

Projektværktøj

Mikkel Schneider Larsen

Kongens Lyngby 2005
IMM-B.Eng.-2005-50

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Summary

Project management might sound like a dry and boring subject. Nonetheless it is not just about optimizing a static system. It is an exiting process. Project management is more than just boring administration. It is about humans and how to gain the best performance by creating the right conditions. In larger projects the role of the project manager, is more administratively focused. It will be necessary to standardize the communication between the individual project participant, the project leader and the top management. This report is about the start up of a development process. It is wanted to construct a tool for standardization of some of the assignments of the project leader. The report tries to show a picture of what a project really is. The report also works as documentation of the work done today. It was chosen to work in iterations using Unified Process. This report is written in Danish.

Resumé

Projektstyring kan måske lyde som et temmelig tørt og kedeligt emne. Ikke desto mindre drejer det sig ikke bare om at optimere på et statisk system. Det er en spændende proces. Projektstyring er mere end bare kedelig administration. Det handler om mennesker, og om hvordan der opnås den bedste ydelse ved at give de rette betingelser. I større projekter vil projektlederens rolle dog være lidt mere administrativt præget. Det vil blive nødvendigt at standardisere kommunikationen mellem den enkelte projektdeltager, projektlederen og den øverste ledelse. Denne rapport handler om opstarten af et udviklingsforløb. Der ønskes konstrueret et værktøj til at standardisere nogle af projektlederens arbejdsopgaver. I rapporten forsøges at give et billede af, hvad et projekt egentlig er. Desuden fungerer rapporten som dokumentation over det per dags dato udførte arbejde på projektet. Det er valgt at arbejde iterativt ud fra "Unified Process". Denne rapport er skrevet på dansk.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfilment of the requirements for acquiring the Bachelor degree in engineering.

The thesis deals with a practical example of how one may implement a project management tool. The main focus is helping the project manager to maintain the overview. The tool is simple, and focused on solving the particular task, in order to maintain user friendliness. Standard forms and reports are used to automate the flow of communication.

The time frame of this project is 10 weeks. It is therefore not realistic to have a product ready for release (although a small scale system ready for use has been implemented). The project should not necessarily be viewed a final product, but as a foundation to build further on.

The report is written in Danish, since this is the first language of the author.

Nærum, January 2006



Mikkel Schneider Larsen

Bilag vedlagt

- A X_1 ¹
Project Tool - Kort Specifikation
- B Mikkel Schneider Larsen
Begrebs og ordliste for diplom eksamensprojektet
- C Mikkel Schneider Larsen
CD med tidsplan, kode, køreklart program, sql, mm.

¹Personen har ønske at være anonym. Personen kaldes derfor X_1 . Der vil blive refereret til dennes arbejdsplads med X_2 af samme årsag.

Taksigelser

Tak til min vejleder, Edward, for at hjælpe mig til at vælge det rigtige eksamensprojekt. Det har været en lang og besværlig proces, med firmaer og et personligt tidspres. Jeg kom igennem med moralen i behold, takket være dig. Jeg er taknemmelig for, at du har båret over med min frustration. Uden din tålmodighed var jeg, med garanti, løbet ind i store vanskeligheder.

Tak til min elskede Louise, fordi du udviser tålmodighed med en distræt kæreste. Du ved heldigvis selv hvordan det er at blive ingeniør.

Tak til dem der har hjulpet mig, men som af den ene eller anden årsag ikke nævnes her.

Indhold

Summary	i
Resumé	iii
Preface	v
Bilag vedlagt	vii
Taksigelser	ix
1 Indledning	1
1.1 Problemformulering	1
1.2 Motivation	2
1.3 Projektet generelt	3
1.4 Projekt begreber	7
1.5 Sektionsopdeling	12

2 Business Model	15
2.1 Konklusion på modellen	17
3 Kravspecifikation	19
3.1 Indledning - hvordan er projektet blevet grebet an?	19
3.2 Hvad fokuseres der på i forhold til domain-modellen	19
3.3 Definition af kravene i forhold til IEEE standard	20
3.4 Use cases	22
3.5 Konklusion på kravspecificeringen	26
4 Design	27
4.1 Det grafiske design	27
4.2 Design patterns	29
4.3 Klasser og filer	29
4.4 Databasen	39
5 Implementering	45
6 Testning af systemet	47
7 Brugervejledning	49
7.1 Nødvendige værktøjer	49
7.2 Sådan virker programmet	49
8 Konklusion	57

Figurer	xiii
<hr/>	
A Project Tool - Kort Specifikation	59
A.1 Målsætning	60
A.2 Overvejelser i forbindelse med modelvalg.	60
A.3 Sheets	61
B Begrebs og ordliste for diplom eksamensprojektet	65

Figurer

1.1	Viser hvordan et projekt typisk foregår.	6
1.2	Viser brainstormdiagram over emnet projektstyring.	8
2.1	Domain Model	16
3.1	Use case diagram	25
4.1	Hvordan hovedvinduet er designet	28
4.2	UML klassediagram for selve programmet	31
4.3	UML klassediagram for generel projektinformation	32
4.4	UML klassediagram for assignlib	33
4.5	UML klassediagram for helpprojectmanager namespace	33
4.6	UML klassediagram for SaveYesNo dialogen	34
4.7	UML klassediagram for newproject	34
4.8	UML klassediagram for openproject	35

4.9	UML klassediagram for MyCommand (command pattern)	36
4.10	UML klassediagram for gant namespace	37
4.11	UML interaktionsdiagram for gant namespace	38
4.12	assignedhr som ER	39
4.13	assigneditems som ER	39
4.14	items som ER	40
4.15	personnel som ER	40
4.16	projects som ER	41
4.17	tasks som ER	41
4.18	Databasen i sin helhed	42
4.19	Xml visning i Visual Studio	43
5.1	Sådan er hovedvinduet implementeret	46
5.2	Hvordan about er designet	46
5.3	Sådan er filteret implementeret	46
7.1	Opstart af program	50
7.2	Nyt projekt	50
7.3	Klar til indtastning	51
7.4	Vælg en dato	52
7.5	Save as	53
7.6	Parat til task	53
7.7	Indtast tasks	54

7.8 Åben xml projekt 55

KAPITEL 1

Indledning

Dette kapitel har til formål, at give læseren en general introduktion til den problemstilling, der behandles rapporten igennem. Kapitlet er delt op i fem afsnit. I afsnit 1.2 gøres kort rede for, hvorfor det er valgt at arbejde med et projektstyringsværktøj. I afsnit 1.3 på side 3, gives en generel introduktion til projektet. I afsnit 1.4 på side 7, identificeres generelle begreber omkring projekter. I afsnit 1.5 på side 12, forklares lidt om hvordan det er valgt at opdele informationer og begreber. Desuden kan den egentlige problemformulering findes i afsnittet 1.1. Målgruppen for kapitlet er læsere, der ønsker en introduktion til de tanker, der danner fundament til specificeringen og designet af projektværktøjet.

1.1 Problemformulering

Her følger det overordnede problem, som skal løses rapporten igennem: Hvordan konstrueres et software produkt eller værktøj, som kan hjælpe projektlederen med at udføre dennes opgaver. Produktet skal ikke være begrænset som det i afsnit A på side 59 beskrevet regneark. Hvilke informationer er der brug for? Produktet skal defineres således, ud fra brug af use cases¹. Det bør fremgå hvorfor

¹En form for kravdefinition kendt fra Unified Process.

en person behøver noget given information. Der lægges vægt på, at projektlederen hurtigt kan danne sig overblik, hvorfor en form for grafisk fremstilling er nødvendigt. Værktøjet skal på sigt hjælpe med at identificere risici i form overbelastning.

1.2 Motivation

Hvorfor er det valgt at arbejde med et projektstyringsværktøj? Salget og brugen af projektstyringsværktøjer udgør et relativt lille marked. Dette er lidt mærkværdigt, eftersom der ofres penge på dygtige projektledere. Den primære motivationsfaktor i forhold til at udvikle et projektstyringsværktøj kommer af troen på, at det i fremtiden er et marked der vil vokse. Det er forfatterens opfattelse, at de programmer der findes på markedet ikke giver optimal udnyttelse af ressourcer. Der er en række hovedkriterier, som det bør forventes at kunne finde i et værktøj til styring af projekter generelt.

Først og fremmest ønskes det at undgå den fælde, som store projektværktøjer falder i. Problemet består først og fremmest i, at brugere har forskellige krav til hvad værktøjet skal kunne. En projektleder knyttet til et softwareprojekt ønsker dels værktøjer til de overordnede projektt tekniske emner (læs planlægning og risikohåndtering), samt egenskaber, som bruges inden for software industrien. Hvis en værktøj forsøger at løse begge disse kategorier af problemer, vil det indeholde egenskaber, som ikke bruges af en projektleder indenfor byggebranchen. Nogen vil hævde, at dette ikke gør noget, da værktøjet kan udvides til også at dække byggeprojekter. Problemet i dette er imidlertid, at kunder ikke gider betale for egenskaber, som de alligevel ikke bruger. Desuden vil det, når programmet når en vis størrelse, være vanskeligt at finde frem til de egenskaber, som den enkelte bruger ønsker. Nogen software virksomheder forsøger at løse disse problemer, enten ved at lade brugere kunne købe moduler til en grundpakke/installation af deres produkt, eller bruge filtre².

Det kan imidlertid være svært som kunde at vælge hvilke moduler, denne har brug for. En dygtig sælger vil hurtigt kunne sælge moduler, som kunden aldrig kommer til at benytte. En software virksomhed skal ligesom alle andre virksomheder tjene penge. Derfor forsøges, der at udvikle eksisterende produkter til at have egenskaber, som ikke oprindeligt var tiltænkt. Et produkt sælges ved at have en lang liste med egenskaber. En fast kunde opnås ved at levere de egenskaber, som denne har brug for.

²Med filter menes en eller anden mulighed for at fravælge visning af bestemte kategorier af information.

Forfatteren tror at software værktøjer i fremtiden vil dele sig i to grupper. Store produkter med mange, både nyttige og unyttige, egenskaber vil kunne sælges til store virksomheder. Beslutningstageren i forbindelse med indkøb af produkter vil i større virksomheder sjældent være den, der rent faktisk benytter produktet. En sådan person beslutter sit køb ud fra hvordan produktet overordnet set tager sig ud, da personen ikke sidder tæt på processen. Her menes antallet af egenskaber, hvordan sælgeren præsenterer produktet og hvor imponerende produktet er i førstehåndsindtrykket. Derimod vil beslutningstagere i mindre virksomheder ofte sidde tættere på processen. Disser personer vil være mere engageret i om produktet løser lige netop den opgave, som det er sat til. Prisen er selvfølgelig en vigtig faktor i begge situationer, men en person der sidder i processen vil være mere fokuseret på cost-benefit for denne i forhold til kun at beslutte ud fra hvad der virker imponerende.

I fremtiden vil der være store værktøjer, som kan løse mange opgaver, der ikke nødvendigvis er relateret. Men det er forfatterens opfattelse, at mindre værktøjer, der dels kan løse den givne opgave, og dels arbejde sammen med større værktøjer, i højere grad vil vinde indpas.

Værktøjet som præsenteres i denne rapport er et eksempel på udviklingen af et mindre værktøj. Selvom det måske ikke er realistisk at lave et færdigt produkt indenfor de 10 uger er det påbegyndelsen af et værktøj, der ikke findes i så stor grad på markedet. Det vil i fremtiden kunne bruges måske som open source af mindre virksomheder eller non-profit organisationer. Det er vigtigt at der ikke bare findes store systemer, men også mindre værktøjer, som løser en generelt veldefineret opgave.

Dette er begyndelsen på en ukendt, men forhåbentlig spændende rejse, som næppe afsluttes indenfor det næste halve år efter afleveringen af denne rapport.

1.3 Projektet generelt

Dette afsnit giver en meget overordnet beskrivelse af, hvorfor der bør udvikles projektværktøj. Der gives en kort introduktion til projektlederens arbejdsopgaver. Afsnittets målgruppe er personer, som ønsker at læse en introduktion til hele rapporten. Læsere der ønsker en kort beskrivelse og derfor ikke ønsker at læse rapporten i sin helhed, kan få en kort beskrivelse af, hvorfor ikke alle projektmedlemmer behøver alt information.

Citat fra Wikipedia, the free encyclopedia(<http://www.wikipedia.org/>): *Project management is the discipline of defining and achieving targets while opti-*

mizing the use of resources (time, money, people, materials, energy, space, etc).

God planlægning har altid været en udfordring især i store projekter. Projektledere kan notere aftaler i kalenderen, men at danne sig et godt overblik over arbejdsbyrden hos hver enkelt teammedlem kan være en næsten umulig opgave. I ethvert projekt, uanset hvilket forretningsområde der er tale om, bør projektledere kunne identificere sådanne risikogrupper på forhånd, med henblik på at implementere foranstaltninger som begrænser omfanget af problemet.

I takt med at projekter vokser i størrelse, skifter projektlederens rolle fra motivationsgiver, ekspert og egentlig team medlem til administrative opgaver. Der er ikke nogen forskel på at lede små eller store hære, bortset fra at lederens rolle ændrer sig. Store projekter kræver dog mere opmærksomhed omkring kommunikationen. Modsat små projekter med tæt samarbejde, og derfor også større social indbyrdes forståelse, er der en risiko for at der laves såkaldte elektroniske grænser. Ved tæt samarbejde opbygger mennersker naturlig forståelse for hinanden. Kropssprog kan have meget stor betydning. Når der skrives e-mails frem og tilbage kan det ske, at forfatteren til en e-mail antager at noget viden er alment kendt. Dette kan medføre tab af information og dermed introducere en risiko. I store projekter bør projektlederens beslutningsområde derfor være mere generelt. Da projektlederen ikke kan have føling med alle dele af projektet, bør dennes informationstilgang være præget af standard rapporter. Oplysninger skal være meget konkrete, og det kan ikke forventes at projektledere ved noget om detaljer diskuteret på uformelle møder. Her og nu, læs detaljerede, beslutninger bør tages af folk med indsigt i netop deres del af processen. Dermed begrænses risikoen for tab af kritisk information. Samtidig medfører det at projektlederen kan holde sit fokus på dennes egentlige arbejdsområde, planlægning og optimal udnyttelse af tid, resurser osv.

For at standardisere kommunikationen bør der opnås enighed om nogen fælles regler. Standard formularer kan med fordel anvendes for at tvinge teammedlemmer til at begrænse deres kommunikation til det der er essentielt for projektlederen. Et projektstyringsværktøj skal hjælpe med at danne et hurtigt overblik over projektet. Værktøjet skal hjælpe projektlederen med at holde fokus på de kritiske dele af projektet. Der hvor problemer med en vis sandsynlighed kan forventes at opstå. Først og fremmest skal et sådant værktøj tjene med dels at inddele informationer i en logisk struktur således, at der hurtigt kan findes bestemte kategorier af informationer, uden nødevendigvis at vide hvor disse informationer befinder sig.

For at give læseren lidt information om hvad et projekt er for en generisk størrelse følger her lidt beskrivelse af et projektforbøb.

