

## REQUIREMENT MANAGEMENT

Maria Teresa Palomba – A-Key S.r.l.

[t.palomba@a-key.it](mailto:t.palomba@a-key.it)



## Riferimenti:

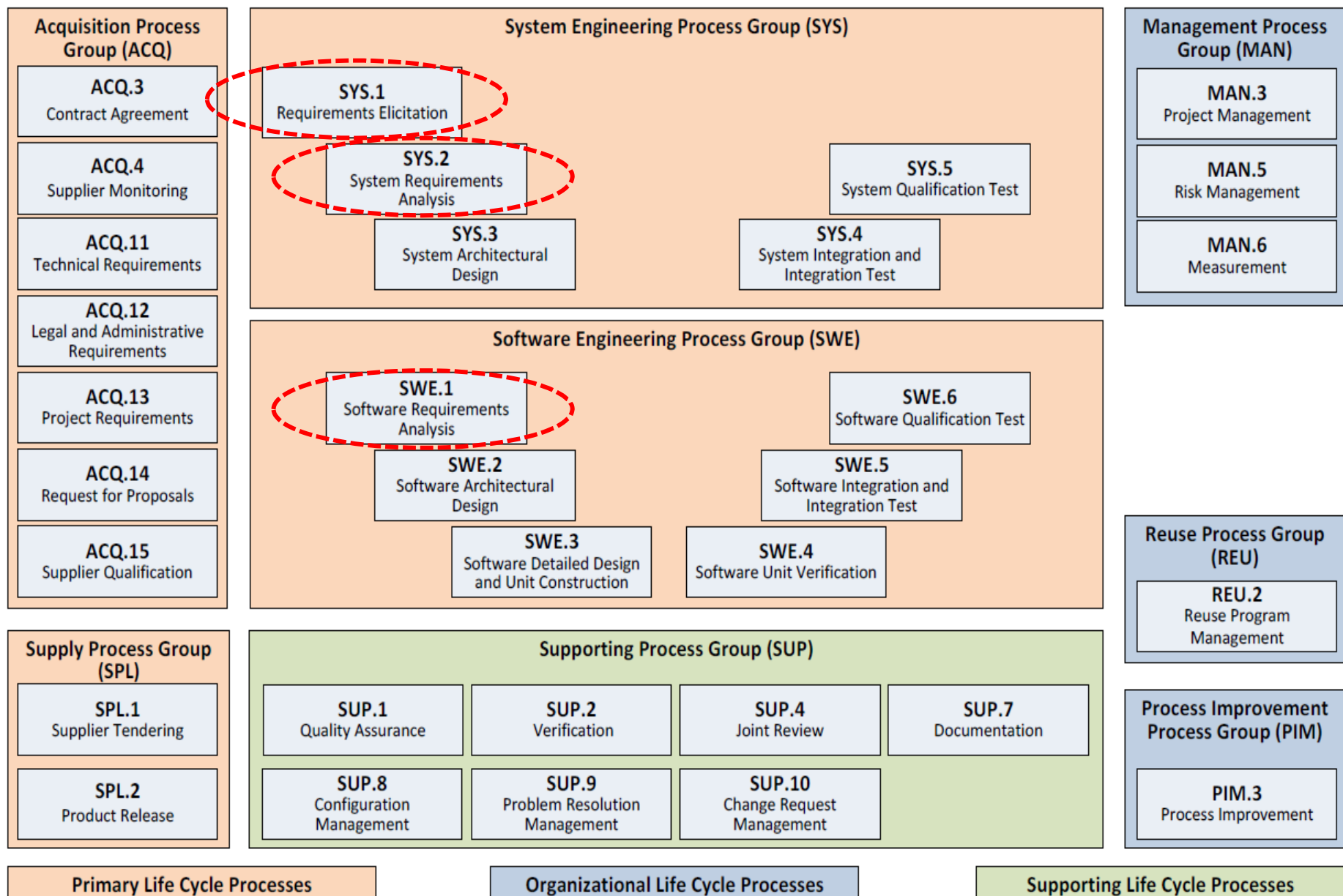
- A. <http://www.computer.org/web/swebok> - Software Engineering Body of Knowledge (SWEBOK), version 3.0
- B. <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm> modello CMMI-DEV v. 1.3
- C. [http://www.automotivespice.com/fileadmin/software-download/Automotive SPICE PAM 30.pdf](http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf)  
- Automotive SPICE PAM PRM v.3.0
- D. <http://www.ibm.com/developerworks/rational/library/04/r-3217/> - Traceability from Use Cases to Test Cases
- E. <http://space.se.spacegrant.org/uploads/Requirements-Writing/Writing%20Good%20Requirements.pdf>  
- Writing Good Requirements (A Requirements Working Group Information Report)

