

# Introduktion til systemudvikling (SY)



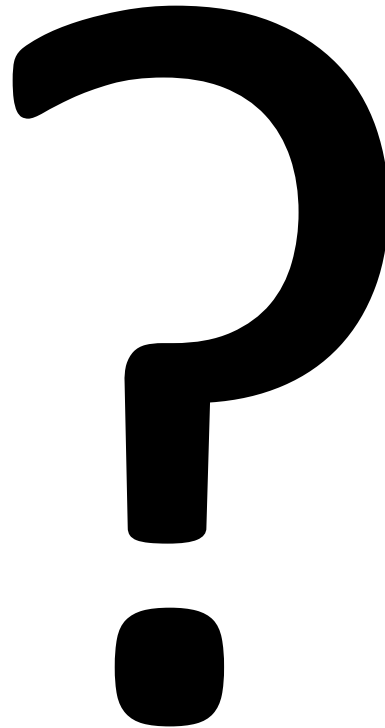
# Dagsorden

- Intro
- Canvas
- IT-forundersøgelser
  - Interview, Workflow, Medarbejder-Opgave-Mål tabel
  - Use Case, Klasser
- Mock up – Prototyper
  - Bemærk: Vi holder tænke højt test, når vi mødes næste gang

# Præsentation

- Anita Lykke Clemmensen
- Email: [alcl@ucn.dk](mailto:alcl@ucn.dk)
- Lektor, studievejleder – ansat siden 2012
- Master IKT og læring, datamatiker, lærer

Så er det jeres tur...



# Hvordan hænger fagene sammen?

## Virksomheden

Hvorfor bygge huset?

Hvad skal huset bruges til?



## Udvikling

Hvilket hus skal vi bygge?

Hvordan skal huset designes?