Et typisk projekt er illustreret i figur 1.1 på side 6. Som det ses på diagrammet

er der en køber og en sælger. Produktet kan være mange ting, men er typisk noget der forekommer i forbindelse med en "value chain"³. Det skal nævnes, at projekter naturligvis kan foregå internt i en virksomhed. "Kunden" vil i disse tilfælde typisk være ledelsen. De fleste projekter vedrører dog en ekstern kunde, da interne projekter som bekendt ikke giver nogen direkte profit. Som det fremgår af diagrammet 1.1, kan projektlederen have mange projektmedlemmer under sig. Derfor er det ikke sikkert, at han eller hende kan nå rundt og tale med alle projektmedlemmer. Dette skaber et behov for at automatisere kommunikationsprocessen. Der findes den dag i dag allerede visse værktøjer, selvom det er forfatteren af denne rapport opfattelse, at en del forbedringer kunne foretages inden for dette felt. De økonomiske omkostninger for sådanne værktøjer er desuden temmelig høje i forhold til hvad de tilfører projektet. Nogle virksomheder, f.eks. X_2 ⁴ og ABB - Danmark, bruger Microsoft Excel til at overvåge stressniveauet (regneark typisk udviklet af interne konsulenter). Dette til trods for, at Excel slet ikke er beregnet til store mængder af records (se afsnit A.1. Dertil kommer sikkerheden i et flat regneark). Siebel⁵ og Microsoft arbejder sammen om at lave et nyt kæmpe projektsystem. Forfatteren er ikke gennemført bekendt med Siebel, men rygter i markedet påstår, at Siebel forsøger at blande alt for mange, meget forskellige værktøjer i et stort produkt. Brugere mener derfor, at værktøjet er for komplekst til generel brug og de føler desuden, at de betaler for mere end det som de har brug for. Dette tegner, ifølge forfatterens opfattelse, til at være et voksende problem inden for software industrien. Noget som bør løses på en eller anden måde inden problemet vokser sig rigtig stort.

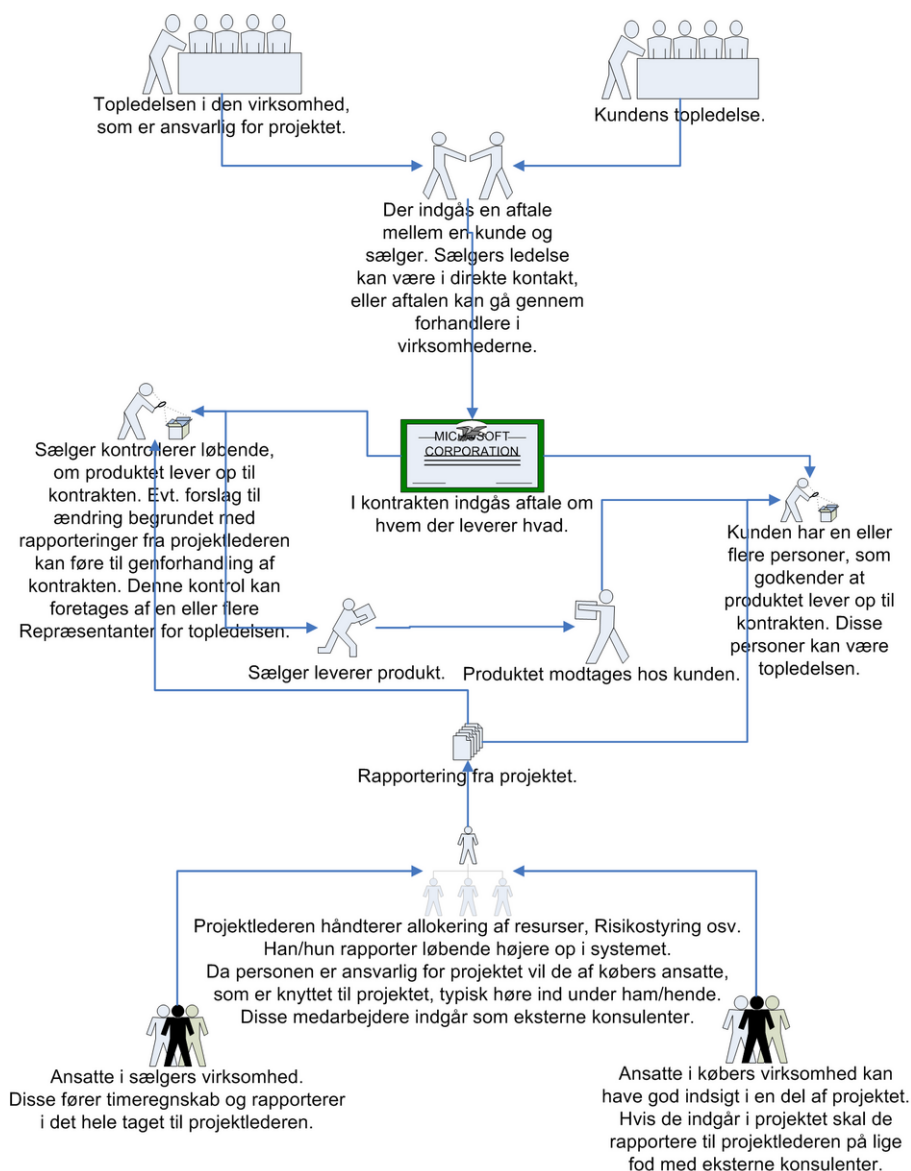
Denne rapport har til formål at beskrive hvorledes et værktøj fokuseret på at løse generelle projekt problemstillinger, og ikke software design og implementering, kan implementeres. Overvejelse gøres omkring hvilke personer, der behøver at vide hvad. Dette skal ikke opfattes som sikkerhedsovervejelser (det vil det imidlertid blive på sigt), men er i højere grad en anerkendelse af at hvert enkelt teammedlem ikke må overbebyrdes med ligegyldig information. Ligegyldig information kan være årsag til, at det enkelte teammedlem ikke handler korrekt i tide.

Værktøjet som udvikles kan frit benyttes, ligeledes kan alt dokumentation. Kildedekoden og systemet må modificeres og videreudvikles så længe, at der refereres til den oprindelige kilde.

³se forklaring i begrebslisten i bilag B på side 65

⁴Firmaets navn er fortroligt efter aftale med X_1 der er forfatter til bilag A

⁵En virksomhed som udvikler projektværktøjer. Se [5]



Figur 1.1: Viser hvordan et projekt typisk foregår.

1.4 Projekt begreber

I dette afsnit kan der læses om nogle af de begreber, som eksisterer i forbindelse med projektarbejde. Afsnittet kan læses uden forudgående kendskab til projektledelse. Det skal ikke ses som nogen udtømmende forklaring på hvilke begreber, der findes i alle projekter. Det er resultatet af dels samtaler med X_1 ⁶, samt en brain-storm foretaget over emnet ”*projektstyring*” ses på figur 1.2 på side 8). Det vedlagte bilag A på side 59, er en specifikation af et værktøj implementeret i Microsoft Excel af X_1 . Afsnittet er ment som en slags rød tråd i forhold til kapitel 3 på side 19. Dette afsnit kan umiddelbart læses af alle, der ønsker en kort introduktion til nogle af de begreber, som findes i forbindelse med projekter.

1.4.1 Projektdeltagere

Af figur 1.2 fremgår det, at mennesker indgår som en projektresurse. For at kunne definere hvilke personer, der har brug for hvilke informationer, er det nødvendigt at identificere de forskellige grundlæggende arbejdsområder i projekter. Der tages udgangspunkt i bilag A, samt i det hele taget den typiske opbygning af projekter illustreret på figur 1.1.

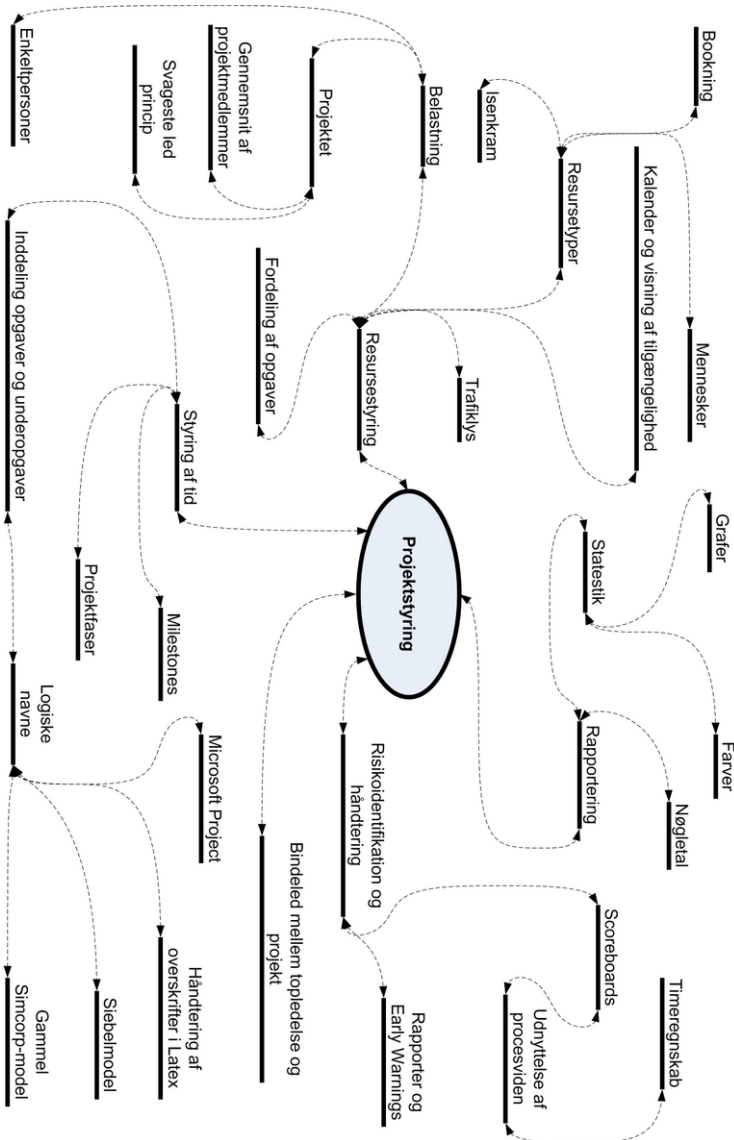
Ved iagttagelse af afsnit A.2 på side 60 identificeres tre grupper af mennesker.

- Styregruppemedlemmer
- Projektleder og projektgruppe (få nøglepersoner med central indsigt)
- Projektdeltagerne

1.4.1.1 Styregruppemedlemmer

Ved styregruppemedlemmer menes ledelse. Disse personer har ikke nødvendigvis noget direkte med projektet at gøre. Styregruppemedlemmer kan være med til at forme det overordnede i kontrakter (se figur 1.1). De kan være interesserede i hvorvidt mål, fastsat som en milestone, bliver overholdt. Disse personer ønsker overblik og rapporter, ikke detaljer om hvorvidt én medarbejder er overbebyrdet. Det blev overvejet hvorvidt denne gruppe burde hedde administration, da der er tale om administrative ledelsesorienterede personer. Navnet kan imidlertid få en utiltænkt klang, da administratorer i et it-system typisk er folk med

⁶En forretningskonsulent i X_2 . Efter aftale med X_1 holdes dennes navn fortroligt.



Figur 1.2: Viser brainstormdiagram over emnet projektstyring.

flere rettigheder. Den egentlige administrator i dette system bør være den, som administrerer resurser (dvs. projektlederen eller projektgruppen). Betegnelsen styregruppe er måske heller ikke ideel da navnet, hvis det står alene, kan opfattes som gruppe der styrer projektet, hvilket ikke nødvendigvis er tilfældet. Derfor blev det besluttet at bruge udtrykket "topledelsen". En topledelse er, uanset om de måtte have indflydelse på det enkelte projekt, nogen der sætter overordnede strategier. En topledelse tager sig normalt ikke at fordelingen af arbejdsopgaver. Topledelsen er vist øverst på figur 1.1.

1.4.1.2 Projektleder og projektgruppe

Projektlederen er den der styrer projektet og fordeler arbejdsopgaver. Projektlederens placering i projekter er vist på figur 1.1. Hvis der er tale om en projektgruppe, eller en på anden måde matrixorienteret ledelseform⁷, bør alle projektlederne have rettigheder til at redigere i fordelingen og prioriteringen af arbejdsopgaver. For at opnå konsistens i kommunikationen med medarbejderstaben kræver det, at gruppen enten udpeger en kommunikationsansvarlig (dermed også den person, som har projektlederegenskaber i forhold til systemet) eller bliver fuldstændig enige om alting under formelle møder. Der blev valgt betegnelsen "Projektleder". Det er valgt udfra, at alle medlemmer i en projektledergruppe har lige ansvar, og derfor hver i sær fungerer som en projektleder. Hvis en projektgruppe har valgt en ansvarlig, vil denne have funktion som en projektleder.

1.4.1.3 Projektdeltagerne

Projektdeltagere er generelt personer, som arbejder på projektet. Det kan være eksterne konsulenter eller folk fra medarbejderstaben. Hver især kan disse personer have specielviden. Nogen gange kan tidsplanen for en arbejdsopgave skride. Derfor er det vigtigt, at de enkelte projektdeltagere melder tilbage til projektlederen, så snart de opdager at planen skrider. Projektlederen kan så vælge at bytte rundt på rækkefølgen af opgaver, for at løse problemet. Uden projektdeltagere er der derfor ikke noget projekt. Projektdeltagere er vist nederst på figur 1.1.

⁷ se B på side 65

1.4.2 Opdeling af opgaver

Fordeling af opgaver er vigtig og opdelingen er opgaver er ligeså vigtig. På figur 1.2 fremstår opdeling af arbejdsopgaver som noget meget centralt i forhold til styring af tid. I forbindelse med beskrivelsen af "value chain" i afsnit B, forklares hvordan et produkt nogen gange deles op i underleverancer. I stedet for bare at opfatte et stort projekt, bliver projekter opdelt i mindre delprojekter, der måske igen kan deles op i underopgaver. Fordelene ved at opdele arbejdsopgaver er flere. For det første kan skred i tidsplanen hurtigere identificeres således, at projektlederen kan tage hånd om problemet så tidligt i processen som muligt. For det andet bliver det nemmere at sætte projektdeltagere til at arbejde parallelt⁸ med relaterede underopgaver. Derved øges produktionshastigheden. Der kan oprettes en egentlig pipeline⁹ af opgaver, hvorved produktionshastigheden øges. I de følgende underafsnit defineres metoder til at opdele et projekt.

1.4.2.1 Siebel, Project og overskrifter i Latex

Som det fremgår af afsnit A.1 er der en del forvirring, omkring navngivning af forskellige niveauer i opdelingen af opgaver. Der er i det nævnte regneark benyttet Siebels model frem for X_2 s. Det blev besluttet helt af droppe begge modeller til navngivning da der, i princippet, kræves henvisning til en regel i stedet for almindelig logisk tænkning. Det faktum, at forvirringen opstod i første omgang må være rigelig argumentation. Det er ikke utænkeligt at nye medarbejdere, som ikke har lært at tænke "rigtigt", kan blive forvirret. I Microsoft project deles opgaver op i underopgaver(subtasks). Der kan opbygges en rimelig overskuelig struktur, med en task delt op i subtasks.

Denne indeling mangler dog lidt information om, hvor projektet er nået til. Det blev besluttet at benytte en struktur der minder om opdeling af en bog i kapitler. Ideen kom egentlig, da de indledende kommentarer til denne rapport, som er produceret med Latex, blev skrevet. Når en bog skrives i Latex benyttes "kommandoer" som: `\chapter`, `\section`, `\subsection` og `\subsubsection`. På samme måde kan et projekt opdeles i faser, opgaver/tasks, underopgaver/subtasks og underunderopgaver/subsubtasks.

I stedet for bare at have tasks og subtask i designet af project manager, er det valgt at bruge phase, task og milestone. Begrebet phase kan her dække over, hvad der svarer til en task i MS Project. En task opfattes som en konkret opgave, hvorimod en fase er en periode med et overordnet mål, uden at være

⁸se definition af "parallelt arbejde" i afsnit B

⁹se definition, og hvorfor parallelt arbejde ikke er det samme som pipelining, i afsnit B

nogen konkret opgave.

Det bør være muligt at angive forskellige deadlines, hvor der kan holdes statusmøde. Disse deadlines i projekter bliver typisk kaldt milepæle eller milestones. Dette begreb eksisterer i det fleste projektværktøjer, og er derfor også designet ind i project manager.

I det følgende defineres de nævnte opdelinger.

1.4.2.2 Tasks

Tasks er i project managers forstand det nederste niveau. Hvis man ønsker tasks og subtask som i Microsoft Project kan der benyttes faser. Rapporten igennem vil en task være en konkret arbejdsopgave.