## Modello a cascata

Modello di sviluppo basato sulla **esecuzione in sequenza di passaggi** da una fase alla successiva in un **ordine strettamente lineare**

### CICLO A CASCATA (WATERFALL)





**REQUIREMENT** (ieee610)

A “requirement” is:

- 1 )A condition or capability needed by a user to solve a problem or achieve an objective.
- 2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
- 3) A documented representation of a condition or capability as in 1 or 2

I requirements quindi, descrivono le funzionalità, i servizi, le modalità operative e di gestione, i vincoli di un sistema software, il cui soddisfacimento possa essere verificato attraverso dei test.

Il processo di Requirements Analysis **identifica, descrive e traduce** le “**esigenze di business**” e i **vincoli** espressi dai “**requirements providers**” verso il **team di progetto** incaricato della realizzazione di un sistema software.

La gestione e analisi dei requisiti è l'area di processo dedicata ad assicurare la **consistenza tra i requisiti e le caratteristiche del software prodotto**; essa è un elemento fondamentale di qualsiasi progetto.



How the customer explained it



How the Project Leader understood it



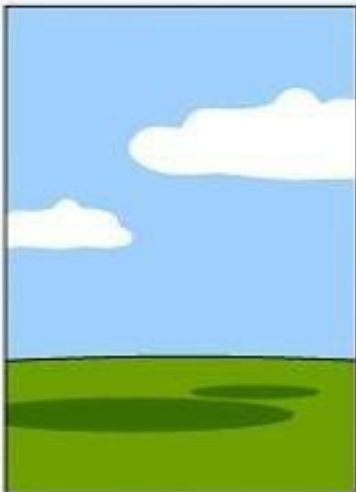
How the System Analyst designed it



How the Programmer wrote it



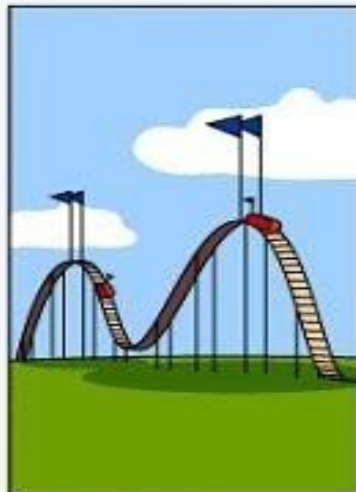
How the Business Consultant described it



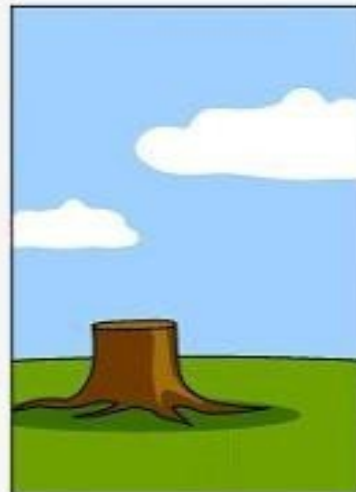
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

