la pianificazione, il controllo e la sequenza di applicazione nella pratica dei concetti e delle tecniche di Requirements Engineering trattati. L'obiettivo non è quello di enfatizzare un approccio specifico per il Requirements Engineering o per il Software Engineering in generale.

Definisce cosa costituisce la conoscenza del Requirements Engineer, ma non le interfacce precise con altre discipline e processi del Software Engineering.

### Livello di dettaglio

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

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

### Obiettivi formativi/ livelli cognitivi di conoscenza

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

- **L1: Conoscere** (descrivere, enumerare, caratterizzare, riconoscere, nominare, ricordare, ...) – Il candidato è in grado di ricordare o recuperare materiale appreso in precedenza.
- **L2: Capire** (spiegare, interpretare, completare, riassumere, giustificare, classificare, confrontare, ...) – Il candidato è in grado di cogliere/costruire il significato da materiale o situazioni specifiche.

- **L3: Applicare** (specificare, scrivere, progettare, sviluppare, implementare, ...) – Il candidato è in grado di applicare la conoscenza e le competenze in determinate situazioni.
- **L4: Analizzare** (indagare, trarre conclusioni, fornire argomentazioni, ...) – Il candidato è in grado di analizzare un problema dato, argomentare ciò che dovrebbe/potrebbe essere fatto, scomporre il problema in parti, applicare il pensiero critico, argomentare su cause ed effetti.
- **L5: Valutare** (criticare, giudicare) – Il candidato è in grado di fornire una critica ben argomentata di un artefatto specifico; esprimere un giudizio approfondito su un dato caso. Si noti che un obiettivo formativo al livello cognitivo Ln contiene anche elementi di tutti i livelli cognitivi inferiori (da L1 a Ln-1).

Esempio:

Un obiettivo formativo del tipo "Applicare la tecnica di RE xyz" è al livello cognitivo (L3).

Tuttavia, la capacità di applicare richiede che gli studenti conoscano prima la tecnica di RE xyz (L1) e capiscano per cosa viene utilizzata la tecnica (L2).



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

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

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

### Struttura del syllabus

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

Esempio:

Capitolo 2: Sorgenti dei requisiti (L3)

Durata: 2.5 ore

Termini: stakeholder, sorgente dei requisiti, gestione delle relazioni (relationship management), utente, Persona

Questo esempio mostra che il Capitolo 2 contiene obiettivi formativi di livello L3, e che due ore e mezza sono ritenute necessarie per insegnare il materiale di questo capitolo.

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

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

Esempio: EO 2.1.1

L'esempio mostra che l'obiettivo formativo EO 2.1.1 è descritto nel sottocapitolo 2.1.

## L'Esame

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

- Elicitazione dei Requisiti Practitioner
- Elicitazione dei Requisiti Specialist

L'esame per il conseguimento della certificazione Elicitazione dei Requisiti – Practitioner consiste in un **esame a scelta multipla**.

L'esame per il conseguimento della certificazione Elicitazione dei Requisiti – Specialist consiste in un **compito scritto**.

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

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

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

- Verifica tutti gli obiettivi formativi del Syllabus. Tuttavia, per gli obiettivi formativi relativi ai livelli cognitivi L4 e L5, le domande di esame sono limitate agli elementi relativi ai livelli cognitivi da L1 a L3.
- Può essere svolto subito dopo un corso, ma anche indipendentemente da esso (ad esempio, a distanza o presso un test center).

Il **compito scritto** per la **certificazione Specialist**

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

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

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

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

Un elenco degli Organismi di Certificazione autorizzati da IREB è disponibile sul sito web <https://www.ireb.org>.

### Storia delle Versioni

Versione	Data	Commento
3.1.0	1 Marzo 2024	Pubblicazione della versione italiana del syllabus basata sulla versione inglese del curriculum 3.1.0
3.2.0	18 Giugno 2024	I Livelli Cognitivi di Conoscenza sono stati corretti. EO 1.3.4, 4.3.1 e 4.3.2 modificati, aggiunti i nuovi EO 3.1.5 e 3.2.5.

# Contenuti

1	Un framework per strutturare e gestire l'elicitazione dei requisiti e la risoluzione dei conflitti (L4) .....	9
1.1	L'ambito di elicitazione dei requisiti e di risoluzione dei conflitti nel Requirements Engineering (L2) .....	9
1.2	Fattori rilevanti per l'approccio di pianificare le attività di elicitazione dei requisiti e di risoluzione dei conflitti (L2) ....	10
1.3	Pianificare ed eseguire le attività di elicitazione dei requisiti ed i conflitti (L4) .....	10
1.4	Pattern (Modelli) di processo (L2) .....	11
2	Sorgenti dei requisiti (L3) .....	13
2.1	Fondamenti sulle sorgenti di requisiti (L3) .....	13
2.2	Identificare, classificare e gestire gli stakeholder (L3) .....	14
2.2.1	Identificare e selezionare gli stakeholder come sorgenti dei requisiti (L3).	14
2.2.2	Gestione delle relazioni (relationship management) con gli stakeholder (L3).	15
2.2.3	Schema di documentazione per gli stakeholder coinvolti (L3) .....	15
2.2.4	L'utenti come gruppo di stakeholder speciali (L3) .....	16
2.3	Identificare, classificare e gestire i documenti (L3) .....	16
2.3.1	Identificare e selezionare i documenti come sorgenti dei requisiti (L3) .....	17
2.3.2	Schema di documentazione per i documenti (L3) .....	17
2.4	Identificare, classificare, gestire i sistemi (L3) .....	18
2.4.1	Identificare e selezionare i sistemi come sorgenti dei requisiti (L3) .....	18
2.4.2	Schema di documentazione per i sistemi (L3) .....	19
3	Tecniche di elicitazione (L4) .....	20
3.1	Tecniche di raccolta (L4) .....	20
3.1.1	Tecniche di indagine (L3) .....	20
3.1.2	Tecniche di osservazione (L3) .....	21
3.1.3	Tecniche di collaborazione (L3) .....	22
3.1.4	Tecniche basate sugli artefatti (L3) .....	23

3.2	Tecniche di progettazione e generazione di idee. (L4)	23
3.2.1	Brainstorming (L3)	24
3.2.2	Tecniche per analogia (L2)	24
3.2.3	Prototipazione (L3)	24
3.2.4	Scenari e storyboard (L3)	25
3.3	Strumenti di riflessione (thinking tool) (L2)	25
3.3.1	Pensare secondo i livelli di astrazione (L2)	25
3.3.2	Pensare in termini di problemi e obiettivi (L2)	25
3.3.3	Evitare gli effetti trasformativi (L2)	26
3.3.4	Pensare in termini di modelli (L2)	26
3.3.5	Mind mapping (mappe mentali) (L3)	27
3.4	Esempio di strutturazione delle tecniche di elicitazione: attributi (L2)	27
4	Risoluzione del conflitto (L4)	31
4.1	Identificazione del conflitto (L2)	31
4.2	Analisi del conflitto (L3)	32
4.3	Risoluzione del conflitto (L4)	33
4.4	Documentazione della risoluzione del conflitto (L2)	34
5	Competenze del Requirements Engineer (L3)	35
5.1	Competenze richieste nelle aree di elicitazione (L2)	35
5.2	Teoria della comunicazione e modelli di comunicazione (L2)	35
5.3	Auto-riflessione sulle competenze personali nell'elicitazione dei requisiti (L3)	36
5.4	Opportunità di sviluppo personale (L2)	36
5.5	Apprendere dall'esperienza precedente (L2)	38
6	Riferimenti bibliografici e ulteriori letture	39



# 1 Un framework per strutturare e gestire l'elicitazione dei requisiti e la risoluzione dei conflitti (L4)

Durata: 1.5 ore

Termini: Attività di elicitazione, attività di risoluzione dei conflitti, tecnica, pattern (modello) di processo

## Obiettivi Formativi

- EO 1.1.1 Comprendere l'ambito di elicitazione dei requisiti e di risoluzione dei conflitti nel Requirements Engineering
- EO 1.2.1 Comprendere le sfide di pianificare l'elicitazione dei requisiti e la risoluzione dei conflitti
- EO 1.2.2 Comprendere i fattori rilevanti per l'approccio di pianificare le attività di elicitazione dei requisiti e di risoluzione dei conflitti
- EO 1.3.1 Applicare la struttura informativa per le attività di elicitazione dei requisiti e di risoluzione dei conflitti
- EO 1.3.2 Comprendere la differenza tra attività di elicitazione dei requisiti e di risoluzione dei conflitti a breve e a lungo termine
- EO 1.3.3 Comprendere l'importanza di una fase di setup per l'elicitazione dei requisiti e la risoluzione dei conflitti
- EO 1.3.4 Analizzare le attività di elicitazione e di risoluzione dei conflitti per selezionare e utilizzare quelle giuste
- EO 1.4.1 Comprendere l'importanza di adattare le tecniche di elicitazione dei requisiti e di risoluzione dei conflitti a specifici contesti
- EO 1.4.2 Comprendere il concetto di pattern di processo

## 1.1 L'ambito di elicitazione dei requisiti e di risoluzione dei conflitti nel Requirements Engineering (L2)

In accordo con la definizione di Requirements Engineering specificata in [PoRu2015], l'obiettivo dell'elicitazione dei requisiti e della risoluzione dei conflitti è "conoscere i requisiti rilevanti", "raggiungere un consenso tra gli stakeholder su questi requisiti" e "comprendere [...] i desideri e le esigenze degli stakeholder".

Durante l'elicitazione, 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. Questo include l'identificazione di queste sorgenti, la comprensione della natura e dell'importanza dei differenti tipi di requisiti, e l'applicazione di tecniche appropriate per la loro elicitazione. Un punto importante nell'elicitazione è la trasformazione delle richieste, dei desideri e delle aspettative implicite in requisiti espliciti [ISO29148].

Durante l'elicitazione, spesso si identificano requisiti in conflitto tra loro provenienti da sorgenti differenti. Questi conflitti devono essere risolti, per poter creare un insieme unico, consistente e concordato che possa servire come input per uno sviluppo, una manutenzione e un utilizzo efficienti di un sistema efficace.

## 1.2 Fattori rilevanti per l'approccio di pianificare le attività di elicitazione dei requisiti e di risoluzione dei conflitti (L2)

La letteratura sulla stima del software [McCo2006] e i risultati dalla realtà industriale attribuiscono una grande responsabilità alla disciplina del Requirements Engineering per il raggiungimento degli obiettivi dello sviluppo complessivo. Dal punto di vista del Requirements Engineering, una parte significativa di questa responsabilità deve essere attribuita all'elicitazione dei requisiti e alla risoluzione dei conflitti.

