**Unified process**

Iterative development, hvilket betyder en serie af små tidsbestemte mini projekter også kaldet iterations. (Skal være et tested, integreret og brugbart system bagefter). (side 14 larman).

Man starter med at lave noget design forberedelse, evt. noget UML, alt behøver ikke være perfekt i detaljer før programmeringen. Produktet er ikke en throw-away prototype, eller eksperimentel. Efter en iteration er det en subset af det færdige system, dette gør man så flere gange indtil programmet er færdigt.

I hver iteration vælger man en lille del af de requirements der er, som man så hurtigt kan designe implementere og teste. Det behøver ikke være perfekt eller præcis det de ønsker, men tidlig udgivelse af noget brugbart giver mulighed for meget feedback. Undgår at skulle spekulere på hvad de korrekte design er, hvis man kan give noget de kan prøve.

**Virksomhed:**

Stakeholders, udvidet risko anaylse

**Business modeling:** domain model.

**Requirements:** Use-Case diagram, use case beskrivelser,

**Design:** Supplementary specification (identifikation af ikke funktionelle krav) <https://www.guru99.com/non-functional-requirement-type-example.html> – System sequence diagram  
<https://stackoverflow.com/questions/16889028/difference-between-sequence-diagram-sd-and-a-system-sequence-diagram-ssd> - Class diagram (samt et opdateret som slut dokument) – Pakkediagram over systemsts logiske arkitektur – Statemachine diagram – ER Diagram

**Implementation:**

**Project management:**

**Testing:**

**Enviroment:**