

Analyse av prosjekt

INF5700

Amir Hasanbegovic



07. 12. 2008

Innholdsfortegnelse

Innledning	s. 3
Prosjektet i INF5420	s. 3
Prosjektledelse og metodikk	s. 3
<i>Planlegging</i>	s. 4
<i>Utforming av prosjektplan basert på SCRUM</i>	s. 4
<i>Utforming av ny prosjektplan basert på waterfall modell</i>	s. 7
<i>Estimering, scope og risikoanalyse</i>	s. 8
Prosjektledelse og metodikk i HW utvikling og SW utvikling	s. 11
Konklusjon	s. 12
Referanser	s. 14
Vedlegg	s. 15

Innledning

Prosjektledelse og metode har vist seg å være en veldig viktig del av prosjektarbeid innen alle felt. Før et prosjekt settes igang burde det alltid være en plan tilstede som viser til hvordan man skal gå frem for å løse oppgavene på en mest mulig effektiv måte. Dette gjelder også tekniske prosjekter innen IT hvor det har vært en stor utvikling innen prosjektledelse og metode de siste årene. Denne rapporten skal ta for seg viktige prinsipper fra prosjektarbeid innen IT, med hovedvekt på hardware design. Det er i utgangspunktet lite informasjon å finne når det kommer til prosjektledelse og design metodikk innen hardware utvikling og følgelig vil det bli gjort et forsøk på å anvende prinsipper fra software prosjekter på et hardware prosjekt jeg har dette semesteret i INF5420. I den sammenheng vil det bli drøftet hvorvidt utvalgte metoder som har blitt anvendt på organisering av prosjektarbeidet er effektive og om det er elementer som burde unngås i fremtidig prosjektarbeid. Prosjektet i INF5420 er preget av et eksperimentelt aspekt og det vil dermed bli gjort et forsøk på å finne en god måte på å håndtere slike prosjekter. Siden metodene som blir brukt i denne rapporten hovedsaklig stammer fra software utvikling, vil det også bli sammenlignet hvorvidt prosjektledelse og design metodikk innen hardware skiller seg fra prosjektledelse og design metodikk innen software.

Prosjektet i INF5420

Studieretningen for Mikroelektronikk har et obligatorisk kurs, INF5420 i 7. semester som går ut på å fullføre et bedriftsrelatert prosjektarbeid. I utgangspunktet skal prosjektarbeidet utføres i en relevant bedrift, men siden det bare er meg som tar kurset, ble prosjektet utført under oversyn av veilederen i kurset, som da tok rollen til oppdragsgiveren. Formålet med denne prosjektoppgaven er å få erfaring i hvordan profesjonelt prosjektarbeid i hardware design skal gjennomføres og hvordan man best skal kunne forholde seg til oppdragsgivers kvalitetskrav. Prosjektarbeidet gikk ut på å lage en *nivåkonverter* som er i stand til å konvertere lave spenninger til høye spenninger. Design av en slik krets er hovedsaklig basert på analog design. Formålet med en nivåkonverter er å koble sammen komponenter som har forskjellige forsyningsspenninger, noe som gir mulighet for redusert effektforbruk i kretsen. Siden nivåkonvertere som konverterer subterskel spenninger til høye spenninger fortsatt er et relativt nytt tema, så har ikke oppdragsgiver gitt noen strengt spesifiserte spesifikasjonskrav til den endelige kretsen. Den viktigste spesifikasjonen er at nivåkonverteren skal kunne integreres i en industriell applikasjon. Nivåkonverteren skulle tilfredsstillende følgende kvalitetskrav:

- Nivåkonverteren skal være i stand til å konvertere 180mV signaler til 1.2V signaler.
- Nivåkonverteren skal fungere på et temperaturområde mellom -40 grader celsius og +85 grader celsius.

Ytterligere relevante spesifikasjoner i et bedriftsrelatert prosjekt er blant annet hastighet, effektforbruk og arealforbruk. Disse egenskapene skulle jeg vurdere selv og foreta en endelig avgjørelse til slutt.

Prosjektledelse og metodikk

I denne delen av rapporten skal jeg ta for meg de forskjellige prosjektplaner og design metoder. I prosjekter som skal tilfredsstillende industrielle krav kreves det stort apparat og mye ressurser for å planlegge og utføre et potensielt vellykket prosjekt. I mitt tilfelle er disse kravene redusert og tilpasset det som har blitt gjort i dette spesifikke prosjektet.

