

Vejledning til udviklingsprocessen for semesterprojekt 3 (PRJ3)

Indholdsfortegnelse

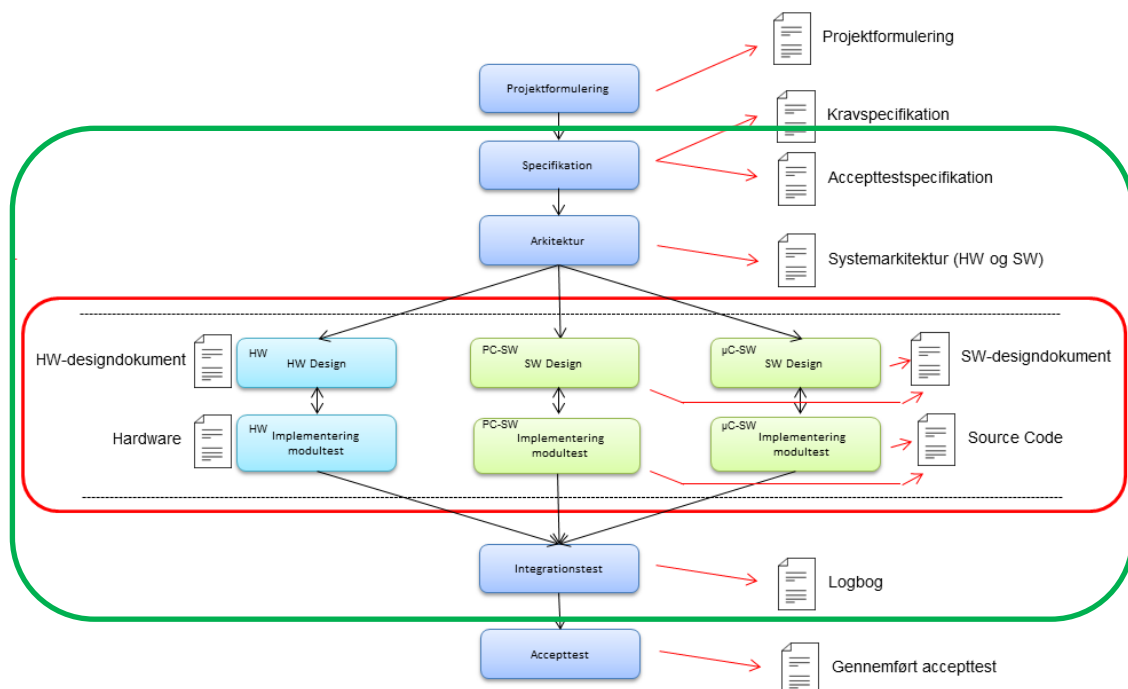
Indledning.....	3
Baggrund	3
Iterativ udvikling med ASE-modellen	4
Udvikling i PRJ3.....	6
Milestones	6
Udviklingsfaser	6
Inception.....	6
Elaboration	6
Construction	7
Metoder til iterativ udvikling.....	7
Kanban.....	7
Scrum.....	7
Eksempel på brug af Scrum i PRJ3.....	8
Referencer	8

Indledning

Denne vejledning har til formål at beskrive den udviklingsproces, som anvendes på 3. semesters semesterprojekt (PRJ3). Projektet er fælles for E-, EP- og IKT-studerende på diplomingeniøruddannelsen på Ingeniørhøjskolen Aarhus Universitet. Fokusområdet er udvikling og implementering af et selvvalgt produkt, med udgangspunkt i en given hardwareplatform. Udviklingsprocessen gennemføres ved hjælp af en iterativ metode og samtidigt under anvendelse af procesmetoden, som er lært på studiets 2. semester. Projektets indhold er beskrevet detaljeret i projektoplægget for PRJ3 [1].

Baggrund

”ASE-udviklingsmodellen” [2], som er vist i figur 1, er introduceret på 2. semester og giver en struktureret model for udvikling af et system bestående af hardware og software.



Figur 1: ASE-Modellen (Kilde: Kim Bjerger, Vejledning til udviklingsprocessen for projekt 2)

Modellen er velegnet til udvikling af software og hardware for projekter for hvilke man har et godt domænekendskab inden projektet startes op. Med et godt domænekendskab er det muligt at beskrive præcise krav, formulere tests og designe komplette use cases på et tidligt tidspunkt. Når arkitekturen er fastlagt, kan der i 2. semesterprojektet (PRJ2) arbejdes ”begrænset” iterativt på design, implementering og modul-test indtil den ønskede funktionalitet er opnået. Som helhed er udviklingsprocessen på 2. semester inkrementel, den er ikke som helhed iterativ. Den del af udviklingsprocessen på 2. semester, der er iterativ, er markeret med det røde rektangel på Figur 1.

