



Emnekode : 15-402
Kandidatnr. : 5625
Dato : 8/12
Ark nr. : 1 av 9

Oppgave 1

Planleggende metoder er metoder som er strukturerte. Motivasjonen til å utvikle mer formelle og planleggende metoder kom fra en rekke problemer man hadde med systemutvikling etterhvert som kompleksiteten til systemene økte.

Tidligere hadde systemutvikling vært en mer tilfeldig og uformell prosess, der de aller fleste systemer ble utviklet til egen bruk av utviklere som var selvopplært. Programmeringsspråk og ~~utviklings~~ omgivelser var også primitive, i tillegg til at teknologien var begrenset. Det fantes heller ingen standarder som utviklerne kunne støtte seg til, slik at all kode ble veldig forskjellig, og derfor vanskelig å tyde for utenforstående.

Etterhvert når man ~~skulle~~ begynte med å utvikle ~~kunder~~ ~~ble~~ ~~disse~~ ~~problemerkene~~ programvare til kunder, ble disse problemene enda større. Utviklere var vant til å jobbe alene, ~~og~~ ~~ikke~~ og var derfor ukjent med kommunikasjon, ~~for~~ ~~å~~ ~~blå~~ men deling av kode. Etterhvert ble dette et så stort problem at uttrykket "software crisis" oppstod for å beskrive den situasjonen man var i (1960).

Som et svar på dette ble man mer og mer åpen for analyse og design, og sette dette i mer kontrollerte former. Man så på individualitet og kryptisk programmering som et problem.



Oppgave 9 forts.

Med andre ord ville man løse disse problemene med individualisme, "dårlig" kode, økende kompleksitet og lite struktur. Derfor kom man frem til en systemutviklings utviklings livssyklus (SDLC), som delte opp prosessen i forskjellige faser, ~~der~~ der en fase må avsluttes, før man starter på den neste. Dette ~~gjorde~~ delte opp prosjektene i mindre og mer håndterlige biter for å takle økende kompleksitet.

"Problemet" med individualiserte løserplandrevne metoder med å ~~være~~ være veldig beskrivende for hva ~~for~~ enhver person skal gjøre. Holdningen er at det ikke skal bli så mye å si for hvem som ~~gjør~~ jobber med prosjektet ~~for~~ så lenge de følger metoden til punkt og prikke.

Plandrevne metoder forsøker også å løse problemet med å kommunisere ved å ~~være~~ ~~veldig~~ ~~krav~~ være veldig mye dokumentasjon, slik at man kan bytte ut folk underveis.

Agile metoder begynte å ta form på 90-tallet som en reaksjon på de tunge og dokumentorienterte plandrevne metodene. Et av problemene de skulle løse var at nesten ingen fulgte de ~~de~~ formelle metodene 100%, som kan bli en sammenheng med at de ikke er utviklet av praktiskere.

Agile metoder ville også adressere problemet med ustabile krav som endrer seg utover prosjektet. De plandrevne metoder fastsetter omfanget i starten av prosessen endres dette konstant ved bruk av agile metoder.



Oppgave 7 fortsetter

Hensikten med agile metoder er oppå å kutte ned på ~~cost~~ på ressursbruk. Dette gjøres ved holdningen "You aren't gonna need it" (YAGNI), slik at man kun gjør ting (som dokumentering, funksjoner) som man vet man trenger i dag, da fremtiden er så usikker.

Det at agile metoder opprinnelig ble kalt lettvektsmetoder sier noe om hva tankegangen bak de er, man ville bli mer agil, altså smidig og "på luggen" da man så at nesten alle ISD-prosjekter feilet.