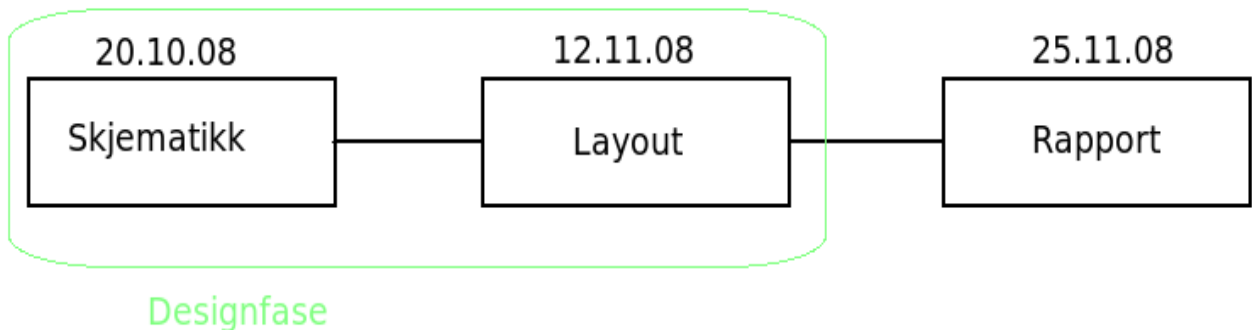
Planlegging

Planleggingsfasen er en av de viktigste og mest omfattende delene i et prosjektarbeid, og her er det viktig å bruke litt tid. Det er ofte fristende å begynne med selveste designet før en plan er på plass og dette er ikke så rart siden teknologien er i så hurtig endring, spesielt innen hardware utvikling. Prosjektarbeid uten en gjennomtenkt form for plan kan lett føre til et mislykket prosjekt. I starten av hvilken som helst designfase kreves blant annet stor investering, måtte dette være utstyr, personell eller andre behov. Uten en plan på plass er det lett å miste både kontrollen og oversikten over de forskjellige aspektene i prosjektet og dermed ende med å kjøre prosjektet i grus.[3] Prosjekter, uansett omfang, burde planlegges til en passende grad. I tilfellene der det ikke er nøye planlagt hva som trenger å bli gjort og når det skal gjøres, kan føre til unødvendig bruk av ressurser på feil områder i prosjektarbeidet. Dermed er det viktig å planlegge nøye, og på en måte som passer prosjektet best.

Siden prosjektoppgaven min hovedsaklig bestod av en design- og en rapportfase valgte jeg å basere planleggingsfasen på disse to aspektene. Kort sagt var planleggingen basert på følgende prinsipper og gjaldt for prosjektet i sin helhet samt hver deloppgave;

- 1) Hva trenger å bli gjort?
- 2) Når skal det være ferdig?

Før prosjektet var i gang ble det satt opp tre tidsfrister i samarbeid med oppdragsgiver. Disse tidsfristene skulle kontrollere hovedpunktene i prosjektet, utenom disse hadde jeg frie tøyler til å kontrollere resten av prosjektet slik jeg selv ville.



Figur 1) Høy-nivå plan for hardware prosjektet

Utforming av en prosjektplan basert på SCRUM

Fremgangsmåten min gikk ut på å prøve å anvende prosesser fra agil metodikk på hardware prosjektet mitt, hovedsaklig på grunn av at det så ut til å gi grei struktur i prosjektet. Jeg valgte å prøve meg på prosesser fra SCRUM metodikken.

Selv om jeg fikk bekreftet at SCRUM har blitt brukt i digital hardware utvikling, så viste det seg å til at SCRUM medførte til komplikasjoner i analog/mixed-signal design, noe som ble oppdaget ganske tidlig i designfasen.[10] Fremgangsmåten min gikk ut på å sette opp *product backlog* og *sprint backlog* for å ha en detaljert oversikt over oppgaver som skulle gjøres og hvor lang tid som var satt av til hver oppgave.[8] Se Tabell 1 og Tabell 2.

Prioritet	ID	Beskrivelse	Est.	Av
Sprint 1 Skjematikk				
M	1	Design av standard nivåkonverter	10	A.H
H	2	Teori og diverse arkitekturer	20	A.H
M	3	Implementasjon av alle komponenter	7	A.H
M	4	Vurdere tradeoffs og implementere	15	A.H
H	5	Mitt personlige bidrag	25	A.H
Sprint 2 Layout				
H	6	Standard layout prosedyre	20	A.H
M	7	Optimalisering av arealforbruk	10	A.H
M	8	Verifisering og håndtering av diverse problemer	18	A.H
Sprint 3 Rapport				
H	9	Relevante simuleringer og oppsett	10	A.H
M	10	Skrijving	16	A.H

Tabell 1) Product backlog

ID	Beskrivelse	Est	Tid brukt
Sprint 1			
1			
	Research	6	
	Design	4	
2			
	Research	10	
	Design av flere arkitektruer	7	
	Verifisering	3	
3			
	Bestemme beste arkitektur	2	
	Implementasjon av beste arkitektur	2	
	Verifisering	3	
4			
	Research	5	
	PSRR	3	
	Lekasje	3	
	Effektforbruk	3	
5			
	Ytterligere research på valgt arkitektur	6	
	Desing	10	
	Tradeoffs	7	
	Verifisering	3	
Sprint 2			
6			
	Deign	10	
	DRC	3	
	LVS	7	
7			
	Research - teknikker og strukturer	5	
	Desing	5	
8			
	Nødvendige endringer	10	
	Verifisering og simulering	8	
Sprint 3			
9			
	Matlab oppsett	2	
	Respons	2	
	Diverse	6	
10			
	Generell skrijving IMRaD	16	