Ved at trække en gennemført, iterativ arbejds metode ned over ASE-modellen, kan vi gøre ASE-modellen mere robust overfor manglende domænekendskab og fremtidige tilpasninger. Dette er hensigten med 3.

semesterprojektet PRJ3. Den gennemførte, iterative arbejdsmetode åbner mulighed for at tilføje/ændre krav og udvide/ændre systemets arkitektur flere gange gennem projektets forløb. Denne iterative arbejds- metode er markeret med det grønne rektangel i Figur 1.

Iterativ udvikling er kendetegnet ved at man gennemfører sit projekt i små bidder (iterationer). Hver itera- tion indeholder elementer fra flere af processerne beskrevet i ASE modellen. En iteration kan på PRJ3 der- for indeholde både krav, arkitektur, design, udvikling og test, men kun for en lille bid af projektet.

Semesterprojektet på 3. semester indeholder mange ubekendte og det er derfor en god ide at bruge en iterativ arbejdsmetode konsekvent.

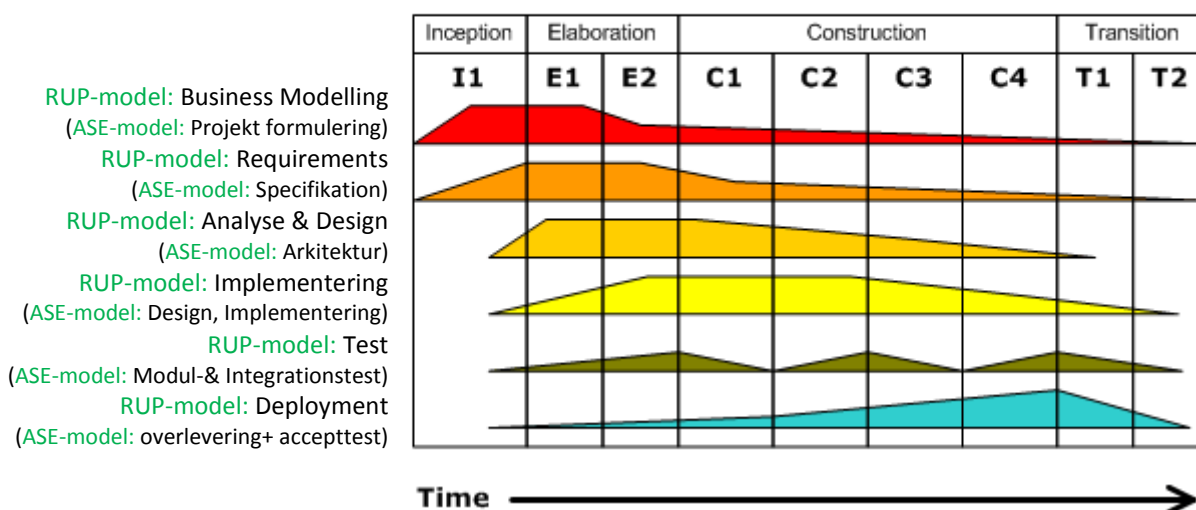
Iterativ udvikling med ASE-modellen

Selvom I arbejder iterativt, har projektet stadig forskellige faser. Processerne omkring ideudvikling og krav- specifikation betones meget i de første faser, som vist i figur 2, mens detailudvikling fylder mindre. Beto- ningen forandrer sig gennem projektforløbet til at være mere udviklingsorienteret og senere orienteret mod produktoverlevering. Forskellen fra den oprindelige ASE-model i figur 1, er at I arbejder parallelt på flere processer indenfor hver iteration. I gør dette for at bygge og forbedre på baggrund af erfaringer og erkendelser fra tidligere iterationer.

Vi indleder for eksempel projektet med at opstille de første og vigtigste krav for projektet. Når vi har styr på disse, begynder vi at grave os ned i problemområdet, f.eks. "Hvordan måles en afstand uden berøring?", "Hvordan implementeres et GUI på en embedded platform?"

Iterative Development

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



Figur 2: Udviklingsfaser og Iterationer. ASE termer i parantes (Kilde: Wikipedia, Iterative and incremental development)

Et iterativt projektforløb kan indledes som vist i figur 3, som viser RUP-udviklingsmodellen (RUP = Rational Unified Process). Figuren er inspireret af Larman: "Applying UML and Patterns", kap. 6). Bemærk hvorledes discipliner og artefakter mapper til ASE-modellen i figur 1. Bemærk desuden at der arbejdes på næsten alle

discipliner i hver iteration og at I allerede efter et par uger kan "prøve noget af". Dokumentation udvikler sig i løbet af projektforløbet og skal derved opdateres løbende.

