

3. Nøglepunkter

- **Abstraktionsniveau:** Find det rette niveau af abstraktion og identificer relevante synspunkter.
- **Fælles sprog:** Dokumentation handler om kommunikation, hvilket kræver fælles terminologi og notation.
- **C4 model:** Bruges til at dokumentere forskellige abstraktionsniveauer i et system.

4. C4 Model

- **Niveau 1: System Context Diagram**
 - Viser et overordnet billede af systemlandskabet, fokuserer på personer (aktører) og softwaresystemer.
- **Niveau 2: Container Diagram**
 - Viser den højniveau form af softwarearkitekturen og hvordan ansvar er fordelt.
- **Niveau 3: Component Diagram**
 - Zoomer ind på hver container og viser de vigtigste strukturelle byggesten og deres interaktioner.
- **Niveau 4: Kode (Valgfri)**
 - Zoomer ind på hver komponent og viser hvordan den er implementeret som kode, ved hjælp af UML-klassediagrammer eller lignende.

5. Diagram Retningslinjer

- Diagrammer skal være selvforklarende og ledsages af forklarende tekst.
- Brug farver og former for at optimere læsbarheden, men kun som optimeringer.

6. Recap af Arkitekturprocessen

- En iterativ og inkrementel proces, der testes mod krav, kendte begrænsninger og kvalitetsattributter.

7. Praktisk Øvelse

- Tegn arkitekturen af dit projekt og forklar det til en anden person ved hjælp af papir og blyant eller en whiteboard.

Referencer og Billedkilder

- Kilder: c4model.com

Noter til "4-SW-Architecture - Documentation 2"

1. Introduktion til Softwarearkitekturdokumentation - Del 2

- Forelæser: Claudio Gomes
- Version: 1.0.3

2. Agenda

- Softwarearkitekturdokumentation
- Recap af Simon Browns C4 model

- N+1 Views
- UML diagrammer
- Hvordan man bliver en god arkitekt

3. N+1 Views og UML Diagrammer

- **4+1 View Model:** Brug forskellige visninger til at dokumentere forskellige aspekter af systemet.
 - **Logical View:** Vis systemets funktionelle krav.
 - **Development View:** Vis systemets statiske organisation i udviklingsmiljøet.
 - **Process View:** Vis systemets dynamiske aspekter.
 - **Physical View:** Vis systemets fysiske implementering.
 - **Scenarios:** Vis hvordan systemet fungerer i forskellige scenarier.
- **UML Diagrammer:**
 - **Strukturelle Diagrammer:** Brugerpakke-, komponent- og implementeringsdiagrammer til at vise systemets statiske struktur.
 - **Adfærdsmæssige Diagrammer:** Bruger sekvens-, kommunikations-, state machine- og aktivitetsdiagrammer til at vise systemets dynamik.

4. Hvordan man bliver en god arkitekt

- **Faser i Udviklerens Liv:**
 - Entusiastisk udvikler, desillusioneret udvikler, entusiastisk arkitekt, desillusioneret arkitekt, og "vis" arkitekt.
- **Øv:** Lær af andre, læs og se materialer fra kendte arkitekter som Martin Fowler og Simon Brown.
- **Praktisk Øvelse:** Forklar din softwarearkitektur til en anden person uden brug af computer eller forberedte diagrammer.