1.4.2.3 Phases

Faser kan her være flere ting. Det er det øverste niveau inden for project manager. En fase kan være meget overordnet eller det kan være en egentlig opgave som er delt op i underopgaver for at opnå en hurtigere arbejdsproces.

1.4.2.4 Milestones

En milestone er en skærringsdato. Der arbejdes måske med en lidt løs tidsplan, da det er undefinerbart hvad resultat der ønskes (Det kan være at en kunde, som køber et stort it-system ikke helt ved hvilke muligheder denne har). Det kan dog defineres at til en bestemt dato, skal systemet kunne noget med at beregne ansattes løn. Således sættes defineres en milestone til en bestemt dato. Når der implementeret mulighed for at beregne løn er denne milestone opfyldt og der gøres status. Hvordan projektdeltagerne komme frem er egentlig underordnet. Det kan være, at der ikke er nogen egentlig slutdato for projektet. Hvis der forskes indenfor et eller andet, kan det nogen gange være svært at definere en egentlig slutdato. Der kan så alligevel defineres en milestone. Det er ikke en egentlig slutdato, men et tidspunkt, hvor der gøres status over, hvordan projektet skal fortsætte.

1.5 Sektionsopdeling

I dette afsnit beskrives nogen overordnede tanker, som der er gjort.

Generelt om logisk godt opbyggede systemer er der et begreb, der går igen, i alt fra operativ systemer til små applikationer. Træstrukturer kan hjælpe med at organisere, selv store mængder data, på en logisk og overskuelig måde, således at brugeren kan finde rundt i systemet. Det blev derfor besluttet at opdele de forskellige oplysninger i en træstruktur. Læsere der har kendskab til udvikling af grafiske brugergrænseflader, vil desuden vide at informationer og funktionalteter opdelt med træstruktur nemt kan overføres til specificering af design og implementering. Fra de fleste nyere programmeringssprog, læs tredje generation og op efter, er der adgang til grafiske komponenter som menuer og træstruktur¹⁰.

I de følgende afsnit er der grupperet data, som et projekt indeholder, i en trælignende struktur. Afsnittet kan af læseren bruges til at danne sig overblik over, hvorledes projektinformation kan grupperes i en overskuelig struktur. Der kræves ikke teknisk indsigt, for at forstå hvorfor, store mængder data bør grupperes på en overskuelig måde.

1.5.1 Overall data

Denne sektion bruges til at udfylde generel information om projektet. Det bør kun være projektledere, som har adgang til at udfylde eller redigere disse oplysninger. Top ledelsen bør gives læserettigheder således, at de kan følge projektlederens arbejde uden fejlagtigt at pille ved oplysningerne. Sektionen bør bestå af følgende felter.

Project id: Det er nødvendigt at angive et informativt navn til et projekt. Dette navn behøver ikke at være unikt. Det kan være at et rådgivende ingeniørfirma, som rådgiver om statik, vinder to udbudte projekter. Det kan være, at der skal bygges to sportshaller med hule bærende limtræsbjælker to forskellige steder i verden. Med en vis sandsynlighed kunne to projektledere uafhængigt af hinanden give hver deres projekt samme navn. Det kan være at begge ledere vil kalde netop deres projekt for: "Sådan undgås en kopi af Siemen arenaen". Der er vil næppe nogen der vil fortænke projektlederne det. For at kunne adskilde de to projekter i en projektdatabase er det nødvendigt med et unikt id. Id'et er noget som system bør angive, således at der ikke opstår inkonsistens. Id'et skal altså ikke, uden videre, kunne modificeres af brugere.

¹⁰Ligesom den grafiske struktur set i Microsoft Windows Explorer

Project name: Brugere skal kunne angive et informativt navn til projektet. Der skal således være en mulighed for at angive dette. Dette navn kan evt. bruges, når der senere søges efter lignende projekter.

Keywords: Det skal være muligt at angive nogle nøgleord om et projekt. Således skal der senere kunne søges på fagemner, som indgår i de enkelte projekter.

Description: Det bør være muligt at indtaste en kort tekst, som beskriver projektet uden at dække alle detaljer.

Starting date: Her bør der kunne angives en startdato for hvornår projektet starter.

Ending date: Her bør der kunne angives en slutdato for projektet. Slut datoen kan aldrig komme før startdatoen.

Average working time per week (hours): Dette skal kunne bruges til at angive hvor mange timer der i gennemsnit forventes lagt i projektet per medarbejder per uge. Dette skal senere kunne bruges i forbindelse med at planlægge hvilke medarbejder, som skal knyttes til projektet. Set lidt egoistisk vil en projektleder kunne vælge at overbebyrde en medarbejder på halv tid i en kortere periode. Denne vil, selvfølgelig afhængig af grunden til at denne arbejder på halv tid, måske have mere overskud i forhold til en medarbejder der i forvejen arbejder fuld tid.

Assign resources: Resurser kan deles ind i to kategorier, mennesker og udstyr. Mennesker kan være eksterne konsulenter eller almindelige medarbejdere. På samme måde kan udstyr være noget som findes i den enkelte virksomhed eller det kan være noget, som skal lejes eller købes.

Mennesker: De fleste virksomheder har oplysninger om f.eks. en medarbejders uddannelse osv. En personale database burde indeholde informationer om de enkelte medarbejders initialer, navne, tilhørsforhold eller afdeling.

Det bør i det endelige system kun være muligt at planlægge arbejde med resurser, der er tildelt til projektet.

1.5.2 Registration

Der bør være en form for gant chart, som det kendes fra Microsoft Project og andre projektstyringsværktøjer. Desuden skal dette punkt dække over registrering af status for den enkelte arbejdsopgave. Dette punkt dækker således over,

hvad der måtte være af registreringer i forbindelse med arbejdsopgaver.

1.5.3 Risk management

Under dette punkt vil det være muligt, at se hvor meget det enkelte projekt-medlem er belagt med arbejdsopgaver. Hvor belastet vedkommende er. Desuden skal det udfra belastningen på enkelte medarbejdere være muligt at se om projektet generelt er i fare for at springe tidsplanen. Dertil vil der i fremtiden blive implementeret en form for score board, hvor medarbejdere kan indrapportere risici. Belastningen af medarbejder skal både angives med tal og farvekoder. Noget virksomheder vælger at beligge deres medarbejdere med 80 %, men dette bør kunne reguleres. Ud fra spredningen, i forhold til normal belægning, bruges farverne sort, rød, orange, gul og grøn.

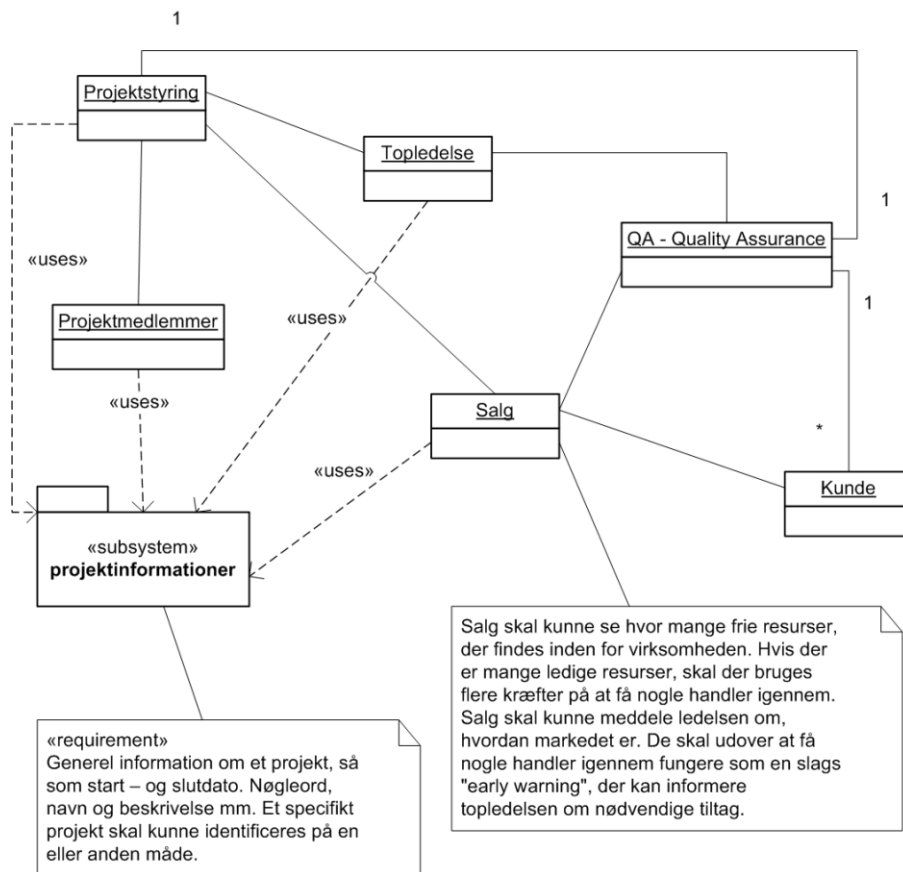
1.5.4 Project report

Dette skal fungere som en automatisk genereret rapport til ledelsen. Her skal der stå navnet på projektet, en beskrivelse af projektet, status for projektet samt risikostatus. Dertil skal der være et skema som viser planed hours, actual hours, forskellen på actual og planed i timer samt procent. Der skal være farvekoder med farverne sort, rød, orange, gul og grøn udfra spredningen i forhold til det planlagte. Det skal være en prognose for hver dag. Prognosen beregnes som totalforskellen i procent multipliceret med planlagte timer divideret med 100. Desuden angives for hver dag det forventede som er total antal faktiske timer plus planlagte timer i fremtiden. Der skal være en piechart, som angiver hvor stor en del af arbejdstiden der er lagt i opgaven, salg mm., planning, udvikling af kompetancer samt administration.

Business Model

Det er, som tidligere nævnt, valgt at arbejde efter en arbejdsproces, der hedder unified process. Herunder indgår noget, der hedder business modeling. Kort sagt handler det om at forstå den målgruppe, som softwaren skal produceres til. Der er allerede i Indledningen til denne rapport foretaget en mindre analyse af, hvad et projekt egentlig er. I den forbindelse blev der identificeret forskellige grupper af personer. I det følgende vil disse grupper af personer kort blive opridset, da de vil indgå i en såkaldt domain model. Bemærk i øvrigt: En domæne-model repræsenterer klasser, moduler og objekter fra den virkelige verden og ikke hvad der findes i softwaren. Det er altså ikke en model der viser klasser og deres ansvar. Det er valgt at bruge domain modellen som en slags identifikation af projektlederen, og dermed det nuværende projekts (applikation) rolle, i forhold til den virkelige verden.

Der vil i et projekt normalt være en række projektdeltagere, der arbejder på projektet. Der vil også være en eller flere projektledere, som styrer projektet og planlægger opgaver og rækkefølgen af opgaver. Da et projekt som regel foregår i et firma, vil der sædvanligvis være en topledelse som holder overordnet øje med projekter, og ligger strategi for hvilke projekter der skal prioriteres højt. Der vil for det meste være en salgsafdeling, som står for at sælge projekter til eksterne virksomheder (dette gælder dog ikke for interne projekter). I alle projekter vil der være en kunde. Hvis et projekt foregår internt i virksomheder vil de, i virksomheden, som har bestilt projektet, være kunden. De fleste virksomheder



Figur 2.1: Domain Model

har desuden en eller anden form for kvalitetssikring. Nogen gange vil der være en speciel afdeling, der tager sig af at kvalitetssikre. Andre gange vil der være tale om retningslinjer, som der udstukket fra ledelsen.

I alle projekters uanset størrelse, vil der være kommunikation mellem de nævnte grupper. I mindre projekter vil der måske være en tendens til, i højere grad, at benytte uformelle samtaler. Uanset størrelsen af et projekt vil denne kommunikation dog kunne gøres formel, læs dokumenteret i rapporter eller en anden form for informationssamling. Kommunikationen og de forskellige grupper af personer er illustreret i domain modellen figur 2.1 på side 16. På figuren 2.1 ses det hvordan de forskellige grupper kommunikerer indbyrdes. Det ses at der med fordel kan være en fælles dokumentbase, hvilket også eksisterer i de fleste projektorienteret virksomheder, som forfatteren har kendskab til. De fleste virk-

somheder har i dag denne base elektronisk implementeret. Heri vil der kunne findes generelle oplysninger om projektet, for eksempel start - og slutdato. Der kan være flere kunder inde i billede på samme tid. Hvilket er illustreret med en stjerne. Her følger kort en beskrivelse af hvad det er valgt at fokusere på i første omgang.

Som problemformuleringen udtrykker, er det værktøj, der i første omgang ønskes udviklet, primært tiltænkt at hjælpe projektlederen. Domain modellen, og dermed virkeligheden som den er et billede af, lægger mere eller mindre op til en række selvstændige applikationer, som arbejder sammen i et distribueret system. Ses der på systemet vil fremtidig udvikling blandt andet bestå i at konstruere værktøjer til ledelse osv.

2.1 Konklusion på modellen

Der blev i dette kapitel defineret en række grupper af personer, og deres indbyrdes kommunikation. Samhørigheden er illustreret i figuren 2.1, der er en domain model over den virkelige verden. Det blev kort redegjort hvilken del af denne virkelige verden, som der sættes fokus på, men det blev samtidig klargjort, at der i fremtiden kan eller bør udvikles applikation som dække disse andre gruppers behov. Disse vil sammen med applikationen, der behandles i denne rapport udgøre et distribueret system.

KAPITEL 3

Kravspecifikation

3.1 Indledning - hvordan er projektet blevet grebet an?

Det er i dette projekt valgt at arbejde overordnet efter Unified Process. Dette betyder, at der blandt andet er arbejdet i itterationer. Det er i denne rapport valgt kun at vise den reviderede udgave af kravspecifikationen. I dette kapitel vil der blive defineret de krav, der er til programmet på nuværende tidspunkt.

3.2 Hvad fokuseres der på i forhold til domainmodellen

Det blev i forbindelse med business modelleringen klargjort, at der overordnet set fokuseres på de arbejdsopgaver, som projektlederen står over for. Inden for dette område er der defineret en række krav til hvad denne applikation skal have af egenskaber.

3.3 Definition af kravene i forhold til IEEE standard

I det følgende er det valgt at benytte IEEE standard 830 fra 1993. Dette er en standard for formulering af en kravspecifikation. Standarden er beskrevet i [9] på side 226. Det er dog ikke nødvendigt at kende til standarden for at forstå dette afsnit. Der er vise punkter i standarden, der bevidst er undladt, da der ikke umiddelbart er noget information under disse punkter.

3.3.1 Introduktion

3.3.1.1 Formål

Denne kravspecifikation bekendtgør de foreløbige krav til et system, der skal hjælpe en projektleder med dennes arbejde. Herunder gemme oplysninger og planlægning af projekter generelt.

3.3.1.2 Målgruppe

Det forventede produkt skal på sigt anvendes af projektledere og projektgrupper i forbindelse med planlægning og løbende risikohåndtering. Den foreløbige version af produktet skal kunne bruges af projektledere til at planlægge opgaver, som kan illustreres med et gant-diagram.

3.3.1.3 Overblik

De to følgende afsnit i denne kravspecifikation handler om følgende. Først kommer et afsnit, der har til formål at give et generelt overblik over produktet. Dernæst gives en mere udførlig beskrivelse af tilgængelig funktionalitet. Denne information er opdelt efter de brugere, der skal tænkes at benytte den.

3.3.2 Overordnet beskrivelse

3.3.2.1 Produktperspektiv

Den udviklede applikation tænkes at kunne gemme informationer på to måder. Der er en database del, som i de første versioner er valgt til at være en typisk MySQL databaseservice. Desuden skal der kunne gemmes lokalt i en flad fil. Det tænkes at være gemt som xml. Produktet bygges op som en grafisk brugerflade med kontakt til det underliggende datalag. Model view control skal overholdes, hvilke vil sige separering af præsentation, logik og data.