Tabell 2) Sprint backlog

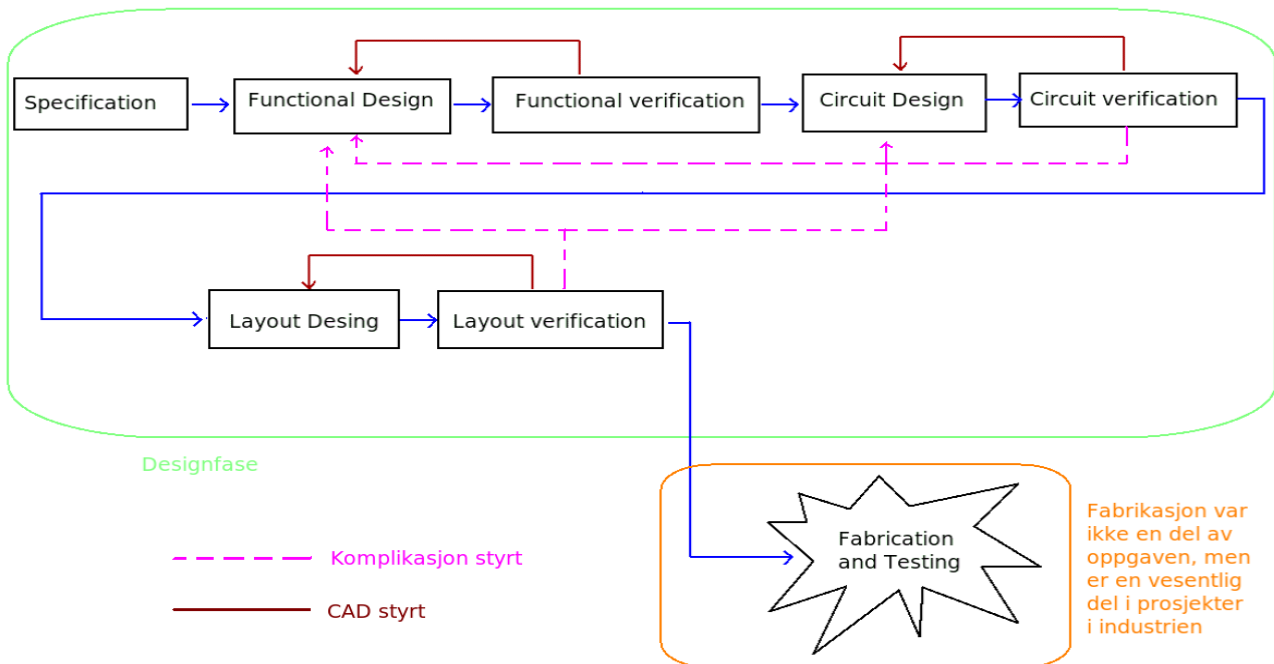
I teorien så det ut til at agil metodikk kunne fungere på et analog/mixed-signal prosjekt. Problemene oppstod da jeg oppdaget at det var vanskelig å holde oversikt over interaksjonen mellom planen og prosjektet. Dette fant jeg ut kunne skyldes flere grunner. For det første var spesifikasjonene fra oppdragsgiver litt tynne noe som førte til at prosjektet hadde et vesentlig eksperimentalt aspekt til det. Fra starten av prosjektet var flere deloppgaver i endring samtidig. Dette gjorde kontinuerlig oppdatering av planen vanskelig, og virket som en unødvendig prosedyre som bare tok opp tid og samtidig reduserte oversikten. Dette impliserer også et stort behov for feedback i planen og selv om SCRUM i utgangspunktet har innebygd feedback, ble det vanskelig å opprettholde oversikten på grunn av at flere deloppgaver var i endring til samme tid. Bidraget fra sprint backloggen gjorde at det ble oversikt over hva som trengte å bli gjort i hver deloppgave, men samtidig reduserte den oversikten i den forstand at estimeringen ble en unødvendig detalj. Siden oppgaven var uforutsigbar medførte estimeringen i sprint backloggen ble en distraksjon og dermed var til liten nytte.

En annen grunn til at SCRUM tilnærmingen feilet kan ligge i feilestimering av tiden, noe som førte til redusert kontroll på prosjektet. En løsning kunne ha vært å overestimere tiden på hver deloppgave og dermed øke kontrollen ved å gi rom for at feedback ble gjennomført. Overestimering er derimot en gode man ikke alltid har tilgjengelig, men i de tilfellene man har muligheten til det burde det utføres med varsomhet siden det kan fort føre til negative effekter som økt "time to market". Underestimering er dessuten ganske vanlig i hardware prosjekter, og er kjent til å kunne være så lavt som tre ganger lavere enn det endelige tidsforbruket i prosjektet. (personlig samtale med prof. Sorre Aunet 21.10.08) I eksperimentelle prosjekter er det vanskelig å estimere tiden som det regnes med å bruke på per deloppgave. Dette er en uforutsigbar prosess i seg selv, og når den blandes med design av høy uforutsigbarhet, som analog design, så er det vanskelig å holde seg til planen på en hensiktsmessig måte.

Bruk av SCRUM er hovedsaklig rettet mot software utvikling hvor den har vist seg å fungere godt. Det virker derimot som at analog/mixed-signal hardware design er alt for uforutsigbart til at SCRUM skal være anvendbart på en hensiktsmessig måte. Det ble for mye å holde styr på både når det gjelder forskjellige faser i prosjektet og selve prosjektplanen, noe som kanskje er unødvendig med tanke på at det bare er en person som deltar i prosjektarbeidet. Med flere deltagere i prosjektarbeidet kunne SCRUM muligens vært mer appellerende også innen analog/mixed-signal design, forbeholdt en større prosjektoppgave der det var større behov for organisering. I små prosjekter der det bare er en deltager kan det tenkes at noen få retningslinjer og begrensninger burde være nok til å utføre et vellykket prosjekt. SCRUM tilnærmingen fremkommer som overdådig i forhold til et lite prosjekt som går ut på design av en nivåkonverter. Dermed ble det gjort et forsøk på å lage en ny prosjektplan med hovedfokus på to prinsipper, nemlig retningslinjer og begrensninger.

Utforming av ny prosjektplan basert på waterfall modell

At SCRUM tilnærmingen i utgangspunktet feilet ble observert veldig tidlig i prosjektarbeidet og dermed ble det utarbeidet en ny prosjektplan og satt opp en ny modell, se Figur 2.



Figur 2) Design flyt for hardware prosjektet

Modellen jeg valgte er en iterativ modell med feedback inspirert av *waterfall* modellen.[4][2] Feedbacken i denne modellen er ikke bare styrt av komplikasjoner i designfasen. Denne modellen tar også hensyn til utstyret (CAD verktøyet) som brukes under designfasen, som da også er med på å styre feedbacken.

