

# Domande Orale APS

Elia Ronchetti

@ulerich

2022/2023

# Indice

1	Domande Capitolo 1	3
2	Capitolo 2 - Analisi dei requisiti e Casi d'uso	7

# Capitolo 1

## Domande Capitolo 1

### Che cosa sono l'analisi e la Progettazione

- Analisi - Enfatizza l'investigazione di un problema e dei suoi requisiti, anzichè di una soluzione
- La progettazione enfatizza una soluzione concettuale che soddisfa i requisiti del problema

### UML è una metodologia?

No è un linguaggio visuale

### Cos'è un processo software

É un approccio disciplinato per la costruzione, il rilascio e la manutenzione del codice. Definisce chi (ruoli) fa cosa (attività), quando (organizzazione temporale) e come (metodologie) per raggiungere un certo obiettivo. Vi sono vari tipi di processi per lo sviluppo, ma la maggior parte hanno (o rielaborano) le attività fondamentali.

**Come ricordarsi la risposta** Parallelo con la costruzione di una casa, costruzione-rilascio(vendita)-manutenzione.

Il chi sono le varie aziende che operano (ruoli), il cosa sono le attività (caldaia, pannelli solare, cappotto, ecc.), il quando è (quando l'azienda arriva, organizzazione temporale) e come il metodo che utilizzano per il lavoro.

## Quali sono le attività fondamentali di processo?

Le seguenti attività sono solitamente incluse in ogni processo software (che tuttavia può cambiarle e gestirle come vuole):

- Requisiti
- Analisi
- Progettazione
- Implementazione
- Validazione
- Rilascio e installazione
- Manutenzione ed evoluzione
- Gestione del progetto

**Come ricordarsi la risposta** Paralelo con richiesta PC assemblato dove, chiedo quali sono le componenti, analizzo se sono ok, produco un preventivo (progettazione), costruisco la macchina (implementazione) Testo se è tutto oK (Validazione), installo il sistema operativo (rilascio e installazione), il cliente poi dovrà mantenere la macchina (manutenzione ed evoluzione) ed eventualmente effettuare upgrade in futuro, quindi gestirla (gestione del progetto).

Altro esempio valido è la costruzione di una casa.

## Che cos'è il processo a cascata

Il processo a cascata è un processo software sequenziale che prevede le seguenti fasi:

- Definizione dei requisiti
- Design del sistema e del software
- Implementazione e testing unitario
- Integrazione e testing di sistema
- Rilascio e manutenzione (che fa ripartire la cascata da uno degli step precedenti)

Si è rivelato essere un approccio fallimentare dato che il suo presupposto è che i requisiti siano stabili nel tempo, assunzione falsa dato che possono variare e anche di molto nel tempo (a seconda per esempio delle esigenze del cliente).

## **Che cos'è UP e quali sono le sue fasi?**

UP (Unified Process) è un processo per lo sviluppo software di tipo iterativo, incrementale ed evolutivo, dato che si basa su iterazioni time-boxed alla fine delle quali si verifica se i requisiti definiti sono corretti e se ciò che è stato prodotto durante l'iterazione (software eseguibile) è ok. Incrementale dato che nella successiva iterazione si riparte dai risultati prodotti dalla precedente ed evolutivo perchè è in grado di cambiare ed evolversi secondo le esigenze del cliente grazie alle iterazioni. Le sue fasi sono le seguenti

- Ideazione - Analisi e stime iniziali per avviare il progetto (stime sia temporali che economiche)
- Elaborazione - Realizzazione del nucleo dell'architettura, visione raffinata, identificazione di gran parte dei requisiti
- Costruzione - Implementazione delle capacità operative iniziali e successiva preparazione al rilascio
- Transizione - Completamento del progetto

## **Che cos'è un'iterazione**

Si tratta di una finestra temporale (2-6 settimane) all'interno della quale si produce un mini progetto (tranne nella fase dell'ideazione), ogni iterazione include:

- Pianificazione
- Analisi e Progettazione
- Costruzione
- Implementazione e Test
- Release

## Che cos'è il Metodo Agile

Lo sviluppo Agile è una forma di sviluppo che incoraggia l'agilità, ovvero una risposta rapida e flessibile ai cambiamenti, adottabile da qualsiasi processo iterativo. UP può essere reso agile aggiungendo:

- Un piccoli insieme di attività ed elaborati
- I requisiti e la progettazione non vengono completati prima dell'implementazione, ma emergono in modo adattivo durante una serie di iterazioni, anche sulla base di feedback
- Applicazione di UML in stile agile (fare solo ciò che serve)

## Capitolo 2

# Capitolo 2 - Analisi dei requisiti e Casi d'uso

### Che cosa sono i requisiti e di che tipo possono essere?

Un requisito è una capacità o condizione a cui il software deve essere conforme. Ogni software deve possedere delle funzionalità o caratteristiche di qualità.

I requisiti possono essere:

- Funzionali - Definiscono servizi o funzionalità che soddisfano le richieste dal cliente. Sono descritti dai **Casi d'uso**
- Non funzionali - Requisiti legati alle proprietà del sistema (velocità, sicurezza, affidabilità, ecc.)

I requisiti funzionali sono spesso più vincolanti dei funzionali, un requisito non funzionale per essere rispettato può generare la creazione di una serie di requisiti funzionali.

I requisiti sono estremamente importanti in un progetto, considerando che la loro definizione incompleta è una delle cause principali del fallimento di progetti.

In UP vengono definiti inizialmente durante l'ideazione (insieme al cliente) e successivamente nella fase di elaborazione. Non è da escludere che possano essere modificati anche più tardi, ma con UP tendono a stabilizzarsi nel tempo. Consideriamo che il 25 % dei requisiti cambia dopo l'ideazione. UP incoraggia un'acquisizione dei requisiti agile, attraverso la scrittura dei **Casi d'uso** con i clienti, anche tramite interviste, workshop dei requisiti con gli sviluppatori e clienti, e feedback dai clienti dopo ogni iterazione.

## **Cosa sono i casi d'uso?**

I casi d'uso sono storie scritte in formato testuale (preferibilmente in forma ridotta) e costituiscono un dialogo fra uno o più attori e un sistema che svolge un compito. Sono utilizzati per scoprire e registrare i requisiti funzionali (possono anche descrivere requisiti non funzionali).

Se presenti il caso d'uso prevede anche il SSD (System Sequence Diagram - Diagrammi di Sequenza di Sistema).

## **Perchè scriviamo i casi d'uso**

- Perchè ci aiutano a identificare e descrivere i requisiti funzionali
- Sono comprensibili dal cliente dato che sono in formato testuale e sono privi di gergo informatico
- Mettono in risalto gli obiettivi dell'utente e il loro punto di vista
- Sono utili per produrre test e la guida utente

## **Che cos'è un attore e quali sono i vari tipi**

Un attore è qualcosa dotato di un comportamento, anche il Sistema in discussione è considerato un attore quando ricorre ai servizi di altri sistemi.

I vari tipi di attori sono:

- Attore primario - Utilizza direttamente i servizi del sistema in discussione affinché vengano raggiunti gli obiettivi dell'utente
- Attore finale - Vuole che il sistema venga utilizzato per raggiungere i propri obiettivi (spesso coincide con il primario)
- Attore di supporto - Offre un servizio al Sistema (es. Servizi esterni di pagamento)
- Attore fuori scena - Attori interessati al comportamento del Caso d'uso, ma non è nessuno dei 3 sopra elencati, e non interviene nel caso d'uso (es. Stato per quanto riguarda il rispetto di una normativa).