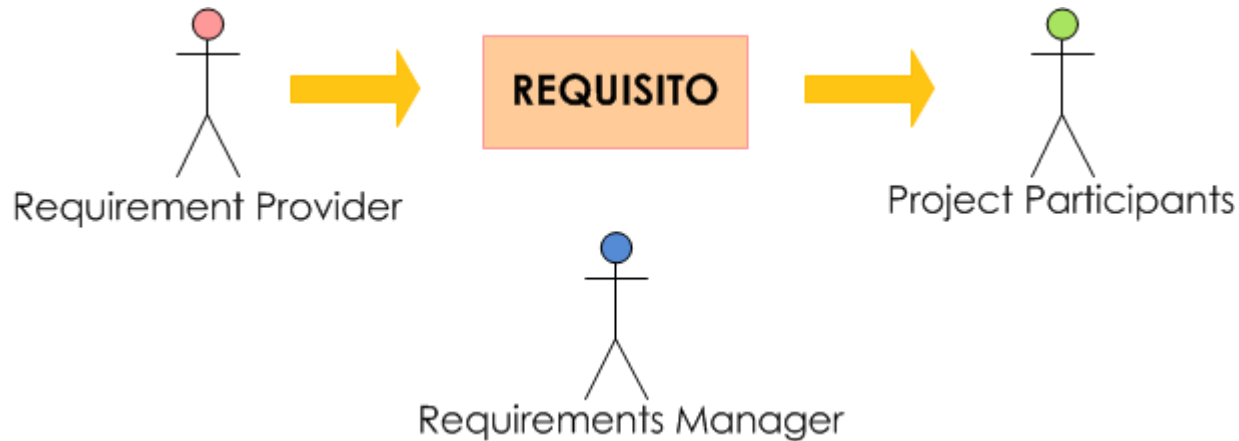
<http://projectcartoon.com/cartoon/1078>



Elementi base del processo sono:

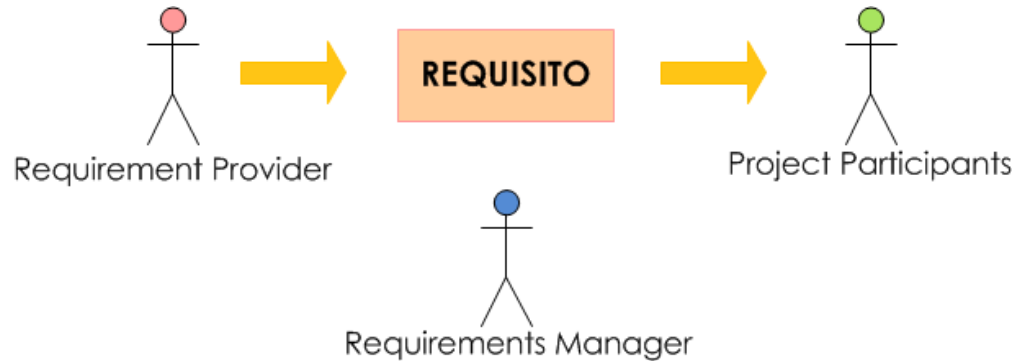
- ❑ Identificare e mantenere aggiornata la definizione di **cosa il sistema software deve fare**, in accordo con i “Requirement Providers” di progetto;
- ❑ Supportare il processo di **revisione, consolidamento e convalida** dei requisiti in collaborazione con i Requirements Providers;
- ❑ Gestire la **condivisione e la presa in carico dei requisiti** per la loro implementazione da parte del team di progetto;
- ❑ **Supportare le fasi di progettazione e realizzazione del software**, e le **verifiche di congruenza del sistema finale rispetto ai requisiti**, mantenendo la **tracciabilità** tra i requisiti e le altre componenti (design, sviluppo, test);
- ❑ Supportare la **gestione delle modifiche** ai requisiti e la **rilevazione di inconsistenze e defects** che dovessero generarsi rispetto ai piani e agli elaborati di progetto.





Ad ogni requisito corrispondono specifici ruoli coinvolti nella sua definizione ed elaborazione.

Per ciascun livello possono essere coinvolti attori diversi.



## Requirement Provider

- Responsabile della formulazione e approvazione del requisito come parte interessata

## Requirement Manager

- Responsabile della comprensione e gestione del requisito per conto del progetto
- Fa da tramite tra il Requirement Provider ed i Project Participants

## Project Participants

- Coinvolti nell'analisi ed elaborazione dei requisiti
- Partecipano, con varie responsabilità, alla implementazione della soluzione richiesta

La descrizione completa di tutti gli aspetti di un sistema richiede di identificare diverse **tipologie di requisiti**, classificabili in diversi modi, a seconda dell'elemento che si vuole evidenziare.

