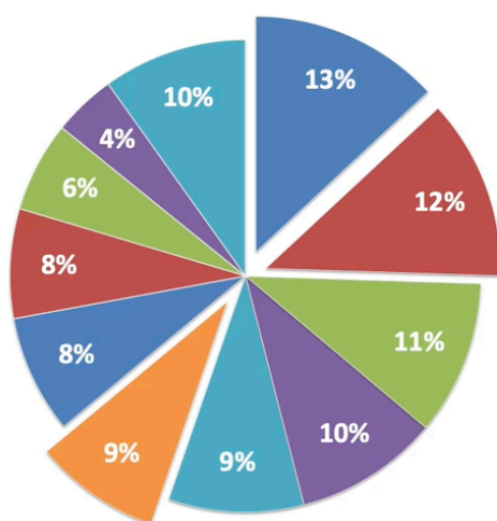


# Requisiti

Friday, 17 March 2023

18:41

- Sono delle funzionalità/capacità o condizione che un progetto deve essere conforme per risolvere determinati problemi  
Ex. Sviluppo in Java
- I requisiti sono importanti siccome i requisiti sono il motivo della maggior parte dei fallimenti



- Requisiti incompleti
- Mancanza di coinvolgimento dell'utente
- Mancanza di risorse
- Aspettative non realistiche
- Mancanza di supporto dell'esecutivo
- Cambiamento dei requisiti & specifiche
- Mancanza di pianificazione
- Caduta di interesse
- Assenza del management IT
- Carenza di conoscenza tecnologica
- Altro

il 34% delle cause dei fallimenti dei progetti software riguardano l'attività dei requisiti

- 2 tipologie:
  - Funzionali
    - Funzionalità o servizi da fornire
    - Risposte che l'utente si aspetta + input -> output
    - Essi devono essere precisi
      - Se ambigui possono essere interpretati in maniera errata
    - Devono essere completi, aka tutti i servizi richiesti
    - Devono essere coerenti tra di loro
  - Es.
    - Una cassiera (quello dei supermercati, quell'oggetto elettrico che determina il prezzo degli oggetti) chiamato POS nextgen deve verificare disponibilità un prodotto
    - Il sistema deve aggiornare ogni vendita + ogni vendita è identificato con un codice
  - Non funzionali
    - Proprietà e vincoli sistema
      - Es sicurezza, prestazione, linguaggio utilizzato, legali, portabilità su altri sistemi
    - Potrebbero essere più critici di quelli funzionali
      - Immagina un qualcosa che è lentissimo
    - Essi potrebbero influire sull'architettura
    - Un requisito non funzionale possono nascere requisiti funzionali

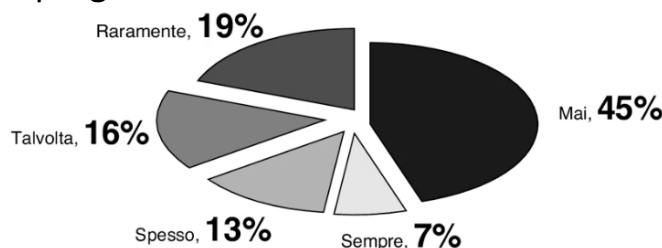
- Deve essere preciso affinché sia possibile verificare se è stato raggiunto
  - Es. Verificare facilità uso, da questo noi dobbiamo creare requisiti non funzionali verificabili  
 Noi vogliamo far sì che il personale riesca ad utilizzarlo in modo da n  
 minimo di errori = **Obiettivo**  
 => Non facilmente verificabile, quindi creiamo un requisito non funz  
 verificabile = **Requisito**  
 = Il personale, dopo 4 ore di formazione, il numero di errori commes  
 deve superare 2 per ora di utilizzo = **Requisito**
  - Quindi, obiettivo è ciò che noi vogliamo ottenere  
 Mentre un requisito (non funzionale verificabile) è come possiamo  
 raggiungere questo obiettivo, e deve essere misurabile
- Obiettivo utile ma non necessario

Es.

- Il Sistema POS nextgen deve essere disponibile sempre
- Deve autenticare gli utenti attraverso tessera identificativa
- Soddisfare leggi italiane

- Informazione interessante:

In un processo a cascata, questa è la percentuale che i requisiti trovati inizialmente saranno utili nel progetto finale:



E questo perché i requisiti cambiano molto nel tempo.

- Come trovarli quindi?
  - Interviste clienti/scrivere casi d'uso con cliente
  - Gruppi di lavoro con clienti come rappresentanti/feedback di ogni iterazione
- Documentazione dei requisiti:
  - Modello dei casi d'uso
    - Insieme scenari tipici dell'utilizzatore del sistema
  - Specifiche supplementari
    - Ciò che non rientra nei casi d'uso, aka requisiti non funzionali
  - Glossario
    - Termini del dominio e problema
    - Avere chiaro in maniera non ambigua delle terminologie
  - Visione

- visione
- Regole di business

- Descrizione:

- Scritti in linguaggi naturale da diagrammi e tabelle
  - Linguaggio standard/consistente
  - Tenere le frasi semplici e comprensibili da tutti
  - Evidenziatore in parti di testo importanti
  - Spiegare il perché del requisito
  - Potremmo avere linguaggio strutturato oppure notazioni grafiche  
Per la spiegazione dei requisiti
- Comprensibile da utenti e clienti, quindi deve essere intuitivo