3.3.2.2 Funktionalitet

Applikationen skal i første omgang varetage to områder. For det første skal der kunne administreres generel projektinformation. Dernæst skal det være muligt at registrere opgaver, faser og milestones. Det sidste skal designes i et modul, med mulighed for grafisk afbildning.

3.3.2.3 Brugerkarakteristik

Det kan ikke umiddelbart forventes, at brugerne af systemet er eksperter til it, men de fleste arbejder med it i hverdagen.

3.3.2.4 Begrænsninger i adgang

Der skal foreløbig kun implementeres et filte, så brugeren har mulighed for at vælge om denne vil se oplysninger som: Alle, Top leder, projektleder eller almindelig projektmedlem. Hvis filteret alle vælges skal der senere være særlig adgang til opsætning af systemet. Det vil i fremtiden være muligt at indsætte noget adgangsbegrænsning ved hjælp af Windows sikkerhedssystem.

3.3.2.5 Antagelser

Det antages at brugeren enten har sat en MySQL database op med de tabeller, som programmet måtte kræve, eller benytter xml-filer.

3.3.3 Specifikke krav

3.3.3.1 Brugerflade

Der kan benyttes en almindelig pc. Pc'en kan enten koble op til en ekstern database, lokal database eller benytte xml filer. Dersom der ønskes ekstern database skal computeren have adgang via tcp netværk.

3.3.3.2 Software

Project manager antages at køre på et Windows operativsystem med: Bemærk! Engelsk dato-format. Dele af programmet består af en særlig dot Net komponent, som ikke bare får oversat datoformatet. I værste fald kan programmet ikke gemme.

3.3.3.3 Kommunikationsflader

Der er ikke umiddelbart nogen krav, ved mindre der ønskes adgang til ekstern datakilde.

3.3.3.4 Funktionelle krav

Brugeren skal kunne foretage følgende: Starte et nyt projekt, gemme et projekt, åbne et projekt, indtaste generelle oplysninger om projektet herunder startdato, slutdato og projektets navn. Indtaste opgaver, faser og milestones, som vises på et gant diagram. De oplysninger som brugeren taster skal vises på samme måde hvis projektet er gemt, lukkes og åbnes igen.

3.4 Use cases

I de følgende tabeller er krav til programmet defineret i form af Use cases. Use case er beskrevet i [6]. Kort fortalt beskriver krav i form af handlingsforløb. På figur 3.1 side 25 ses sammenhængen mellem de forskellige use cases.

Primær aktør:	Projektlederen
Interessegrupper:	Projektleder ønsker at oprette et nyt projekt, som denne senere kan gemme med information omkring et projekt.
Forudsætninger:	Programmet skal være startet.
Succes kriterium:	Der bliver oprettet et nyt projekt, som senere kan gemmes. Der kan tastes data ind i det nye projekt.
Succes scenarium:	Projektlederen start nyt projekt dialogen ved at klikke på nyt projekt i menuen, bruge genvejstast (ctrl+n) eller bruge toolbaren. Der vælges xml projekt. Der trykkes på create new. Et nyt projekt starter, og der kan indtastes information.
Alternative forløb:	Hvis der vælges database projekt, fremkommer en række databaser, som der kan vælges i mellem. Disse databaser er kan tilføjes fjernes i en fil eller lignende. Der oprettes et nyt projekt, med et nyt unikt id.
Bemærkninger:	Det antages at en database er installeret.

Tabel 3.1: Use case 1: Nyt projekt

Primær aktør:	Projektlederen
Interessegrupper:	Projektlederen ønsker at gemme de information, der er afgivet i et nyt eller åbent projekt.
Forudsætninger:	Enten skal der være oprettet et nyt projekt, eller også skal et tidligere projekt være åbent.
Succes kriterium:	Dataene gemmes og kan hentes ved at åbne projektet på ny.
Succes scenarium:	Projektleder vælger gem. Projektlederen lukker enten projektet eller programmet helt. Projektlederen åbner det gemte projekt. De data der fremkommer, er dem der tidligere i processen blev tastet.
Alternative forløb:	
Bemærkninger:	Hvis projektet blev gemt som et database projekt, skal det åbnes som et database projekt. Hvis projektet er gemt i en flad fil, skal det åbnes som en flad fil.

Tabel 3.2: Use case 2: Gem projekt

Primær aktør:	Projektlederen
Interessegrupper:	Projektlederen ønsker at åbne et tidligere gemt projekt.
Forudsætninger:	Der er adgang til et tidligere gemt projekt. Såfremt der ikke tidligere er gemt et projekt, skal der oprettes et nyt projekt, som så gemmes.
Succes kriterium:	Projektlederen kan åbne et tidligere gemt projekt.
Succes scenarium:	Projektleder vælger at åbne et tidligere projekt. Der vælges at hente et xml projekt. Der peges, via en dialog, på den fil, som skal åbnes. Projektet åbner med tidligere gemte data.
Alternative forløb:	Hvis vælges åben databaseprojekt, skal der via en dialog peges på den rigtige database og derfra vælges et projekt. Projektet åbner med tidligere gemte data.
Bemærkninger:	Ekstra krav, som har indflydelse på systemet.

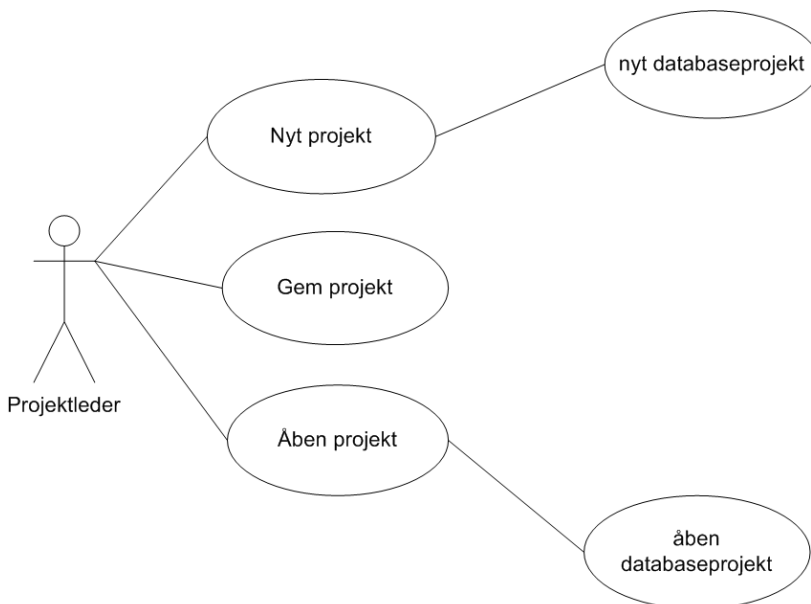
Tabel 3.3: Use case 3: Åben projekt

Primær aktør:	Projektlederen
Interessegrupper:	Projektlederen ønsker at åbne et projekt fra en database.
Forudsætninger:	Der skal være adgang til databasen. Databasen skal være tilføjet systemet, og databasen skal være udformet så den passer til designet af project manager.
Succes kriterium:	Der kan oprettes et nyt projekt i databasen, som åbnes i project manager.
Succes scenarium:	Projektlederen vælger nyt projekt. Projektlederen vælger nyt databaseprojekt. Projektlederen vælger en database, hvorefter et nyt projekt med et unikt id åbner.
Alternative forløb:	
Bemærkninger:	

Tabel 3.4: Use case 4: Nyt projekt fra database

Primær aktør:	Projektlederen
Interessegrupper:	Projektlederen ønsker at åbne et projekt, der tidligere er gemt i en database.
Forudsætninger:	Der skal være adgang til en database, som er udformet med tabeller, så den passer til project manager. Databasen skal være tilføjet project manager.
Succes kriterium:	Et tidligere gemt projekt hentes fra databasen og åbnes.
Succes scenarium:	Projektlederen vælger at åbne et projekt. Der vælges åben via database. Der fremkommer en dialog hvori der kan vælges database og efterfølgende projekt. Der accepteres og det tidligere projekt åbner.
Alternative forløb:	
Bemærkninger:	

Tabel 3.5: Use case 5: Åben database projekt



Figur 3.1: Use case diagram

3.5 Konklusion på kravspecificeringen

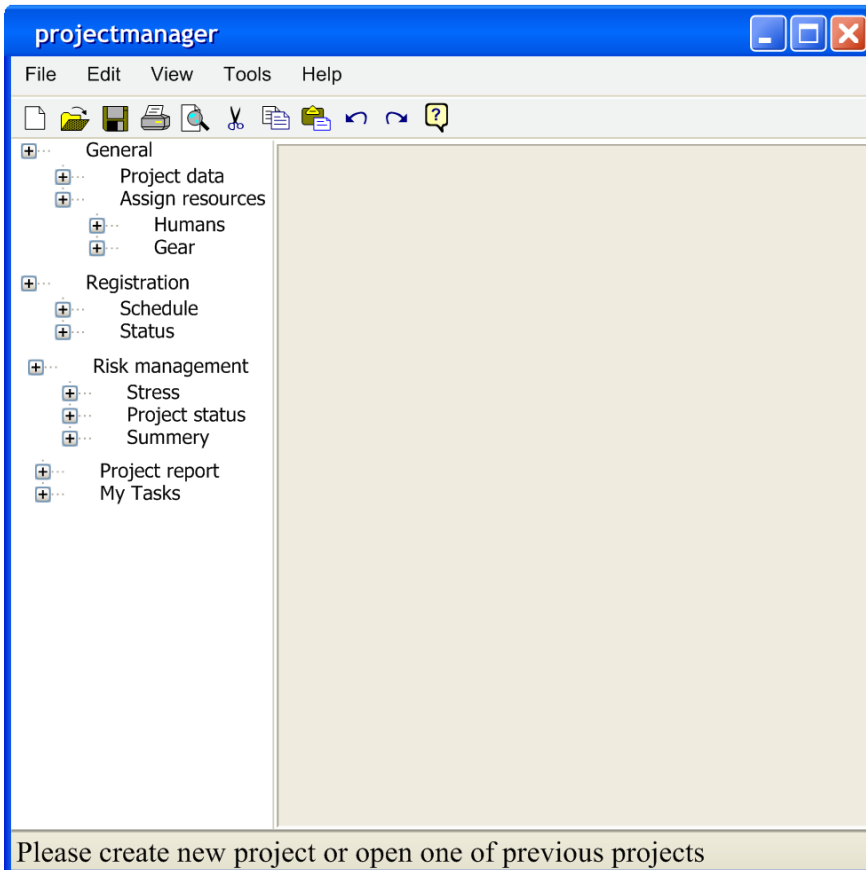
Der er fremsat en række krav som det implementerede software forventes at opfylde. Kravene er defineret dels ud fra IEEE standard 830 fra 1993 og dels i form af handlingsrelaterede use cases kendt fra Unified Process.

KAPITEL 4

Design

4.1 Det grafiske design

På figur 4.1 ses hvorledes hovedvinduet er designet. Det er valgt både at benytte toolbar og mainmenu. Når for eksempel der vælges opret nyt projekt, vil der uanset om handlingen sker via menuen eller toolbaren blive brugt en fælles hjælper funktion. Dette er valgt, ikke bare for at spare kode, men for at sikre ens handlingsforløb. Det blev valgt at indsætte en statusbar, hvori der kan placeres tekst, som hjælper brugeren. I venstre side er der en træstruktur, som brugeren kan bruge når et projekt er åbent. I højre side ses et indholdspanel. Indholdspanelet får tilføjet/fjerne brugerkontroller alt efter hvilket punkt i træstrukturen, der er valgt. Bemærk i øvrigt, at hovedvinduet som det eneste er vist i dette kapitel. De dialoger osv. der ellers bruges, er valgt placeret i kapitel 5. Dette er gjort, da implementeringen af dette ligger 100 procent op af designet, hvorfor der ville forekomme redundante oplysninger i rapporten.



Figur 4.1: Hvordan hovedvinduet er designet

4.2 Design patterns

Det er valgt at bruge design patterns flere steder i applikationen. Der skal ikke gives nogen trættende opremsning af hvor og hvordan, men i det følgende vil der blive beskrevet hvad design patterns er, samt hvilke patterns der er benyttet.

En del mennersker går rundt og tror, at pattern er en stump smart kode. Design patterns har ikke noget med kode at gøre, det er simpelthen et generelt løsningsforslag til et generelt problem. Hvis læseren ønsker at starte med at lære patterns, der typisk er vist implementeret i et sprog, kan det varmt anbefales at kigge på [4].

I dette projekt er der brugt følgende patterns: Observer, Command, Singleton og Facade. Observer er godt, hvis en handling i et vindue eller panel skal overvåges et andet sted, samtidig med, at der ønskes lav kobling høj samhørighed. Command bruges i forbindelse med undo operationer. I det hele taget, når der skal huskes program states. Singleton bruges, når der kun må eksistere en instans af en klasse. Dette er typisk tilfældet i forbindelse med printning. Det har desværre ikke inden for tidsrammen været muligt at implementere printdialog mm. Ikke desto mindre er der tilføjet en klasse til fremtidig udvikling. Facade bruges for at skabe en pæn flade til datalaget. I domain modellen blev det berørt, at der er brug for en fælles dataflade. Hvad enten der er tale om xml eller en database, er ansvaret for dette uddelegeret til en særlig klasse. Nogen vil påstå og til dels med god grund, at ADO.NET er en facade. Dette er strengt taget sandt, men i traditionel forstand vil en facade være en klasse, som tager sig af dataadgangen. I ADO.NET findes et hav af klasser og interfaces, som bruges i forskellige situationer. En af de ting som forfatteren blev særligt bekendt med under dette projekt er, ADO.NET's evner til at håndtere både databaser og xml på en elegant og simpel måde. Det blev besluttet ikke at skrive nogen udtømmende forklaring om ADO.NET. Hvis læseren ønsker at komme igang med ADO.NET henvises til [8] samt [3]. Begge kilder giver en god og relativt god introduktion, og emnet er for stort til at behandle i denne rapport.

4.3 Klasser og filer

4.3.1 Brug af dll'er

Det er valgt at benytte dll'er. Dette kan opfattes lidt ligesom en jar-fil, men er noget der længe har eksisteret i Microsofts verden. Kort sagt er det valgt at

ligge en række klasser i forskellige biblioteksfiler. En fordel ved dll og dot NET er, at en dll-fil ikke nødvendigvis behøver at være skrevet i det sprog, som der implementeres i. I dette projekts tilfælde er de imidlertid brugt for at indkapsle klasser. Dette er på sin vis også opnået ved at bruge forskellige namespaces, men på sigt vil det være lettere at opdatere programmet, da dll'er loades dynamisk.

4.3.2 Beskrivelse af klasser og kommunikation

Selve programmet består af af en stor klasse, som nedarves fra Form. Derfra er en reference til facade. Der bruge forskellige konstruktører fra facade afhængig af hvad der skal kommunikeres med datalaget. På baggrund heraf oprettes nogle dataset, som er ADO.NET's svar på en offline version af data. Facade udgør i det hele taget en "pæn" flade ned mod datalaget. Andre klasser behøver ikke direkte at kende forskellen på, om der er tale om en xml fil eller en database. Selve programmet er illustreret på figur 4.2 side 31.

Indholdspanelet i hovedprogrammet tilføjes dynamisk brugerkontroller afhængig af valget i træstrukturen. Disse bruger kontroller er lavet som paneler, dvs nedarvet fra panel klassen. Kontrollen til at udfylde generel projektinformation ses på figur 4.3 side 32.