- ❑ **Need** (esigenza): richiesta proveniente da un Requirement Provider (Cliente, Utente, altre parti interessate);
- ❑ **Feature**: caratteristica che il sistema deve fornire e che soddisfa un Need
- ❑ **Requisito funzionale**: funzionalità del sistema descritta in termini di input, comportamenti e risultati. Implementa una o più features
- ❑ **Requisito non-funzionale**: caratteristica del sistema da cui derivano vincoli alla progettazione o alla esecuzione (prestazioni, affidabilità ..)

- ❑ **Requisiti Cliente o di Business** (esigenze che devono essere soddisfatte dal sistema software)
- ❑ **Requisiti di Sistema** (elementi HW e SW; interazioni e vincoli fra questi)
- ❑ **Requisiti Software** (funzionalità che devono essere possedute e le condizioni che devono essere soddisfatte dal sistema software)

## Requisiti Non Funzionali (1)

**Performance:** Requisiti non funzionali che indicano tempistiche di risposta del modulo a sollecitazioni esterne

**Safety:** Requisiti non funzionali che indicano vincoli di incolumità legati in particolare a ISO26262.

**Scalabilità:** Requisiti non funzionali che indicano caratteristiche che si possono abilitare a compile time o in configurazione del modulo in modo parziale o completo

**Sicurezza:** Requisiti non funzionali che indicano caratteristiche di non vulnerabilità e di protezione dei dati del sistema

**Diagnostica:** Requisiti non funzionali che consentono di rilevare e comunicare all'esterno dei fault o dei parametri ritenuti rilevanti dal cliente.

## **Requisiti Non Funzionali (2)**

**Vincoli:** Requisiti non funzionali che indicano i range dei parametri di funzionamento del modulo ( HW/funzionale )

**Documentazione:** Requisiti che indicano come il software deve essere documentato

**Ergonomia:** Requisiti che indicano caratteristiche di interazione del sistema con l'utente umano

**Interfaccia:** Requisiti che indicano come il modulo software deve interagire e comunicare con il resto del sistema

# Documentazione dei Requisiti del SW

-----

**SRS**

**Software**

**Requirement**

**Specification**



**Esempio**

1. Scopo del documento
2. Definizioni
3. Descrizione del sistema
4. Descrizione del prodotto
  - a. Interfacce di sistema
  - b. Interfacce utente
  - c. Interfacce Hardware
  - d. Interfacce software
  - e. Interfacce di comunicazione
  - f. Vincoli di memoria
  - g. Ambito operativo
  - h. Funzioni del prodotto
  - i. Caratteristiche dell'utente
  - j. Vincoli, presupposti, dipendenze
5. Requisiti del Software
  - a. Requisiti di interfaccia esterna
  - b. Requisiti Funzionali
  - c. Requisiti di Performance
  - d. Vincoli di progettazione
  - e. Conformità a Standard
  - f. Requisiti dei dati
  - g. Requisiti del sistema software
    - i. Affidabilità
    - ii. Disponibilità
    - iii. Sicurezza
    - iv. Manutenibilità
    - v. Portabilità
  - h. Altri Requisiti
6. Storia delle modifiche
7. Revisioni e Approvazioni
8. Riferimenti



Problema → scrivere efficacemente i requisiti, in modo tale che sia ridotta al minimo **l'interpretabilità** di quanto descritto.



Scrivere efficacemente i requisiti richiede l'utilizzo di un **linguaggio formalizzato** che permetta di **esprimere con esattezza** e **in modo univoco** le richieste funzionali e i vincoli di qualunque tipo che devono essere rispettati dal sistema.

- 1) Un requirement deve specificare **una e una sola funzione**
- 2) Un requirement deve specificare “**cosa è richiesto**” e non “come si fa”
- 3) Un requirement deve essere scritto con una frase comprendente **soggetto** e **predicato** (verbo):
  - a) Il **soggetto** deve essere un umano che interagisce con il sistema oggetto della specifica dei requisiti, oppure un altro sistema che interagisce con quello oggetto di definizione, oppure uno stakeholder, oppure il sistema stesso;
  - b) Il **predicato** deve specificare una condizione, un'azione, un risultato atteso che deve eseguito per il soggetto, o dal soggetto, o con il soggetto