Entrambi richiedono uno specifico approccio di pianificazione a causa delle seguenti sfide:

- L'elicitazione dei requisiti non può essere pianificata solo in base alle dimensioni attese del risultato, poiché all'inizio dell'elicitazione non sono disponibili aspettative realistiche.
- I conflitti tra requisiti non possono essere pianificati o previsti. Il Requirements Engineer deve reagire al conflitto non appena si presenta.

Di conseguenza, è consigliabile evitare una pianificazione dettagliata e definire invece un piano upfront (nelle prime fasi di progetto) di alto livello per le attività di elicitazione dei requisiti e di risoluzione dei conflitti. La pianificazione e l'esecuzione dell'elicitazione dei requisiti e della risoluzione dei conflitti dovrebbero essere eseguite in modo simile a un progetto di ricerca. Questo significa che il piano viene rivisto iterativamente man mano che le attività procedono e che diventano disponibili maggiori informazioni.

## 1.3 Pianificare ed eseguire le attività di elicitazione dei requisiti ed i conflitti (L4)

Sebbene l'elicitazione e la risoluzione dei conflitti richiedano uno specifico approccio di pianificazione, la pianificazione e l'esecuzione non possono essere trattate separatamente dalle altre attività di sviluppo del sistema. Per la definizione di un framework di pianificazione, si assume che ogni sviluppo che includa le attività di elicitazione e risoluzione dei conflitti utilizzi un qualsiasi tipo di piano per strutturare l'effort e i suoi compiti. Man mano che il lavoro procede, il piano deve essere mantenuto e aggiornato.

Due tipi di attività possono essere incluse in qualsiasi tipo di piano:

- *Attività di elicitazione*: Identificazione delle sorgenti dei requisiti ed elicitazione dei requisiti.
- *Attività di risoluzione dei conflitti*: Azioni necessarie a risolvere i conflitti tra requisiti e arrivare a un unico insieme di requisiti concordato.

Un'attività di elicitazione dovrebbe fornire le seguenti informazioni: *obiettivo dell'elicitazione*, *qualità del risultato* desiderata, *sorgenti* coinvolte e *tecnica di elicitazione* selezionata.

Un'attività di risoluzione dei conflitti dovrebbe fornire le seguenti informazioni: *requisito/requisiti* impattati, *sorgente/sorgenti* impattate, *tecnica di risoluzione del conflitto* selezionata e *risultato ottenuto*.

Oltre alle informazioni relative all'elicitazione e alla risoluzione dei conflitti, entrambe le attività possono fornire informazioni sulla gestione delle tempistiche e delle risorse.

In generale, si possono distinguere tre diversi gruppi di attività di elicitazione e risoluzione dei conflitti:

- Gruppo1 – Attività di elicitazione e risoluzione dei conflitti *eseguite*
- Gruppo2 – Attività di elicitazione e risoluzione dei conflitti *a breve termine*
- Gruppo3 – Attività di elicitazione e risoluzione dei conflitti *a lungo termine*

Durante lo sviluppo, il gruppo di attività eseguite crescerà, man mano che verranno eseguite le attività a breve termine. Le attività a lungo termine saranno dettagliate e diventeranno attività a breve termine, oppure saranno raffinate da diverse attività a breve termine, oppure potranno essere abbandonate del tutto se non hanno più senso. Si raccomanda di distinguere tra la *fase di setup* e la *fase di esecuzione* delle attività di elicitazione e risoluzione dei conflitti.

Le seguenti linee guida per la *fase di setup* sono importanti:

- Ottenere una overview della situazione del progetto (il business case)
- Determinare gli obiettivi di elicitazione
- Pianificare l'analisi sistematica del contesto di sistema
- Pianificare l'identificazione sistematica di (diversi tipi di) sorgenti dei requisiti
- Considerare i pattern (modelli) di processo rilevanti per definire le attività
- Fornire tempo e budget per le attività di risoluzione dei conflitti

Le seguenti linee guida per la *fase di esecuzione* sono importanti:

- Considerare l'elicitazione e la risoluzione dei conflitti come attività time-boxed
- Verificare il piano dopo ogni attività (e rivederlo se necessario)
- Schedulare in modo preventivo, utilizzando attività a breve e a lungo termine
- Incorporare tempi morti (slack) per lasciare il tempo alla creatività e agli imprevisti
- Parallelizzare attività indipendenti
- Combinare le attività di elicitazione che indirizzano la stessa sorgente dei requisiti
- Cercare i conflitti per reagire secondo una strategia concordata

Inoltre, è buona pratica aggiungere una *fase di chiusura* che si focalizzi sull'apprendimento dal progetto e sul miglioramento delle competenze dei partecipanti al progetto. Le linee guida sono riportate in 5.

## 1.4 Pattern (Modelli) di processo [L2]

Ogni progetto è un evento unico, quindi non esiste un approccio generale che si adatti a tutte le esigenze di elicitazione. In questo Syllabus, il concetto di *pattern (modello) di processo* viene utilizzato per identificare le similarità tra specifiche situazioni che possono essere utilizzate come linee guida per le attività di elicitazione. Se un singolo pattern non è adatto, è possibile applicare una combinazione o una sequenza di pattern.

Il concetto di pattern è stato originariamente sviluppato in un contesto architeturale [AII1977]. In un contesto di elicitazione, un pattern descrive un metodo riutilizzabile per l'elicitazione dei requisiti in un determinato ambito (p.e. dominio, situazione di progetto).

Il pattern contiene informazioni sul metodo generale (fasi, sequenza di attività) e fornisce una guida per le attività di elicitazione, che include la definizione degli obiettivi di elicitazione, la selezione delle tecniche, la definizione della qualità dei risultati e le possibili sorgenti dei requisiti.

I pattern evolvono in un contesto specifico. Consideriamo tutti i pattern che potenzialmente portano a requisiti nuovi o migliorati. Possono anche includere altre attività (p.e. testing, progettazione, risoluzione dei conflitti).

Esempi di pattern di processo sono:

- Sviluppo a cascata/milestone-driven (guidato dalle milestone)
- Sviluppo software Lean
- Sviluppo software Agile
- Progettazione human-centered
- Design thinking
- Sviluppo di sistemi embedded
- Manutenzione di sistema

Il Requirements Engineer dovrebbe cercare in modo attivo i pattern rilevanti per la propria situazione. Tenere presente che:

- I pattern di processo sono buone pratiche tratte dalla letteratura e dal lavoro pratico, che forniscono una overview che può essere utilizzata come punto di partenza per definire le attività di elicitazione in una situazione analoga.
- In genere, le informazioni fornite non sono sufficienti per un'esecuzione immediata del processo. L'analisi delle somiglianze e delle differenze tra l'ambito del pattern e la situazione reale aiuta a identificare un approccio appropriato e a selezionare le tecniche utili.
- La lista dei pattern sopra riportata non è completa e neanche esaustiva. Inoltre, i pattern possono e spesso dovrebbero essere combinati in vari modi.
- i Requirements Engineer con esperienza sono incoraggiati a sviluppare e condividere i propri pattern.

## 2 Sorgenti dei requisiti (L3)

Durata: 2.5 ore

Termini: stakeholder, sorgente dei requisiti, gestione delle relazioni (relationship management), utente, Persona

### Obiettivi Formativi

- EO 2.1.1 Comprendere l'importanza dell'identificazione sistematica e pragmatica delle sorgenti dei requisiti nel contesto di sistema
- EO 2.2.1.1 Comprendere i gruppi di stakeholder tipici
- EO 2.2.1.2 Applicare l'identificazione e la selezione sistematica degli stakeholder
- EO 2.2.2.1 Applicare la gestione delle relazioni con gli stakeholder per prevenire e risolvere i problemi con gli stakeholder
- EO 2.2.3.1 Applicare uno schema di documentazione per gli stakeholder coinvolti
- EO 2.2.4.1 Comprendere l'importanza dell'utente come stakeholder
- EO 2.2.4.2 Applicare le Personas
- EO 2.3.1.1 Comprendere i documenti candidati tipici
- EO 2.3.1.2 Applicare l'identificazione e la selezione sistematica dei documenti
- EO 2.3.2 Applicare uno schema di documentazione per i documenti considerati
- EO 2.4.1.1 Comprendere le tipiche tipologie dei sistemi
- EO 2.4.1.2 Applicare l'identificazione e la selezione sistematica dei sistemi
- EO 2.4.2 Applicare uno schema di documentazione per i sistemi considerati

### 2.1 Fondamenti sulle sorgenti di requisiti (L3)

La qualità e la completezza dei requisiti dipendono in larga misura dalle sorgenti dei requisiti coinvolte. Dimenticare una sorgente rilevante porterà a una comprensione incompleta dei requisiti. Durante lo sviluppo, il Requirements Engineer deve identificare e coinvolgere tutte le sorgenti dei requisiti rilevanti. Come spiegato nel Syllabus CPRE Foundation Level [IREB2022], le tre tipologie più importanti di sorgenti dei requisiti sono gli stakeholder, i documenti e i sistemi. L'identificazione delle sorgenti dei requisiti è un processo iterativo e ricorsivo [ISO29148], che richiede una rivalutazione costante.

Il Requirements Engineer può scegliere tra due approcci differenti per l'identificazione delle sorgenti dei requisiti:

- Identificazione *pragmatica*: Il Requirements Engineer utilizza la propria conoscenza ed esperienza del progetto e del suo contesto (p.e. la conoscenza di dominio) per identificare gli stakeholder, i documenti e i sistemi rilevanti.
- Identificazione *sistematica*: Il Requirements Engineer applica una strategia specifica per identificare le possibili sorgenti dei requisiti, definendo specifiche attività di elicitazione che si focalizzano sull'identificazione delle sorgenti dei requisiti.

L'identificazione pragmatica e l'identificazione sistematica sono complementari e generano dei rischi se utilizzate singolarmente. E' fortemente raccomandato di utilizzare un mix di entrambi per identificare le sorgenti dei requisiti in modo efficiente ed efficace.

## 2.2 Identificare, classificare e gestire gli stakeholder (L3)

Nel glossario CPRE [Glin2024], uno stakeholder è definito come "una persona o organizzazione che influenza i requisiti di un sistema o che è impattata da quel sistema". L'influenza può anche essere indiretta. Ad esempio, alcuni stakeholder possono dover seguire le istruzioni specificate dai loro manager o dalle organizzazioni."

### 2.2.1 Identificare e selezionare gli stakeholder come sorgenti dei requisiti (L3)

Il Requirements Engineer deve identificare tutti gli stakeholder rilevanti per lo sviluppo del sistema.