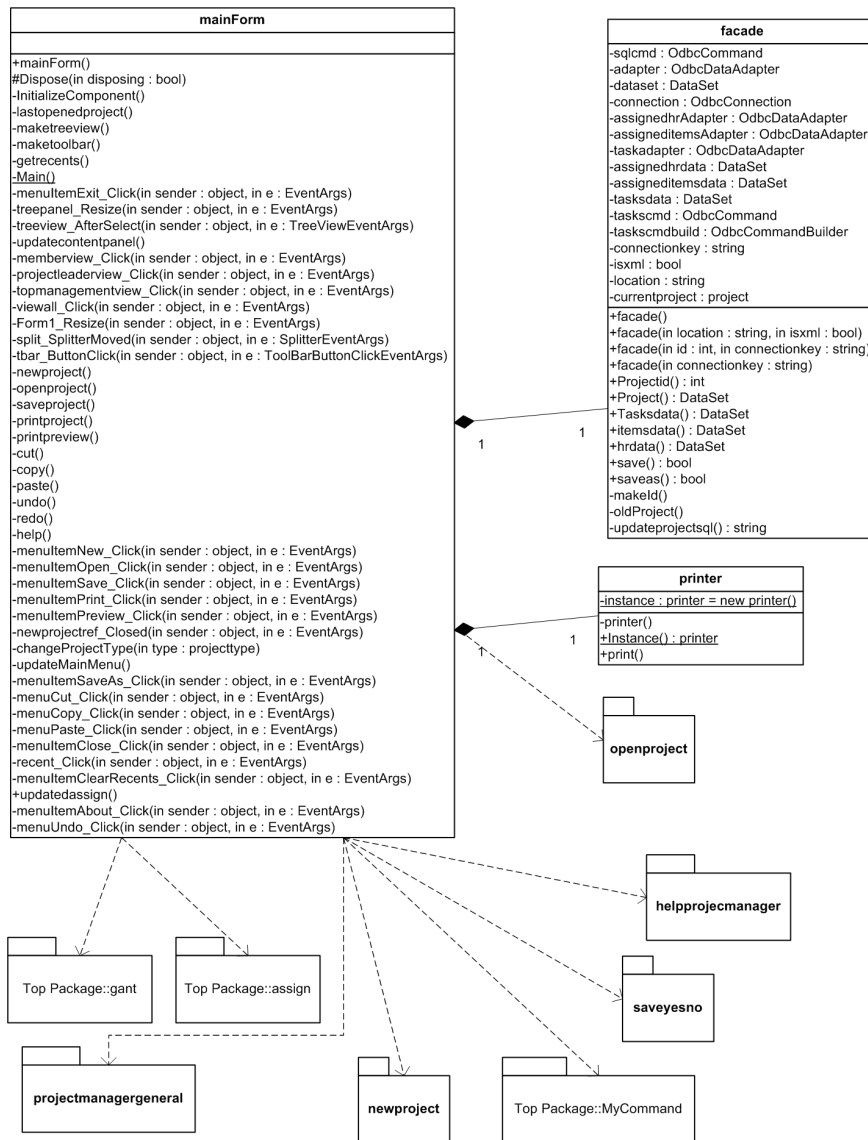
På figur 4.4 ses de klasser, der bruges i forbindelse med at kunne tilføje personer og andre resurser til projektet. Her vil også være klasser til at vise, hvilke resurser der er tilføjet.

På figur 4.5 ses indholdet af den dll, hvor hjælp til programmet skal implementeres. Forløbige er der tilføjet en about box.

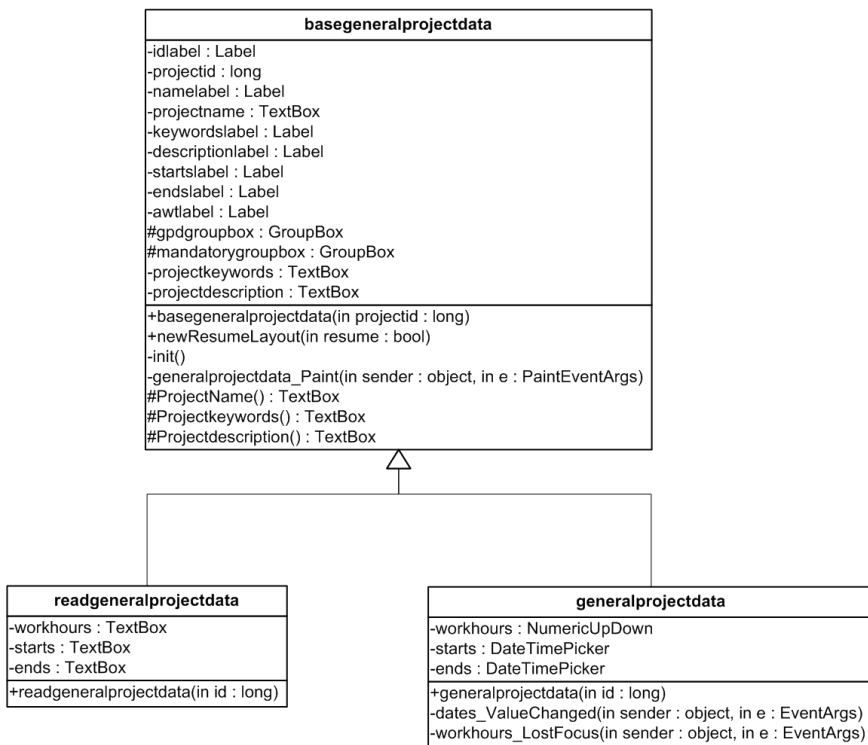
Figur 4.6 viser en dialog, som bruges til at spørge brugeren, om denne ønsker at gemme det nuværende arbejde.

På figur 4.7 ses klasser, som bruges i forbindelse med oprettelsen af et nyt projekt. De to klasser er begge formularer der bruges til at få information om den ønskede projekttype. Alt efter hvilken type projekt, som brugeren vælger, vil der blive vist den ene eller begge forms.

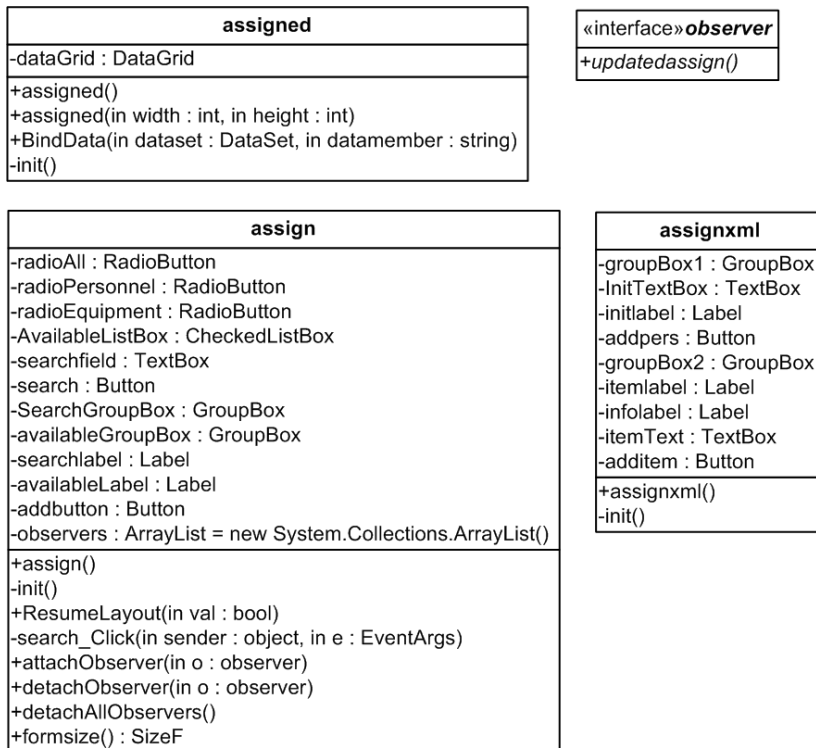
På figur 4.8 side 35 ses de klasser, som bruges i forbindelse med at åbne et nyt projekt. Hvis brugeren ønsker at åbne en xml-fil, vil der blive brugt en standard "open file" dialog. Hvis der vælges at åbne fra en database, vil der fremkomme en form med to lister. Når der vælges en database i den ene liste, vil samtlige projekter fra databasen blive vist i den anden liste. Brugeren kan således vælge et projekt, hvorefter programmet åbner projektet fra databasen.



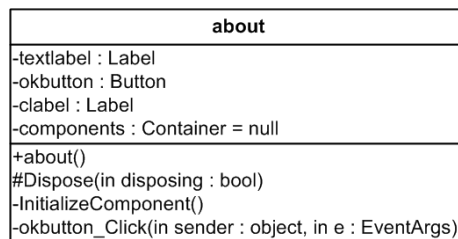
Figur 4.2: UML klassesdiagram for selve programmet



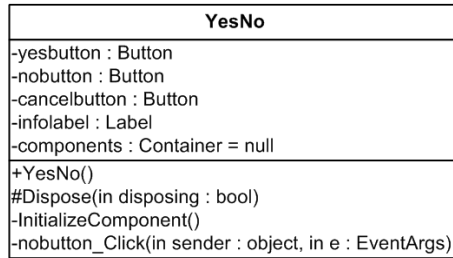
Figur 4.3: UML klassediagram for genel projektinformation



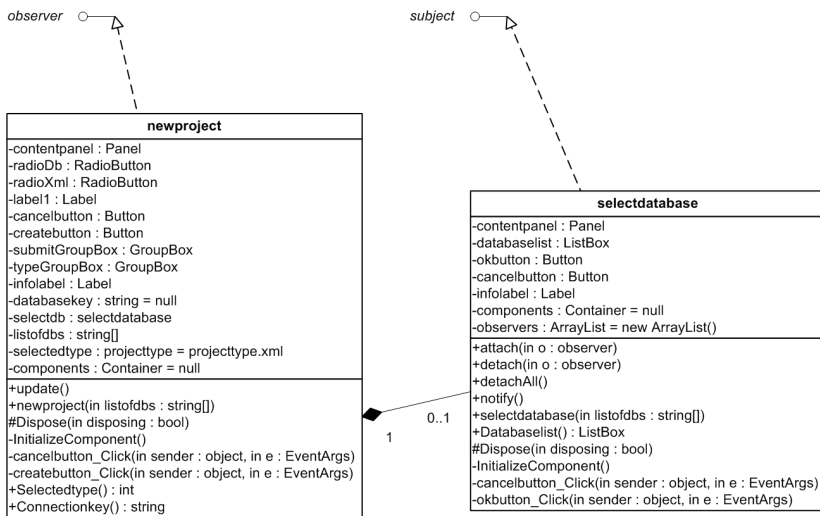
Figur 4.4: UML klassediagram for assignlib



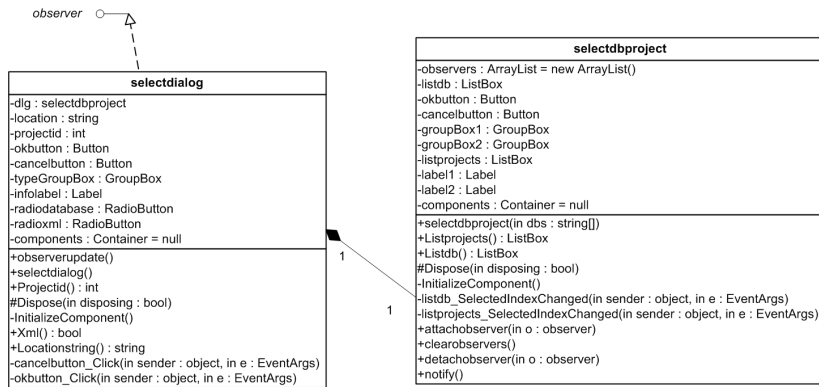
Figur 4.5: UML klassediagram for helpprojectmanager namespace



Figur 4.6: UML klassediagram for SaveYesNo dialogen



Figur 4.7: UML klassediagram for newproject

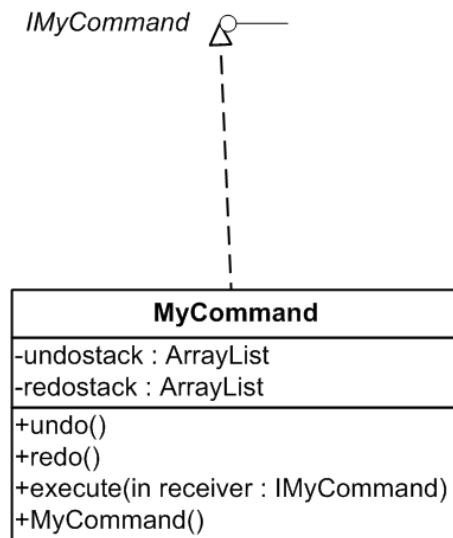


Figur 4.8: UML klassesdiagram for openproject

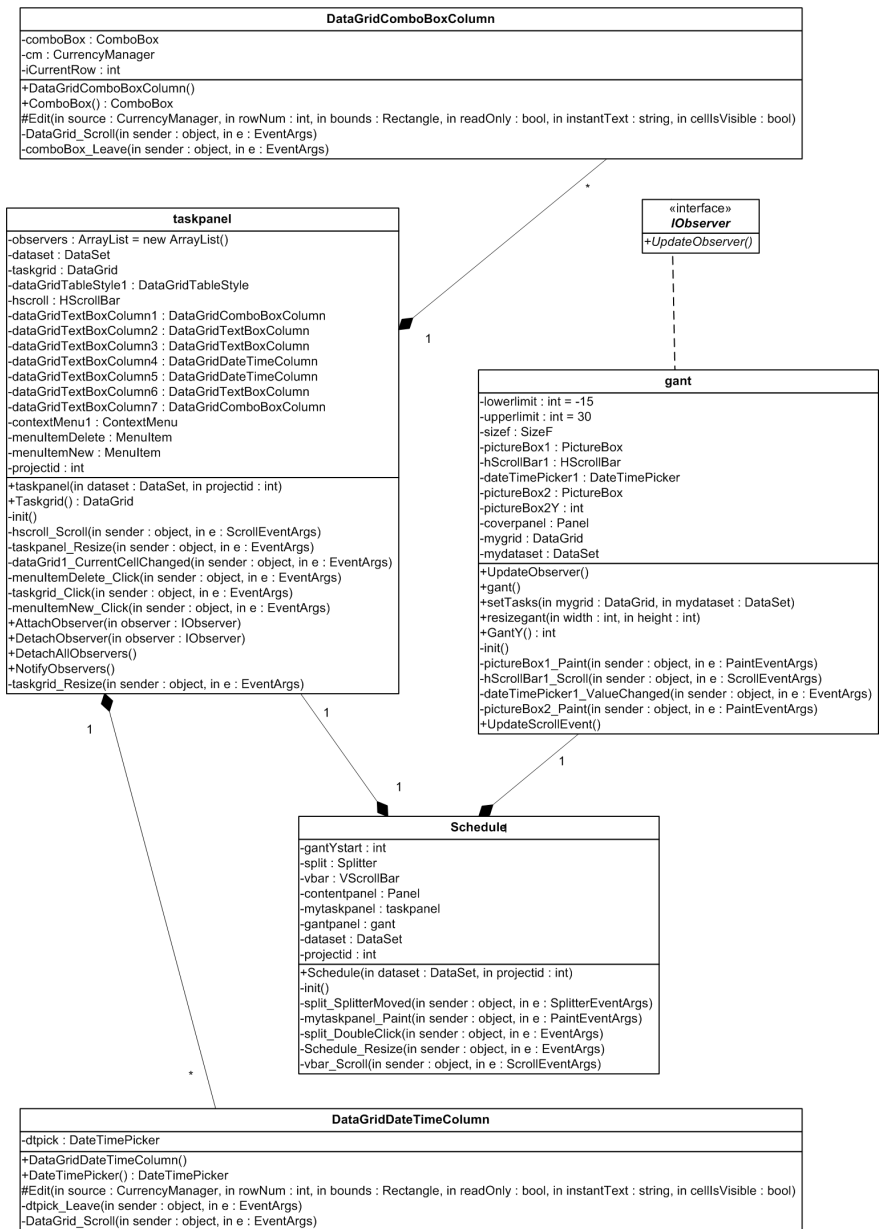
På figur 4.9 er vist en implementering af et command pattern. Dette skal bruges i forbindelse med fortryk operationer. Andre klasser kan eksekvere en kommando og placere en reference i undo-stack'en. Det er krav, at de klasser, som vil benytte undo funktionalitet implementerer det interface, der er vist.