## Programmering

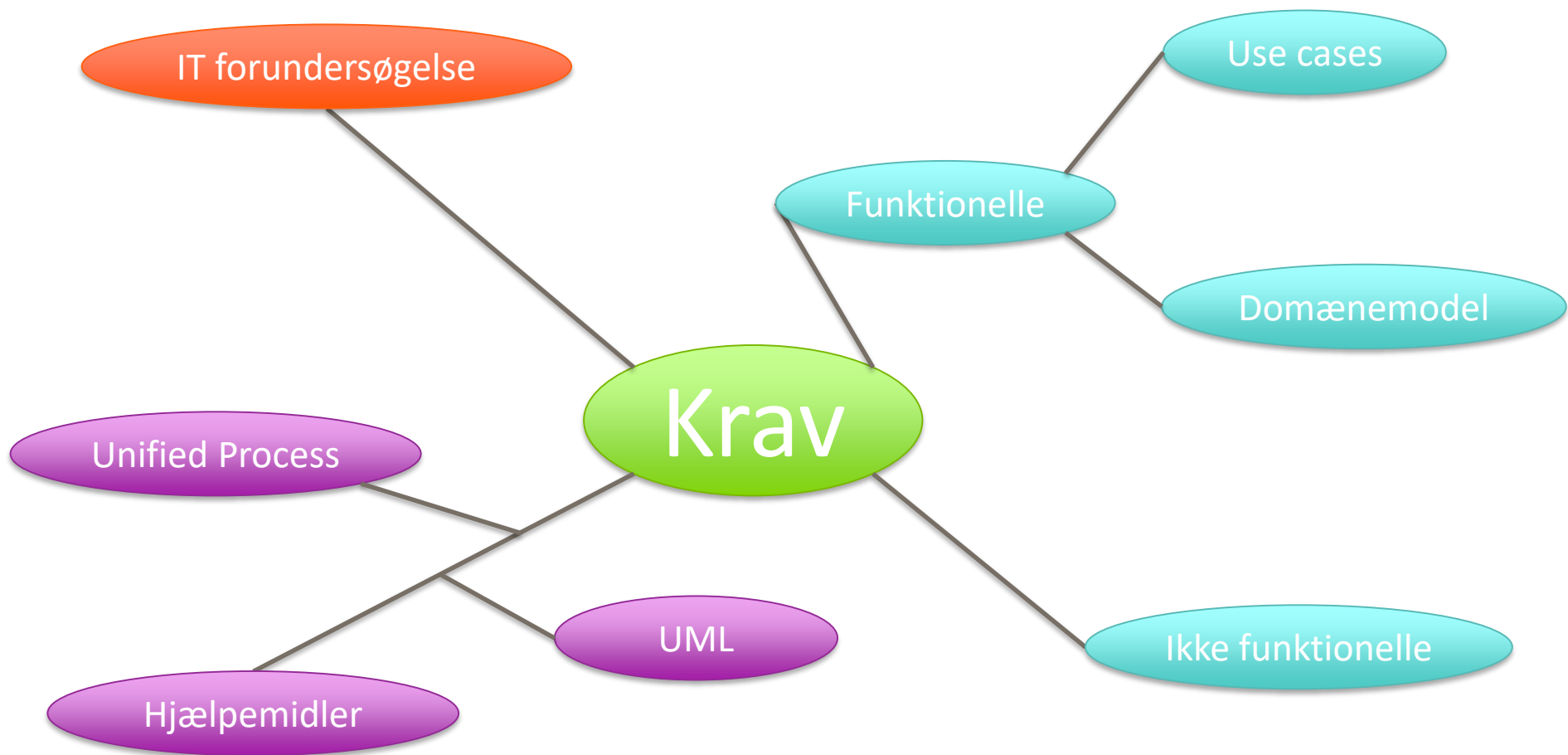
Byg huset



# Hvad er systemudvikling?

- Gå fra ide til færdigt system
- Find krav til systemet
- Analyser krav
- Design løsning (ikke visuel design)
- Realiser ideen
- Vedligehold løsningen

# Krav - sammenhæng



# Eksempel: aktiviteter og diagrammer

## Krav

### Klarlægge funktioner

Use case: find kunde

- 
- 1. En sælger ønsker at finde oplysninger om en kunde
- 2. Sælgeren angiver kunde id
- 3. Systemet returnerer kundeoplysninger

### Klarlægge ikke funktionelle krav

Systemet skal være brugervenligt

### Klarlægge krav til information

#### Customer

id  
name  
address

## Design

### Designe arkitektur

### Designe grænseflade (UI)

### Designe klasser

#### Customer

id : int  
name : String  
address : String

getName()  
getAddress()

### Designe database

#### Table: Customer

id	name	address
----	------	---------

## Realisering

### Skrive kode

```
public class Customer {  
    private int id;  
    private String name;  
    private String address;  
  
    public Customer (int id, String name,  
                    String address) {  
        this.id = id;  
        ...  
    }  
    public String getName() {  
        return name;  
    }  
}
```

### Teste



# Problemstilling

How Projects Really Work (version 1.0)

Create your own cartoon at [www.projectcartoon.com](http://www.projectcartoon.com)



How the customer explained it



How the project leader understood it



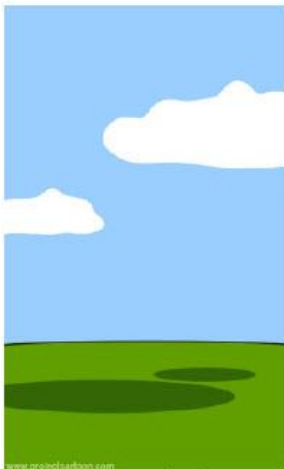
How the analyst designed it



How the programmer wrote it



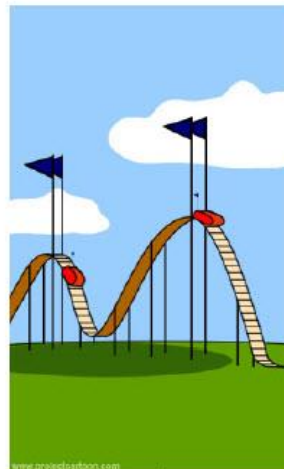
How the business consultant described it



How the project was documented



What operations installed



How the customer was billed















How it was supported



What the customer really needed

# Hvorfor struktur?

- Kunden får det han vil have!
- Systemet bliver færdig til tiden
- Ingen fejl
- Budgettet bliver overholdt
- Dokumentation
- Osv
- Ønsketænkning????

