



Emnekode : 15402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 1 av 9

oppgave 1

Prosjekttrekanten som vises representerer et produkts kvalitet i utviklings kontekst.

Dette vil si at trekanten selv representerer kvalitet og de tre hjørnene bestemmer hvorvidt kvaliteten er god eller dårlig.

For å få en balansert trekant må hvert av hjørnene vektas likt i forhold til hverandre.

Dersom et prosjekts omfang plutselig øker drastisk må man bestemme hvordan man ønsker å takle dette. Man kan da velge å gjøre endringer i kostnad eller tid for å gjøre opp for økningen i omfanget. Det er i stor grad dette systemutviklingsmetoder forsøker å gjøre, men på forskjellige måter i henhold til prosjektets spesifikasjoner.

Plandrevne metoder bruker omfattende dokumentasjon og planlegging for å utarbeide en spesifikasjon som viser prosjektets omfang på forhånd.

Dette innebærer at kunden må være helt inneforstått om krav og løsninger før utviklingen starter. Dette betyr i praksis at kunden ikke kan komme med endringsønsker etter at utviklingen har startet.

Boehm og Turner skriver for eksempel at i plandrevne sammenheng bør man i aller høyeste grad kun godta endringer



lik 1% av prosjektets omfang per måned. Dette vil si at omfang omtrent holdes helt statistisk i en plandreven sammenheng og det er derfor viktig at kunden er klar over dette. Plandrevne metoder vil derfor forsøke å holde kvaliteten god ved å sikre et stabilt og ukaotisk miljø hvor omfanget holdes så statistisk som mulig. På denne måten blir det enklere å planlegge tidsbruk og dermed også kostnader for prosjektet.

Agile metoder (smidige metoder) har en litt annen tilnærming. Agile metoder ønsker endringer velkommen selv i avsluttende faser av utvikling (Agile Manifesto). Dette vil derfor si at omfang er i stadig endring og de må derfor bruke andre verktøy for å sikre kvalitet. Det viktigste i agile metoder er kanskje estimering og planlegging. Boehm og Turner skriver at "agilister" estimerer at de bruker cirka 20% av tiden sin til planlegging. Dette er en måte å sikre at det viktigste gjøres først og at det blir gjort innenfor en gitt tidsramme. I agile metoder kommer det stadige endringsønsker og disse prioriteres og utvikles. Kunden kan på et hvilket som helst tidspunkt vurdere prosjektet som fullført dersom kravene blir oppfylt. Dette som følge av at alt utvikles modulært. Ved bruk av iterasjoner og



Emnekode : 15-402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 3 av 9

planlegging sammen med kundene vil det til enhver tid være klart hvordan prosjektet ligger an i henhold til tid og kostnad. Et annet viktig begrep som Boehm og Turner nevner innenfor agile metoder er "tacit kunnskap". Dette er teamets kunnskap som de deler i et gitt prosjekt. Språket som brukes til å beskrive elementer som kan være vanskelig å illustrere med ord er eksempel på tacit kunnskap. Ved å opparbeide et høyt nivå på denne typen kommunikasjon innad i teamet kan utviklerne løse mer kompliserte oppgaver raskere. Avslutningsvis vil jeg understreke viktigheten av de agile metodenes kontakt med kunden. Cockburn skriver mye om kommunikasjon og tacit kunnskap, men det er like viktig å ha kunden inneforstått med prosjektets retning som teamet. Nettopp derfor involverer agile metoder kunder i aktiviteter som Planning Poker for å gi uttrykk for tids- og omfangsrammene av moduler og elementer. Dermed kan kundene visuelt se hva slags kostnader ønskene deres vil resultere i.



Emnekode : 15-402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 4 av 9

Oppgave 2

Vi har tidligere sett på hvordan plan-drevne metoder og agile metoder håndterer kvalitet i form av tid/omfang/kostnad. Jeg velger å fortsette å sammenlikne disse to metodegruppene da vi nå skal se på hvordan disse har oppstått og hva som skiller dem. Jeg velger å se på det i mer grove trekk enn Fitzgerald gjør i sin bok da det vil være mer hensiktsmessig i denne sammenhengen.

Plandrevne metoder var noen av de første som oppstod. De tidlige versjonene av "systems development life cycle" var forgjengerne til dagens fossefall metoder. Vi kan også si at mange agile konsepter også stammer herfra.

Planlegging → Analyse → Design → Implementasjon
(Fitzgerald 2002)

Disse metodene kom fra ingeniørgrkenes prosesser for utvikling og oppstod i stor grad i vakuumet der det ikke fantes struktur for arbeidsmetoder innen IS-utvikling. Disse har siden utviklet seg via strukturering og data drevet utvikling. Strukturert utvikling var en videreutvikling av SDLC som kom når det ble klart at behovet for tilpassing til IS-utvikling var viktig (Fitzgerald).



Emnekode : 15-402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 5 av 9

Plandrevne metoder slik vi kjenner dem idag bygger i stor grad på et stabilt utviklingsmiljø uten et høyt nivå på endringsprosenten. Metodene bygger på solid dokumentasjon av metoden og passer i stor grad best i større prosjekter. Dette både med tanke på antall utviklere og produktets omfang (Cockburn) (Boehm & Turner). Man kan si at utviklerne i plandrevne prosjekter til enhver tid jobber etter en plan utviklet på forhånd og denne bør/skal ikke vikes fra. Store og komplekse prosjekter som USA's flyprogram (brukes som eksempler i en mengde litteratur) og romfartsprogrammet. Dette er gode eksempler hvor krav og planer ble utviklet på forhånd og firmene som ble vurdert som strukturente og modne nok ble satt til å gjøre jobbene. Boehm og Turner nevner ~~Capability Maturity Model~~ Maturity Model som en stor del av denne ~~vurderingen~~ vurderingen. Dette er et rammeverk som viser et firmas evner og modenhet samtidig som det viser at firmaet stadig jobber for å bedre sine prosesser.