På figur 4.10 side 37 ses en af de absolut mere komplekse brugerkontroller. Af samme årsag er der på figur 4.11 vist hvordan der interageres med kontrollen. Kontrollen, der er et panel, består i virkeligheden af to paneller, delt med en splitter. Det ene panel har et datagrid med særlige kollonestilarter. Der skal ikke her gives en udtømmende forklaring på hvad dette er, men kort sagt er det valgt at implementere datatime-picker komponenter når nogle celler vælges. Der bliver simpelthen vist en særlig kontrol, når for eksempel der udfyldes i start-dato rækken. Selvom eksemplet i [2] angiver datakilde og ikke databinding, giver artiklen en god introduktion til hvad det vil sige, at bruge andre kontroller end text i sit datagrid.

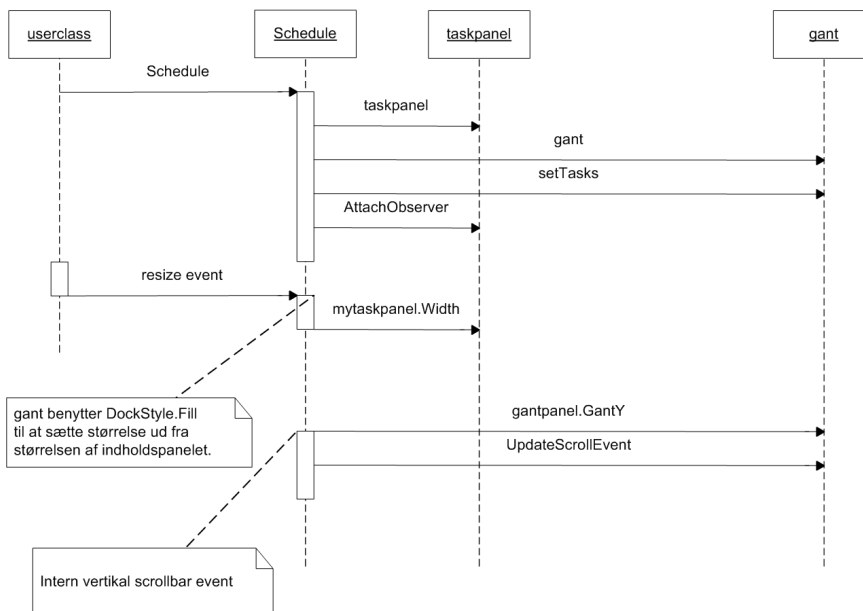
Datagrid'et er løst koblet, ved hjælp af observer pattern, til en særlig "chart"-klasse. I denne klasse kan der vælges en dag, der skal startes ud fra. Der vises så opgaver 15 dage før og 30 dage efter. Øverst vises den dag, hvor opgaven ligger. Hvis det varer flere dage tegnes gant-baren over flere dage. Hvis der er valgt en fase eller milestone i taskgridet, vil disse blive tegnet på en særlig måde.



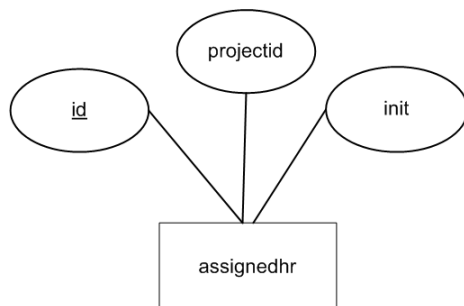
Figur 4.9: UML klassediagram for MyCommand (command pattern)



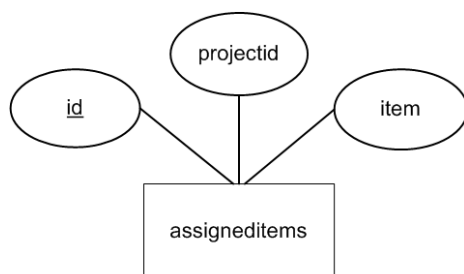
Figur 4.10: UML klassediagram for gant namespace



Figur 4.11: UML interaktionsdiagram for gant namespace



Figur 4.12: assignedhr som ER



Figur 4.13: assigneditems som ER

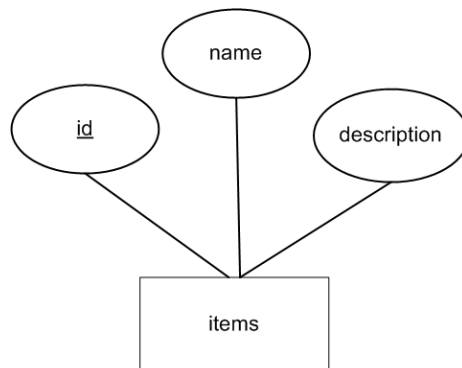
4.4 Databasen

Databasen består indtil videre af seks tabeller. I det følgende er tabellerne i første omgang illustreret individuelt som ER diagram. På figur 4.18 side 42 ses databasen i sin helhed. Bemærk! Selvom der er angivet foreign keys, kan dette ikke implementeres i en standard udgave af MySQL, der er den benyttede database. Relationerne er angivet for at vise, at der er sammenhæng mellem data i de forskellige tabeller. Udvikleren er selv ansvarlig for at opdatere alle felter. Bemærk! Tabellerne overholder Boyce-Codd normalform og dermed også 3. normalform. Se [7] for nærmere beskrivelse af normalisering.

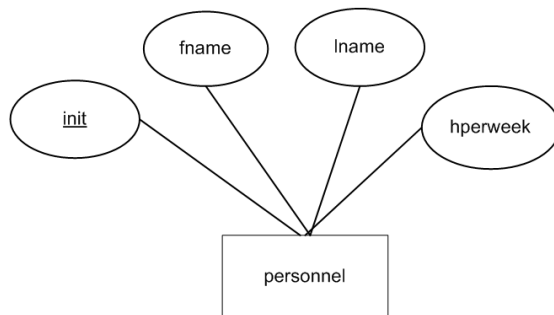
På figur 4.12 ses den tabel, som indeholder personer, der er tilføjet til et projekt.

Figur 4.13 er den tabel, som indeholder udstyr, der er tilføjet et projekt.

Figur 4.14 er en tabel, som indeholder udstyr der kan tilføjes til projekter.



Figur 4.14: items som ER



Figur 4.15: personnel som ER

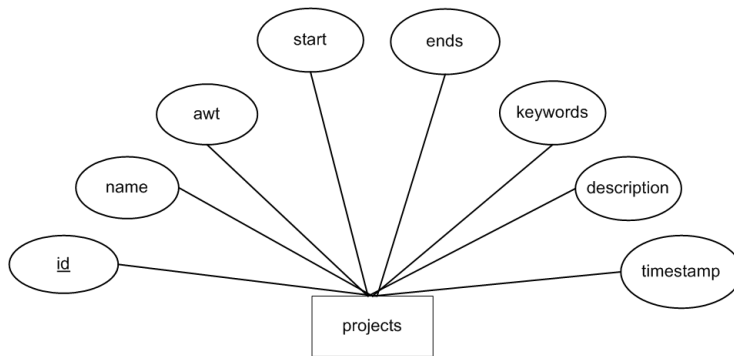
Figur 4.15 er en tabel, som indeholder medarbejder, der kan knyttes til et projekt.

Tabellen vist på figur 4.16 er der, hvor information om selve projektet gemmes.

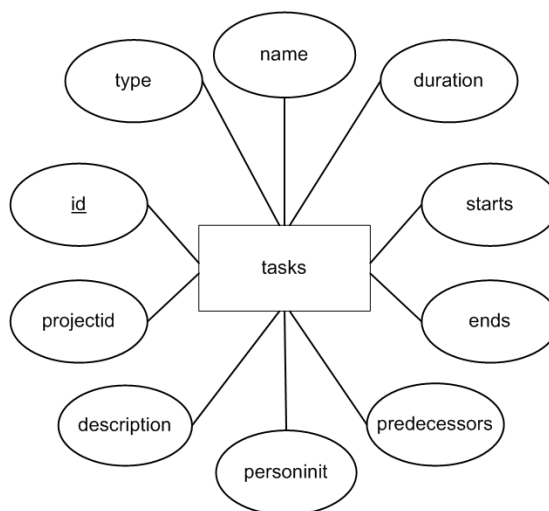
I tabellen figur 4.17 gemmes opgaver, faser og milestones. Som det også fremgår af figur 4.18 er den enkelte task knyttet til projektet via "projectid".

4.4.1 xml strukturen

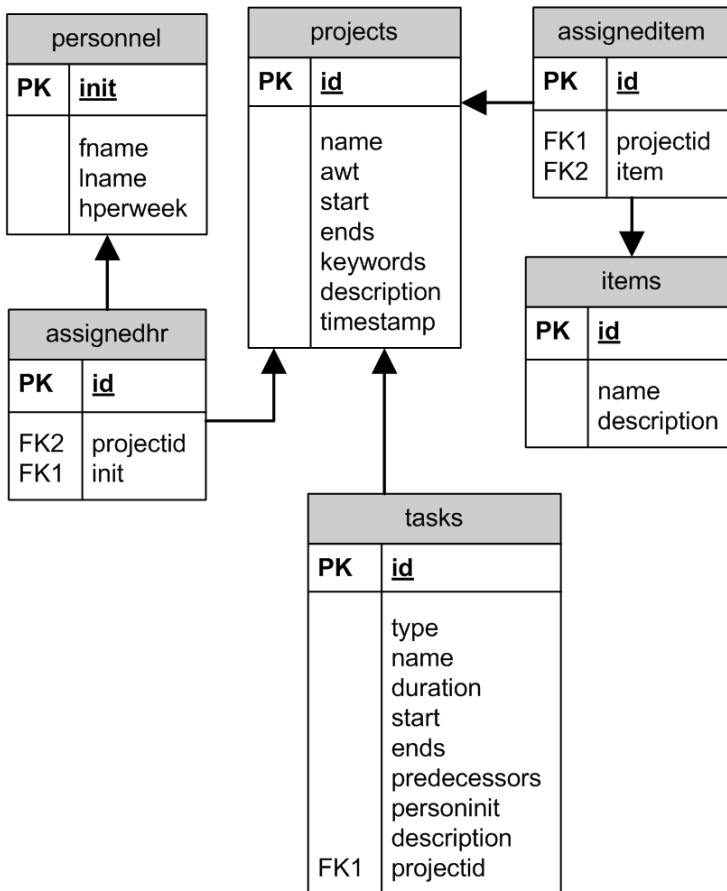
Det smarte ved at benytte xml er, at der kan opbygges en struktur, som er magen til den der findes i database. Det er netop det der sker, når programmet kommunikerer med en xml-fil. For at vise hvor meget de to data typer ligner



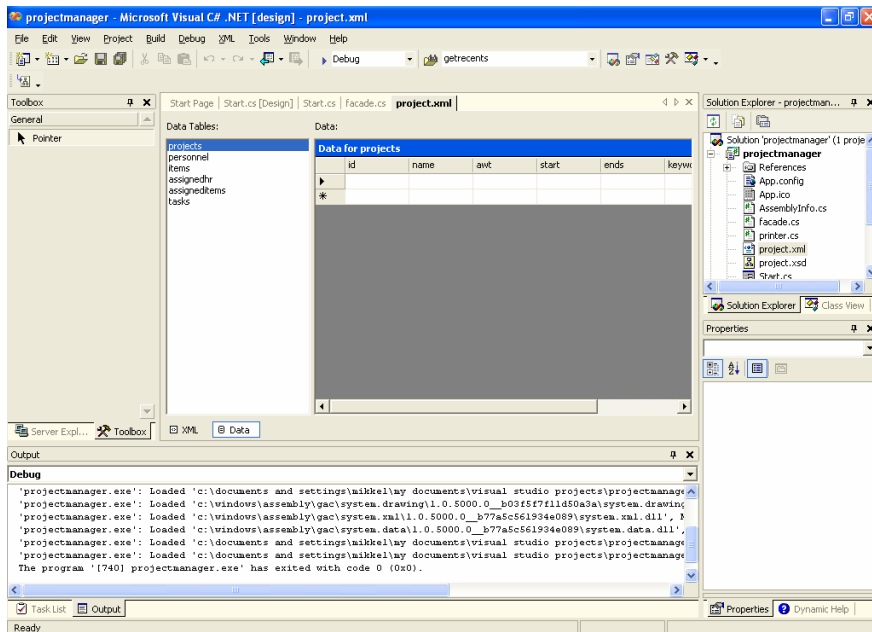
Figur 4.16: projects som ER



Figur 4.17: tasks som ER



Figur 4.18: Databasen i sin helhed



Figur 4.19: Xml visning i Visual Studio

hinanden, kan det på figur 4.19 ses hvordan en xml fil i Visual Studio kan ses som et datagrid, på samme måde som data fra en database vises.

KAPITEL 5

Implementering

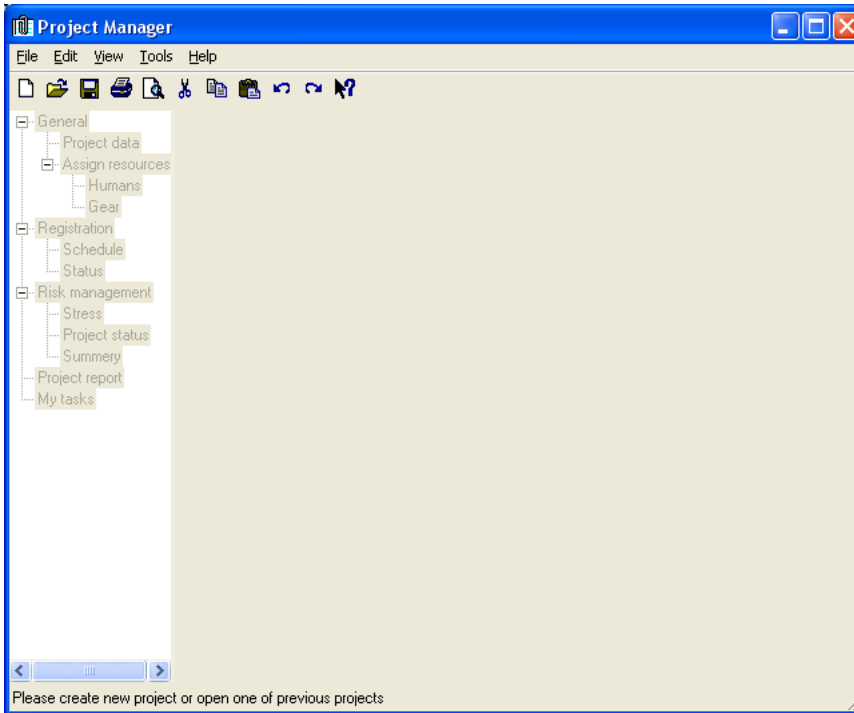
Designet er efter endt implementering blevet opdateret, så det svarer overens med, hvad der på nuværende tidspunkt er implementeret. For at illustrere hvorledes designet er implementeret, følger her en række figurer med beskrivelse.

På figur 5.1 ses hvordan det ser ud, når programmet starter op, såfremt der ikke åbnes et projekt.

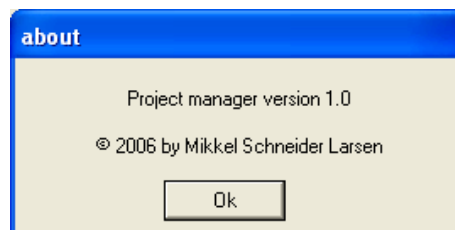
På figur 5.2 ses hvordan about er implementeret. Denne box kan fremkaldes ved at trykke about under help.

På figur 5.3 ses hvordan der kan filtreres i adgang og udseende til de forskellige funktionaliteter.

