

Studieteknik

1. Notatteknik
 - **Elev fremlæggelse** af notatteknik
2. Rapportskrivning
3. Projektplanlægning

Notatteknik: Elev fremlæggelse

En påmindelse

- Husk at upload jeres fremlæggelsesmateriale af notatteknik til itslearning.
 - Lokation af upload er:
Itslearning -> Studieteknik -> Planer -> dag 3 -> Aflevering - Kapitel fremlæggelse

Studieteknik

Rapport skrivning

Rapportskrivning – Dokumentet

- I får udleveret et oplæg for hvordan en rapport skal udarbejdes for jeres programmeringsstudie:
 - Hent dokument "Den gode rapport – oplæg.pdf" fra undervisningsplan for faget på itslearning (under dag 4) og referer til den når der er behov.
 - Dokumentet er et *forslag* til opbygning af rapporter over mindre programmeringsprojekter. At det er et forslag betyder, at man ikke nødvendigvis skal medtage ALLE punkter fra dokumentet med i sin rapport, kun de punkter aktuelle for jeres projekt.
 - I programmeringsstudie, dokumenter i jeres projekt og dens proces i en rapport som typisk afleveres sammen med en produkt (et styk software). *Det samme gælder i softwareudviklings erhverv, hvor man udarbejder en dokumentation.*
 - Man supplerer typisk sin rapport/dokumentation med
 - kode (i **source control repo**),
 - brugsvejledning, beskrivelser af produktet o.s.v. eventuelt (med **WiKi**).
 - Man kan dokumentere hvordan opgaverne bliver fordelt i gruppen med eventuelt en **board** (Kanban board).
 - Man anvender typisk UML og grafisk eksempler som hjælp til at beskrive sin produkt/proces/kode:
 - Under ide fase (**Inception**): Use case
 - Under design fase (**Elaboration**): mock-ups
 - Under implementation (**Implementation**): prototyping
 - Under test/deploy (**Test**): functional/non-functional testingFor at få overblik over de for ovennævnte faser, skal vi gennemgå **projektplanlægning**.
 - Brug dag 5 til at læse dokument "Den gode rapport – oplæg.pdf" igen, denne gang reflekter hvorledes og i hvilket punkt i dokumentet, at UML og udviklingsprocessen kan anvendes til at supplere rapporten.

Rapportskrivning

Krav til forside

Oftest skal den studerende selv formulere en projekttitel, der matcher projektets indhold.

- Det er vigtigt, at titlen er så klar og retvisende som muligt.

Det er vigtigt at følgende oplysninger fremgår af forsiden:

- Titel – og evt. undertitel
- forfatternavn(e)
- dato for aflevering
- Institution
- Opgavetype
- Fag
- Semester(skole forløb) og vejleder(eller lærer).

Studieteknik

Projektplanlægning

(Vandfaldmodel / V-model / Agile)

Softwareudviklingsmodeller

Model vs. proces

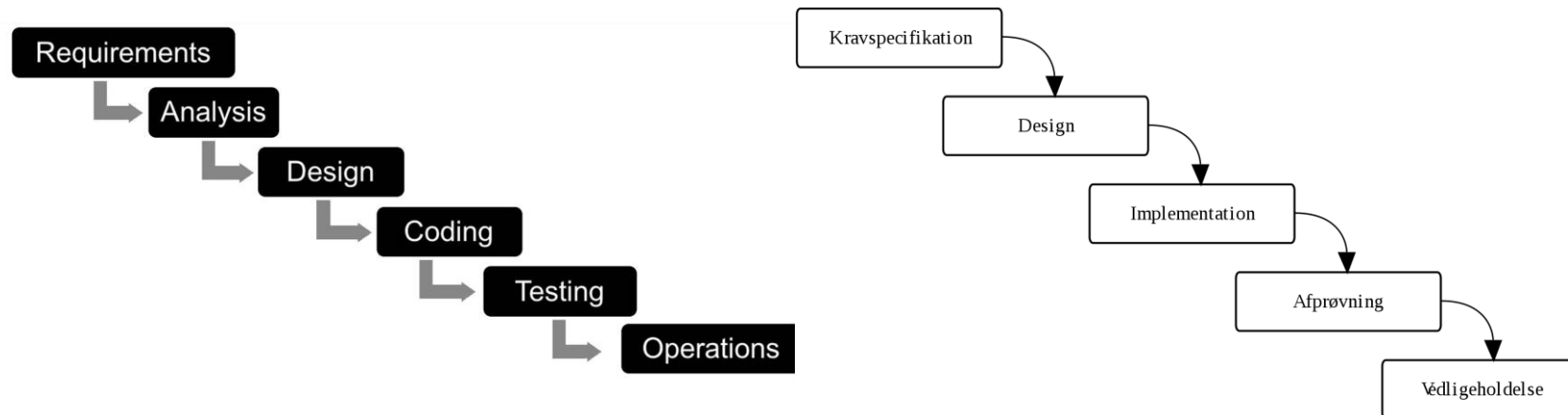
Model

- Udviklingsmodel er en abstraktion af udviklingsprocessen. 
- 'Generelt', alle kan sige at de implementer en model.*
- Modeller:
 - Vandfald
 - Agile

Proces

- Fysisk trin som skal gennemgås for at producere et stykke software.
- 'Custom', Meget afhængigt af virksomhedskultur.
- Grundlæggende process som definere mange modeller:
 - Unified Process

Eksempel, ens model med trinvis gennemførelse af opgaverne, men med forskellige processer:



Vandfaldsmodel

Den komplette proces er opdelt i flere faser. Den ene fase skal afsluttes for at nå den næste fase.

