



Oppgave 1

I dette prosjektet finner oppdragsgiver etter 3 måneder ut at fremdriften er for dårlig og utvider derfor prosjektgruppen fra 4 til 7-8 utviklere.

Kravene til systemet er kun beskrevet med brukerhistorier.

Design/brukerinterface snitt og arkitektur er ikke beskrevet nærmere i systemkrav.

Utvidelsen av prosjektet medfører at gruppen må splittes på to rom.

Konsekvensene som dette får er først og fremst at kommunikasjon på "tvers av rommene" blir vanskeligere.

Dette er et potensielt stort problem fordi Scrum, som er den smidige eller Agile systemutviklingsmetode er basert på en høy grad av "tacit" - eksplisitt - kunnskap hos utviklerne.

Samarbeid om systemutvikling krever derfor en stor grad av kommunikasjon mellom deltakerne i et prosjekt for å løse utfordringer som oppstår.

Det er flere måter man kan søke å løse kommunikasjons-



Emnekode : 1S-402
Kandidatnr. : 3308
Dato : 26-11-2012
Ark nr. : 2 av 12

problemet som er oppstøtt pga delingen på to rom.
Et sett av løsninger går på å innføre et element for planlegging, som dermed angir en rolledeling mellom rommere, og reduserer behovet for kontinuerlig kommunikasjon på tvers.
Det er mange måter man kan dele oppgaver på. Man kan inndele brukerhistoriere etter funksjonelle og ikke-funksjonelle krav, slik at en del griper jobber med de funksjonelle kravene og en annen med de ikke-funksjonelle (Jeg forutsetter at også de ikke-funksjonelle kravene er uttrykt som bruker-historier - f.eks "som produkt eier ønsker jeg at systemet skal programmeres med Java..."). En litt annen måte er å dele oppgaver etter lag i et lag delt system, der ett delteam jobber med systemstøtte (OS, database etc.) og kjerne forretningslogikk, og det andre delteamet jobber med brukergrensesnitt og brukerautorisasjon. Nå er det ikke oppgitt hva slags arkitektur vil bli akkurat om (MVC, "lagert"



Emnekode : 15-402
Kandidatnr. : 3308
Dato : 26-11-2012
Ark nr. : 3 av 12

- lagedelt - som drøftet her, repositorij, client server-, eller noe annet.

Derfor bør antagelig prosjektet styrkes ved at det lages et minste mål av arkitektur, dvs at Scrum-metoden suppleres med FDD - Feature Driven Development - hvor man jobber etter en felles grupplan for strukturen for hele løsningen og så deler prosjektet inn etter "features" - hovedfunksjoner - som så prioriteres ikket en forretningsanalyse - hvor feature blir deretter gjenstand for systemutvikling etter felles Scrum. FDD skiller seg fra XP - Extreme programming metoden, ved at hele teamet (alle de 7-8 utviklerne) ikke behøver å ha eierskap til hele koden/systemet, men at rollene kan deles etter features.

Det andre hovedsettet av løsning av det oppståtte kommunikasjonsproblemet er å sørge for formalisert kommunikasjon på tross av rommere. En representant fra hvert rom møtes daglig til et scrummøte (daily scrum) der man



Emnekode : 1S-402
Kandidatnr. : 3308
Dato : 26-11-2012
Ark nr. : 4 av 12

drøfter og prioriterer bruker-
historier og tilhørende tasks
etter avhengighet mellom
delteamene og løser utfor-
dringer av felles interesse.
Kanskje bør det oppnevnes en
brukerrepresentant til det
nye rommet også

En fordel ved å innføre en viss
grad av planlegging i dette
scrum-prosjektet vil kanskje
også være forbedret
"expectation management" -
bedre avstemning mellom
forventning til systemkav-
enes oppfyllelse og allokering
av tid og ressurser til ut-
viklerne.



Oppgave 2

Plan drevene utviklingsmetoder er de eldste system utviklingsmetodene og bygger på prinsipper fra engineering fra industri (fly, biler, romfart etc) der omfattende planlegging ble sett på som en forutsetning for å kunne dele opp et stort utviklingsprosjekt i komponenter og kommunisere dette til forskjellige team. Plan drevene system utviklingsmetoder krever en stor grad av styring og organisatorisk innsatsstruktur.

Utover på 60-tallet så man at mange system utviklingsprosjekter fikk problemer. En av årsakene kan være at systemkravene i mange braker organisasjoner hadde en tendens til å endre seg raskt over tid. Plan drevene metoder hadde problemer med å møte disse raskt endrede kravene. Agile metoder utviklet seg etter hvert som en utveksel fra "rapid prototyping" / "rapid development" så vel som fra filosofi som mente at system utvikling er et håndverk - nærmest en kunst -



snarere enn en industriell prosess. En gruppe forskere (Beck, Cockburn) med flere formålte det Agile manifestet

Individer og samarbeid
over prosesser og verktøy
Programvare som virker
over økt tilgjengelige dokumentasjon
Kunde samarbeid
over kontraktforhandlinger
Reaktion på forandring
over det å følge en plan

Dette høres ut som en krigs-erklæring og ble også på mange måter oppfattet slik. Etter hvert har det vist seg at de ulike metodene hver har sine styrker og svakheter på sine respektive anvendelsesområder. Agile metodene har sine "homegrounds" eller "sweetspots" mens Plan dreve metodet har sine. Boehm og Turner (2012) har tatt til orde for å kombinere plan dreve og Agile metoder i ulike "blandingsforhold" i ulike stadier av et prosjekt basert på en risikostyring.