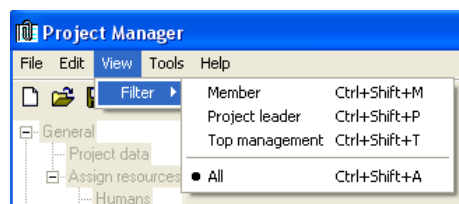
Såfremt læseren ønsker at se mere af programmet, findes det på den medfølgende cd. Desuden er der givet et eksempel på program kørsel i brugervejledningen.



Figur 5.1: Sådan er hovedvinduet implementeret



Figur 5.2: Hvordan about er designet



Figur 5.3: Sådan er filteret implementeret

KAPITEL 6

Testning af systemet

Der er foretaget to former for test, foruden at applikationen løbende er afprøvet af personer uden it relateret baggrund. Dels er foretaget en unit test, hvor de enkelte moduler, der er udviklet, er sat ind i en form, som ikke havde noget med projektet at gøre. Her blev de kørt med debug funktionen i Visual Studio. Fejl er løbende blevet rettet, hvorfor der er ikke på nuværende tidspunkt er kendskab til nogen fejl. Desuden er der foretaget en funktionel test eller black box. Dette er foregået simpelthen ved at eksekvere programmet. Handlinger er blevet testet for at sammenligne med de use cases som er defineret i 3.4 på side 22. I det følgende er kort oplistet hvad der er testet, og om der var fejl i programmet.

- Test om der kan oprettes et nyt projekt, som xml: Ok
- Test om et projekt (xml-fil) kan gemmes: Ok
- Test om et projekt (xml-fil) kan åbnes: Ok
- Test om et projekt (database) kan oprettes med unikt id: Ok
- Test om et projekt (database) kan åbnes: Ok

Desuden er det testet om de tasks, phases og milestones der angives af brugeren fremkommer i den rigtige rækkefølge på gant diagrammet: Ok.

De krav, som er fremsat i 3 er blevet gennemprøvet. Der var ikke umiddelbart nogen fejl i blackbox testen (den funktionelle test). De fejl der derimod løbende er fundet i forbindelse med unit test er rettet.

KAPITEL 7

Brugervejledning

7.1 Nødvendige værktøjer

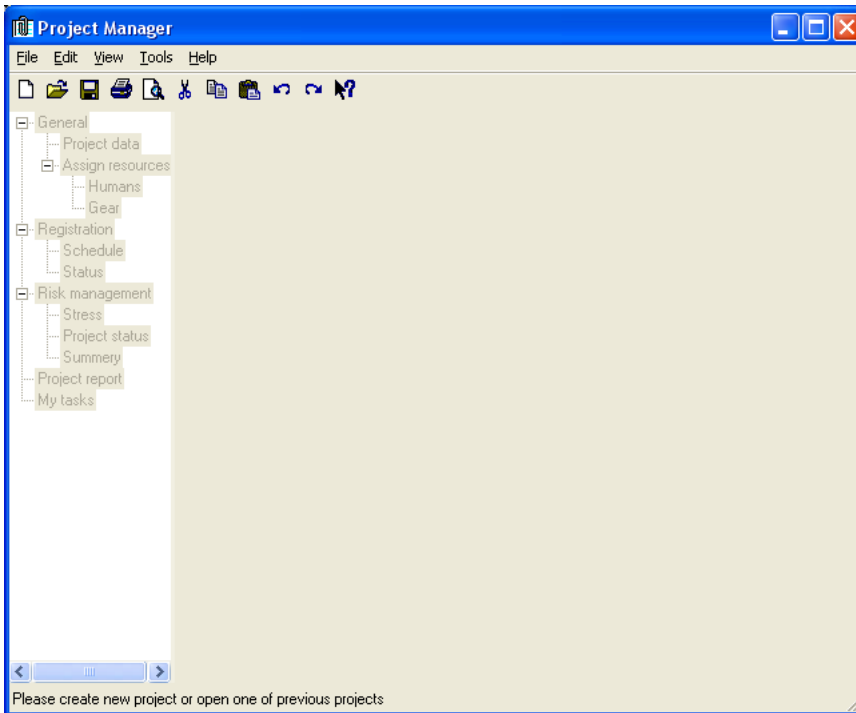
For at kunne gemme projekter i en database, skal der være installeret MySQL. Vejledning i installation og opsætning findes på [1]. På <http://www.mysql.com/> kan der ligeledes downloades en version af MySQL, samt evt. drivere. Hjælp til formulering af sql sætninger, der laver de i 4 nævnte tabeller kan findes på den medfølgende cd. Såfremt der ikke er adgang til en database, vil Project Manager alligevel kunne gemme som xml filer.

7.2 Sådan virker programmet

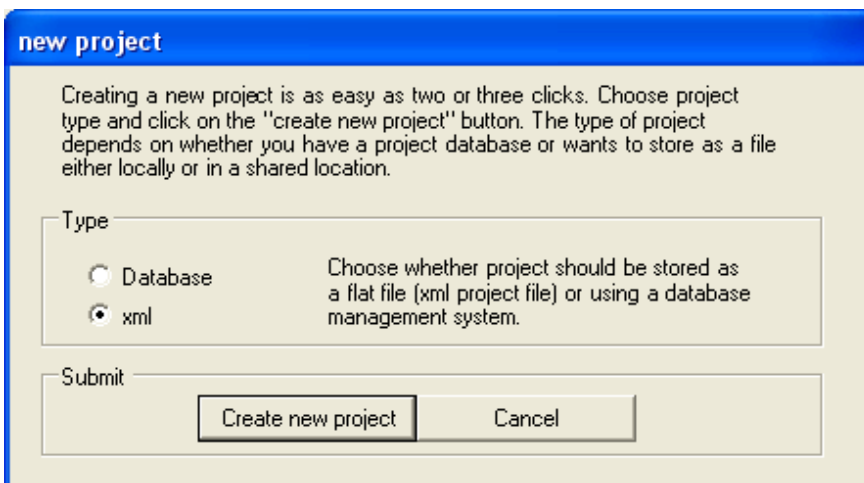
Når programmet startes ses det vindue, der er illustreret på figur 7.1

Klik på "New" ikonet i toolbaren (bemærk holdes musen over toolbarknapperne, vil der fremkomme en kort beskrivende tekst). Der kan også vælges "New" i hovedmenuen eller ved at trykke ctrl+n. Herefter vises dialogen figur 7.2.

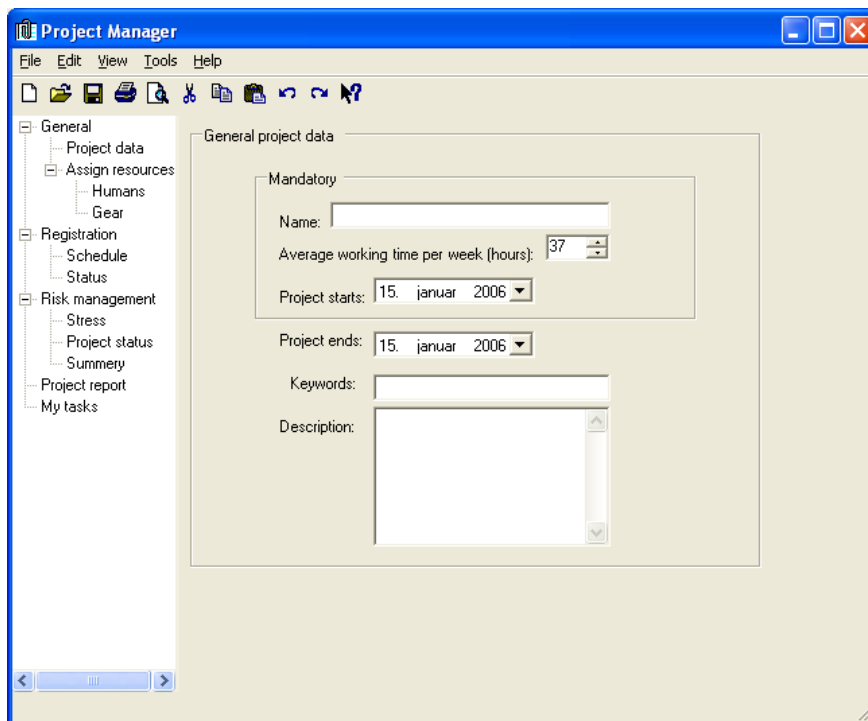
I dette eksempel vil der blive gemt som xml, så vælg xml (standard valg) og tryk



Figur 7.1: Opstart af program



Figur 7.2: Nyt projekt



Figur 7.3: Klar til indtastning

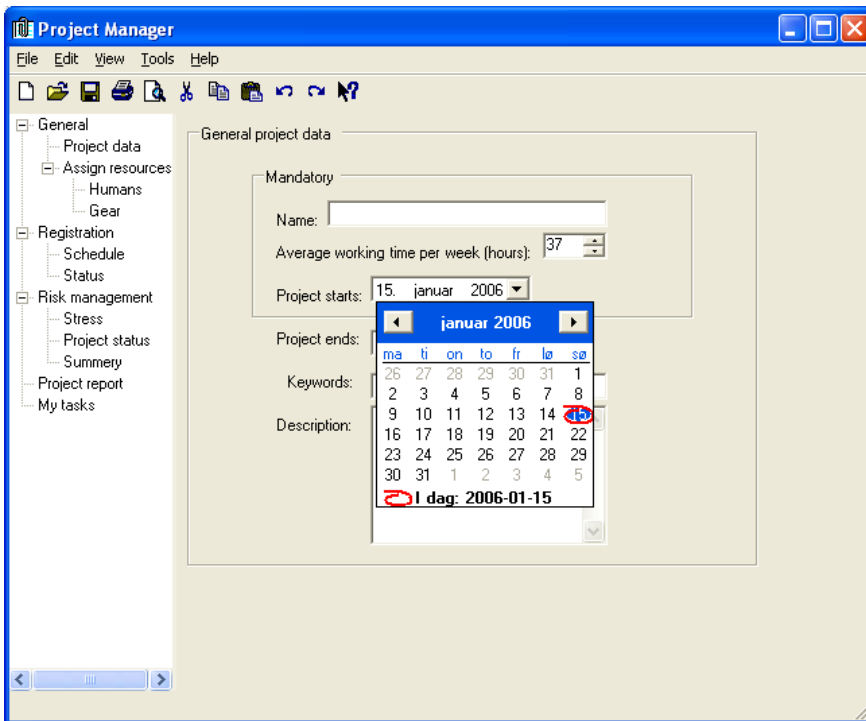
enter eller knappen "Create new project". Der vil fremkomme et billede som vist på figur 7.3. Det er nu muligt at indtaste et navn osv. I dette tilfælde vælges navnet "test".

Start og slutdato sættes ved at trykke på komponenten og vælge en dato i kalenderen (alternativ kan datoen skrives og der kan flyttes om og ned med piltasterne). Se figur 7.4.

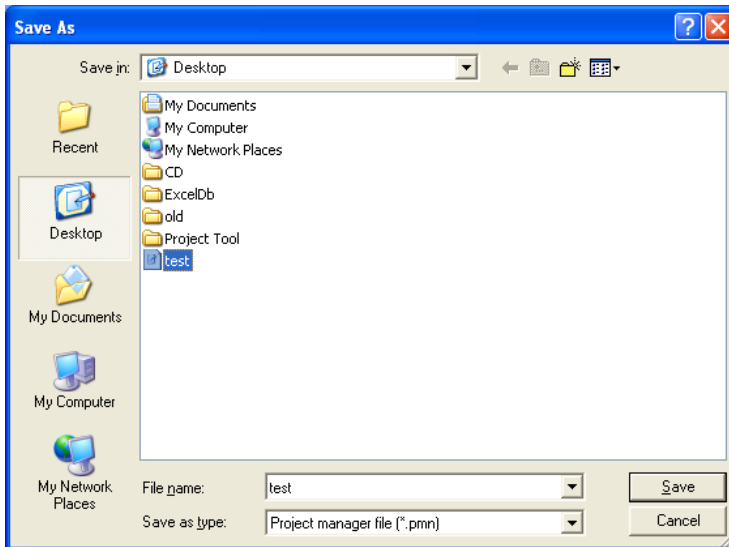
Når formularen er udfyldt trykkes gem (diskette ikon i toolbaren eller ctrl+s). Da projektet ikke har været gemt før fremkommer en "save as" dialog. Se figur 7.5. Vælg det sted, hvor filen skal gemmes.

Prøv evt. efterfølgende at trykke gem, for at se, at fil dialogen ikke kommer frem igen. I træstrukturen vælges nu "Schedule" eller "Registration". Herefter fremkommer et skærbillede som vist på figur 7.6.

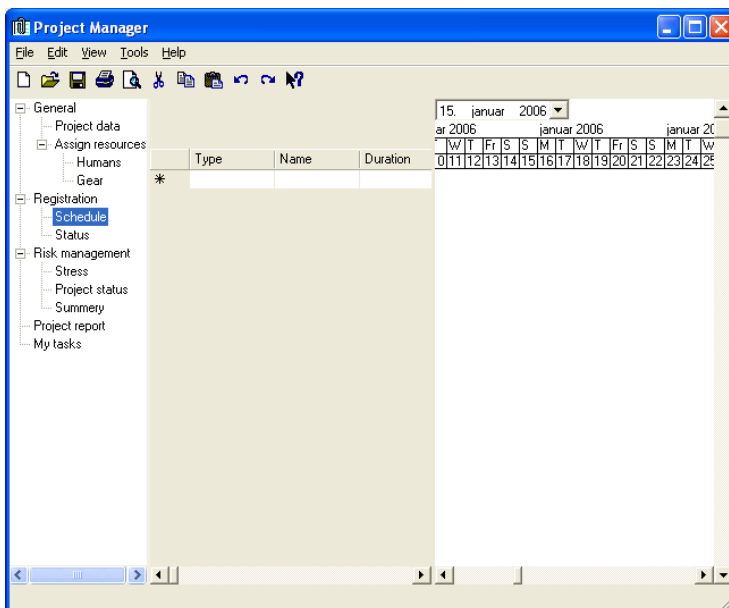
Tryk et sted på datagridet for eksempel på "Type" kolumnen. Der fremkommer



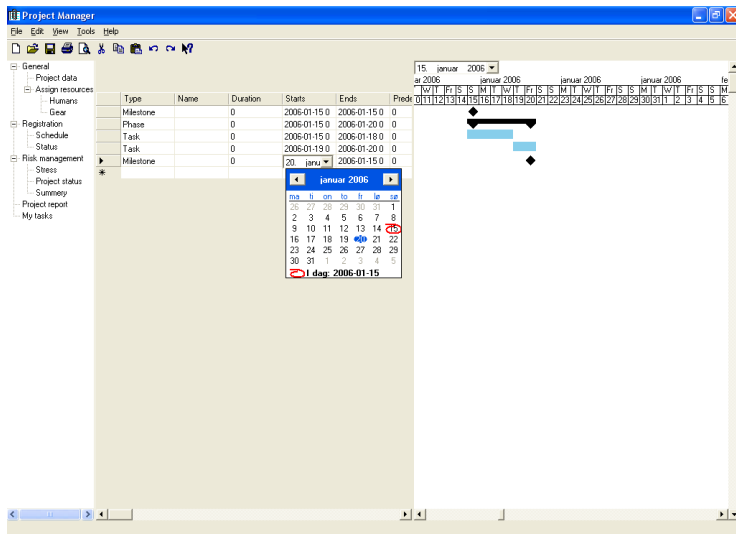
Figur 7.4: Væg en dato



Figur 7.5: Save as



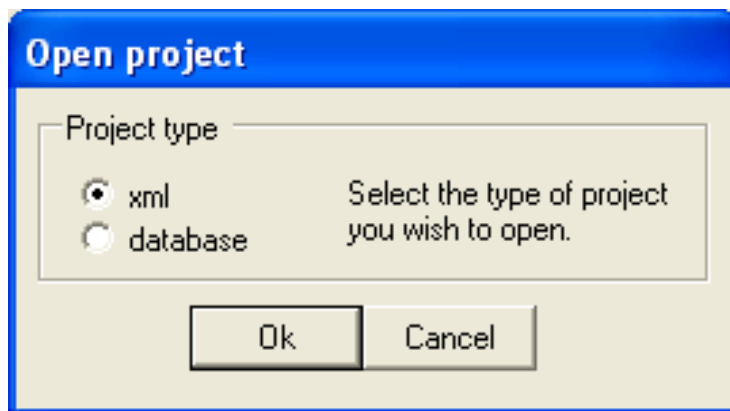
Figur 7.6: Parat til task



Figur 7.7: Indtast tasks

herefter en task. Ved at markere en række og klikke med højre musetast fremkommer en contextmenu. I denne menu kan der vælges at indsætte en ny "entry". Alternativt kan der med musen markeres i "Type" feltet på den række der blev oprettet ved at klikke på grid-kollonen. Indsæt data som vist på figur 7.7. Prøv evt. at vælge anden dato i gant diagrammet til højre. Når dataene er indtastes trykkes gem.