Planbare metoder bygger på antagelsene:

- Standardisering av utviklingsprosessen for å takle kompleksiteten
- ISD er en disiplin som kan sammenlignes med ingeniørarbeid
- ISD er i hovedsak en teknisk prosess
- Det er lettere med prosjektstyring og kontroll med rigide og detaljerte planer,
- Kunnskapsbevaring skjer gjennom omfattende dokumentasjon



Oppgave 1 (fortsetter)

Agile metoder bygger på disse antagelsene =

- Prosjektets kontekst kan forandres kontant, derfor må man være smidig i forhold til endringer
- ~~Det viktigste er selve produktet og ~~etter~~ om det er ~~klart~~ for ~~brukeren~~~~
- Det viktigste er nytteverdien produktet har for kunden/brukeren
- Individuer og interaksjon/kommunikasjon er viktigere enn prosesser og verktøy (Agile manifesto - AM)
- Programvare som virker er viktigere enn dokumentasjon (AM)
- Samarbeid med kunden ~~er~~ for isteden for kontraktforhandlinger. (f.eks en on-site representant for kunden)
- Det er viktigere å reagere på endringer fremfor å følge en plan (AM)
- Korte iterasjoner, tidlig levering av programvare med verdi er viktig
- ISD er et håndtverk, det er viktigere hvem som arbeider fremfor hvilken metode de følger
- Kommunikasjon fungerer best ansikt til ansikt, og dertiligere og mer avstand det blir mellom utviklerne.
- Samarbeid mellom forretningsfolk og utviklere gjennom hele prosjektet
- Enkelhet, gjøre så lite som mulig, men akkurat nok.
- ISD er et samarbeidsspill om kommunikasjon og oppfinnelser (Cockburn)
- Kunnskapsberøring skjer hovedsaklig med tacit knowledge



Emnekode : 15-402
Kandidatnr. : 5625
Dato : 8/12
Ark nr. : 5 av 9

Oppgave 2

Uansett hvilken erfaring og kompetanse en utvikler har vil det ha kausekvenser for metode-bruken. I følge Fitzgerald et al. vil en ~~erfaren~~ utvikler følge en metode ~~mye~~ ~~noye~~ veldig noye. Dette vil føre til at metode-i-bruk blir tilbrøntet til den formelle metoden. Dette vil være en som er på nivået "following" av Cockburns tre nivåer for å mestre ferdigheter. Man har altså så lite erfaring at man bruker metoden som en slags krykke for å støtte seg til

Etter hvert som erfaringen og kompetansen øker og man kommer opp på nivå 2, detaching, vil man kunne se svakheter ved den formaliserte metoden man bruker, slik at man kan "slippe krykken" og heller stole mer på sin egen kompetanse og erfaring.

Det siste av Cockburns nivåer, fluently, er når utvikleren ~~er~~ har opparbeidet seg en så stor kompetanse og erfaring at han "ser det store bildet" slik ~~at~~ at han kan begynne å bruke metoder som han selv skreddersyr, basert på konteksten. Det er ganske interessant at metode-bruken går opp når man kommer til det øverste nivået, ~~men~~ men bruken er helt ~~annetledes~~ annertledes enn noen som er på nivå 1.

Hiditt har jeg brukt utvikleren som et ganske ~~litt~~ begrep som dekker alle som er involvert i prosjektet. Men når det kommer til ~~brukt~~ metode-i-bruk man ~~ikke~~ opp med



Emnekode : 15-402
Kandidatnr. : 5625
Dato : 8/12
Ark nr. : 6 av 9

Oppgave 2 forts.

~~er det hovedsakelig prosjektledere som angir~~
Fitzgerald et al. mener at utviklerens rolle ikke kan overvundes (noe plandrevne metoder ikke tar så mye hensyn til) er det viktig å se på de menneskelige faktorene som avgjør hvordan utviklerne arbeider/opptrer.

Cockburn deler ~~og~~ synet til Fitzgerald et al. om at menneskelige faktorer er avgjørende for ISD. Han beskriver mennesker som "funcky people" som innebærer at de er:

- Spontane
- selvmotsigende
- avhengig av god kjemi
- liker variasjon innen team
- ingen er like / kan ikke generalisere for mye
- liker å jobbe inkrementelt og iterativt
- liker å finne opplysninger framfor å bruke eksisterende
- vi jobber best med konkrete eksempler, noe håndgripelige ved å endre på noe eksisterende, ved å se og lytte, når vi for konsentrere oss og kommunisere når vi trenger det og vi liker konkrete tilbake meldinger.

Cockburn mener også at vi er gode til å se oss omkring, tilpassningsdyktige, ~~bidrar og~~ bidra og ta initiativ og finne feil.