V-model

En udvidelse af vandfaldsmodellen. Det har en tilsvarende testfase for hver udviklingsfase. Derfor er der en tilknyttet testfase for hvert trin i udviklingscyklussen. Den tilsvarende testfase i udviklingsfasen planlægges parallelt.

Agile

En filosofi eller orientering baseret på værdier og principper. Disse værdier og principper er defineret i det agile manifest. Se det selv på: <https://agilemanifesto.org/principles.html>

Agile værktøjer:

- **Scrum**
 - Scrum's 3 roles: **ProductOwner**, **ScrumMaster** og **ScrumTeam**
 - Scrum's 3 ceremonies: **SprintPlanningWorkshop**, **DailyScrum**, **SprintReviewMeeting**
 - Scrum's 3 tools: **ProductBacklog**, **SprintBacklog** og **Project board**
 - sprints
 - Continuous delivery (**DevOps**)
- Lean
- **Kanban**
 - **Extreme programming**
- Crystal
- Dynamic Systems Development Method (DSDM)
- Feature Driven Development (FDD)

Grafisk eksempler af modellerne:

<https://github.com/TEC-joel/StudieteknikH1/wiki/Onsdag---plan%C3%A6gning-af-projekter,-opgaver-m.m.>

Konklusion på model vs. proces

“All models are wrong, but some are useful.”

- Opfylder aldrig 100% ens behov under udvikling.
- Vil altid blive udsat for tilpasning alt efter firmaets kultur, udviklingsteamets kultur, osv.
 - Derfor ser man aldrig organisationer som implementere Agile eller vandfaldsmodel med præcis samme processer.

Øvelse

Læs følgende tekst, brug det i overvejelsen når i skal besvar øvelse på næste slide.

Problemer ved vandfaldsmodellen:

Virkelighedens softwareprojekter følger sjældent et strengt sekventielt forløb - der vil næsten altid være nye krav, viden, teknologi, etc. der skal tages højde for. Men når kravene i et vandfaldsprojekt først er på plads, er projektet **mere eller mindre en "black box" for kunden**, indtil udviklingen er færdig. Dette gør det næsten **umuligt at inddrage arbejdet med UX og usability** i sådanne processer, da der **ikke er "working software" før til slut i projektet** og derfor **ikke noget at teste på**. Desuden er der heller **ingen garanti for at kunden får det ønskede**, da **intet kørende system eksisterer før til slut** i projektforløbet. Derfor er risikoen ved vandfaldsprojekter også utrolig høj.

Agil udvikling er mindre risikabelt

Agil udvikling gør op med disse risici ved at have fokus på transparens, omstillingsparate teams og kontinuerlig levering af *"working software"*. Fordelene ved agil udvikling er evnen til at reagere hurtigt på ændrede krav i projektet – både fra kunden, men også fra markedet. Risikoen ved et agilt projekt er derfor ikke så høj som ved et vandfaldsprojekt, da der pga. transparens og kontinuerlig levering, maksimalt vil være tale om en iteration, der skal kasseres.

[Kilde : <https://www.itu.dk/om-itu/presse/nyheder/2017/hvordan-kommer-man-fra-vandfald-til-agil-ux>]

3 udviklings scenarier:

Med tekst fra forrige slide som udgangspunkt, besvar udefra de 3 udviklings scenarier vist for neden, hvilket passer bedst til udvikling mod vandfaldsmodellen og hvilket passer til Agile (eller skal alle 3 være Agile? Giv en begrundelse).

1. Jeg(eller firmaet) udvikler software til eget intern brug. Der anvendes kun en enkelt udvikler til formålet.
2. En stor gaming firma skal udvikle et nyt spil til spillemarked som firmaet forventer at blive rigtigt populære.
3. Firmaet skal udvikle en app som en udvalgt gruppe af befolkning skal anvende. Firmaet afgøre selv design hvor brugerne bare selv må lære at bruge app'en gennem en serie af vejledninger. (f.eks. itsLearning applikation)

Home

no-tec-cit edited this page now · 15 revisions

Studieteknik

[Se målpind](#)

Markdown: [Tutorial](#)

- [Test](#)
- [Italics and Bold](#)
- [Headers](#)
- [Links](#)
- [Images](#)
- [Blockquotes](#)
- [Lists](#)
- [Paragraphs](#)

Softwareudviklingsmodeller:

- [Besvar her.....](#)

Til besvarelse:

Opret en ny overskrift (Softwareudviklingsmodel) på jeres Wiki side.

Opret et "bullet" punkt for hver besvarelse under overskriften.