Den nye modellen har en kontrollert struktur og tar samtidig en generalisert form når det gjelder design av hardware. Dette gir mer frihet i designfasen og gir mulighet til å foreta relevante og intuitive tiltak underveis rettet mot design av nivåkonvertere som i mitt tilfelle. Egenskapene dette brakte med seg var også at interaksjon mellom plan og prosjektarbeid ble forbedret ved at kontinuerlig oppdatering av planen ble oversiktlig. Den nye modellen skal i utgangspunktet tillate bruk av *ad hoc* prosedyre for å kompensere for den tidligere nevnte høye uforutsigbarheten ved analog/mixed-signal design. Slik prosedyre egner seg hovedsaklig til mindre prosjekter der man ønsker økt fleksibilitet som ikke går utover oversikten i prosjektet. Større prosjekter har mer nytte av generaliserte prinsipper hvor *ad hoc* prosedyre burde unngås så mye som mulig, som da gir øket gjenbruks verdi til design modeller.[10]

Den nye planen ble utarbeidet ved å droppe sprint backlogen og modifisere product backlogen som ble benyttet i SCRUM tilnærmingen. Den nye planen gjorde det lettere å identifisere de forskjellige designfasene, samtidig som at den medførte til både økt kontroll og oversikt. (I motsetning til erfaringene med den agile fremgangsmåten) Planen fikk følgende form:

ID	Beskrivelse	Tidsbegrensning	Ny ID
Skjematikk			
1	Design av standard nivåkonverter	5	ID(1)
2	Teori og diverse arkitekturer	7	ID(2)
3	Implementasjon av alle komponenter	3	ID(2)
4	Vurdere tradeoffs og implementere	7	ID(2),ID(3)
5	Mitt personlige bidrag	6	ID(2),ID(4)
<i>BANK</i>		49	
Layout			
6	Standard layout prosedyre	10	ID(4)
7	Optimalisering av arealforbruk	5	ID(6)
8	Verifisering og håndtering av diverse problemer	8	ID(4),ID(6)
<i>BANK</i>		25	
Rapport			
9	Relevante simuleringer og oppsett	3	ID(4),ID(6)
10	Skriving	10	ID(9)
<i>BANK</i>		13	

Tabell 3) Fungerende plan for designfasen basert på risiko ranking

Som nevnt over, hadde prosjektet mitt et eksperimentelt aspekt til det. I slike prosjekter kan arbeidet bli statisk ved at man setter seg fast og kaster bort tid på oppgaver som ikke fører til noe konstruktivt. Dermed implementerte jeg tidsbegrensninger for hver deloppgave for å gjøre arbeidet mer dynamisk.[11] Dette gjør at prosjektet er i konstant utvikling og man produserer data kontinuerlig. Det kan være litt ekstremt, men eksperimentelle prosjekter har svikt som en del av planen og derfor er det viktig å produsere så mye fornuftig data som mulig iløpet av prosjektarbeidet. Tidsbegrensningen per deloppgave setter restriksjoner på hvor lenge man kan jobbe med en oppgave før man enten går videre eller eventuelt går i feedback til en tidligere deloppgave. Siden planen er delvis risiko drevet kan tidsbegrensningene være relativt strenge og er satt slik at man fort skal komme i gang og fortest mulig produsere resultater. [6]

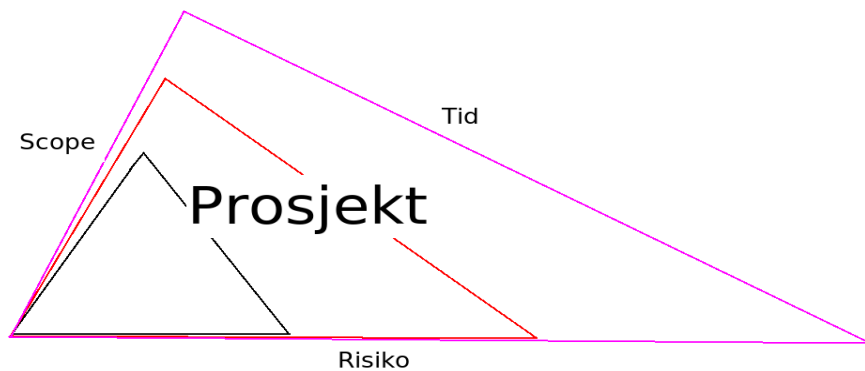
Måten planen fungerer på er ved å ha et gitt antall timer estimert og fordelt på de forskjellige deloppgavene. Deretter setter vi de resterende timene i en "bank" og fordeler dem senere ved behov hvis det skulle hende at en deloppgave går i feedback eller vi velger å gå videre. Dersom det hender at deloppgaven ikke bruker like mye tid som estimert, så legges disse timene tilbake til "banken". Komplikasjonene ved denne planen er ganske åpenbare. Hvordan vet vi hvor mye tid vi kan fordele per deloppgave og hvordan vet vi hvor mange ganger en deloppgave kan gå i feedback?

For å bestemme tidsfordelingen baseres det på risikoanalysen som blir diskutert senere i teksten. Tidsfordelingen er hovedsaklig avhengig av sannsynligheten for at en negativ hendelse inntreffer og dens innvirkning på prosjektet.

Estimering, scope og risikoanalyse

For en prosjektleder er det viktig å vite hvor mye tid som blir investert i et prosjekt og om tiden investert er investert fornuftig. For å få oversikten på dette må prosjektlederen benytte seg av estimering. Fornuftig estimering kan være forskjellen mellom et vellykket og et mislykket prosjekt. Estimering krever mye research og det er mange elementer som spiller inn. Eksempler på disse elementene kan være bredden på de ansattes kunnskap, markedstilstand og mange flere. I dette prosjektets tilfelle er estimeringen hovedsaklig basert på scope og risiko med hensyn på design. Etter at scope- og risikoanalysen ble gjennomført, benyttet jeg meg av min egen erfaring innen analog/mixed-signal design for å estimere antall timer hver deloppgave potensielt kommer til å

kreve. Dette prosjektet som prosjekter flest er avhengig av scope og grovt sett kan innvirkningen fra scope på tid og risiko for prosjektet mitt slik ut: [5]



Figur 3) Forhold mellom tid, risiko og scope med forskjellige nivåer av scope.