Un elenco non esaustivo dei ruoli degli stakeholder include:

- Utenti diretti del sistema
- Business manager/process manager (manager di processo)
- Singoli clienti, clienti rappresentanti di un'organizzazione
- Avversari e competitor (concorrenti)
- Staff IT
- Istituzioni governative e enti legali

Le sorgenti potenziali per i ruoli di stakeholder rilevanti sono:

- Checklist di gruppi e ruoli dei tipici stakeholder (si veda sopra)
- Strutture organizzative (p.e. organigrammi dell'azienda che utilizzerà il sistema pianificato)
- Documentazione dei processi di business (p.e. i processi di business che devono essere supportati dal sistema da sviluppare)
- Schemi di categorizzazione degli stakeholder (p.e. il modello a cipolla di Alexander [AlBe2009] o la mappa generica degli stakeholder di Robertsons [RoRo2013])

Durante l'*identificazione pragmatica* degli stakeholder, i Requirements Engineer usano la loro attuale conoscenza ed esperienza del contesto (p.e. il dominio) per identificare i ruoli degli stakeholder rilevanti e i loro rappresentanti (gli stakeholder).

Durante l'*identificazione sistematica degli stakeholder*, il Requirements Engineer definisce gli obiettivi di elicitazione con un focus dedicato all'identificazione degli stakeholder.

Dovrebbero essere considerati due diversi tipi di obiettivi di elicitazione:

- *Focalizzato sulle informazioni*: identificare i singoli stakeholder richiesti per determinate informazioni
- *Focalizzato sugli stakeholder*: identificare i singoli stakeholder che rappresentano determinati ruoli di stakeholder

Gli stakeholder identificati inizialmente sono sorgenti utili per identificare sorgenti aggiuntive.

## 2.2.2 Gestione delle relazioni (relationship management) con gli stakeholder (L3)

I problemi con gli stakeholder nascono generalmente se i diritti e gli obblighi di uno stakeholder, rispetto al sistema proposto o al progetto attuale, non sono chiari, o se le esigenze dello stakeholder non sono sufficientemente indirizzate. La gestione delle relazioni con gli stakeholder è un modo efficace per rilevare i problemi con gli stakeholder.

[Bour2015] raccomanda il *cerchio degli stakeholder* per una gestione efficace delle relazioni con gli stakeholder. Consiste di cinque passi:

1. Identificazione di tutti gli stakeholder
2. Prioritizzazione per determinare chi è importante
3. Visualizzazione per comprendere l'intera comunità di stakeholder
4. Coinvolgimento attraverso una comunicazione efficace
5. Monitoraggio dell'effetto della comunicazione

La gestione attiva delle relazioni con gli stakeholder [Bour2009] definisce in modo esplicito i diritti e gli obblighi di uno stakeholder rispetto allo sviluppo del sistema proposto. A seconda della natura dello sviluppo, questo può essere formulato come un accordo con gli stakeholder coinvolti.

## 2.2.3 Schema di documentazione per gli stakeholder coinvolti (L3)

Il Syllabus CPRE Foundation Level [IREB2022] definisce quali sono le minime informazioni sugli stakeholder che dovrebbero essere documentate. Si dovrebbero considerare anche le informazioni sulla classificazione degli stakeholder e gli attributi specifici del progetto.

Secondo [AlBe2009], gli stakeholder possono essere classificati in base al loro impatto sul sistema nuovo o modificato:

- Stakeholder del *sistema stesso*: direttamente impattati dal sistema nuovo o modificato (utenti, amministratori, operatori, ...)
- Stakeholder nel *contesto circostante*: indirettamente impattati dal sistema nuovo o modificato (business manager, proprietari del progetto, sponsor, clienti, ...)
- Stakeholder dal *contesto più ampio*: hanno una relazione indiretta con il sistema nuovo o modificato, o con il progetto di sviluppo (enti legali, organismi di definizione degli standard, organizzazioni (non)-governative, competitor, staff IT)

Può anche essere utile documentare ulteriori informazioni rilevanti per lo specifico sviluppo del sistema.

Nel definire le informazioni aggiuntive, occorre considerare le circostanze specifiche del contesto attuale. I possibili fattori di influenza sono:

- *Rilevanza pubblica*: In un contesto di elevata rilevanza pubblica, può essere utile documentare quanto uno stakeholder conosce o può influenzare l'opinione pubblica.

- *Criticità dei tempi*: In un contesto con tempistiche molto strette, la disponibilità o il tempo di risposta di uno stakeholder possono essere informazioni molto importanti quando si devono prendere decisioni critiche.

Durante lo sviluppo, tutte le informazioni sugli stakeholder devono essere continuamente aggiornate e adattate alle specifiche circostanze.

Alcune forme di documentazione comunemente utilizzate sono la tabella degli stakeholder, il database degli stakeholder e la mappa mentale (mind map) degli stakeholder.

#### 2.2.4 L'utenti come gruppo di stakeholder speciali (L3)

In linea di principio, ogni sistema avrà dei possibili utenti. Tuttavia, non tutti i sistemi hanno un'interazione diretta con l'uomo: alcuni rilasciano le loro funzionalità attraverso altri sistemi. Per i sistemi interattivi con un'interfaccia umana, tutti gli utenti diretti del sistema sono di grande interesse per il Requirements Engineer.

Gli utenti in-house (interni all'azienda, conosciuti e coinvolti individualmente) sono significativamente differenti dagli utenti esterni (esterni all'azienda, generalmente non conosciuti individualmente e non direttamente coinvolti; p.e. acquirenti di prodotti di consumo).

Di solito, il numero di (potenziali) utenti non consente di coinvolgere tutti gli individui nel processo di elicitazione. Per questo motivo, gli utenti effettivi possono essere aggregati in user group, sulla base dell'analisi degli utenti o della conoscenza del dominio di altri stakeholder.

Un modo comune per rappresentare gli user group è l'uso di Personas [Coop2004]. Le Personas sono individui fittizi, che rappresentano user group tipici di sistema con esigenze, valori e abitudini simili. Le Personas sono modellate a partire dai dati raccolti sugli utenti reali attraverso la *ricerca sugli utenti* [BaCC2015]. Se non sono (ancora) disponibili dati rilevanti della ricerca sugli utenti, è possibile creare delle *Personas provvisorie*, chiamate anche *Personas ad-hoc* [CRCN2014].

Gli user group o le Personas devono essere prioritizzati per definire gli user group/Personas principali e secondari. Il sistema, in particolare la sua interfaccia utente, sarà ottimizzato per lo user group principale.

Il concetto di User Experience (UX) indirizza soprattutto la creazione di una grande esperienza per gli utenti. Una definizione di User Experience è fornita da uno standard ISO. [ISO9241-210] definisce la User Experience come "le percezioni e le risposte di una persona che si ottengono dall'utilizzo o dall'uso previsto di un prodotto, sistema o servizio". Conoscere le idee e i principi della User Experience è importante per lo sviluppo di sistemi interattivi.

### 2.3 Identificare, classificare e gestire i documenti (L3)

I documenti sono un'altra sorgente importante per i requisiti. Vengono utilizzati per trasferire concetti tra gli esseri umani nel tempo e a distanza.



### 2.3.1 Identificare e selezionare i documenti come sorgenti dei requisiti (L3)

I possibili tipi di documenti utilizzati come sorgenti dei requisiti sono:

- Standard tecnici, normative, regolamenti interni
- Documenti dei requisiti (p.e. di sistemi competitor)
- Manuali utente (p.e. di sistemi competitor)
- Documenti strategici
- Documentazione degli obiettivi
- Documentazione dei processi di business

Durante l'*identificazione pragmatica* dei documenti, il Requirements Engineer utilizza la propria conoscenza ed esperienza attuale del contesto (p.e. il dominio) per identificare i documenti e i tipi di documenti rilevanti.

Durante l'*identificazione sistematica* dei documenti, il Requirements Engineer definisce gli obiettivi di elicitazione con un focus dedicato all'identificazione dei documenti. Devono essere considerati due tipi differenti di obiettivi di elicitazione:

- *Focalizzato sulle informazioni*: trovare documenti per determinate informazioni richieste
- *Focalizzato sui documenti*: trovare documenti di determinati tipi ritenuti rilevanti per lo sviluppo

Durante l'*identificazione sistematica* dei documenti, il Requirements Engineer può:

- Cercare i rappresentanti di categorie di documenti tipici
- Cercare nei documenti già identificati i riferimenti verso altri documenti eventualmente rilevanti
- Chiedere agli stakeholder già identificati la documentazione rilevante
- Cercare la documentazione sui sistemi rilevanti già identificati (si veda 2.4)

Per decidere se un documento è rilevante come sorgente dei requisiti, il Requirements Engineer deve stabilire criteri specifici.

### 2.3.2 Schema di documentazione per i documenti (L3)

Relativamente ai documenti che sono potenzialmente da utilizzare come sorgenti di requisiti, dovrebbero essere memorizzate almeno le seguenti informazioni:

- Titolo del documento
- Dove è memorizzato il documento
- Versione del documento
- Breve descrizione (quale tipo di informazione può fornire il documento)
- Rilevanza

In base al contesto, possono essere rilevanti anche informazioni aggiuntive.

I documenti hanno sempre una certa relazione con gli stakeholder, che può anche essere memorizzata, p.e.:

- Gli stakeholder che ritengono il documento rilevante
- Autore o organizzazione che ha emesso il documento
- Organizzazioni che utilizzano il documento nei loro processi
- Organizzazioni coinvolte nella verifica dell'aderenza

Il Requirements Engineer deve mantenere aggiornate le informazioni sulla documentazione. Questo include anche riconsiderare se sono diventati rilevanti ulteriori documenti aggiuntivi, o se i documenti identificati in precedenza hanno perso rilevanza. Dovrebbe essere prestata una particolare attenzione alle modifiche, agli aggiornamenti e agli aggiornamenti del version number (version numbering).

## 2.4 Identificare, classificare, gestire i sistemi (L3)

Nel contesto (sia in modo diretto che in modo più ampio) di un sistema, altri sistemi possono essere identificati come sorgenti di requisiti.

### 2.4.1 Identificare e selezionare i sistemi come sorgenti dei requisiti (L3)

I possibili tipi di sistemi utilizzati come sorgenti dei requisiti sono:

- Sistemi con cui è definita un'interfaccia, compresi i sistemi legacy
- Sistemi che condividono una piattaforma / ambiente / ecosistema
- Sistemi dei competitor
- Sistemi con dati, funzionalità o interfacce utente simili
- Sistemi predecessori da sostituire
- Sistemi futuri (in costruzione o anche solo pianificati)

Durante l'*identificazione pragmatica* dei sistemi, il Requirements Engineer utilizza la propria attuale conoscenza ed esperienza sul progetto e sul contesto (p.e. il dominio) per identificare i sistemi rilevanti e i tipi di sistemi.

