

Ingegneria dei Requisiti

Corso tenuto dal Professor Mariano Ceccato

Università degli Studi di Verona

Alessio Gjergji

Indice

1	Introduzione	2
1.1	Terminologia di base	2
1.1.1	Le due versioni del mondo	3
1.1.2	I requisiti di sistema e i requisiti software	3
1.2	Scope dell'ingegneria dei requisiti	4

Capitolo 1

Introduzione

1.1 Terminologia di base

Per avere un'implementazione corretta e completa del software dobbiamo fare in modo che il software risolva un problema concreto del mondo reale. Dobbiamo quindi comprendere correttamente il mondo reale, ovvero come funziona, come dovrebbe funzionare, quali sono i vincoli che governano, capire il contesto in cui il software deve operare. Questo è il compito dell'ingegneria dei requisiti.

Esempio: il controllo di un'automobile

- **Problema:** la gestione del freno a mano in un'automobile moderna potrebbe essere automatizzato.
- **Contesto:** se l'automobile è in movimento, se si sta frenando, se è intenzione del guidatore fermarsi, se il guidatore ha premuto il pedale del freno, ...

Quando parliamo di **requisiti** dobbiamo inoltre definire il **mondo**. Quando definiamo il mondo, abbiamo a che fare con elementi molto complessi che contengono di fatto elementi umani (*lo staff, l'utente, il cliente, ...*), elementi fisici (*dispositivi, software legacy, la natura, ...*) e dovrà quindi adattarsi alla situazione esistente.

Dobbiamo definire anche il mondo delle **macchine**. Parliamo quindi di tutto ciò che dovrà essere implementato e/o acquistato, come ad esempio *database, server, client, ...* e i componenti hardware e software che dovranno essere implementati.

L'ingegneria dei requisiti non si limita solamente al mondo delle macchine, ma tiene in considerazione anche gli effetti che il software ha sul mondo reale, che dovrà modellare.

Il mondo e le macchine hanno i propri fenomeni, ma ne condividono anche alcuni. Ad esempio, il mondo ha il *rilascio del freno a mano*, mentre le macchine hanno `errorCode = 013`. Nell'intersezione potremmo avere ad esempio `motor.Regime = 'up'`.

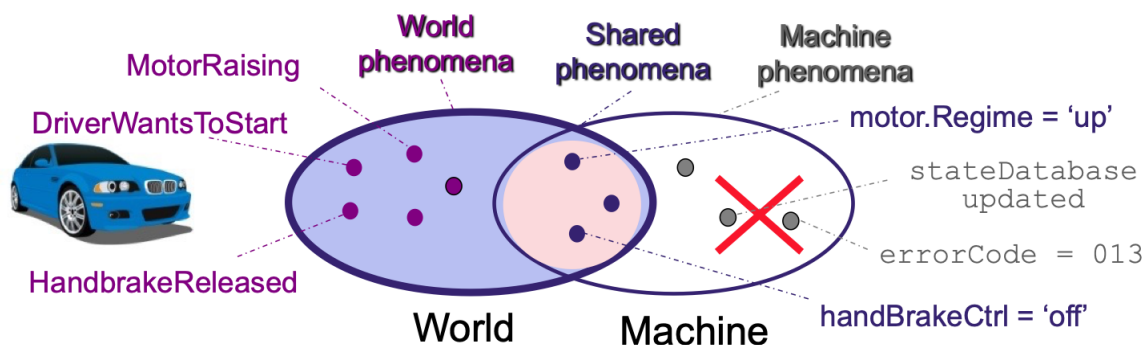


Figura 1.1.1: Il mondo e le macchine

Quando parliamo di requisiti, consideriamo solamente la parte relativa al mondo, che comprende quindi anche l'intersezione. È importante comprendere che non descrivono **nulla** riguardo alle macchine.

1.1.1 Le due versioni del mondo

Il sistema è l'insieme delle interazioni delle componenti che strutturano il mondo.

- Il sistema **as-is**: si tratta del sistema prima che il software venga implementato. Questo sistema è composto da solamente dal mondo.
- Il sistema **to-be**: si tratta del sistema come dovrebbe essere quando il software opererà. Questo sistema è composto dall'unione del mondo e delle macchine.

Definizione preliminare dell'ingegneria dei requisiti

Consiste in una serie di attività collegate tra loro che ci permettono di esplorare, valutare, documentare, consolidare, rivedere e adattare quelli che sono gli obiettivi, le capacità, i vincoli e le assunzioni in un sistema software. Basato sui problemi del sistema **as-is** e le opportunità date dalle nuove tecnologie.

1.1.2 I requisiti di sistema e i requisiti software

Il sistema (*unione del mondo reale e del sistema software*) sono i requisiti che fanno riferimento a tutto, si tratta degli statement che fanno parte del **system to-be**, mentre i requisiti software sono un sottoinsieme dei requisiti di sistema, fanno quindi riferimento solamente al **software to-be**, di fatto le funzionalità che il software che dovrà fornire per rispondere in maniera adeguata ai problemi del nostro utente.

1.2 Scope dell'ingegneria dei requisiti

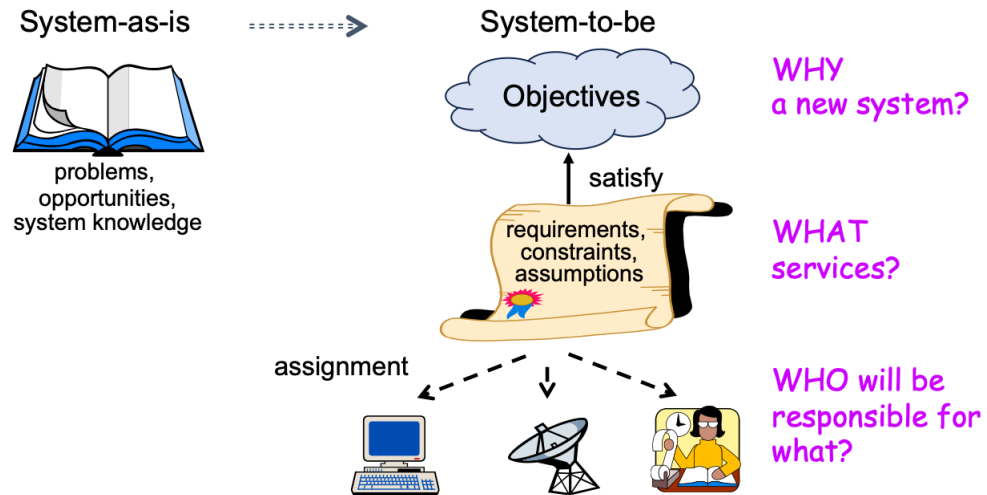


Figura 1.2.1: Scope dell'ingegneria dei requisiti

Prima di iniziare a lavorare sui requisiti, abbiamo il nostro system as-is, che comprenderà i problemi, le opportunità e le conoscenze sul sistema. Da qui mapperemo il tutto nel system to-be. Per comprendere al meglio il system to-be, dobbiamo ragionare in tre prospettive:

- Perché: quali sono gli obiettivi dell'abbinamento tra software e mondo reale. Quindi quali sono le necessità che il software dovrà soddisfare.
- Cosa: quali sono le funzionalità che il software dovrà fornire per soddisfare i bisogni del cliente, quali saranno i vincoli e le assunzioni che il software dovrà rispettare.
- Chi: chi sarà responsabile della fornitura delle funzionalità.