- 4) Un requirement deve essere scritto **utilizzando** la **forma imperativa**;
- 5) **Evitare** la **forma passiva e la forma negativa**;
- 6) **Evitare** l'uso di **congiunzioni** come “e”, “o” che specificano più requirements in un'unica frase;
- 7) **Evitare** l'uso di parole aventi **significati differenti o dipendenti dal contesto** (es. “moto”, “metro”);
- 8) **Evitare** l'uso di parole aventi **significato indeterminato** (es. “semplice”, “adeguato”, “spesso”);
- 9) **Evitare** l'uso di **abbreviazioni e simboli non definiti nel glossario**.

Modello per requisiti di **Comportamento/Performance**

**The** <System\_name>  
**shall** <behavior>  
**if** <conditions>,  
**where** <quality factor>

Modello per requisiti **Non Funzionali**

**The** <System\_name>  
**shall** <instance>  
**with** <feature>,  
**and/or** <constraint>

Modello per requisiti di **Capacità**

**The** <System\_name>  
**shall produce** <output>  
**for use by** <nodes>,  
**if** <conditions>

Modello per requisiti di **Conformità di Processo**

**The** <System\_name>  
**shall be** <process>  
**in accordance with** <document>

Un buon requisito esprime una caratteristica **necessaria, verificabile, raggiungibile**.

L'insieme dei requisiti di un sistema software dovrebbe essere in grado di descrivere **tutti i comportamenti osservabili dall'esterno e tutte le caratteristiche attese** del sistema software da realizzare sulla base di tali requisiti.

La valutazione di tale capacità si basa sulla definizione di un insieme di **criteri di accettazione** dei singoli requisiti e dell'insieme dei requisiti, che ne permettono la validazione.

Descrizione
Una SRS è <b>corretta</b> se e solo se ogni requisito in essa definito rappresenta qualcosa di effettivamente richiesto al sistema da costruire
Una SRS è <b>atomica</b> se e solo se ogni requisito in essa definito non può essere suddiviso in ulteriori requisiti
Una SRS è <b>non ambigua</b> se e solo se ogni requisito in essa definito ha una sola interpretazione

### Descrizione

Una SRS è **completa** se ha le seguenti 4 qualità:

- 1) Tutto ciò che ci si aspetta che il SW debba fare è incluso nella SRS
- 2) Tutte le definizioni di risposta del SW a tutte le classi realizzabili di dati di input, in tutte le classi di situazioni realizzabili sono incluse nella SRS
- 3) Tutte le pagine sono numerate; tutte le figure e le tabelle sono numerate, denominate e referenziate; tutti i termini e le unità di misura sono fornite; tutto il materiale e tutti i capitoli cui viene fatto riferimento sono presenti
- 4) Non vi sono capitoli “Da definire”



### Descrizione

Una SRS è **verificabile** solo se ogni requisito in essa definito è verificabile.

Un requisito è verificabile se esiste un processo definito e realizzabile tramite il quale una persona o una macchina può verificare che il prodotto costruito soddisfi i requisiti.

Una SRS è **consistente** se e solo se nessun sottoinsieme dei requisiti in essa definiti determina una contraddizione.

Es. Conflitto di terminologia, conflitto di caratteristiche, inconsistenza temporale.