PROJEKT FORLØB	IKKE STRUKTURERET	STRUKTURERET
START		
30% AFTIDEN ER GÅET		
70% AFTIDEN ER GÅET		
100% AFTIDEN ER GÅET		
110% AFTIDEN ER GÅET		
200% AFTIDEN ER GÅET		

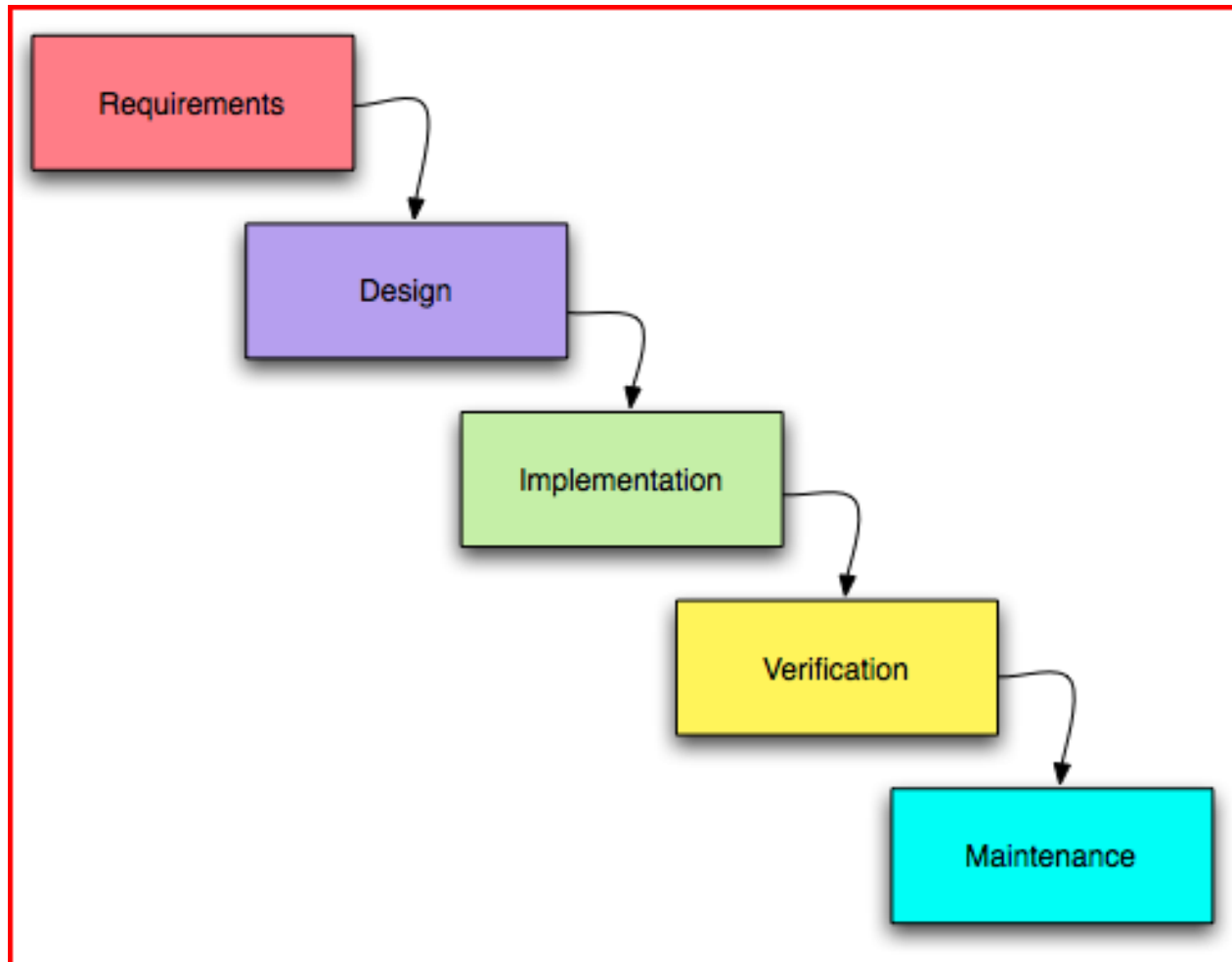
# Software Udviklingsproces

- Beskriver en tilgang til at designe, udvikle, udgive og vedligeholde software.
- En Software Udviklingsproces definere og beskriver aktiviteterne som skal udføres for at transformere brugerkravene til et IT system
- Proces til at kører et softwareprojekt

# Proces-tilgange (udviklingsmodeller)

- Vandfaldsmodel: Et sekventielt gennemløb gennem hovedaktiviteterne krav, design, implementering og test.
  - Et vellykket projekt kræver, at man kan beskrive kravene på forhånd
- Iterativ model: Mange små gennemløb gennem aktiviteterne krav, design, implementering og test (kalder iterationer)
  - De mest kritiske funktioner laves først
  - Derved reduceres usikkerhed
  - Tilpasninger sker løbende