Durante l'*identificazione sistematica* dei sistemi, il Requirements Engineer definisce gli obiettivi di elicitazione con un focus dedicato all'identificazione dei sistemi. Si devono considerare due diversi tipi di obiettivi di elicitazione:

- *Focalizzato sull'informazione*: identificare sistemi che contengano determinate informazioni richieste
- *Focalizzato sui sistemi*: identificare sistemi dei tipi considerati rilevanti per il progetto di sviluppo

Durante l'*identificazione sistematica* dei sistemi, il Requirements Engineer può:

- Utilizzare la documentazione sul contesto di sistema
- Chiedere agli stakeholder già identificati informazioni sui sistemi rilevanti
- Cercare nei documenti già identificati le informazioni sui sistemi rilevanti

- Utilizzare tecniche di generazione di idee per identificare possibili sistemi analoghi
- Condurre ricerche di mercato per identificare i sistemi dei competitor
- Considerare i sistemi legacy

## 2.4.2 Schema di documentazione per i sistemi (L3)

I sistemi utilizzati come sorgente dei requisiti devono essere documentati con almeno le seguenti informazioni:

- Nome del sistema
- Tipo di sistema (p.e. sistema competitor, sistema predecessore, sistema con un'interfaccia, ...)
- Una breve descrizione di dati, funzionalità, processi, gruppi utenti, ...

In base al contesto, possono essere rilevanti informazioni aggiuntive.

Dovrebbe essere prestata un'attenzione speciale ai sistemi che si interfacciano direttamente. Questi possono essere classificati come:

- Sorgenti dei dati, che forniscono i dati
- Destinatari di dati, che utilizzano i dati
- Sistemi a supporto come un sistema operativo (OS) o un DataBase Management System (DBMS)

I sistemi hanno sempre una specifica relazione con gli stakeholder, che può anche essere memorizzata, p.e.:

- Stakeholder/Organizzazioni che utilizzano il sistema in modo diretto o indiretto nei loro processi
- Stakeholder/Organizzazioni che gestiscono il sistema
- Stakeholder/Organizzazioni che progettano, sviluppano o commercializzano il sistema
- Stakeholder/Organizzazioni che si occupano della manutenzione del sistema, offrono supporto o formazione
- Organizzazioni che osservano il sistema (p.e. enti governativi, ONG)

Le informazioni sui sistemi sono generalmente presenti nei documenti. Questi documenti dovrebbero essere gestiti separatamente come sorgenti dei requisiti (si veda 2.3).

Il Requirements Engineer deve mantenere aggiornata la documentazione dei potenziali sistemi sorgente. Questo include anche riconsiderare se sono diventati rilevanti ulteriori sistemi aggiuntivi, o se i sistemi identificati in precedenza hanno perso rilevanza. Dovrebbe essere prestata una particolare attenzione alle modifiche, agli aggiornamenti e agli aggiornamenti del version number (version numbering).

## 3 Tecniche di elicitazione (L4)

Durata: 8.0 ore

Termini: tecnica di elicitazione, attributo, classificazione, strumenti di riflessione (thinking tool)

### Obiettivi Formativi

- EO 3.1 Comprendere la differenza tra le tecniche di raccolta, tecniche di progettazione e generazione di idee e tecniche di supporto/strumenti di riflessione (thinking tool)
- EO 3.1.1 Applicare l'intervista e il questionario come esempi di tecniche di indagine
- EO 3.1.2 Applicare l'osservazione sul campo, l'apprendimento e l'indagine contestuale come esempi di tecniche di osservazione
- EO 3.1.3 Applicare i workshop e il Requirements Engineering crowd-based come esempi di tecniche di collaborazione
- EO 3.1.4 Applicare l'archeologia dei sistemi, la lettura prospective-based e il riutilizzo dei requisiti come esempi di tecniche basate sugli artefatti
- EO 3.1.5 Analizzare le tecniche di raccolta per selezionare quelle più adeguate
- EO 3.2 Applicare le precondizioni per la creatività
- EO 3.2.1 Applicare il brainstorming come esempio di tecniche di generazione di idee
- EO 3.2.2 Comprendere le tecniche per analogia come esempi di tecniche di generazione di idee
- EO 3.2.3 Applicare la prototipazione come esempio di tecniche di progettazione
- EO 3.2.4 Applicare scenari e storyboard come esempi di tecniche di progettazione
- EO 3.2.5 Analizzare le tecniche di progettazione e di generazione di idee per selezionare quelle più adatte.
- EO 3.3.1 Comprendere e utilizzare i livelli di astrazione, i problemi e gli obiettivi, i modelli, gli effetti trasformazionali e il mind-mapping come esempi di strumenti di riflessione
- EO 3.4 Comprendere gli attributi delle tecniche di elicitazione come approccio fondamentale per strutturare le tecniche di elicitazione

Questo capitolo distingue tra tecniche di raccolta (3.1), tecniche di progettazione e generazione di idee (3.2) e strumenti di riflessione (thinking tool) (3.3). Questa differenziazione è ovviamente artificiale. In pratica, non esiste una netta separazione tra le tecniche. Ai fini della presentazione e dell'insegnamento, la differenziazione è comunque importante per strutturare le tecniche e per apprendere il focus principale delle tecniche.

3.4 fornisce le caratteristiche di identificazione tipiche delle tecniche di elicitazione. Queste possono essere utilizzate per descrivere nuove tecniche e per fornire linee guida generali sulle caratteristiche di identificazione che sono potenzialmente utili in una determinata situazione di progetto.

### 3.1 Tecniche di raccolta (L4)

Le tecniche di raccolta sono tecniche consolidate per l'elicitazione dei requisiti. Aiutano a elicitare i Satisfier e Dissatisfier.

#### 3.1.1 Tecniche di indagine (L3)

Le tecniche di indagine hanno lo scopo di porre domande appropriate agli stakeholder. Una importante distinzione è tra domande a risposta aperta e domande a risposta chiusa.

### 3.1.1.1 Intervista (L3)

In un'intervista, il Requirements Engineer pone domande a uno o più stakeholder per poter elicitarne nuovi requisiti o raffinare quelli esistenti. Richiede una preparazione accurata. Durante l'intervista, le risposte devono essere registrate in modo adeguato a supportare la post-elaborazione dei risultati dell'intervista. Esistono diversi tipi di interviste, p.e. interviste con un insieme definito di domande o interviste con un insieme aperto di domande [Port2013], [BaCC2015].

### 3.1.1.2 Questionario (L2)

Viene chiesto a più persone di rispondere per iscritto allo stesso insieme di domande, presentate in modo strutturato. I questionari quantitativi sono utilizzati principalmente per confermare i requisiti precedentemente elicitati, mentre i questionari qualitativi sono più adatti all'elicitazione di nuovi requisiti. I primi possono essere valutati rapidamente e forniscono informazioni statistiche, mentre i secondi tendono a fornire risultati complessi e sono quindi generalmente più lunghi da preparare e da valutare [BaCC2015], [Harr2014].

## 3.1.2 Tecniche di osservazione (L3)

Le tecniche di osservazione hanno lo scopo di estrarre i requisiti dall'osservazione, per esempio, di processi, utenti o situazioni tipiche di utilizzo.

Dovrebbe essere prestata una particolare attenzione al "simplification bias" (pregiudizio di semplificazione) degli osservatori [BaCC2015]: osservatori inesperti (novizi del dominio) hanno la tendenza a semplificare le strategie di problem solving dell'utente esperto, mentre lo osservano. Si raccomanda quindi di parlare con gli esperti in materia (subject matter expert) prima di utilizzare le tecniche di osservazione, e/o di lasciare che gli esperti eseguano la review delle note raccolte dall'osservazione, per ridurre al minimo questo pregiudizio.

### 3.1.2.1 Osservazione sul campo (L3)

Il Requirements Engineer osserva gli stakeholder durante il loro lavoro nel loro ambiente abituale, senza interferire. Le osservazioni fatte vengono utilizzate per derivare i requisiti, che devono essere confermati da una review o da ulteriori tecniche di elicitazione.

### 3.1.2.2 Apprendimento (L2)

Il Requirements Engineer conduce una breve formazione pratica nell'ambiente in cui il sistema da sviluppare/migliorare sarà successivamente utilizzato o è già in uso. Esperti in materia con esperienza insegnano al Requirements Engineer ad acquisire una migliore comprensione del dominio e quindi a elicitarne meglio i requisiti.

### 3.1.2.3 Indagine contestuale (contextual inquiry) (L3)

L'indagine contestuale è una tecnica iterativa di raccolta dei dati sul campo, in cui il Requirements Engineer studia in profondità pochi utenti accuratamente selezionati per arrivare a una comprensione più completa della pratica lavorativa dell'intera base utenti [BeHo1998].

L'indagine contestuale si basa su quattro principi:

- *Contesto*: andare nel contesto dell'utente per osservarlo mentre svolge le sue attività
- *Partnership*: chiedere loro di parlare del proprio lavoro e coinvolgerli nella scoperta di aspetti non descritti del lavoro
- *Interpretazione*: sviluppare una comprensione condivisa con l'utente sugli aspetti rilevanti del lavoro
- *Focus*: durante la preparazione dell'indagine contestuale, definire gli obiettivi di elicitazione e dirigere l'indagine raccogliendo i dati rilevanti per poter raggiungere gli obiettivi

### 3.1.3 Tecniche di collaborazione (L3)

Le tecniche di collaborazione hanno lo scopo di focalizzarsi sulla collaborazione tra gli stakeholder per l'elicitazione dei requisiti. In questo processo di co-creazione, gli stakeholder con differenti prospettive, spesso utenti di sistema, sono direttamente coinvolti nella raccolta, evoluzione o raffinamento dei requisiti. Questo tipo di tecniche fornisce una piattaforma di discussione e consente un feedback immediato da parte degli stakeholder. Esempi di tecniche di collaborazione sono i workshop sui requisiti, il Requirements Engineering crowd-based e i laboratori attivi.

#### 3.1.3.1 Workshop sui requisiti (L3)

Workshop è un termine che racchiude le tecniche orientate al gruppo (group-oriented). I workshop possono essere condotti in modalità molto differenti e possono includere altre tecniche di elicitazione o modelli di processo (p.e. un workshop di design thinking all'interno di uno sviluppo agile). I formati di un workshop possono variare da piccoli meeting informali a eventi organizzati con diverse centinaia di stakeholder [Gott2002].

#### 3.1.3.2 Requirements Engineering crowd-based (L2)

Per alcuni sistemi (p.e. le applicazioni mobili), i requisiti possono essere raccolti dalla "folla" (crowd). Possono essere dati espliciti (p.e. feedback, review) o dati impliciti (p.e. dati di utilizzo, log dei difetti) [MNJR2015], [GrDA2015].