Når vi ser på alle egenskapene Cockburn mener at vi som mennesker har er det lett å trekke den slutningen at de som jobber med prosjekt vil påvirke hvordan metode-i-bruk er. Spesielt når man tenker på mennesker er minst, om ikke mer viktig (i alle fall etter ~~er~~ et agilt tankesett),



Emnekode : 1S-402
Kandidatnr. : 5625
Dato : 8/12
Ark nr. : 7 av 9

Oppgave 2 forts.

, enn den tekniske biten i ISD.

Når man skal velge en formell metode (so eller velge å ikke bruke en formell metode), vil de erfaringer prosjektlederen har også spille en stor rolle. Kanskje har han brukt en metode som han har hatt suksess med tidligere slik at han velger å ~~gjøre det samme~~ gjøre det likt på det neste prosjektet (eller motsatt). Kanskje har han så mye erfaring at han klarer å skreddersy metoden perfekt i forhold til konteksten. Den utdanningen han har ~~vil~~ vil også påvirke ~~hø~~ fremgangsmåten under prosjektet, eller om han har en agil eller strukturert tilnærming til ISD.

For å oppsumere så vil all bagasjen til utviklere (alle som deltar) påvirke måten man faktisk ~~opp~~ utfører arbeidet i ISD på, men et viktig faktum her er at agile metoder tar hensyn ~~til~~ til dette og gjenkjenner dette som en sannhet, mens planrevne metoder prøver å fjerne graden menneske påvirker prosessen.



Oppgave 3

Hvis man ser på Cockburns definisjon av Systemutvikling: "Goal directed game of communication and invention", er det tydelig at kommunikasjon tildeles en stor rolle i ISD.

Grunnen til at kommunikasjon er så viktig er at ISD er veldig kompleks, og uten god nok kommunikasjon vil man ikke klare å gjennomføre prosjektet på en god måte. Kommunikasjon er veldig viktig uansett hvor mange som jobber sammen, eller hvor langt man er fra hverandre, bare at vanskeligheten med å kommunisere godt nok øker.

Effekten av kommunikasjonen er viktigere enn hvordan man kommuniserer, og vi ser at agile og planbare metoder har to ulike tilnærminger til problemet. Agile metoder baserer seg på "tacit knowledge" gjennom tett samarbeid, utermet kommunikasjon og lite dokumentasjon. Planbare metoder gjør det motsatt og baserer seg på mye dokumentasjon.

I følge Bohem og Turner er dårlig kommunikasjon grunnen til at de fleste prosjekter feiler. Grunnen til dette kan være at mangel på kommunikasjon fører til misforståelser, man drar ikke lasset i samme retning eller at man bruker for mye tid kommunikasjonen i forhold til hva man får igjen.



Emnekode : 15-402
Kandidatnr. : 5625
Dato : 8/12
Ark nr. : 9 av 9

Oppgave 3 fortsetter.

Cockburn sine ERG-seconds, som er mått for både energi og tid er en god pekepinn på at man ~~ikke bør~~ bør kommunisere så effektivt som mulig. Jo mer tid og ressurser man bruker i et prosjekt på andre ting enn selve produktet, jo større er sjansen for å ha en bakpåk, kostnadsmessig eller i forhold til tidsbruk.

Cockburn sier også at hurtigheten/effektiviteten til de som jobber på et prosjekt ikke er rasere enn hastigheten på kommunikasjonen. Men det er viktig at man ikke ofrer kvalitet, for hastighet, men og følge prinsippet om at kommunikasjonen skal være god nok.

I lys av at MSD er et spill basert på samarbeid er det vanskelig å tenke seg at kommunikasjon ikke er kritisk for et projekts suksess. Det vil være vanskelig å ~~samarbeide~~ samarbeide uten å kommunisere på noe vis.

For å oppsumere kommunikasjonen så vil jeg konkludere med at kommunikasjonen i MSD er viktig for det er umulig for mennesker som skal jobbe sammen for å levere ett produkt å ikke kommunisere seg i mellom. Man kan si at kommunikasjon er linet som holder menneskene i prosjektet sammen, og styrer de mot et felles mål. Utan tilstrekkelig kommunikasjon vil man ende opp med at utviklere blir i forskjellige nettinger og ikke klarer å levere et godt nok produkt til rett tid og til rett pris.