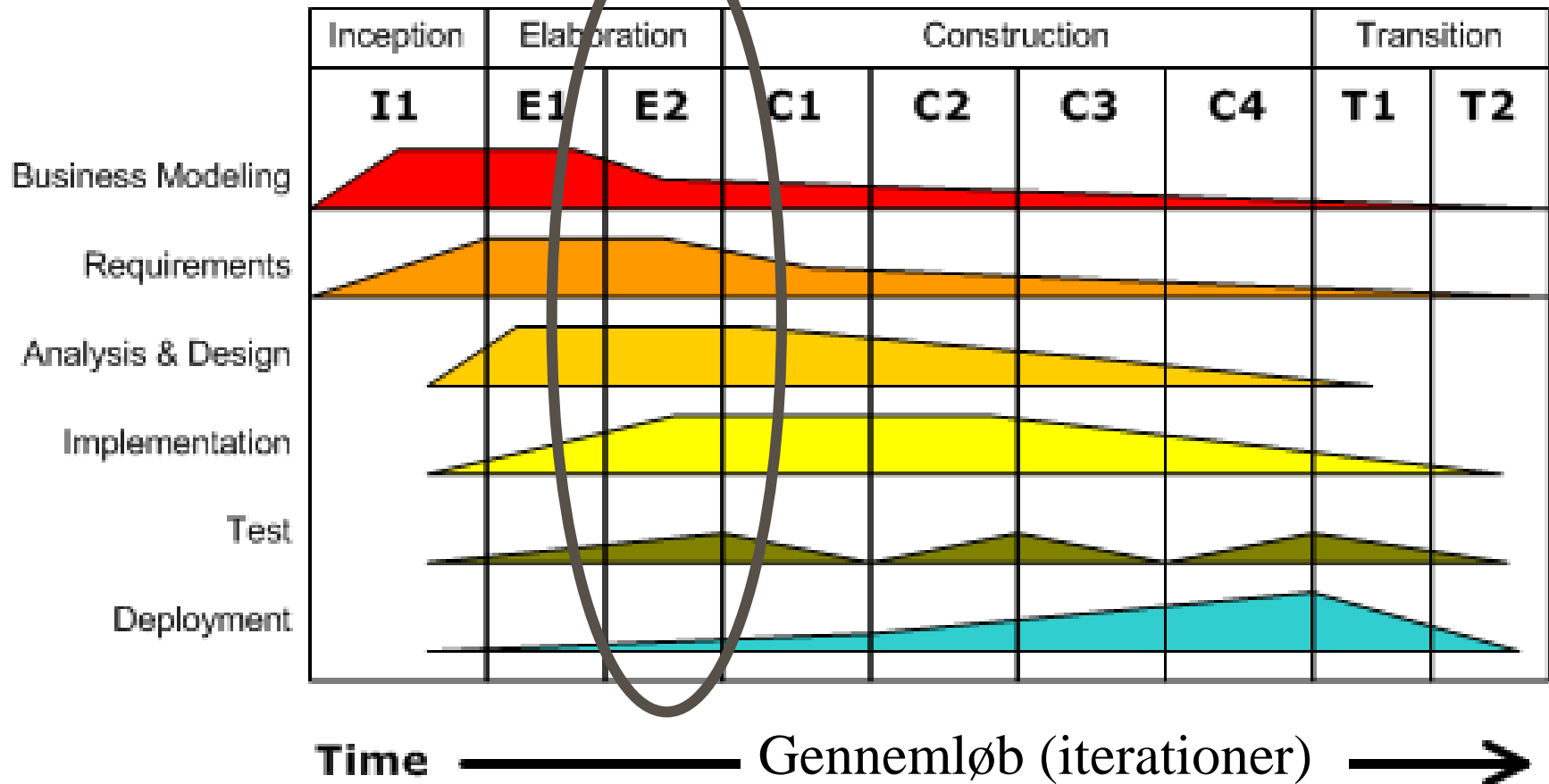
# Traditionel vandfaldsmodel



# Unified Process – UP.