Iteration		1	2	3	4..n
Mål		Klarhed om projekt ide samt de vigtigste krav	Opnå erfaring med problem-domænet for konkretisering af krav	Tidlig implementering og stabilisering af krav	Design, implementering og test af funktionalitet
Længde		1 uge	2 uger	2 uger	2 uger
Disciplin	Artefakt	Inception 1	Elaboration 1	Elaboration 2	Construction 1..n
Projekt Formulering	Projekt-formulering	Udled projekt-formulering. Anvend MosCoW til at identificere og prioritere krav.			
Specifikation	Kravspecifikation & Accept-test	Udvælg få "Must-have" krav der detaljeres som Use-Cases og ikke-funktionelle krav. Disse skal være dem som tilføjer projektet mest værdi	Fasen afsluttes med en opdatering af krav. Halvdelen af Use-cases kan nu præciseres. Hardware blokke (BDD) og eksterne interfaces kan ligeledes præciseres.	Afsluttes igen med opdatering af krav og præcisering af yderligere Use-Cases. Ved Scrum er backlog ved at tage form nu.	Krav opdateres om nødvendigt, men disse burde være rimelig stabile nu. Ved Scrum re-prioriteres backlog.
Arkitektur (System Analyse)	System-arkitektur	Ingen	Analysér hvad som skal til (hw/sw/mek) for at kunne implementere de få udvalgte krav	Analysér for de næste krav og opdater system arkitektur med BDD, eksterne interfaces og SW arkitektur	Opdater ved ændringer i sw arkitektur, eksterne interfaces.
Design	Design Dokumentation	Ingen	Lav design for de få udvalgte, men essentielle krav	Design for de næste krav	Design for de næste krav, medtag erfaringer fra sidste iteration
Implementering og modul test	HW/SW/Mekanik	Ingen	Implementér disse. vha. mock-ups, eval boards, virtual machines, emulatorer etc. Således at vi får et proof-of-concept	Implementér og modultest disse. 5% bygget	Implementér og test disse. xx% bygget
Projekt Styling	Projekt plan	Udled grov overordnet plan	På baggrund af erfaringerne opdateres projektplan	Projektplan kan opdateres og er mere solid	

Figur 3: Eksempel på iterativ gennemførelse af projekt

Udvikling i PRJ3

Semesterprojektet på 3. semester udvikles med de processer og artefakter, som også gjorde sig gældende i semesterprojektet på 2. semester.

Det iterative arbejde skal dokumenteres og det er vigtigt at I for hver opstart af en iteration gør jer klart hvad målet for den pågældende iteration er, samt hvilke opgaver som skal søges løst af hvem. Ligeledes er det ved afslutning af hver iteration vigtigt at gøre sig klart, hvad der er færdigt (dvs: specificeret, designet, implementeret og testet), og hvad der ikke er færdigt, inden I planlægger næste iteration.

Det anbefales at anvende en struktureret metode til at implementere den iterative udvikling. Der findes mange forskellige metoder og det er op til projektgruppen at vælge en passende metode. Et eksempel er Scrum, som beskrives i afsnittet ”Metoder til iterativ udvikling”.

Milestones

Der er følgende milestones for PRJ3 (relativt til semester start):

- Uge 2: Gruppedannelse afsluttet
- Uge 3: Projektformulering afleveret til vejleder (efter *inception1*)
- Uge 7: Review af Kravspecifikation, Accepttestspecifikation og Systemarkitektur (efter *elaboration2*)
- Uge 9: Review af Implementation og Proces (efter én *construction* iteration)
- Uge 14: Aflevering af projektartefakter

Udviklingsfaser

Udviklingsarbejdet tager udgangspunkt i Figur 3 og anvender termene herfra.

Inception

I denne første fase af projektet forventes det at projektgruppen når frem til en grundidé for projektet som ønskes gennemført.

Projektformuleringen skal formuleres som kendt fra vejledningen for 2. semester[2]. Til forskel fra 2. semester skal projektformuleringen også indeholde en MoSCoW analyse, som præciserer projektets mål, dvs. hvad er vigtigt og hvad er mindre vigtigt.

Projektformuleringen skal sammen med en første overordnet tidsplan godkendes af gruppens vejleder ved afslutning af iterationen.

Kravsspecifikationen påbegyndes som beskrevet i Figur 3.

Elaboration

Denne næste fase har som mål at gøre projektgruppen bekendt med problemområdet, samt at få en forståelse af hvilke teknologier og metoder som der skal arbejdes med. Fasen gør det desuden muligt bedre at vurdere kompleksitet og omfang – og dermed risici i projektet. Fasens iterationer tager udgangspunkt i centrale use cases og ikke-funktionelle krav, hvis mulige løsninger undersøges ved hjælp af f.eks. mock-ups, tests på PC etc.