Vælg herefter close via mainmenuen. Projektet lukker og der kan oprettes et nyt projekt. Vælg åben projekt ved at klikke på folderikonet i toolbaren (der kan også trykkes ctrl+o). Vælg xml og peg herefter via dialogen på den fil der tidligere blev gemt (se figur 7.8). Når filen åbne konstateres det i første omgang at de generelle data blev gemt. Vælg "Schedule" via træstrukturen. Det konstateres nu, at også gant diagrammet kan gendannes.



Figur 7.8: Åben xml projekt

Konklusion

Der er konstrueret en foreløbig version af applikationen Project Manager. Applikationen har til formål at hjælpe en projektleder i forbindelse med planlægning af projekt. Denne applikation er testet i forhold til de krav, som er opstillet i 3. Projektlederens rolle er defineret og det blev beskrevet hvad begrebet projekt dækker over. Der er i den forbindelse lagt op til en masse fremtidig udvidelser ikke bare i forbindelse med Project Manager, men ligeledes udvikling af andre applikation, som hjælper med indrapportering i forbindelse med projekter. Det blev til dette projekt valgt at arbejde efter unified process, der er en arbejds-metode til softwareudvikling. Der blev fra start af lagt en tidsplan for dette projekt (se bilag C). Planen er i store træk overholdt, dog var der lidt stress i anden iteration, da der opstod problemer i forbindelse med formatet af en dato. Dette er pga. opdatering af kravene til brugerens systemopsætning midlertidigt løst, men i fremtiden bør det overvejes om commandbuilder er værd at bruge i specielle situationer. For at imødegå, at der kun har været en person tilknyttet dette projekt, er der løbende foretaget mindre brugertest, af folk uden nogen form for it relateret baggrund. Det antages at applikationen, som denne rapport omhandler, har en udvikling på yderligere seks måneder eller mere.

BILAG A

Project Tool - Kort Specifikation

A.1 Målsætning

Project Tool.xls (herefter kaldet PT) er et Excelbaseret værktøj til styring og analyse af aktiviteter og ressourcer i et projekt, baseret dels på import fra andre systemer, i dette tilfælde Siebel, dels egen registrering. Værktøjet er pr. november 2005 kun ca. 80-85 % færdigudviklet.

PT er ikke nødvendigvis et alternativ til MS Project, projektplaner kan udmærket udvikles i MSP, med efterfølgende import til PT. Denne version er dog baseret på manuel flytning af projektdata fra MSP til Siebel, med efterfølgende download til PT. Årsagen til dette mellemed er, at der forventes en integrationsløsning mellem MSP og Siebel inden for ca. 1 år. På den måde sikres konsistens i tekststrengene mellem Siebel og PT, i forbindelse med månedlig download af Actual Hours på aktiviteter, dermed sikres mod fejl i lookup funktioner og IF statements. PT er designet til, af hensyn til filstørrelse, max. 2.000 projektklinier i planen, ca. 30 konsulenter, og en max løbetid på 47 uger.

I afsnit A.2 lidt om overvejelserne i forbindelse med valg af model. I afsnit A.3 gennemgås de enkelte afsnit

A.2 Overvejelser i forbindelse med modelvalg.

Efter Siebels model, består et projekt af 3 niveauer.

- Milestone
- Task
- Activity

Bemærk en del andre steder defineres en Activity som indeholdende X antal Tasks, hvorimod Siebel er modsat - det har voldt nogle problemer i X_2 ¹ at tænke "modsat" - Project Tool er specificeret ud fra Siebels model.

Endvidere mangler jeg nok Phase i toppen, dette er i eksempel filen indbygget i Task teksten. Du bør nok i din model tage hensyn til almene projektledelsesstandarder, hvis usikkert, specificere dig ud af det.

Efter X_2 's projektmodel, opererer man med 3 niveauer i projektorganisationen:

¹Firmaets navn er fortroligt efter aftale med forfatteren af dokumentet.

- Styregruppemedlemmer
- Projektleder og projektgruppe (få nøglepersoner med central indsigt)
- Projektdeltagerne

Det grundlæggende i værktøjet var, at kunne booke ressourcer under de enkelte tasks, og dermed fastlægge overfor egne projektressourcer og kunden, hvilke ressourcer der er til rådighed hvornår. D.v.s. et værktøj primært tiltænkt som støtte til Projektlederen. Den væsentligste forskel i forhold til MSP på dette punkt er at i MSP knyttes ressourcer til som % under en aktivitet, hvor PT arbejder med timer.

Alle andre faciliteter i værktøjet er egentlig en organisk vækst af dette grundlag: Hvor meget belaster vi projektdeltagerne i planen?

Hvordan er fremdriften målt matematisk i projektet i forhold til planen, totalt og på taskniveau? Kan vi producere automatiske "Early warnings" til projektlederen, ved store projekter?

Kan vi lave ekstrakt af data, der letter adgangen til at producere styregruppe materiale.

En generel advarsel - adskillige steder i PT ligger der skjulte værdier, evt. i skjulte kolonner og rækker. Sheetet er ikke protected, og enhver Delete aktion er derfor forbundet med risiko. Hvor der i forbindelse med en handling skal deles noget, er det macrostyret.

A.3 Sheets

A.3.1 Static Data

Dette sheet benyttes til at registrere grunddata i projektet, start- slutdato, projektnavn, projektgrupper og ressourcer. Flere dropdownmenuer i andre sheets genbruger disse data, samt datoer. Her er et mindre problem jeg har nedprioriteret p.t. Sheet "Resources" henter datoerne, og producerer kalenderen, incl. ugenumre. Men ugenumrene over en årsultimo skal justeres i formlen, da Excel som standard bruger amerikanske ugenumre, der afviger 1 fra europa.

A.3.2 Resources

Sheet Resources indeholder hele projektplanen incl. Milestones, Tasks og evt. Activities, og tilføres dernæst ressourcer.

Projektplanen kan produceres på to måder, enten gennem import fra Siebel (macrobaseret, men skal først bygges nu) eller manuelt gennem makroknapperne i toppen af sheetet. På Taskniveau periodiseres det totale antal timer ud på uger.

Efter planen er registreret, kan man begynde at lægge ressourcer ind under de enkelte Tasks, og specificere yderligere i Activity. Ressourcen får timer allokeret ved markering af en given dato ud for navnet, eller markering af flere datoer, og benytte de farvede makroknapper.

Hele workbooken opdateres automatisk, herunder en kontrol mod overbooking, hvis ressourcen i samme uge er allokeret andre tasks. (Den grønne cirkel bliver rød)

En lidt høkeragtig løsning er kontrol inden for dagen. Vælg et navn i dropdown-menuen. Subtotal funktionen i Excel tæller i øverste linie, og en conditional formating farver subtotalen over en given dato, hvis mere end 8 timer allokeret.

A.3.3 Ressource Totals

Sheetet komprimerer data per ressource, og beregner den procentuelle belastning per uge i forhold til standard arbejdsugen på sheet Static Data. Igen, conditional formating giver et visuelt overblik over, hvem der er belastet. Det er faktisk denne tabel, som Sheet Resources' macrobaserede kontrol tilknyttet farveknapperne smager på under overbooking kontrollen.

A.3.4 Sales Opportunies and Credits

En "flad" side uden nogen ben i - en slags Scoreboard, Hvad kan vi sælge af tillægsydelser, og hvad har vi foræret dem, som vi kan bruge senere overfor dem i en forhandlingssituation.

A.3.5 Steering Group I + II

Steering Group I er en statusrapport, baseret på udtræk fra sheet Resources, hvor man efter importen tilføjer afvigende datoer og smileys. Udsendes til styregruppen i forbindelse med de månedlige statusmøder.

Steering Group II er en Risk Matrix, hvor man kategoriserer mulige risici i projektet. Chartet dannes automatisk på baggrund af de indtastede koordinater i tabellen. Dette punkt forsømmes ofte i et projektet, at få struktur og synlighed på risici alle mere eller mindre ved eksisterer i et givet projekt. Risici skal defineres bredt, ikke kun som et givet moduls funktionalitet i forhold til kundens specification, men også menneskelige faktorer.

A.3.6 Task Analysis

Er egentlig en sandhed med modifikationer - jeg importerer godt nok tasks (Resultatet af importen kan ses i pivot tabellerne via View Details), men analysen beskæftiger sig p.t. kun med totalerne - Dette skal jeg have løst inden udgangen af december.

Efter importen, som sker via en .csv fil genereret i Siebel, der derefter lander i en af de to næste sheets (grå tabs), som er rådata, dannes de to pivottabeller. Dernæste løber macroen indexet igennem, og farver cellerne afhængig af afvigelserne i forhold til grundtabellen øverst. Sheetet indeholder endvidere muligheden for at indtaste en prognose, der så lægger sig i forlængelse af Actual Hours.

På den måde har man 3 faktorer at forholde sig til.

- Planen
- Forbrugte timer
- Prognose

På task niveau er det tanken at afvigelserne grafisk skal illustreres, om det er via farvede tabeller eller charts, a la Risk Matrix'en, ved jeg ikke endnu, men der skal nok testes på følgende

- Actual startdate <> Planned start date mere end x % = status på igangværende tasks

- Actual enddate $\langle \rangle$ Planned end date mere end $x\%$ = status på afsluttede tasks
- Actual hours $\langle \rangle$ Planned hours på tidspunkt y mere end $x\%$ = status på arbejdsmængden.

Den sidstnævnte kan måske siges at være lidt kontroversiel. Man kan umiddelbart kun udgå fra en lineær betragtning, men hvem siger arbejdsmængden i en given Task er jævnt fordelt over perioden? Ikke desto mindre fastholder jeg den tese, som er bedre end ingenting. Projektlederen har en procentsats, han kan forholde sig til, og enten selv, eller den faglige ansvarlige for Tasken vurdere, om afvigelsen i forhold til den lineære beregning er inden for forventet.

Alternativet - lidt mere sofistikeret, er at man har mulighed for på en task at indtaste:

Expected Start Date Expected End Date Estimated Hours Estimated hours in percent period I, II, III, IV (de 4 er naturligvis = 100 %)

Endelig kunne man også indlægge en prognose mere sofistikeret end mine procenter. Men nu er vi ovre i noget af det som MSP som det er nu, håndterer bedre, det med at flytte planens tasks og dates. Disse afhængigheder kan dog bygges ind i en Excelkolonne, såvel som en database.

Men min prognose udtrykt i ord: If startdate $<$ today - Expected deviation on remaining hours from today - enten som index, dvs. $110\% = 10\%$ mere end planen, eller som \pm timer.

BILAG B

Begrebs og ordliste for diplom eksamensprojektet

Value chain: I dag bliver en del arbejde, inden for næsten alle brancher, udlisteret. For at en person kan købe eksempelvis en bil, skal bilen produceres og sælges. Typisk specialiserer forhandleren sig inden for salg. En producent specialiserer sig i den overordnede produktion. Producenten udlister måske produktionen af motoren til en underleverandør, der kan levere billigere og bedre motorer. Denne underleverandør udlister produktionen af bolte, da det er billigere end egen produktion. Således opstår der en kæde af virksomheder, som hver især kan producere billigere eller bedre indenfor lige deres område. Alle virksomhederne i kæden modtager derfor hver i sær en del af den endelige købers betaling. Heraf kommer udtrykket "value chain".

Pipelining: Pipelining er det der sker, hvis der er flere maskifulde vasketøj, som skal vaskes hurtigst muligt ved brug af én vaskemaskine og én tørretumbler (de færreste bryder sig vel om at stå og vente på ingenting). I stedet for at arbejde med hele processen på en gang, foretages én vask med vaskemaskinen. Når maskinen har vasket første maskinfuld flyttes dette tøj til tørretumbleren inden næste vask påbegyndes. Når anden vask er overstået flyttes tøjet fra tørretumbleren. Tøjet i vaskemaskinen flyttes til tørretumbleren og tredje vask kan begynde samtidig med, at der tørres tøj og lægges tøj sammen. Da alle led i "produktionen" af rent vasketøj er i gang samtidig, forøges den overordnede produktionshastighed. Dette kaldes også forbedring af throughput. Det mest langsomme led i kæden er afgørende for produktionshastigheden. Optimering af et sådan led kaldes optimering af flaskehalse.

Parallelt arbejde: Pipelining er parallelt arbejde, men parallelt arbejde er ikke nødvendigvis pipelining. Parallelt arbejde er et lidt bredere begreb. Dele af et system kan arbejde parallelt, med forskellige typer opgaver, som i pipelining. To dele af et system kan også arbejde parallelt med samme type arbejde. Hvis der eksisterer n produktionslinjer øges produktionen med en faktor n . Altså flere produktionslinjer giver mere produktion sammenlignet med én produktionslinje.

Matrixorienteret ledelseform Matrixorienteret ledelseform er når to ledere placeret forskellige steder i en organisation, har indflydelse på en gruppe mennesker. Antag f.eks. at der er to afdelinger i et firma. Hver afdeling har en afdelingsleder. Denne ene leder giver en ordre til sin afdeling. Lederen fra den anden afdeling indser, at dette medfører problemer i den del af projektet, som dennes afdeling arbejder med. Derfor giver den anden afdelings leder en kontraordre til den første afdeling. Dette skaber dynamik i firmaet, og sparer måske projektet for nogle problemer. Det er dog ikke uden problemer. Kontreorder skaber forvirring. Hvad hvis lederen for den anden afdeling forsøger at flytte noget arbejde over til en, i forvejen,

overbelastet afdeling? Matrixorienteret ledelseform skaber dynamik, men kan også skabe kaos, hvis ikke kommunikationen fungerer optimalt.

Unified Process Unified process er en arbejdsform indenfor software udvikling. Det forudsættes blandt andet, at der arbejdes itterativt. En række værktøjer blandt andet UML diagramtyper bruges i forbindelse med dokumentationen af de enkelte faser. Hvis det undervejs i implementeringen opdages, at designet bør laves om, behøver testfasen ikke at komme lige efter implementeringen. Der kan skiftes mellem design og implementering inden der udføres test. Unified process er nærmere beskrevet i [6].

Litteratur

- [1] <http://dev.mysql.com/>.
- [2] <http://msdn.microsoft.com/msdnmag/issues/03/08/datagrids/default.aspx>.
- [3] <http://www.c-sharpcorner.com/>.
- [4] <http://www.dofactory.com/patterns/patterns.aspx>.
- [5] <http://www.siebel.com/>.
- [6] Craig Larman. *Applying UML and Patterns - An introduction to object-oriented analysis and design and the unified process*. Prentice Hall, 2nd edition, 2002.
- [7] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. McGraw-Hill Higher Education, third edition, 2003.
- [8] Jacob Hammer Pedersen Ranga Raghunathan Scott Allen, Syed Fahad Gilani and Jon Reid. *Beginning C# Databases*. Wrox, 2002.
- [9] Hans Van Vliet. *Software Engineering - Principles and Practice*. McGraw-Hill Higher Education, third edition, 2003.