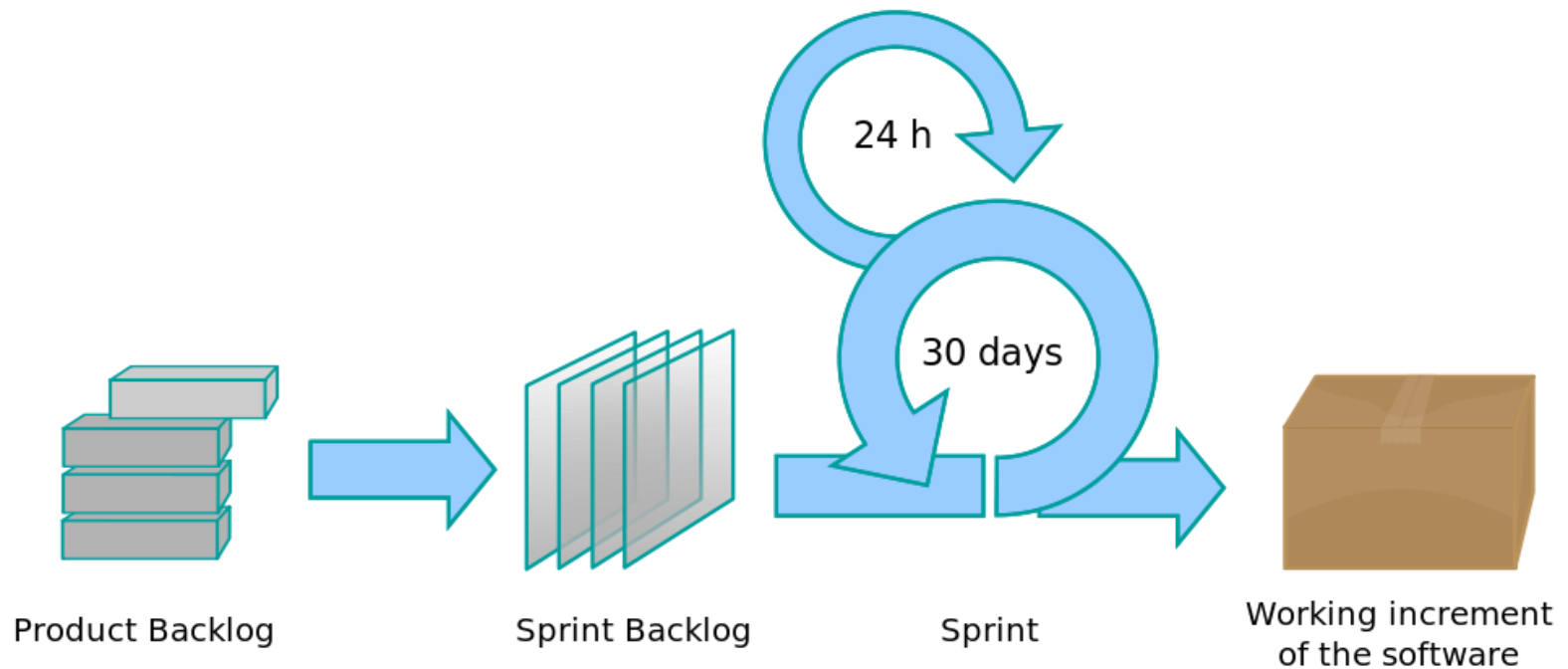
## Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