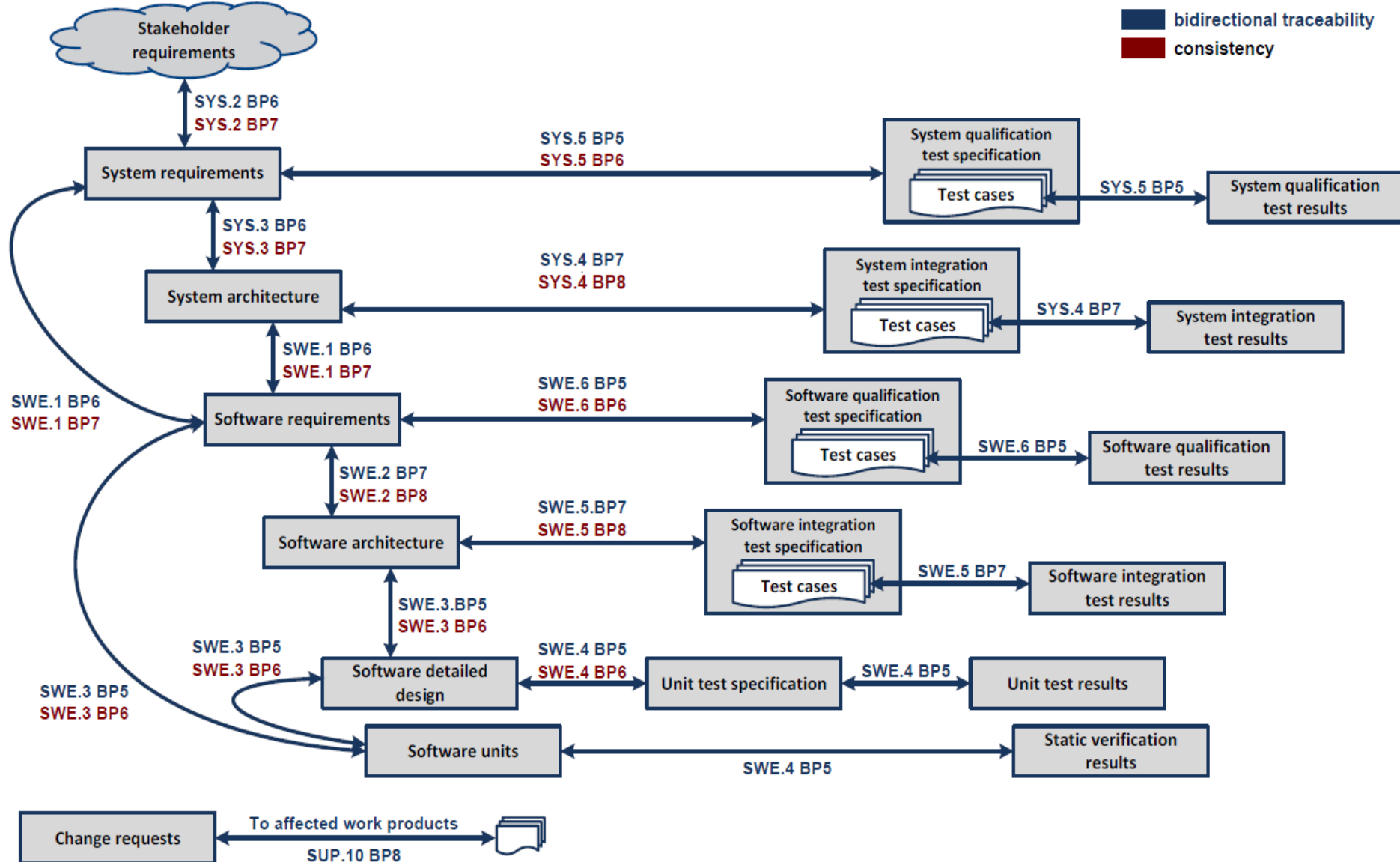
**Descrizione**

Una SRS è **tracciabile** se l'origine di ciascuno dei requisiti in essa definiti è chiara e se la SRS facilita il riferimento di ogni requisito nella documentazione di sviluppi e modifiche successive

Una SRS è **fattibile** se ogni requisito in essa definito è realizzabile entro i vincoli del progetto e della organizzazione

Le attività di un progetto procedono attraverso la definizione, produzione ed elaborazione di **diversi elementi collegati tra di loro** secondo varie relazioni: requisiti (in vari livelli), design, coding, test, defect, change requests etc.

Per questo motivo ogni progetto deve mantenere una **Matrice di Tracciabilità**, in cui sono documentati i dettagli di tutte le relazioni rilevanti, e le attività di progetto devono prevedere operazioni di registrazione e aggiornamento puntuale di queste informazioni.