Ut ifra Figur 3 kan vi observere at dess større scope, så øker risikoen og dermed også estimert tidsforbruk i prosjektet med ca det dobbelte av det opprinnelige. Scope i dette prosjektet beskriver hvor mange og hvor strenge spesifikasjonene for nivåkonverteren blir. Det som er greit å spesifiserer her er at når det gjelder eksperimentelle prosjekter, så kan scope være vanskelig å determinere. Kreativitet og utforskning spiller en stor rolle underveis i prosjektet og dermed kan innvirkningen fra scope på tid og risiko være eksponentiell. [11]

Fremgangsmåten for risikoanalysen er delvis motivert ut fra fremgangsmåten brukt i The Spiral Model [2] flettet sammen med en kvalitativ ranking metode. [6] I risikoanalysen baserte jeg meg på følgende punkter:

- 1) Identifiser prosjektets 10 viktigste risiko elementer.
- 2) Planlegg og vurder en handlingsplan for risiko elementene.
- 3) Utfør passende handling for å håndtere eller eventuelt kontre risiko elementene.

Det kan tenkes at i større prosjekter skal disse punktene gjentas flere ganger i løpet av prosjektløpet. Risiko elementene burde igrunn overvåkes og sammenliknes, men siden prosjektet mitt kan anses som et mindre prosjekt, så er ikke dette nødvendig. Ved å sett opp de 10 viktigste risikoelementene får man er ganske god oversikt over hva som kan gå galt og får samtidig muligheten til å håndtere de viktigste forekomstene av negative hendelser på en effektiv måte. Når det kommer til eksperimentelle prosjekter, burde denne prosedyren gjøres mer dynamisk, vet for eksempel bruk av post it lapper som man har synlig til enhver tid under prosjektarbeidet.[11] Dette ble praktisert i prosjektarbeidet, men forholdsvis avslappet nivå.

Etter at risiko elementene var plukket ut, vurderte jeg hvert risiko element basert på min erfaring innen analog/mixed-signal design. Dette tilsvarte rankingen i risikoanalysen hvor omstendigheten ble vektet med enten L, M eller H . Det er ranking metoden som forteller oss hvor mange timer vi burde investere i en deloppgave som går i feedback. Det som er viktig er å identifisere de negative hendelsene i designfasen med de oppførte risikoelementene i risikoanalysen. Etter å ha vurdert mulige negative inntreffelser, kom jeg frem til følgende tabell:

#	Risiko	Sanns.	Innvirk.	Handling
1	Urealistisk å møte gitte spesifikasjoner	L	H	Konsultere med oppdragsgiver
2	Manglende kunnskap innen analog design	M	M	Investere mere tid; Konsultere med diverse professorer/studenter
3	Urealistisk tidsplan	M	H	Minke spesifikasjoner (scope)
4	Mangel på diverse informasjon	H	M	Søke på IEEE Xplore, ACM, Inspec, subterskel bøker
5	Ikke møte <i>ekstra</i> spesifikasjoner	H	L	Investere mere tid; Droppe ekstra spesifikasjoner; Handling(#4)
6	Eget preg på kretsen ikke realiserbart	H	L	Investere mere tid; Handling(#4); Droppe eget preg på kretsen
7	Skjematiikk stemmer ikke med layout	M	H	Konsultere med diverse professorer/studenter; Cadsoft manual
8	Uforutsigbare nye layout teknikker	L	M	Finne en vei rundt dem med minst mulig ytelse tap
9	CAD verktøy simulering ikke tilfredsstillende	M	H	Manuelle kalkulasjoner
10	Komplikasjoner med Matlab	L	M	Cadence grafer; Finne annet matematisk verktøy; Handling(#9)

Tabell 4) Risikoanalyse med sannsynlighets- og innvirknings ranking

Vektene i risikoanalysen i Tabell 4 tar for seg sannsynligheten for at et risikoelement inntreffer og innvirkningen risiko elementet har på prosjektet. For å gjøre designprosessen mer systematisert vurderte jeg meg frem til vektene, som fikk følgende verdier:

$$L = 0.15, M = 0.25 \text{ og } H = 0.35$$

For å illustrere hvordan vektene i risikoanalysen fungerer, kan vi se på et eksempel. Vi ser for oss situasjonen der vi bruker mer enn 3 timer på ID(3) i Tabell 3. Dette fører oss i feedback til ID(2) hvor antall timer vi skal invester i feedbacken er gitt av vektningen i risikoanalysen. Det kan tenkes at forsinkelsen skyldtes Risiko #2, som har vektene M og M. De timene vi skal legge til i denne deloppgaven kan da lett beregnes via en enkel formel:

$$\text{Tid som investres i ny ID} = (\text{opprinnelig}) \text{ BANK} * \text{Sanns.(\#)} * \text{Innvirk.(\#)}$$

som blir

$$\text{Tid som investres i ny ID} = 49 * 0,25 * 0,25 = \underline{3 \text{ timer.}}$$

For å hindre at vi setter oss fast i ID(3), så brytes feedbacken når vi har investert opp til 25% av opprinnelig timer i BANK. Hvis feedbacken ikke brytes der, så har vi fått avvik fra planen. Det mystiske tallet 25% kommer fra opprinnelige tidsbegrensningen i ID(3) dividert med tidsbegrensningene totalt for BANKEN som i dette tilfellet svarer til 11,76 timers total investering i ID(3).