# SCRUM

SCRUM – iterative og agil proces



# Udviklingsmodel, metode og notation

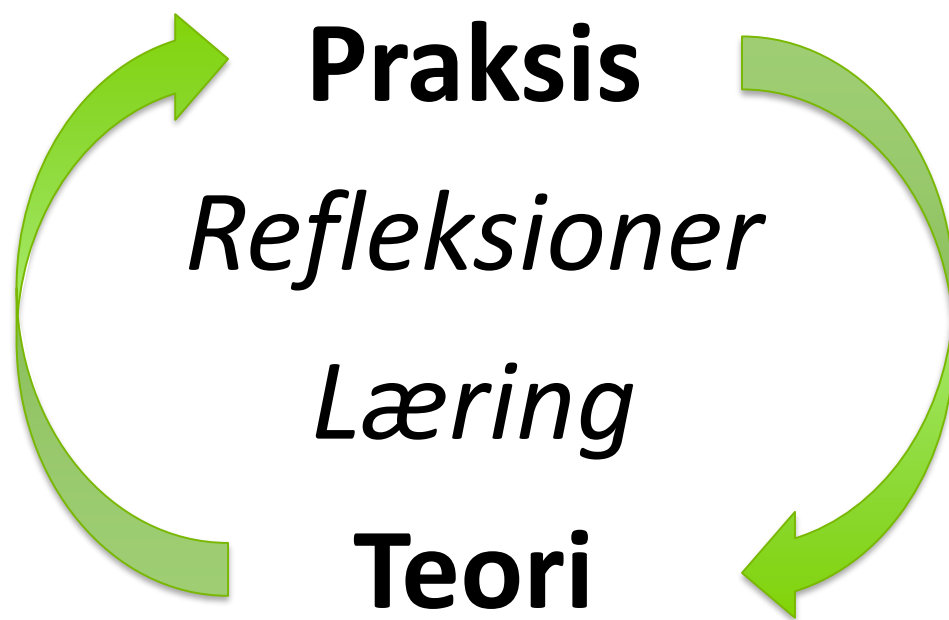
- En udviklingsmodel er en samling af aktiviteter, der fører frem til det færdige IT system
  - UP (Unified Process) er en udviklingsmodel
- En metode beskriver i detaljer hvordan de forskellige aktiviteter udføres og dokumenteres
- En notation er et sprog til beskrivelse og visualisering
  - Unified Modeling Language (UML) er en fælles standard for konstruktion af diagrammer til objekt-orienteret udvikling af softwaresystemer. Understøttes af forskellige case-værktøjer (vi vil her bruge UMLet)
  - UP og mange andre metoder benytter UML.