### 3.1.4 Tecniche basate sugli artefatti (L3)

Gli artefatti sono prodotti del lavoro umano (chiamati anche prodotti del lavoro), come sistemi IT, documenti, immagini, file audio e video, ecc. Alcuni di questi prodotti di lavoro sono sorgenti rilevanti dei requisiti. In genere, l'attività di analizzare i prodotti di lavoro in dettaglio richiede molto tempo.

#### 3.1.4.1 Archeologia dei sistemi (L3)

L'archeologia dei sistemi è una tecnica per elicitare informazioni su un nuovo sistema dalla documentazione, dall'interfaccia utente o dal codice di un sistema legacy o un sistema competitor. Si raccomanda di iniziare prima dall'analisi di documenti come le specifiche, la documentazione di test o i manuali utente, perchè contengono informazioni similari ai requisiti. Con l'aiuto dell'archeologia dei sistemi, si può garantire che nessun requisito implementato nel sistema attuale venga perso.

#### 3.1.4.2 Lettura perspective-based (basata sulla prospettiva) (L3)

Il Requirements Engineer utilizza una prospettiva specifica, ad esempio una prospettiva di utilizzo o una prospettiva dei dati, per poter identificare i requisiti rilevanti da un documento [Pohl2010].

#### 3.1.4.3 Riutilizzo dei requisiti (L3)

Se all'interno dell'azienda esistono progetti similari, o versioni precedenti del sistema che deve essere sviluppato, i requisiti di questi progetti possono essere riutilizzati. I requisiti considerati per il riutilizzo devono essere confermati da una review o da ulteriori tecniche di elicitazione.

## 3.2 Tecniche di progettazione e generazione di idee. (L4)

Tradizionalmente, la letteratura sul Requirements Engineering parla di tecniche creative. Il loro scopo è di creare idee per trovare soluzioni per una determinata domanda, problema o obiettivo. Esempi di tali tecniche sono il brainstorming [Osbo1979] oppure 6-thinking hats [DeBo2006]. Durante l'elicitazione dei requisiti, le tecniche creative vengono utilizzate per creare requisiti nuovi o innovativi, che spesso sono Delighter.

Aldif fuori del software e del Requirements Engineering, viene utilizzato il termine più ampio di tecniche di progettazione. Le tecniche di progettazione si riconducono alle tecniche creative per la generazione di idee, e forniscono tecniche aggiuntive o combinate per elaborare le idee e per ottenere ulteriori approfondimenti di una specifica idea [Kuma2013]. Tecniche comuni includono la prototipazione (prototyping, p.e. mock-up), gli scenari e gli storyboard.

## Precondizioni per la creatività

La creatività non nasce a comando, ma per caso. La creatività è più probabile che accada quando sono soddisfatte tutte e quattro le seguenti precondizioni [KrSc2017]:

- *Possibilità* – e quindi tempo – per far nascere un'idea
- *Conoscenza* dell'argomento, che aumenta le probabilità di un'idea che faccia la differenza
- *Motivazione*, poiché il nostro cervello può essere creativo solo se esiste un beneficio diretto per il suo proprietario
- *Sicurezza e safety*, poiché le idee inutili non devono avere conseguenze negative

Le tecniche di progettazione e generazione di idee aiutano in alcuni o in tutti questi aspetti a creare un ambiente adeguato allo sviluppo di nuove idee e innovazioni.

### 3.2.1 Brainstorming (L3)

Il brainstorming è stato sviluppato negli anni 1940–1950 da Alex F. Osborn [Osbo1979]. Come la maggior parte delle tecniche creative, il punto cruciale del brainstorming è la separazione della ricerca di idee dall'analisi delle idee. Viene condotta in gruppi di circa 5–10 persone e dura circa 20 minuti. Un moderatore assicura lo svolgimento ordinato del brainstorming e la documentazione delle idee.

Negli anni si sono sviluppate diverse varianti, come ad esempio il paradosso del brainstorming, il metodo 6–3–5, il brainwriting.

### 3.2.2 Tecniche per analogia (L2)

Le tecniche per analogia (p.e. bisociazione [Koes1964]) aiutano a trovare idee per argomenti critici e anche complessi. Usano le analogie per supportare il pensiero e generare idee. Il loro successo o fallimento è influenzato principalmente dalla qualità dell'analogia. La rilevanza di sistemi simili è discussa in 2.4.

### 3.2.3 Prototipazione (L3)

Prototipazione è un termine generico che si riferisce alla creazione di vari tipi di campioni o modelli iniziali che vengono costruiti per fare un'esperienza reale e visualizzare un concetto o un processo.

Nell'elicitazione dei requisiti, il termine prototipo non si riferisce solo all'implementazione di prototipi nel software. Si riferisce a tutto quello che può rappresentare i requisiti di un sistema da sviluppare (p.e. schizzo dell'interfaccia utente, mock-up fisico, video). Lo scopo della prototipazione nell'elicitazione dei requisiti è la simulazione del nuovo sistema e l'esplorazione dei requisiti, aiutando a stimolare l'accordo, l'obiezione o il chiarimento, ma anche il miglioramento.



Un prototipo può essere valutato attraverso l'applicazione di un walkthrough utente [ShRP2007] o di un testing utente/di usabilità [RuCh2008]. Spesso il risultato di questa valutazione è un insieme di nuovi requisiti.

### 3.2.4 Scenari e storyboard (L3)

Gli scenari di utilizzo descrivono, sotto forma di esempi realistici, come un utente interagirà con il sistema proposto [RoCa2002].

Uno storyboard è uno scenario visualizzato. Ha l'aspetto di un fumetto, con una serie di immagini e/o schermate, e visualizza come un sistema o un prodotto deve essere utilizzato. La riflessione su un esempio concreto permette a clienti e agli utenti di immaginare i requisiti nella situazione applicativa reale e quindi di eseguire la review e modificarli [RiFi2014].

## 3.3 Strumenti di riflessione (thinking tool) (L2)

Le tecniche di elicitazione descritte finora rappresentano tecniche che descrivono un modo di raccogliere le informazioni o di creare uno specifico prodotto di lavoro ai fini dell'elicitazione dei requisiti.

In questo sotto-capitolo vengono presentate tecniche trasversali a questi tipi di tecniche, poiché favoriscono un certo modo di pensare. Sono definite tecniche a supporto e strumenti di riflessione (thinking tool), perché non vengono applicate da sole, ma insieme ad altre tecniche.

### 3.3.1 Pensare secondo i livelli di astrazione (L2)

I livelli di astrazione sono un potente strumento di riflessione nell'elicitazione dei requisiti [GoWo2005], [Laue2014]. Spesso vengono utilizzati come un tipo di modello di processo per strutturare il lavoro di elicitazione, cioè per elicitare prima solo i requisiti al livello più alto di astrazione e continuare poi con i livelli successivi. Può essere utilizzato anche per strutturare le informazioni sui requisiti, per identificare lacune nei requisiti oppure requisiti non necessari, e per focalizzare le attività di elicitazione a un certo livello di astrazione. Ad esempio, in un workshop con gli utenti, è consigliabile focalizzarsi sul contesto di sistema, poiché gli utenti sono gli esperti del contesto di sistema. Parlare delle strutture dati di un sistema può non essere appropriato, poiché agli utenti non interessano le strutture dati interne.

### 3.3.2 Pensare in termini di problemi e obiettivi (L2)

Pensare in termini di problemi e obiettivi è una competenza fondamentale per il Requirements Engineer.

Un *problema* è "una difficoltà, una domanda aperta o una condizione indesiderata che richiede un'indagine, una considerazione o una soluzione" [Glin2024]; in altre parole, lo stato di un aspetto nel contesto di uno stakeholder, che viene vissuto come negativo. Un problema può esistere nel presente (un problema reale).

Un obiettivo è "uno stato desiderato (che uno stakeholder vuole raggiungere)" [Glin2024]: lo stato di un aspetto nel contesto di uno stakeholder che si aspetta sia positivo. Un obiettivo esiste solo nel futuro.

I problemi e gli obiettivi non esistono nel mondo reale: sono costruzioni mentali degli stakeholder. Lo stesso argomento può essere concepito come un problema da uno stakeholder ed essere visto come obiettivo da un altro stakeholder. I problemi e gli obiettivi possono essere conosciuti solo comunicando con gli stakeholder appropriati.

Problema e obiettivo sono interconnessi da un altro costrutto mentale: La *soluzione* è una roadmap per un determinato intervento nel contesto dello stakeholder. Di solito, più di una soluzione può risolvere il problema e raggiungere l'obiettivo (in una certa misura). Per ulteriori informazioni, si veda [LoLS2017].

Pensare in termini di problemi e obiettivi consente al Requirements Engineer di analizzare e scoprire la rete completa di problemi, soluzioni e obiettivi.

In letteratura esistono diversi approcci che si focalizzano sui problemi, p.e. Problem Frames [Jack2001], o sugli obiettivi, p.e. KAOS [Lams2009].

### 3.3.3 Evitare gli effetti trasformativi (L2)

Nel syllabus CPRE Foundation Level [IREB2022], gli effetti trasformativi (o insidie nell'informazione) sono discussi nel contesto dell'output del Requirements Engineering, che è la documentazione.

I Requirements Engineer dovrebbero essere consapevoli della presenza di questi (e altri) effetti trasformativi anche nei loro input, poiché si verificano frequentemente durante le attività di elicitazione, in fase di comunicazione con gli stakeholder o durante la lettura dei documenti. Rilevare la presenza di questo tipo di effetti è un trigger per eseguire ulteriori attività di elicitazione, che probabilmente riveleranno requisiti aggiuntivi o dettagliati.

### 3.3.4 Pensare in termini di modelli (L2)

Il syllabus CPRE Foundation Level [IREB2022] introduce diversi tipi di modelli (p.e. class diagram, activity diagram) per documentare i requisiti. I modelli permettono di focalizzarsi su una specifica prospettiva di un sistema: dati, funzione, comportamento. I modelli possono anche servire come strumento di riflessione se il Requirements Engineer vuole focalizzarsi su una specifica prospettiva durante una particolare attività di elicitazione, p.e. discutere un activity diagram durante un'intervista con lo stakeholder oppure sviluppare un class diagram durante un workshop con gli stakeholder. Il Requirements Engineer deve comunque tenere presente che i modelli sono utili solo se il linguaggio di modellazione è compreso da tutti gli stakeholder coinvolti.

### 3.3.5 Mind mapping (mappe mentali) (L3)

Il mind mapping è uno strumento per pensare grafico [Buza1993]. Posizionando un argomento principale al centro e distribuendo le idee nei rami, i pensieri e le idee possono essere ordinati e strutturati. È opportuno utilizzare sia il testo sia le immagini sia il colore. Le rappresentazioni "noiose" (linee rette, un solo colore) dovrebbero essere evitate per rendere la rappresentazione più "stimolante" per il cervello.