Metodene er i stor grad bygd på konseptet om en fossefallsmodell som de første metodene som ~~er~~ nevnes tidligere. Disse er riktig nok videreutviklet og tilpasset IS-utvikling som jeg nå skal se nærmere på.



Emnekode : 15-402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 6 av 9

På slutten av 90-tallet og begynnelsen av 2000-tallet begynte en ny form for IS-utviklingsmetoder å oppstå. Den agile gruppen oppstod. Min mening er at ettersom statistiske resultater nå viste i hvilken grad plandrevet utvikling lyktes i IS-prosjekter var det mange som så behovet av en endring i hvordan IS utvikles.

Agile (eller smidige) metoder bygger på et miljø med mye større grad av endring og selvstendig arbeid. Dette i motsetning til plandreven utvikling hvor man til enhver tid har en oppgave utfra en plan. Innen agile metoder kan planen forandre seg ukjentlig eller kanskje til og med daglig. Boehm og Turner skriver at disiplin er svært viktig i et slikt miljø. Dette fordi at med en større grad av frihet må utviklerne være i stand til å se behovene, kommunisere effektivt (faglig kunnskap) og endre metoder for å tilpasse dem til nye situasjoner. Dette henger sammen med Cockburns klassifisering av utviklere hvor man i aller høyeste grad burde ha level 3 og 2-utviklere i agile prosjekter. Level 2 og 3 er begge i stand til å tilpasse metoder etter nye situasjoner, og mine erfaringer gjennom arbeidet med prosjekt oppgaven i faget har vist at et høyt antall slike utviklere i stor grad kan erstatte mye av dokumentasjonen som plandrevne metoder krever.



Emnekode : 15-402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 7 av 9

Sweetspots og home grounds nevnes av Cockburn og Boehm & Turner. Dette dreier seg i stor grad om hvorvidt metoden du bruker passer i ditt miljø og til ditt prosjekt. Begge gir deg en oversikt over hvor du befinner deg og hvordan du eventuelt kan justere for å passe bedre til metoden eller å få metoden til å passe deg bedre.

Cockburn lister opp fem sweet spots og bruker disse til å vurdere hvorvidt man kan bruke agile metoder. De fem er:

- 2-3 utviklere i samme rom
- Onsite bruker-eksperter
- En-måneders iterasjoner
- Fullt automatiserte regresjonstester
- Erfarne utviklere

Boehm og Turner gjør noe liknende ved å lage home grounds eller hjemmebaner for hver av ~~de~~ de to metode-gruppene. Dette sammentattes i flere tabeller som kan brukes til å vurdere metodens passform til prosjektet. Jeg vil komme tilbake til hvordan vi kan bruke disse elementene til å vurdere metoden man bruker senere.



Emnekode : 15-402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 8 av 9

Dersom vi gjør forutsetningen om at man enten driver plandrevet eller agil utvikling kan det oppstå to situasjoner med tanke på feil bruk av metode. Enten kjører man for agilt i en setting som ville ha vært bedre med plandrevet ellers gjør man motsatt.

I det første tilfellet kan man oppleve at man kjører en agil form for utvikling i et miljø med for mange utviklere, for lite kontakt med kunde, varierende eller ikke-eksisterende iterasjoner osv. Ved å legge opp til en agil struktur uten å følge opp rundt disse nøkkelementene kan føre til at man mister kontroll, dette ved at omfanget blir for stort. En annen problemstilling kan være et overtall av undererfarne utviklere. Som nevnt tidligere bør agile prosjekter ha et overtall av Cockburns level 2 eller 3-personer. Boehm og Turner forteller at dersom dette ikke følges opp blir de level 2 og 3-utviklerne som jobber i prosjektet helt opptatt av å forsøke å lære opp utviklere på de lavere nivåene og derfor ikke i stand til å utføre egne oppgaver.

Det andre tilfellet jeg nevner ovenfor handler om at man bruker en plandrevet metode i et miljø som passer bedre i agil setting. Resultatene av dette kan være at mange høyt kvalifiserte utviklere føler at jobbing med prosessen tar mer tid enn faktisk



Emnekode : 15-402
Kandidatnr. : 7669
Dato : 6.12.2011
Ark nr. : 9 av 9

Utvikling og derfor føler seg lammet av dette. Dersom endringsprosenten er høy og man driver plandrevet vil dette mest sannsynlig føre til at prosjektets tidsaspekt blir for langt og går over tiden. Bermød vil prosjektet kanskje avsluttes dersom kunden ikke er fornøyd med resultatet.

Man kan bruke Cockburns sweet spots sammen med Boehm og Turners home ground tabeller for å vurdere hvor man ligger i forhold til bruk av metode. Man kan bruke Boehm og Turners stjernemodell til å se sammenhengen og visuelt fremstille miljøet sitt for å deretter gjøre endringer i metoden.

Det må sies at verken plandrevet eller agile metoder er noen "silver bullet" for IS-utvikling og at ingen av dem kan brukes som de er. Alle må skreddersys etter organisasjonens behov og verktøyene jeg skisserer ovenfor kan brukes til dette.