

Noter til "1-SW-Architecture - Process 1"

1. Introduktion til Softwarearkitektur

- Forelæser: Claudio Gomes
- Institution: Aarhus Universitet
- Version: 1.0.1

2. Hvad er Softwarearkitektur?

- **Definitioner:**
 - Martin Fowler: Beslutninger, du ønsker at tage tidligt.
 - Simon Brown: Alt relateret til betydelige elementer i et softwaresystem, fra struktur og fundament til vellykket implementering.

3. Hvorfor arkitektur?

- Forudsigelse af fremtiden er vanskeligt; softwarearkitektur handler om at forberede sig på fremtidige ændringer.
- Eksempler på dårlig arkitektur, der har negativ indvirkning på virksomheder (f.eks. Pinterest).

4. Input til arkitekturen

- **Funktionelle krav:** Hvad systemet skal gøre.
- **Ikke-funktionelle krav:** Hvordan systemet skal gøre det. Disse krav er ofte meget følsomme over for arkitekturen.

5. Kvalitetsattributter

- Vigtige egenskaber som skalerbarhed, tilgængelighed, vedligeholdelse, ydeevne osv.
- Eksempler: Systemet skal kunne afspille videoer for 1000 brugere samtidigt.

6. Arkitektur som kompromis

- En god arkitektur skal afveje krav og forventninger fra forskellige interessenter, beskrevet som brugerhistorier og kvalitetsattributter.

7. Struktur og adfærd

- Arkitektur består af både struktur (moduler, komponenter) og adfærd (hvordan data flyder mellem moduler).

8. SOLID-principper i arkitektur

- Anvendelse af SOLID-principper på arkitekturniveau (f.eks. modulært design, lav kobling, høj samhørighed).

9. Hvordan arkitekturer?

- Brug af Rational Unified Process (RUP) og Microsoft Application Architecture Guide som vejledninger.

10. Arkitekturprocessen

- **1. Arkitektur mål:** Definer input og output for arkitekturprocessen.
- **2. Nøglescenarier:** Identificer kritiske funktionelle og ikke-funktionelle krav.

- **3. Skab et applikationsoverblik:** Forslag til arkitektur baseret på applikationstype, implementeringsbegrænsninger og relevante teknologier.
- **4. Identificer nøgleproblemer:** Løs vigtige problemer og overvej fremtidige ændringer.
- **5. Definér kandidatløsninger:** Evaluér mod baseline arkitektur og test gennem arkitekturmæssige spikes.

11. Kommunikér arkitektur

- Kommunikation af arkitektur til interessenter og teammedlemmer er afgørende.

Øvelser og lektier

- Identificer nøglescenarier og kvantificer kvalitetsattributter for en ny videostreamingtjeneste.

Noter til "2-SW-Architecture - Process 2"

1. Introduktion til Softwarearkitektur

- Forelæser: Claudio Gomes
- Institution: Aarhus Universitet
- Version: 1.0.3

2. Ingen Sølvkugle

- **Essentiel Komplexitet:** Problemet er komplekst og kan ikke refaktoreres væk.
- **Tilfældig Komplexitet:** Løsningen er kompleks, men kan refaktoreres væk.

3. Arkitekturstilarter

- **Lag:** Bekymringer opdelt i lag.
- **Klient-server:** Klienten anmoder serveren.
- **N-tier:** Lignende lag, men hvert lag er på en separat computer.
- **Pipes and Filters:** Data strømmer og transformeres i en pipeline.
- **Message Bus:** Applikationer interagerer via en kommunikationskanal.

4. Recap af arkitekturprocessen

- Iterativ og inkrementel proces, der tester mod krav, kendte begrænsninger og kvalitetsattributter.

5. Kvalitetsattributter

- Eksempler: Tilgængelighed, konceptuel integritet, interoperabilitet, vedligeholdelse, ydeevne, sikkerhed osv.
- Specifikke kvalitetsattributter for kurset inkluderer udviklingsomkostninger, udviklingstid, testbarhed, vedligeholdelse osv.

6. Arkitekturproces Trin

- **1. Arkitekturmål:** Input og output af arkitekturprocessen.
- **2. Nøglescenarier:** Identificer kritiske funktionelle og ikke-funktionelle krav.

- **3. Skab et applikationsoverblik:** Forslag til arkitektur baseret på applikationstype, implementeringsbegrænsninger og relevante teknologier.
- **4. Identificer nøgleproblemer:** Løs vigtige problemer og overvej fremtidige ændringer.
- **5. Definér kandidatløsninger:** Evaluér mod baseline arkitektur og test gennem arkitekturmæssige spikes.

7. Arkitekturstilarter

- **Lag:** Logisk opdeling af bekymringer i lag.
- **Klient-server:** Klient anmoder, server behandler.
- **N-tier:** Fysisk opdeling af præsentation, applikationsbehandling og datastyring.
- **Pipes and Filters:** Data behandles i sekventielle trin.
- **Message Bus:** Applikationer kommunikerer gennem en meddelelsesbus.

Øvelser og lektier

- Skab et applikationsoverblik for en prototype og identificer kvalitetsattributter for en videostreamingtjeneste.