## 3.4 Esempio di strutturazione delle tecniche di elicitazione: attributi (L2)

I Requirements Engineer dovrebbero selezionare con cura le tecniche di elicitazione da utilizzare in base al contesto e alle esigenze specifiche della situazione. Per supportare questa selezione, le tecniche possono essere classificate in base a determinati attributi. Un esempio di attributi utili è presentato in Tabella 1.

La natura di una tecnica di elicitazione può essere descritta da una combinazione di questi attributi.

Ad esempio, la tecnica *"intervista"* è caratterizzata dagli attributi *"conversazionale"* e *"interrogativa"*. Un'intervista può anche essere *"osservativa"* nel caso in cui il Requirements Engineer conduca l'intervista presso la sede dell'utente finale previsto.

Tuttavia, *"osservativa"* non è un attributo basilare delle interviste, in quanto potrebbero essere condotte anche telefonicamente o in altri luoghi dove non è possibile fare osservazioni.

Tabella 1 definisce gli attributi rilevanti. Nella scelta di una tecnica di elicitazione, il Requirements Engineer dovrebbe considerare la disponibilità e le caratteristiche degli stakeholder, le esigenze del cliente, gli obiettivi e i vincoli di progetto, il dominio e il contesto in cui lavora (cfr. 1.3). Classificare un lungo elenco di tecniche disponibili in base agli attributi rilevanti può aiutare a selezionare le tecniche da utilizzare in una situazione specifica. *"Esistono buone pratiche nel contesto, ma non esistono migliori pratiche"* [KaBa2012]: ogni situazione può richiedere una combinazione specifica di tecniche per avere successo.

Tabella 1 Attributi per la classificazione delle tecniche di elicitazione.

Attributo	Descrizione	Obiettivi da raggiungere	Adatto nelle seguenti situazioni
<b>Conversazionale</b>	Un dialogo tra il Requirements Engineer e gli stakeholder	Comprendere il contesto di sistema; elicitare gli obiettivi e ottenere un overview dei Satisfier (Kano)	Quando gli stakeholder (rilevanti) sono disponibili per uno scambio di informazioni oralmente
<b>Interrogativa</b>	Porre agli interlocutori domande (almeno in parte) preparate per conoscere i fatti o le loro opinioni	Elicitare gli obiettivi e i Satisfier; verificare i Dissatisfier; ottenere l'opinione degli stakeholder o ulteriori informazioni sui requisiti precedentemente elicitati; elicitare informazioni dettagliate; chiarire requisiti specifici	Quando le domande rilevanti possono essere formulate in anticipo; quando è possibile una comunicazione con gli stakeholder; quando si tratta di argomenti complicati
<b>Osservativa</b>	Osservare i comportamenti degli stakeholder in una situazione reale, mentre operano su un sistema esistente o mentre svolgono compiti specifici	Raccogliere informazioni sul comportamento effettivo dello stakeholder; elicitare i Dissatisfier; analizzare i requisiti di usabilità; raccogliere dati sul contesto dell'utente	Quando non è possibile rivolgersi direttamente agli stakeholder o se questi non sono in grado di esplicitare le proprie necessità e le proprie azioni (in modo sufficientemente dettagliato); quando ci sono dubbi sulla congruenza tra la situazione reale e quella dichiarata; quando occorre migliorare la comprensione delle necessità degli utenti; quando occorre migliorare la comprensione del progetto (p.e. in preparazione ad altre tecniche di elicitazione)

Attributo	Descrizione	Obiettivi da raggiungere	Adatto nelle seguenti situazioni
<b>Provocare (dis)accordo</b>	Dimostrare gli aspetti rilevanti di una soluzione per ottenere un feedback affermativo o contraddittorio dagli stakeholder	Rendere i requisiti tangibili per gli stakeholder; valutare i requisiti precedentemente elicitati; ottenere un feedback sulle varianti di una soluzione	Se gli stakeholder hanno difficoltà a immaginare cosa desiderano; se il Requirements Engineer può spiegare o mostrare aspetti della soluzione proposta agli stakeholder (o addirittura farla utilizzare); se gli stakeholder hanno difficoltà a spiegare le loro necessità
<b>Basata sugli artefatti</b>	Analizzare i prodotti di lavoro (p.e. documenti, modelli, prodotti o sistemi in uso)	Derivare i requisiti dai prodotti di lavoro esistenti; elicitare i Dissatisfier/Satisfier, in particolare i vincoli	Quando i prodotti di lavoro rilevanti sono disponibili e accessibili; per migliorare la comprensione del progetto e del dominio (p.e. in preparazione ad altre tecniche di elicitazione); se gli stakeholder non sono direttamente disponibili
<b>Stimolare la creatività</b>	Promuovere la creatività e l'innovazione	Elicitare Delighters; proporre approcci innovativi	Quando è necessaria l'innovazione; quando manca una direzione predeterminata; quando altri approcci falliscono
<b>Sperimentale</b>	Sperimentare l'ambiente e lo spazio del problema dove verrà utilizzato il sistema da sviluppare	Derivare i requisiti dalle situazioni reali; comprendere il problema da risolvere grazie agli utenti nel loro contesto lavorativo; ottenere empatia	Quando gli utenti e l'usabilità sono aspetti chiave del progetto; quando è possibile accedere all'ambiente dove sarà effettivamente utilizzato

Esistono anche altri modi per classificare le tecniche di elicitazione, ad esempio:

- Modello Kano (si veda [IREB2022])
- Processo di Design Innovation Kuma2013]

## 4 Risoluzione del conflitto (L4)

Durata: 2.0 ore

Termini: necessità, consistenza, completezza, fattibilità, conflitto tra requisiti, conflitto sociale

### Obiettivi Formativi

EO 4.1.1 Comprendere la differenza tra conflitti tra requisiti e altri conflitti sociali

EO 4.1.2 Applicare l'identificazione dei conflitti

EO 4.2.1 Applicare la classificazione dei tipi di conflitto

EO 4.2.2 Comprendere, in qualità di Requirements Engineer, quali conflitti risolvere e quali delegare

EO 4.3.1 Applicare l'agreement (accordo), il compromesso, la definizione di varianti, la votazione e l'imposizione come esempi di tecniche di negoziazione

EO 4.3.2 Analizzare le tecniche di negoziazione per selezionare quelle più appropriate in base alle caratteristiche del conflitto

EO 4.4.1 Comprendere la documentazione della risoluzione dei conflitti tra requisiti

Durante l'elicitazione, il Requirements Engineer scopre, raccoglie e progetta un'ampio insieme di requisiti. Le tecniche di elicitazione non garantiscono che questo insieme nel suo complesso sia chiaro, completo, consistente, non ambiguo e accettabile. Per ottenere l'insieme finale dei requisiti, tutti gli stakeholder devono raggiungere una comprensione condivisa e un accordo su tutti i requisiti ritenuti rilevanti. Se alcuni stakeholder non sono d'accordo, questa situazione deve essere riconosciuta come un conflitto tra requisiti, che dovrebbe essere risolto di conseguenza.

La risoluzione dei conflitti consiste di quattro attività:

- Identificazione del conflitto
- Analisi del conflitto
- Risoluzione del conflitto
- Documentazione della risoluzione del conflitto

L'identificazione e l'analisi del conflitto sono attività sistematiche nel Requirements Engineering e sono un prerequisito per risolvere qualsiasi conflitto. Una volta identificato un conflitto tra requisiti, il Requirements Engineer dovrebbe iniziare le attività di risoluzione del conflitto per selezionare una tecnica di risoluzione adeguata e per documentarne l'esito.

### 4.1 Identificazione del conflitto (L2)

I conflitti in generale sono un argomento sociale e vengono tipicamente indicati come "conflitto sociale" per indicare che un conflitto nasce tra persone. Un conflitto sociale può essere definito come segue: "... un'interazione tra attori (individui, gruppi, organizzazioni e così via), in cui almeno un attore vede un'incompatibilità con un altro attore (altri attori) rispetto al pensiero, all'immaginazione, alla percezione, al feeling e/o al desiderare, in un modo tale che la sua realizzazione sia compromessa dall'altro attore (altri attori)."

[Glas2004]

Un conflitto tra requisiti può essere interpretato come un particolare tipo di conflitto sociale e viene definito come segue: "Un conflitto nel Requirements Engineering (conflitto tra requisiti) è un'incompatibilità tra requisiti, basata su una percezione contraddittoria di due o più stakeholder." [RueA2014]. Esistono diversi indicatori che permettono di identificare i conflitti. Gli indicatori possono essere osservati nella comunicazione e nella documentazione.

Gli indicatori comunemente osservati nella *comunicazione* sono:

- Rifiuto
- Indifferenza
- Pedanteria
- Domande di dettaglio
- Interpretazione errata
- Occultamento
- Delegazione

Gli indicatori comunemente osservati nella *documentazione* sono:

- Dichiarazioni contraddittorie degli stakeholder
- Risultati ottenuti dall'analisi di documenti o sistemi in conflitto tra loro
- Requisiti inconsistenti nel dettaglio
- Uso inconsistente dei termini nella specifica

La maggior parte dei conflitti tende a essere nascosto e può essere identificato solo monitorando attentamente questi indicatori. Se si verifica uno degli indicatori, questo non significa che sia presente un conflitto tra requisiti. Il Requirements Engineer dovrebbe quindi prestare continuamente attenzione. Nella maggior parte delle attività di elicitazione dei requisiti, il Requirements Engineer stimola gli stakeholder a esprimere chiaramente le loro posizioni, rivelando in alcuni casi problemi inaspettati o conflitti esistenti.

## 4.2 Analisi del conflitto (L3)

Una volta identificato un conflitto, il Requirements Engineer deve chiarire se il conflitto identificato è un conflitto tra requisiti. Questa distinzione è importante perché la risoluzione di un conflitto tra requisiti è di responsabilità del Requirements Engineer, mentre gli altri conflitti devono essere risolti da altri partecipanti (p.e. un project manager).

L'analisi delle caratteristiche di un conflitto tra requisiti aiuta il Requirements Engineer a comprenderne la natura. Le seguenti caratteristiche [RueA2014] di un conflitto possono aiutare a comprenderne la natura e a trovare una soluzione appropriata:

- Tipo di conflitto
- Argomento del conflitto
- Requisiti impattati
- Stakeholder coinvolti
- Opinioni dei diversi stakeholder
- Causa del conflitto



- Stato di avanzamento/storia del conflitto
- Conseguenze del conflitto
- Rischi derivanti

Il tipo di conflitto è importante per decidere se un determinato conflitto è un conflitto tra requisiti. Si distinguono sei diversi tipi di conflitti (cfr. [IREB2022]):

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

La maggior parte dei conflitti tra requisiti può essere classificata come conflitto sui dati, conflitto di interessi e conflitto di valori. I conflitti sull'argomento, se presenti, vengono spesso identificati nelle prime fasi del progetto. I conflitti strutturali e i conflitti di relazione non sono di solito correlati ai requisiti e dovrebbero essere risolti da altri partecipanti.