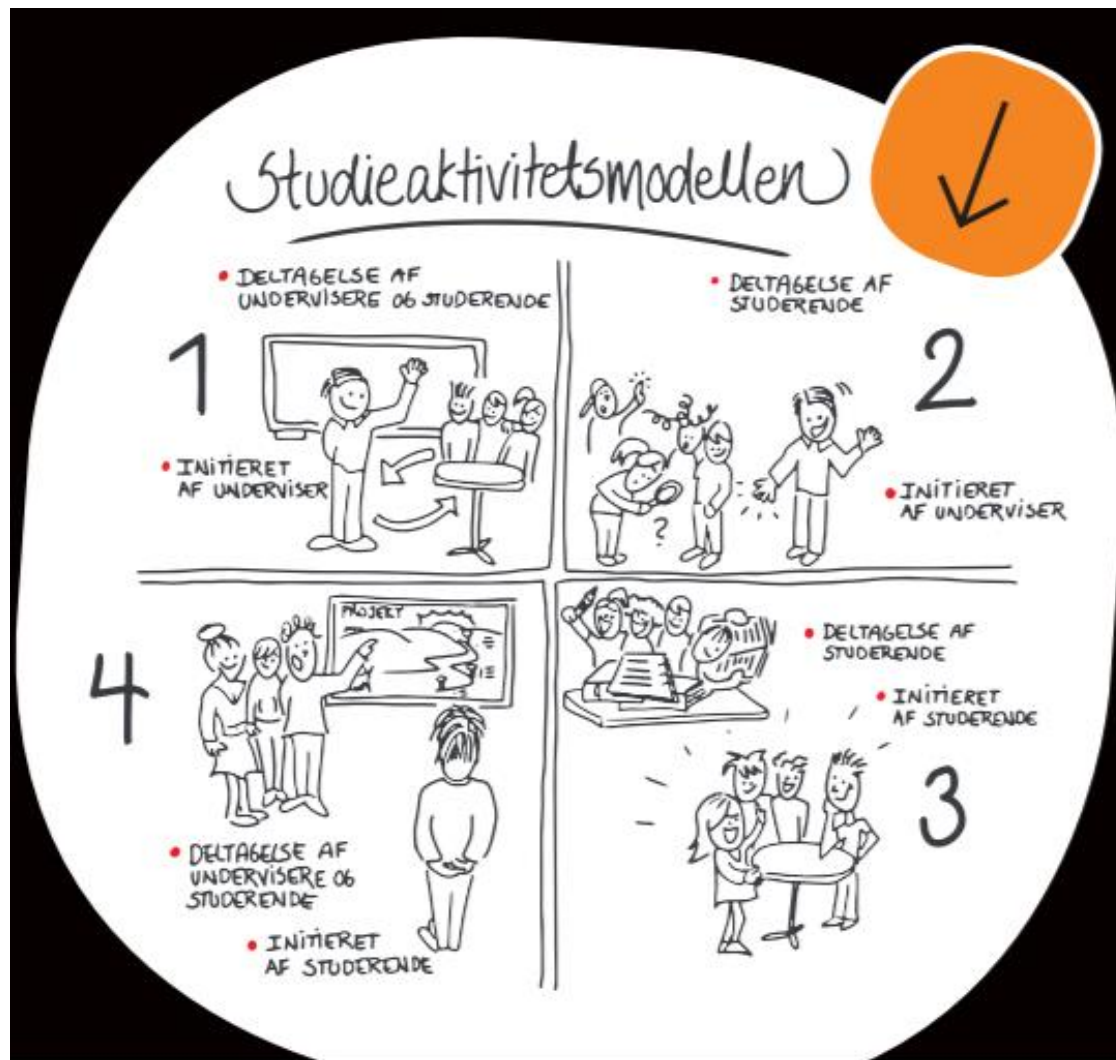
# Faget systemudvikling på 1. semester

- **Krav :**
  - Hvilke informationer skal håndteres i systemet (klasser)
  - Hvilke funktioner skal der være i systemet (use cases)
  - Hvilke krav er der til kvalitet (svartider, sikkerhed...)
- **Design løsning:**
  - Hvordan skal programmet overordnet struktureres (arkitektur)
  - Hvordan skal objekterne designes?
- **Realiser**
  - Programmering
- **Idriftsættelse**
- **Vedligehold**

## Refleksiv Praksislæring



# SAM



# Litteratur

- Applying UML and Patterns by Craig Larman
  - Larman har arbejdet med systemudvikling siden 1980'erne og B.S. og M.S. i computer science
  - Bogen bruges også på 2. semester – er et godt opslagsværk
- John W. Satzinger
  - Object-Oriented Analysis and Design
  - Udleveres som PDF
- Øvrig materialer ses i [Canvas](#)