ANNO ACCADEMICO 2025/2026

Tecnologie e Architetture Avanzate di Sviluppo Software

Teoria

Altair's Notes

DIPARTIMENTO DI INFORMATICA

Premessa

Licenza

Formato utilizzato

Box di "Concetto sbagliato":

Concetto sbagliato 0.1: Testo del concetto sbagliato

Testo contente il concetto giusto.

Box di "Corollario":

Corollario 0.0.1 Nome del corollario

Testo del corollario. Per corollario si intende una definizione minore, legata a un'altra definizione.

Box di "Definizione":

Definizione 0.0.1: Nome delle definizione

Testo della definizione.

Box di "Domanda":

Domanda 0.1

Testo della domanda. Le domande sono spesso utilizzate per far riflettere sulle definizioni o sui concetti.

Box di "Esempio":

Esempio 0.0.1 (Nome dell'esempio)

Testo dell'esempio. Gli esempi sono tratti dalle slides del corso.

Box di "Note":

Note:-

Testo della nota. Le note sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive.

Box di "Osservazioni":

Osservazioni 0.0.1

Testo delle osservazioni. Le osservazioni sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive. A differenza delle note le osservazioni sono più specifiche.

1 Introduzione

1.1 Intro al Corso

Parole chiave:

- Web Apps.
- Mission Critical.
- DevOps.
- Cloud Native.

Definizione 1.1.1: Mission Critical Applications

Un'applicazione o sistema le cui operazioni sono fondamentali per una compagnia o un'istituzione.

Osservazioni 1.1.1

- Enfasi sui requisiti non funzionali: i requisiti funzionali sono la baseline, ma ci si aspetta di più per rimanere competitivi.
- Da non confondere con life critical: non muore nessuno.

Definizione 1.1.2: Enterprise Application Integration (EAI)

Tutto l'insieme di pratiche architetturali, tecnologie, patterns, frameworks e strumenti che consentono la comunicazione e la condivisione tra diverse applicazioni nella stessa organizzazione.

Si ha enfasi sull'infrastruttura:

- *Data Integration:* combinare dati da più moduli diversi (coinvolge database).
- *Process Integration:* le interazioni tra più moduli.
- *Functional Integration:* si vuole fornire una nuova funzionalità sfruttando funzionalità già esistenti.

1.1.1 Esempio e Requisiti Non Funzionali

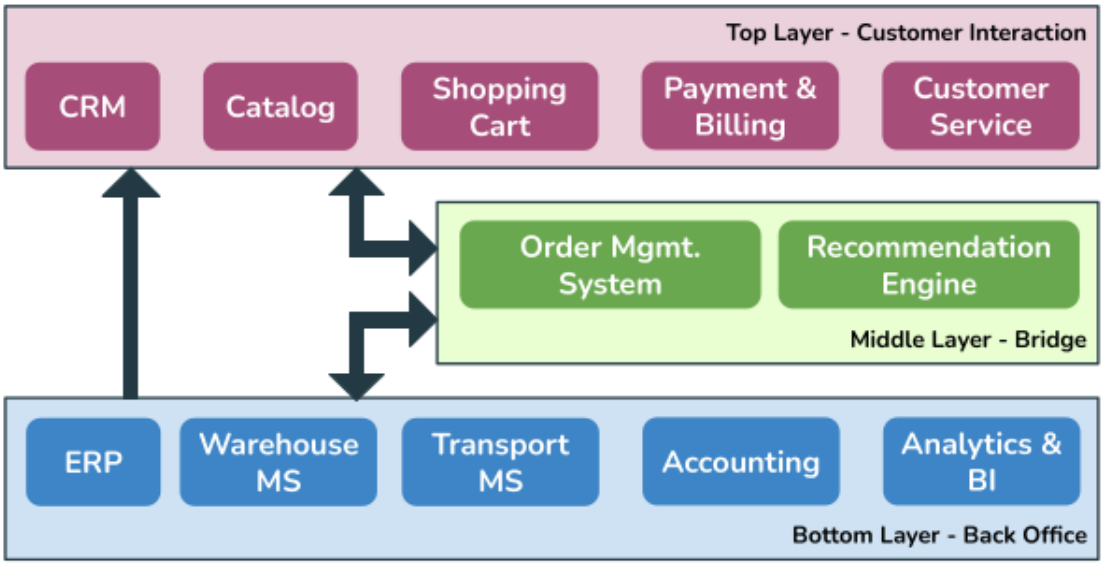

Figure 1.1: Esempio di e-commerce.

Commento dell'esempio:

- Ci sono tre livelli:
  - Top Layer: moduli che si rivolgono al cliente.
  - Middle Layer: gestione della comunicazione tra cliente e azienda.
  - Bottom Layer: moduli interni aziendali.

Requisiti non funzionali:

- High availability/zero downtime: l'applicativo deve essere sempre o quasi sempre disponibile.
- Affidabilità: in caso di interruzione di workflow si deve far sì che non ci siano stati danni (e.g. un'interruzione durante una transazione).
- Consistenza dei dati.
- Integrità dei dati.
- Low latency: per avere una buona performance, tutto deve essere fluido.
- Scalabilità.
- Sicurezza.
- Resilienza: capacità di reagire agli errori.
- Mantenibilità: quanto un pezzo di software sia mantenibile o riutilizzabile.
- Osservabilità: per comprendere eventuali problemi in un sistema distribuito.
- Auditability: le verifiche di qualità fatte su software[1].

[1] Meglio visto in "Etica, Società e Privacy".

1.1.2 Panoramica Storica

Definizione 1.1.3: Waterfall

Le metodologie a cascata[a] sono metodologie in cui ci sono fasi ben distinte e separate tra loro.

[a]Viste a "Sviluppo delle Applicazioni Software".

Note:-

È un modello prevedibile, ma lento a gestire i cambiamenti.

Osservazioni 1.1.2

- Software on the shelf: una volta acquistato è proprio.
- Software custom: prodotto su richiesta, ha bisogno di tutto un servizio di manutenzione.

Definizione 1.1.4: Lean

Metodologie nate negli anni '50 alla Toyota, verranno applicate al software dagli anni '90. Si basa su tre principi:

- Muda[a] (waste): si deve stare sui requisiti, non mettere troppe funzioni non necessarie.
- Mura (unevenness): è necessaria consinstenza per aumentare la prevedibilità.
- Muri (overburden): non sovraccaricare le persone o le macchine. Non progettare software utilizzando strumenti greedy di risorse.

[a]JOJO'S Reference

Note:-

Lo strumento fondamentale è il *kanban*: la lavagna, per organizzare il lavoro.

Definizione 1.1.5: Siloed

Organizzazione aziendale a silos: si comunica poco e male. CI sono 4 gruppi:

- BA Team: relazioni con gli stakeholders, requisiti, specifiche, documentazione.
- Dev Team: programma e fa un minimo di unit testing.
- Test Team: testa e decide se il sistema è pronto.
- Ops Team: si occupa del deployement.

2
Test2