Den nye modellen viste seg å være ganske effektiv, spesielt når risikoanalyse ble implementert. Dette førte til at prosjektet hadde god flyt siden det på forhånd hadde blitt reflektert over muligheter for å kontre potensielle problemer. Etter hver prosjektdag ble det loggført utført arbeid på prosjektet og samtidig lagt planer for videre arbeid. Dette ble samkjørt med oppdatering av prosjektplanen der også oppdragsgiver hadde en rolle. Det var ukentlige møter med oppdragsgiver som til tider førte til at nye kriterier ble satt for prosjektet på basis av både eksisterende problemer og potensielle problemer som etterhvert kunne forekomme. Den nye modellen demonstrerte god robusthet når det kom til endringer i planen, noe som tok relativt lite tid å implementere. Ut ifra egenskapene den nye modellen demonstrerer er det sannsynlig at den er anvendbar også på andre hardware prosjekter,

spesielt prosjekter som er preget av et eksperimentalt aspekt. Modellen ble i utgangspunktet utformet til å tilfredsstill prosjektets behov, noe som er vanlig i hardware industrien der bedrifter lager sine egne modeller som de tilpasser prosjektene sine.[10]

Prosjektledelse og metodikk i HW utvikling vs SW utvikling

Opp til i dag har det blitt skrevet lite om forskjellen mellom design metodikk og prosjektledelse innen software og hardware prosjekter. Tydeligvis har software industrien vært mye flinkere på å gi ut informasjon angående prosjektledelse og design metodikk enn hardware industrien. Etter en lang jakt på relevant stoff om hardware utvikling kom jeg i kontakt med noen hardware ingeniører fra Texas Instruments i Oslo. Her fikk jeg bekreftet at hardware industrien er varsomme med å gi ut informasjon angående prosjektledelse metoder og design metodikk de selv tar i bruk. Derimot var det en ting de ville understreke, og det er at design metodikken ofte tilpasses prosjektet.

Prosjektledelse er generelt sett relativt likt innen hardware og software utvikling.[10] Store software prosjekter er som kjent vanskelig å lede. Det er også hardware prosjekter, men det som gjør hardware prosjekter lettere å lede er nemlig observasjons muligheten til lederen.[1] I hardware prosjekter er det lettere for lederen å identifisere om de ansatte gjør det de skal, rett og slett ved å observere måten de ansatte jobber på og hva som blir produsert. På andre siden har vi software, hvor dette ikke alltid er tilfellet siden software produkter krever mer innsikt for å få kartlagt hva som egentlig har blitt gjort og om det tilfredsstill lederens krav. Konsekvensene av dette er at innen hardware, kan arbeid som ikke tilfredsstill lederens krav lett bli identifisert og på denne måten også unngå tap i tid og penger uten at det går lang tid. Ser man på software, kan den reduserte observasjons mulighet føre til at det går lang tid før feil blir observert eller at feilen utvikler seg på en så negativ måte at da den endelig identifiseres lar den seg ikke reparere av tids- eller økonomisk grunn.

En annen ting som skiller prosjektledelse innen hardware og software er kontakten med kunden. Kunden skal vite hva de vil ha og samtidig ha en viss forståelse på det. Dette er lettere med software utvikling enn i hardware utvikling. Grunnen ligger i at software er mer fleksibelt enn hardware og kundens tanker kan lettere anvendes på software.[8] Innen hardware industrien slites det ofte med at kunden har for høye krav. Dette er siden det relativt sett kreves mer innsikt av kunden når det kommer til hardware, noe som det ofte er mangel på. Dette er noen av forskjellene mellom prosjektledelse innen hardware og software, men mange av prosedyrene man finner i software utvikling kan kjennes igjen i hardware utvikling, måtte det være markedsundersøkelser, risikoanalyse eller aspekter som jevnlig kontakt med kunder osv.

Når det kommer til design metodikk, kan det være stor skille mellom software og hardware industrien. Agil metodikk som fungerer på software prosjekter, er ikke like anvendbar på alle typer hardware prosjekter. De seneste åra har det blitt veldig populært å bruke HDL (hardware description language). Dette er en type ”programmering språk” som benyttes i digital design som gir ingeniører muligheten til å gjøre programkode til digitale elektroniske kretser. I slike prosjekter kan agil metodikk fungere, men om det er anbefalt diskuteres fortsatt. Texas Instruments i Oslo har hatt ett prosjekt der de testet ut bruk av agil metodikk på et HDL prosjekt.[10] Dette viste seg til å ha både fordeler og ulemper ifølge en av ingeniørene. Når det gjelder hardware design på transistor nivå, så må det spesiell design metodikk til. Her kreves det mer av ingeniøren siden en liten feil forårsaker ofte store økonomiske og tidsmessige tap. Siden kravene er så høye, benyttes det ofte spesialutviklede metoder og verktøy for å sikre effektivitet og sikkerhet i utviklingsfasen. Dette er derimot avhengig av prosjektoppgaven og er ikke tilstrekkelig i alle tilfeller. Noen ganger er god

kompetanse hos de ansatte som er det eneste skillet mellom et vellykket og mislykket prosjekt.

Innen hardware design er det først og fremst spørsmål om det er mulig å *systematisere* design fasen i et prosjekt.[9] Dette kommer an på type design man holder på med som hovedsaklig dreier dette seg enten om digital eller analog design. I mitt tilfelle faller designet inn under analog/mixed-signal design, altså en hybrid av digital og analog design, men der analog design er dominerende. Da blir det spesielt vanskelig å ha en elegant og systematisert måte å jobbe på. I analog design er det mange uregelmessigheter og begrensninger, noe som gjør design fasen uforutsigbar. [7] Dermed faller denne type design vanligvis på en enkel person som forholder seg til ad hoc prosedyrer og samtidig greier å ha oversikten over det som blir gjort under design forløpet. Sammenlikner man denne type metodikk med dagens design metodikk i software, så må man gå mange tiår tilbake i tid for å være istand til å se noen likheter. Mange vil påstå at analog design henger mange tiår etter når det gjelder design metodikk, og må se seg slått av design metodikk brukt i digital design som har hatt en laber men merkbar utvikling etter at HDL språk ble etablert.