## Esempio Rappresentazione Tracciabilità Requisiti-Test

	SW_REQ_3421	SW_REQ_3422	SW_REQ_3423	SW_REQ_3424	SW_REQ_3425
SW_TST_1032	X	X			
SW_TST_1033			X		
SW_TST_1034		X			
SW_TST_1035				X	
SW_TST_1036					X
SW_TST_1037					X

**CMMI- DEV  
AUTOMOTIVE SPICE**

## Process areas

### REQUIREMENTS DEVELOPMENT

“The purpose of Requirements Development (RD) is to elicit, analyze, and establish customer, product, and product component requirements. “

### REQUIREMENTS MANAGEMENT

“The purpose of Requirements Management (REQM) is to manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products. “



## REQUIREMENTS DEVELOPMENT - RD

### Specific Goal and Practice Summary

#### ❑ SG 1 Develop Customer Requirements

- SP 1.1 Elicit Needs
- SP 1.2 Transform Stakeholder Needs into Customer Requirements

#### ❑ SG 2 Develop Product Requirements

- SP 2.1 Establish Product and Product Component Requirements
- SP 2.2 Allocate Product Component Requirements
- SP 2.3 Identify Interface Requirements

#### ❑ SG 3 Analyze and Validate Requirements

- SP 3.1 Establish Operational Concepts and Scenarios
- SP 3.2 Establish a Definition of Required Functionality and Quality Attributes
- SP 3.3 Analyze Requirements
- SP 3.4 Analyze Requirements to Achieve Balance
- SP 3.5 Validate Requirements

## REQUIREMENTS MANAGEMENT - REQM

### Specific Goal and Practice Summary

#### ❑ SG 1 Manage Requirements

- SP 1.1 Understand Requirements
- SP 1.2 Obtain Commitment to Requirements
- SP 1.3 Manage Requirements Changes
- SP 1.4 Maintain Bidirectional Traceability of Requirements
- SP 1.5 Ensure Alignment Between Project Work and Requirements

## Processes

### **SYS.1 Requirements Elicitation**

“The purpose of the Requirements Elicitation Process is to gather, process, and track evolving stakeholder needs and requirements throughout the lifecycle of the product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed work products. “

### **SYS.2 System Requirements Analysis**

“The purpose of the System Requirements Analysis Process is to transform the defined stakeholder requirements into a set of system requirements that will guide the design of the system .”

### **SWE.1 Software Requirements Analysis**

“The purpose of the Software Requirements Analysis Process is to transform the software related parts of the system requirements into a set of software requirements. “

## Requirements Elicitation Process

### Process outcomes

- 1) **continuing communication with the stakeholder** is established;
- 2) **agreed stakeholder requirements** are **defined and baselined**;
- 3) a **change mechanism is established** to evaluate and incorporate changes to stakeholder requirements into the baselined requirements based on changing stakeholder needs;
- 4) a **mechanism** is established **for continuous monitoring of stakeholder needs**;
- 5) a **mechanism** is established for ensuring that **customers can easily determine the status and disposition of their requests**;
- 6) **changes arising from changing technology and stakeholder needs** are **identified, the associated risks assessed and their impact managed**.

Base Practices from standard

## System Requirements Analysis Process

### Process outcomes

- 1) a **defined set of system requirements** is established;
- 2) system requirements are **categorized and analyzed for correctness and verifiability**;
- 3) the **impact of system requirements on the operating environment is analyzed**;
- 4) **prioritization for implementing** the system requirements is defined;
- 5) the **system requirements are updated** as needed;
- 6) **consistency and bidirectional traceability** are established between **stakeholder requirements and system requirements**;
- 7) the **stakeholder requirements are evaluated** for **cost, schedule and technical impact**;
- 8) the **system requirements are agreed and communicated to all affected parties**.

## Software Requirements Analysis Process

### Process outcomes

- 1) the **software requirements** to be **allocated to the software elements** of the system and **their interfaces are defined**;
- 2) software requirements are **categorized and analyzed for correctness and verifiability**;
- 3) the **impact of software requirements on the operating environment** is analyzed;
- 4) **prioritization for implementing** the software requirements is defined;
- 5) the **software requirements are updated as needed**;
- 6) **consistency and bidirectional traceability** are established between **system requirements and software requirements**; and consistency and bidirectional traceability are established between **system architectural design and software requirements**;
- 7) the **software requirements are evaluated for cost, schedule and technical impact**;
- 8) the **software requirements are agreed and communicated to all affected parties**.

Base Practices from standard