La maggior parte dei conflitti presenta caratteristiche di più di una tipologia, poiché le diverse cause potrebbero interagire. I Requirements Engineer dovrebbero quindi prestare attenzione a tutti i tipi di conflitto, anche se la soluzione non è di loro responsabilità.

### 4.3 Risoluzione del conflitto (L4)

Un prerequisito per la selezione di una tecnica di risoluzione appropriata è una comprensione approfondita della natura del conflitto tra requisiti. Si possono distinguere le seguenti tecniche di risoluzione (si veda [IREB2022]):

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

Esistono ulteriori tecniche ausiliarie, ad esempio:

- Comunicazione non violenta [Rose2015]
- Tecniche di negoziazione [FiUP2012]
- Valutazione di tutti i fatti [DeBo2006]
- Interesse più-meno [DeBo2006]
- Matrice decisionale [BiAB2006], [IsNe2013]

In base alle caratteristiche di un conflitto, dovrebbero essere selezionate le tecniche di risoluzione del conflitto adeguate.

## 4.4 Documentazione della risoluzione del conflitto (L2)

Dopo la sua risoluzione, il conflitto deve essere documentato in modo appropriato. Oltre alle caratteristiche del conflitto riportate in 4.2, dovrebbero essere incluse:

- Assunzioni sul conflitto e sulla sua risoluzione
- Vincoli che influenzano la scelta della tecnica di risoluzione del conflitto e/o della risoluzione del conflitto
- Potenziali alternative considerate
- Risoluzione del conflitto, incluse le ragioni della risoluzione scelta
- Responsabili delle decisioni (decision-maker) e altri collaboratori

Se non documentate, gli stakeholder potrebbero semplicemente dimenticare o ignorare le decisioni prese, o cercare di modificarle a posteriori. Questo si verifica spesso in situazioni in cui il conflitto tra requisiti viene risolto, ma non si risolve un (altro) conflitto sociale sottostante.

## 5 Competenze del Requirements Engineer (L3)

Durata: 1.5 ore

Termini: Competenze, modelli di comunicazione, interpretazione, risposta, auto-riflessione

### Obiettivi Formativi

EO 5.1.1 Comprendere le competenze richieste nelle aree di elicitazione

EO 5.2.1 Comprendere i fondamenti della teoria della comunicazione

EO 5.3.1 Applicare l'auto-riflessione sulle competenze personali nell'elicitazione dei requisiti

EO 5.4.1 Comprendere le necessità per lo sviluppo personale

EO 5.5.1 Comprendere l'apprendimento da esperienze precedenti

### 5.1 Competenze richieste nelle aree di elicitazione (L2)

Nel Syllabus CPRE Foundation Level [IREB2022], le competenze comunicative, il pensiero analitico, l'empatia, la capacità di risolvere i conflitti, la capacità di moderazione, la fiducia in se stessi e la capacità di convincere sono presentate come competenze (soft) richieste a un Requirements Engineer. Per l'elicitazione dei requisiti a livello Practitioner/Specialist, sono rilevanti anche le seguenti caratteristiche:

- Consapevolezza di sé
- Consapevolezza del contesto
- Natura motivazionale
- Leadership
- Flessibilità
- Riflessione
- Neutralità
- Competenza interculturale
- Coscienza etica

Di tutte queste competenze, le competenze comunicative sono il fattore chiave di successo per il Requirements Engineer. Tutte le interazioni tra il Requirements Engineer e gli stakeholder, che sono le sorgenti principali dei requisiti, sono una forma di comunicazione e tutte le competenze sopra menzionate giocano un ruolo in questa.

### 5.2 Teoria della comunicazione e modelli di comunicazione (L2)

La comunicazione consiste nella *condivisione di concetti significativi* tra individui. Durante la comunicazione, le informazioni possono essere perse, aggiunte, distorte o non interpretate correttamente. Il Requirements Engineer dovrebbe fare in modo di evitare il più possibile questi problemi.

La comprensione della teoria e dei modelli, e la capacità di integrare queste conoscenze nelle attività di comunicazione quotidiana, miglioreranno la comunicazione del Requirements Engineer e porteranno a risultati migliori.

Una comprensione della teoria della comunicazione può essere ottenuta studiando i seguenti modelli di comunicazione:

- Il modello Shannon–Weaver [ShWe1971]
- Il modello circolare di comunicazione [Schr1971]
- Il modello "a quattro facce" di Schulz von Thun [Schu1981]

### 5.3 Auto-riflessione sulle competenze personali nell'elicitazione dei requisiti (L3)

Questo Syllabus e la relativa formazione gettano le basi per un'applicazione di successo delle tecniche presentate. Lo sviluppo e il miglioramento delle competenze personali per l'elicitazione dei requisiti è un processo di apprendimento a lungo termine.

Anche se il Requirements Engineering di uno sviluppo è stato un successo, in genere esistono diverse opportunità di miglioramento. Ad esempio:

- Una tecnica ha rilasciato i risultati attesi / ha contribuito allo sviluppo?
- Gli stakeholder hanno accettato le tecniche di elicitazione o di risoluzione dei conflitti applicate?
- L'effort per una tecnica era giustificabile rispetto al contributo allo sviluppo?
- Quale tecnica avrebbe potuto consentire di elicitare in anticipo i requisiti emersi in ritardo durante lo sviluppo?

La valutazione appropriata delle proprie abilità può essere fatta da un lato attraverso l'osservazione diretta del comportamento e dall'altro attraverso un'analisi successiva. In un'osservazione diretta il focus dovrebbe essere posto su una o al massimo due caratteristiche per ottenere un risultato di monitoraggio accurato e affidabile (p.e. l'osservazione della propria comunicazione riflessiva durante un'intervista). Per valutare le proprie competenze in un'analisi successiva, la risposta di altre persone è una sorgente importante (p.e. il feedback a 360° [LeLu2009]). Anche una scheda di valutazione delle capacità precedentemente definite è uno strumento di misura adeguato [SmMa2004].

### 5.4 Opportunità di sviluppo personale (L2)

Molto spesso un'esperienza pratica insufficiente viene presentata come una ragione per non applicare una specifica tecnica di elicitazione o di risoluzione dei conflitti. Questo atteggiamento può essere comprensibile in termini di successo del progetto (il Requirements Engineer applica le tecniche che conosce meglio per garantire il successo del progetto); in termini di sviluppo personale, questo atteggiamento non è utile. Un'alternativa comprovata è l'applicazione di tecniche non familiari in un contesto a basso rischio (p.e. eseguire l'apprendimento con un piccolo sottinsieme di stakeholder).

È possibile applicare una tecnica non familiare in parallelo a una tecnica familiare (p.e. un questionario viene applicato in parallelo a una serie di interviste).

## 5.5 Apprendere dall'esperienza precedente (L2)

I componenti fondamentali di un processo di formazione personale che favorisca l'apprendimento dalle esperienze precedenti sono:

- Miglioramento nel lavoro quotidiano
- Misure del proprio profilo di abilità (svolte in modo regolare)
- Misure della formazione
- Misure del mentoring

## 6 Riferimenti bibliografici e ulteriori letture

- [AlIS1977] C. Alexander, S. Ishikawa, M. Silverstein: A Pattern Language – Towns – Buildings – Construction. Oxford University Press, New York, 1977.
- [Alex2005] I.F. Alexander: A Taxonomy of Stakeholders – Human Roles in System Development. International Journal of Technology and Human Interaction, Vol 1, 1: 23–59, Hershey, 2005.
- [AlBe2009] I.F. Alexander, Ljerka Beus–Dukic: Discovering requirements: how to specify products and services. John Wiley & Sons Ltd, Hoboken, 2009.
- [BaCC2015] K. Baxter, C. Coraggio, K. Caine: Understanding Your Users: A Practical Guide to User Research Methods, 2nd edition. Morgan Kaufmann, Burlington, 2015.
- [BaGr1975] R. Bandler, J. Grinder: The Structure of Magic I – A Book About Language and Therapy. Science and Behavior Books, Palo Alto, 1975.
- [BaGr1976] R. Bandler, J. Grinder: The Structure of Magic II – About Communication and Change, Science and Behavior Books, Palo Alto, 1976.
- [BeHo1998] H. Beyer, K. Holtzblatt: Contextual Design: Defining Customer–Centered Systems. Morgan Kaufmann, Burlington, 1998.
- [Beve1957] W. I. B. Beveridge: The Art of Scientific Investigation. The Blackburn Press, Cambridge, 1957. Il testo completo di questo libro è disponibile su [www.archive.org](http://www.archive.org). Ultima revisione febbraio 2019.
- [BiAB2006] S. Biffi, A. Aurum, B. Boehm: Value–Based Software Engineering. Springer–Verlag, Berlin, 2006.
- [Bour2009] L. Bourne: Stakeholder Relationship Management: A Maturity Model for Organizational Implementation. Gower Publishing Ltd, Burlington, 2009.
- [Bour2015] L. Bourne: Making Projects Work: Effective Stakeholder and Communication Management. CRC Press, Boca Raton, 2015.
- [Buza1993] T. Buzan: The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain’s Untapped Potential. BBC Books, London, 1993.
- [CaGr2012] E. Cameron, M. Green: Making Sense of Change Management: A Complete Guide to the Models Tools and Techniques of Organizational Change. 3rd ed., Kogan Page, London, 2012.
- [Cock2001] A. Cockburn: Writing Effective Use Cases. Addison–Wesley, Boston, 2001.
- [CRCN2014] A. Cooper, R. Reimann, D. Cronin, C. Noessel: About Face: The Essentials of Interaction Design. 4th Edition, John Wiley & Sons, Inc, Hoboken, 2014.
- [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.
- [CoSh2007] T. Colburn, G. Shute: Abstraction in Computer Science. Minds & Machines, Vol. 17, pp. 169–184, Springer Science+Business Media B.V., Dordrecht, 2007.
- [CrOB2006] O. Creighton, M. Ott, B. Bruegge: Software Cinema: Video-based Requirements Engineering – 14th IEEE International Requirements Engineering Conference. IEEE Computer Society, Washington DC, 2006.
- [DeBo2006] E. DeBono: Edward DeBono's Thinking Course – Powerful Tools to Transform Your Thinking. BBC Active, London, 2006.
- [GrDA2015] Eduard C. Groen, J. Doerr, S. Adam: Towards Crowd-Based Requirements Engineering – A Research Preview– Requirements Engineering– Foundation for Software Quality – 21st International Working Conference – REFSQ 2015. Essen, Germany, March 23–26, 2015. Proceedings. pp. 247–253., Cham, 2015.
- [FiUP2012] R. Fisher, W. Ury, B. Patton: Getting to Yes – Negotiating an agreement without giving in, 3rd rev. ed. Random House Business, New York, 2012.
- [Glas1982] F. Glas: The process of conflict escalation and roles of third parties. In: G.B.J. Bomers and R.B. Peterson, (eds) Conflict management and industrial relations, Springer–Science+Business Media, Dordrecht, 1982.
- [Glas2004] F. Glas: Konfliktmanagement: Ein Handbuch für Führungskräfte, Beraterinnen und Berater. 11th ed., Freies Geistesleben, Stuttgart, 2004.
- [Glin2024] M. Glinz: A Glossary of Requirements Engineering Terminology (Version 2.1.0). IREB e.V., Karlsruhe, 2021. <https://www.ireb.org/en/downloads/#cpre-glossary>. Ultima revisione febbraio 2024.
- [Gott2002] E. Gottesdiener: Requirements by Collaboration: Workshops for Defining Needs, Addison–Wesley Professional, Boston, 2002.
- [GoWo2005] T. Gorschek, C. Wohlin: Requirements Abstraction Model. Requirements Engineering Journal Vol. 11, No. 1, pp. 79–101. <http://dx.doi.org/10.1007/s00766-005-0020-7>, 2005. Ultima revisione febbraio 2019.
- [Harr2014] D. F. Harris: The Complete Guide to Writing Questionnaires– How to Get Better Information for Better Decisions, I&M Press, North Carolina, 2014.
- [HoDC2007] P. Holman, T. Devane, S. Cady: The Change Handbook. The Definitive Resource on Today's Best methods for Engaging Whole Systems. McGraw–Hill Professional Pub Group West, New York, 2007.
- [IGKCD2002] A. Iles, D. Glaser, M. Kam, J. Cannyand, E. Do: Learning via Distributed Dialogue: Live Notes and Handheld Wireless Technology. Proc. Conf. Computer Support for Collaborative Learning, Hillsdale, 2002.
- [IREB2022] IREB Certified Professional for Requirements Engineering Foundation Level Syllabus (version 3.1.0). IREB e.V., Karlsruhe, 2022.