Design metodikk i software har hatt en jevn utvikling de siste årene. Kjente modeller som The Spiral Model og SCRUM er bare få av dem som har blitt brukt flittig. Som nevnt over har hardware industrien vært litt forsiktige når det kommer til å gi ut informasjon om design metodikken de tar i bruk. Derimot er det fortsatt kjent hvordan prosessen i hardware utvikling vanligvis går for seg. I hardware utvikling kan man ofte finne en iterativ tilnæringsmåte på design prosessen. Denne prosessen består av design og verifisering med feedback. For å illustrere hvorfor det kan være ugunstig i mixed-signal hardware å ta i bruk for eksempel The Spiral Model kan vi se på følgende. I The Spiral Model benyttes det av for eksempel flere prototyp faser.[2] Et punkt som direkte fører til at denne modellen kan være ugunstig for hardware utvikling er at produksjon av prototyp kretser er veldig dyrt og tar lang tid. Det tar ofte et par måneder før prototyp kretsen blir ferdigprodusert og i tilfelle man vil at kretsen skal bli ferdig et par uker før originale tidsskjemaet, så må man ofte ut med store summer. I motsetning har vi software der prototyp er betraktelig billigere å realisere, både med tanke på tid og økonomi, noe som gjør denne modellen appellerende. Hvis man bygger argumentene sine på dette eksempelet forbeholdt hardware design, ville denne modellen sannsynligvis føre til økt kostnad og ikke minst økt "time to market", noe som stadig blir viktigere i en mer og mer kompetitiv industri.

Konklusjon

Det er ofte slik at studenter ikke tar i bruk kjente metoder innen prosjektledelse og design metodikk når de setter igang med diverse prosjektarbeid. Dette skyldes antageligvis at de ikke har blitt introdusert til disse metodene, og dermed går glipp av mye informasjon når det kommer til effektivisering og organisering av prosjektarbeid. I denne rapporten ble det diskutert hvordan en prosjektplan kan organisere arbeidet rettet mot hardware design, eller mer konkret analog/mixed-signal design. Det ble gjort et forsøk på å anvende prinsipper fra SCRUM metodikken på prosjektarbeidet, noe som viste seg å være vanskelig i forhold til uforutsigbarheten analog/mixed-signal design bringer med seg. SCRUM tilnærmingen viste seg lite velegnet for arbeid med et lite hardware prosjekt innen analog/mixed-signal design i den forstand at interaksjon mellom plan og prosjekt ble uoversiktlig. Som et resultat av den mislykkede SCRUM anvendelsen ble en ny prosjektplan utarbeidet som er inspirert av waterfall modellen. Den nye modellen viste seg å gi mer frihet i designfasen samtidig som at den er generalisert med hensyn til hardware design. Nye modellen viste også gode egenskaper når det kom til eksperimentelle prosjekter ved at den gjorde at prosjektet fort kom igang og at det ble kastet bort minst mulig tid på områder som ikke førte til noe konstruktivt. Modellen benytter seg av tradisjonelle metoder innen prosjektledelse, men med ekstra

fokus på utvalget områder for å effektivisere arbeidet. Prosedyren fra prosjektplanleggingen som hadde størst innflytelse var en kvalitativ ranking risikoanalyse. Denne analysen bidro med å gjøre prosjektarbeidet mer forutsigbart og førte til at mange feller ble unngått, noe det igrunn var mye av i prosjektet. Risiko håndteringen ble effektivisert ytterligere ved å gjøre den mer dynamisk. Til slutt i rapporten indikeres det at prosjektledelse innen hardware ikke skiller seg for mye fra prosjektledelse innen software. Design metodikk derimot kan variere ganske mye avhengig av type design (analog/digital), metodikken skal anvendes på.

Prosjektet i INF5420 ble, alt tatt i betraktning, et vellykket prosjekt og har potensialet til å føre til en publikasjon. Mye tyder på at prosjektplanleggingen hjalp en god del med å holde kontrollen på prosjektet samtidig som å føre det unna potensielle feller.

Referanser

- [1] Watts S. Humphrey, "Why Big Software Project Fail: The 12 Key Questions" CrossTalk The journal of defence Engineering., 2005.
- [2] Barry W. Boehm, "A Spiral Model of Software Development and Enhancement" TRW Defence Systems Group., 1988.
- [3] James Taylor, "Managing Information Technology Projects", American Management Association, 2003.
- [4] Alisa McKnight, "Flexible Methodology Management", Design Management Enviroments in CAD, IEE Colloquium on, vol., no., pp.1/1-1/6, 31 Jan 1991.
- [5] Bendik Bygstad, "Two Faces of Project Management", Seminar, 2008.
- [6] Craig Larman, "Agile & Iterative Development: A manager's guide", Addison-Wesley Professional, 2004.
- [7] Louise Joselyn, "Art or Engineering?", New Electronics Magazine, 2007.
- [8] Ida Lau Borch, "Erfaringer fra et IT-prosjekt", Seminar, 2008.
- [9] M. Bozic, OI. L. Grundy, "A Novel Approach to VLSI Analogue Design", Analog Signal Processing (Digest NO:1996/236), IEE Third on-day Colloquium on, vol., no., pp.10/1-10/6, 20 Nov 1996.
- [10] Østby Knut Anders, Samtale over email, 04.12.08 (Vedlegg I)
- [11] Fischman Lee, "Improving Outcomes on Experimental Projects", Galorath Incorporated, presantasjon, [WWW] <http://pmchallenge.gsfc.nasa.gov/Docs/2006attendee-presentations/2006presentationsCD-attendee/Lee.Fischman.pdf>
Podcast intervju: [WWW] <http://studio-5.financialcontent.com/demo?GUID=5503855&Page=MEDIAVIEWER>

Vedlegg

Vedlegg I – Samtale over email med Knut Anders Østby, Prosjektleder i Texas Instruments

Se svar nedenfor.