Til forskel fra 2. semesterprojektet forventes det ikke at man har en komplet og færdig kravspecifikation før arbejdet med systemarkitektur påbegyndes. Som beskrevet i

Figur 3, udvikles kravspecifikationen over tid. I hver iteration udvælges de vigtigste, ikke implementerede krav fra projektformuleringen. Disse krav præciseres efterhånden i kravspecifikationen som fully-dressed use-cases og ikke-funktionelle krav - og som diagrammer, modeller og beskrivelser i systemarkitektur dokumentet.

Efter *Elaboration* fasen bør kravspecifikationen og systemarkitekturen være rimelig stabil.

Construction

I construction fasen er fokus at udvikle produktet. Fasen opdeles i iterationer af 2-3 ugers varighed. Før iterationen påbegyndes, planlægges hvilke opgaver, som søges løst. Hver opgave skal løses helt i løbet af sprintet, dvs. opgaven skal designes, implementeres og modultestes. Skal opgaven/delkomponenten integreres sammen med andre komponenter, bør dette også ske løbende, evt. som en ny opgave i den efterfølgende iteration.

Det er en stor fordel at definere et klart mål, som skal demonstreres ved afslutningen af en iteration. Det er langt sjovere og mere overskueligt f.eks. at udvikle en funktionel, enkel GUI som kan styre en simpel LED, end at arbejde mod at implementere 10% mere af et totalt, men uoverskueligt system.

Metoder til iterativ udvikling

Der findes mange forskellige metoder til at udvikle iterativt. Metoderne beskriver arbejdsgange og typisk ikke et helt procesforløb. Nogle af de mest populære metoder pt. er Kanban og Scrum, som er beskrevet i det følgende.

Kanban

Er den oprindelige Toyota udviklingsmodel. Den har de seneste år fået en renæssance indenfor softwareudvikling, da det er en meget let model, som er nem at komme i gang med. Gode referencer er:

- Wikipedia: Kanban(development) – Fin introduktion. Se også henvisninger i bunden af siden (https://en.wikipedia.org/wiki/Kanban_%28development%29)
- OpenKanban - Introvideoer til Kanban udvikling og adgang til open source værktøj (<http://agilelion.com/agile-kanban-cafe/open-kanban>)

Scrum

Metoden udviklet til softwareudvikling i 1990'erne af Ken Swaber og Jeff Sutherland. Gode referencer er:

- The Scrum Guide - Guide af de oprindelige forfattere, nemt læst og overskuelig (<http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>)
- Mountangoat Software – Har et fint lille overblik over Scrum på deres hjemmeside (<http://www.mountangoatsoftware.com/agile/scrum/overview>)
- Google Tech Talk: Scrum et al - Video med Kenn Swaber som fortæller om Scrum (<https://www.youtube.com/watch?v=lyNPeTn8fpo>)

Eksempel på brug af Scrum i PRJ3

PRJ3 er et typisk In-House udviklingsprojekt, hvilket vil sige at det er firmaet selv som definerer produktets mål. *Product Owner* rollen varetages derfor ikke af en ekstern kunde, men af teamet selv sammen med vejleder.

Den daglige rolle som *Scrum Master* bør varetages af et team medlem, som er i tæt kontakt med udviklings-teamet, hvilket vil sige et medlem af projektgruppen. Rollen kan evt. gå på skift mellem projektdeltagere.

Vejledningssmøderne afholdes typisk i forbindelse med sprint afslutning og indeholder følgende elementer:

- *Demonstration*, hvor gruppen demonstrerer hvad som er opnået i det forgangne sprint.
- *Sprint Retrospective*, hvor vejleder og gruppe ser tilbage på forløbet af det forgangne Sprint og identificerer hvordan det kan gøres bedre i den næste. Det være sig teknisk, procesmæssigt og interpersonelt.
- *Sprint Planning*, hvor vejleder og gruppe prioriterer tasks til den kommende Sprint samt evt. tilføjer nye tasks. Her skal sættes et klart og håndgribeligt mål for det kommende sprint. Eks: Detektér input og tænd LED.

Forud for mødet skal projektdeltagerne have afholdt et *Sprint Review* og opdateret *Product Backlog*, således at gruppen er klar til at planlægge næste sprint med vejleder. Opgaver som ikke er afsluttede i nærværende sprint, flyttes tilbage til *Product Backlog*'en.

Det anbefales at afholde flere ugentlige stå-op Scrum møder, f.eks. i pauserne eller evt. via Skype. Formålet med møderne er at ALLE ved hvad som sker på projektet og at opfange misforståelser, f.eks. omkring interfaces (HW/SW), deadlines og opgaver med mere.

Referencer

[1] Ingeniørhøjskolen Aarhus Universitet, "Projektoplæg for semesterprojekt 3", August 2015

[2] Ingeniørhøjskolen Aarhus Universitet, "Vejledning for gennemførelse af projekt 2", August 2015