<https://www.ireb.org/en/downloads/#cppe-foundation-level-syllabus-3-0>.

Ultima revisione febbraio 2024.

- [IsNe2013] A. Ishizaka, P. Nemery: Multi-criteria decision analysis. Methods and software. John Wiley & Sons, Ltd, Chichester, 2013.
- [ISO29148] ISO/IEC/IEEE29148:2011. Systems and software engineering – Life cycle processes – Requirements engineering, International Organization for Standardization, Geneva, 2011.
- [ISO9241-210] ISO9241-210:2010. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems, International Organization for Standardization, Geneva, 2010.
- [Jack2001] M. Jackson: Problem Frames – Analyzing and structuring software development problems. Addison-Wesley, Boston, 2001.
- [KaBa2012] C. Kaner, J. Bach: The Seven Basic Principles of the Context-Driven School. <http://context-driven-testing.com/>, 2012.
- [Kell1984] J. F. Kelley: An iterative design methodology for user friendly natural language in office information applications. ACM Transactions on Office Information Systems, March 1984, 2:1, 26-41. New York, 1984.
- [Kell2002] H. Kellner: Kreativität im Projekt. Hanser Fachbuch, München, 2002.
- [Koes1964] A. Koestler: The Act of Creation. Penguin Books, London, 1964.
- [KoSo1998] G. Kotonya, I. Sommerville: Requirements Engineering: Processes and Techniques. Wiley Publishing, Hoboken, 1998.
- [KrSc2017] I. Kreß, A. Schwarz: To Brainstorm or Not to Brainstorm – Neuropsychological Insights on Creativity. Requirements Engineering Magazine Vol. 2017-02. <https://re-magazine.ireb.org/issues/2017-02-staying-on-the-right-path/to-brainstorm-or-not-to-brainstorm-neuropsychological-insights-on-creativity/>. Ultima revisione febbraio 2019.
- Kuma2013] V. Kumar: 101 Design Methods – A Structured Approach for Driving Innovation in Your Organization. Wiley, 2013.
- [Kuni2003] M. Kuniavsky: Observing the User Experience: A Practitioners's Guide to User Research, Morgan Kaufmann, Burlington, 2003.
- [Lams2009] A. van Lamsweerde: Requirements Engineering: From System Goals to UML Models to Software Specifications. JohnWiley&Sons, Hoboken, 2009.
- [Laue2014] K. Lauenroth: What does it mean to say "requirement"? – An inquiry into the abilities of the human mind and the meaning of the word "requirement". Requirements Engineering Magazine Vol. 2014-01. <http://re-magazine.ireb.org/issues/2014-1-learning-to-fly/what-does-it-mean-to-say-requirement>, 2014. Ultima revisione febbraio 2019.

- [Leff2011] D. Leffingwell: Agile Software Requirements. Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison–Wesley, Boston, 2011.
- [LeLu2009] R. Lepsinger, A.D. Lucia: The Art and Science of 360 Degree Feedback. 2nd ed., Wiley, San Francisco, 2009.
- [LeWi2003] D. Leffingwell, D. Widrig: Managing Software Requirements: A Use Case Approach. Addison–Wesley, Boston, 2003.
- [LoLS2017] H. van Loenhoud, K. Lauenroth, P. Steiger: The goal is to solve the problem – Some thoughts on problems and goals in the context of Requirements Engineering. Requirements Engineering Magazine Vol. 2017–02. <http://re-magazine.ireb.org/issues/2017-02-staying-on-the-right-path/the-goal-is-to-solve-the-problem/>. Ultima revisione febbraio 2019.
- [MNJR2015] W. Maalej, M. Nayebi, T. Johann, G. Ruhe: Toward Data–Driven Requirements Engineering. IEEE Software Vol. 33, No. 1, pp.48–54, 2015.
- [MaGi2001] N. Maiden, A. Gizikis: Where Do Requirements Come From? IEEE Software Vol. 18, 5: 10–12, Geneva, 2001.
- [Mayh1999] D. J. Mayhew: The Usability Engineering Lifecycle. Morgan Kaufmann, Burlington, 1999.
- [McCo2006] S. McConnell: Software Estimation – Demystifying the Black Art. Microsoft Press, Redmond, 2006.
- [Moor2014] C. W. Moore: The Mediation Process – Practical Strategies for Resolving Conflicts. 4th ed., John Wiley & Sons, Hoboken, 2014.
- [Niel1993] J. Nielsen: Usability Engineering. Morgan Kaufmann, Burlington, 1993.
- [Osbo1979] A. F. Osborn: Applied Imagination. 3rd rev. ed., Charles Scribner’s Sons, New York, 1979.
- [Parn1972] D.L. Parnas: On the Criteria To Be Used in Decomposing Systems into Modules (PDF). Communications of the ACM 15 (12), pp.: 1053–58. doi:10.1145/361598.361623, ACM New York, 1972.
- [Pohl2010] K. Pohl: Requirements Engineering – Fundamentals, Principles, and Techniques. Springer, Berlin, 2010.
- [PoRu2015] K. Pohl, C. Rupp: Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam – Foundation Level – IREB compliant, Rocky Nook, Santa Barbara, 2015.
- [Port2013] S. Portigal: Interviewing Users: How to Uncover Compelling Insights. Rosenfeld Media, Brooklyn, 2013.
- [Reif 2012] D.J. Reifer: Software Change Management: Case Studies and Practical Advice. Microsoft Press, Redmond, 2012.
- [RiFI2014] M. Richter, M. Flückiger: User–Centred Engineering: Creating Products for Humans. Springer, Heidelberg, 2014.

- [Robs2011] C. Robson: Real World Research. John Wiley & Sons, Hoboken, 2011.
- [RoCa2002] M. Rosson, J. M. Carroll: Usability Engineering: Scenario-Based Development of Human Computer Interaction. Morgan Kaufmann, Burlington, 2002.
- [Rohr1969] B. Rohrbach: Kreativ nach Regeln-Methode 635, eine neue Technik zum Lösen von Problemen. Absatzwirtschaft 12, Heft 19:73-75, Meedia GmbH & Co.KG, Amburgo, 1969.
- [RoRo2013] S. Robertson, J. Robertson: Mastering the Requirements Process: Getting Requirements Right. Third Edition, Pearson Education, London, 2013.
- [Rose2015] M. B. Rosenberg: Nonviolent Communication – A Language of Life. 3rd rev. ed., Puddle Dancer Press (US), Encinitas, 2015.
- [Royc1972] W. Royce: Managing the Development of Large Software Systems. Technical Papers of Western Electronic Show and Convention (WesCon) August 25-28, Los Angeles, 1972.
- [RuCh2008] J. Rubin, D. Chisnell: Handbook of Usability Testing- How to Plan, Design, and Conduct Effective Tests. Wiley; Indianapolis, 2008.
- [RueA2014] C. Rupp, i SOPHISTI: Requirements-Engineering and -Management – Aus der Praxis von klassisch bis agil. 6th ed., Carl Hanser Verlag, München, 2014. (capitoli selezionati versione inglese si veda <http://www.sophist.de/en/infopool/downloads/>). Ultima revisione febbraio 2019.
- [Schr1971] W. L. Schramm: How communication works. in W. L. Schramm, ed., The Process and Effects of Mass Communication. rev. ed., University of Illinois Press, Champaign, 1971.
- [Schu1981] F. Schulz von Thun: Miteinander reden 1 – Störungen und Klärungen. .Psychologie der zwischenmenschlichen Kommunikation. Rowohlt, Reinbek, 1981.
- [Shac1991] B. Shackel: Usability – Context, Framework, Definition, Design and Evaluation. In B. Shackel, S. Richardson (Eds.): Human Factors for Informatics Usability (p. 21-37), University Press, Cambridge, UK, 1991.
- [ShRP2007] H. Sharp, Y. Rogers, J. Preece: Interaction Design: Beyond Human-Computer Interaction. John Wiley & Sons, Hoboken, 2007.
- [ShWe1971] C. E. Shannon, W. Weaver: The Mathematical Theory of Communication University of Illinois Press, Champaign, 1971.
- [SmMa2004] S. Smith, R. Mazin: The HR Answer Book: An Indispensable Guide for Managers and Human Resources Professionals, AMACOM, Hertogenbosch, 2004.
- [Wieg2003] K. E. Wiegers: Software Requirements. Microsoft Press, Redmond, 2003.