-----Original Message-----

From: Amir Hasanbegovic [<mailto:amirh@ifi.uio.no>]

Sent: 4. desember 2008 21:26

To: Ostby, Knut Anders

Subject: RE: Design metodikk

> Hei igjen.

>

> Jeg setter stor pris på at du tar deg tid til et par spørsmål.

>

>

> 1) Mitt første spørsmål er om du har vært borti agil metodikk innen

> hardware utvikling?

> Det har blitt godt brukt innen software utvikling(SCRUM, XP), men jeg

> slet med å finne informasjon angående anvendelser på hardware

> utvikling. Spesielt med tanke på analog / mixed-signal design.

Ja, jeg har vært litt borti dette. Har ikke brukt det selv, men en annen prosjektleder (Trygg Eliassen, t.eliassen@ti.com) her hos oss har kjørt deler av et

prosjekt vha scrum metodikk. Jeg har heller ikke funnet noe informasjon om hvordan

scrum bør brukes for hardware utvikling. Jeg har flere ganger lett etter dette, men

ikke funnet noe jeg synes var spesielt nyttig.

Min bakgrunn er fra digital design, og innenfor det området tror jeg det er forholdsvis greit å bruke scrum teknikker direkte. Når det gjelder analog/mixed signal er jeg mer usikker.

>

> 2) Generelt sett, er det noen "kjente" design metodikk/modeller som

> blir brukt ofte innen hardware utvikling? Eller er det slik at

> bedrifter tilpasser egne modeller til de forskjellige

> prosjektoppgavene?

Jeg tror de fleste bedriftene lager seg sine egne modeller, men at de i stor grad er

basert på eller ligner på waterfall modellen. Vi har jo denne store viktige milepælen i et HW prosjekt, som er at man starter produksjonen av chippene.

Dette er

dyrt og tidkrevende, så det gjør at man må sørge for at det er liten sansynlighet

for å måtte gjøre HW endringer etter at man først har satt noe i produksjon.

> 3) Blir prosjekter ofte basert på ad hoc prinsippet? Eller er det

> slik at prosjekter alltid burde være styrt av prinsippelle eller

> generelle grunnlag.

Her skjønner jeg ikke helt hva du mener. Hvis du tenker på om prosjektgjennomføringen bør følge en metode eller en slags oppskrift, så er svaret

helt klar ja. I en bedrift gjør man gjerne mange prosjekter og det er en stor fordel om man kan gjenbruke metoder/dokumentasjon/estimater osv for flere prosjekter. Det er også lettere å få et prosjekt til å "gå av seg selv" om de som jobber på prosjektet har gjort flere lignende prosjekter før og dermed er godt trent i gjennomføringen.

>
> 4) Jeg vet ikke hvor mye erfaring du har med prosjektledelse innen software, men vil du si at prosjektledelse innen software og hardware er relativt likt? Eller skiller prosjektledelse innen hardware seg fra prosjektledelse innen software?

HW prosjekter har mye større terskel for å gjøre en release. Det lar seg ikke gjøre å komme ut med månedlige service packs eller lignende. HW prosjekter trenger også mer og dyrere verktøy i form av simulatorer, syntese, place and route, etc. Når det gjelder selve prosjektledelsen, så tror jeg ikke forskjellen er særlig stor, men jeg har ikke noe erfaring fra SW prosjekter.

Jeg har nylig tatt PMP (Project Management Professional) kurs i regi av www.pmi.org og her lærer man prosjektledelse som er nærmest helt uavhengig av hva slags type prosjekt som skal gjøres.

>
> 5) Har jeg din tillatelse til å bruke denne mailen som en eventuell referanse i rapporten min?

Ja.

>
>
> Igjen, jeg setter stor pris på alle svar. :)
>
>
> mvh
>
> Amir H.
>
>
>> Hei Amir,
>>
>> jeg forsøker gjerne og svare på noen spørsmål.
>>
>> mvh
>> Knut Anders
>>
>> --
>> Knut Anders Østby
>> Program manager
>> LPW TI Norway
>> phone: (+47) 22958484
>> mobile: (+47) 97032482
>> email: k.a.ostby@ti.com
>>
>>

>> ----Original Message-----
>> From: Amir Hasanbegovic [<mailto:amirh@ifi.uio.no>]
>> Sent: 4. desember 2008 14:07
>> To: Ostby, Knut Anders
>> Subject: Design metodikk
>>
>>> Hei Knut Anders
>>>
>>> Mitt navn er Amir Hasanbegovic, jeg studerer Mikroelektronikk ved
>>> UiO. Jeg var på bedriftsbesøk hos TI på Forskningsparken et par uker
>>> siden der jeg kom i kontakt med en av IC designerene der. Vi hadde
>>> en samtale om prosjektledelse og design metodikk innen hardware
>>> design og i den sammenheng ga han meg email adressen din. Jeg har
>>> akkurat avsluttet et prosjekt som gikk ut på å lage nivåkonverter
>>> som konverterte 180mV spenninger til 1.2V spenninger. Poenget var å
>>> samkjøre prosjektarbeidet med en prosjektledelse del. Etter min
>>> erfaring er det lite stoff å finne når det kommer til
>>> prosjektledelse og design metodikk spesielt rettet mot hardware.
>>>
>>> Da lurte jeg på om jeg kunne spørre deg et par korte spørsmål hvis
>>> du har tid til å svare?
>>>
>>>
>>> mvh
>>>
>>> Amir H.