Emnekode : 15-402
Kandidatnr. : 3308
Dato : 26-11-2012
Ark nr. : 7 av 12

Plan drevede metoder

Planchevise metoder kan brukes til å planlegge og gjennomføre lange iterasjoner og flere systemkrav

↳ system og programvare design

↳ implementasjon og enhets testing

↳ system testing

↳ implementering

(Vannfallsmetoden)

Planlegging gir også en mulighet til å identifisere kompleksitet og risikofaktorer. Man kan dele systemet totalt, inn i mer og mindre verdi- eller sikkerhetskritiske komponenter. For de mer kritiske komponentene vil det være behov for tett kommunikasjon med systemeier/oppdragsgiver for å gjennomgå fremdriften og eventuelt akseptansetester.

Resultater blir analysert og prioritering av funksjoner for neste iterasjon planlegges.



Agile metoder

Agile metoder løser best oppgaven der kravene uttrykkes for ulike endringer, eller der kravene først fremkommer etterhvert som en løsning tas i bruk.

Agile metoder fungerer best der hvor behovet kan løses gjennom en serie med korte iterasjoner (1 sprint kan typisk være 2 eller 4 uker). Agile metoder styrkes av ulike resultater. De to metodene kan kombineres. Begge metodene er utsatt for risiko knyttet til omme- og miljøbetingelser:

- stor teknologisk usikkerhet
- mange interessenter
- stor kompleksitet i strukturen (et system av systemer)

En metodikk for å pilotere/prototype bygge de mest kritiske komponentene, identifisere og prioritere interessenter (stakeholders) ~ spørre for å sikre er godt representert, samt tidlig forpliktelse til et design av grensesnitt kan bidra til å redusere risikoen. Denne metodikken heber elementer fra både plan- og agile metoder.



Oppgave 3

Utvikling av et nytt system for selobehandlingskunder

La oss anta at det er snakk om en engrosbedrift med et lag for bedriftskunder, der varene kan merkes automatisk, og der bedriften ønsker at kundene selv foretar utsjekk (varene er RFID-radobrikke merket - slik at systemet blir selvkontrollerende)

Bedriften engasjerer et team for å lage systemet. I tillegg til de 5 utviklerne bør det være en produkteier fra engrosbedriften og minst 1 (helst 2) representant(er) fra kundebedrifter som skal bruke systemet. De (2) brukers representanter bør være, eller gjerne godt til arbeidssammensatte til sjåførene som bruker varen. Produkteier bør i tillegg med utviklerne være kjøpt med Scrum, som er valgt som utviklingsmetode



Emnekode : 15-402
Kandidatnr. : 3308
Dato : 26-11-2012
Ark nr. : 10 av 12

De bør kunne bryte systemkravene ned til brukerhistorier, der bruker felles for oversikt over varer på lager - kan søke og finne varer - se pris - velge vare(r) og antall - kvittere for vare (når den er lagt på lastebilplanet) etc.

Som produkteier vil antagelig integrasjon med andre systemer slik som ERP, CRM et likn. være viktig, slik at produkteier må ha god kjemiskap til disse. Den ideelle produkteier vil derfor sannsynligvis være bedriftens IT-sjef, økonomisjef eller markedsjef.

Utvikler-tenner

Coaching skilles utviklere etter en skala der level 1 betyr

1a ~ gjennomsnittlig eller undererfarne utviklere. De fungerer godt i stabile situasjoner, men kan få problemer ved store endringer

1b ~ mer kreative enn 1a ~

kan lettere tilpasse seg forandring

2 ~ kan lede små prosjekter

Disse kan med erfaring bli level

3 ~ som kan lede store, nye prosjekter.



Emnekode : 15-402
Kandidatnr. : 3308
Dato : 26-11-2012
Ark nr. : 11 av 12

I et prosjekt som dette som er i overkant stort for et rent spile-prosjekt, trenger vi en klar overvekt av level 2 eller 3 utviklere, og de siste utviklerne bør være minst 1A. Hvis vi forutsetter 5 utviklere og nummererer dem fra 1 til 5 så kan vi tildele dem ulike roller som de bør bemerke sammen med produkt-eier og bruker-representanter:

Produkt-eier - tar beslutninger
- prioriterer brukerhistorier
- bruker-representanter
- hjelper å utvikle brukerhistorier. Tester.
1. Scrim master - kjenner scrum-metodikk - skiller ut tekniske krav
2. Tester - bistår bruker repr. og programmerer med å utføre og gj. fare test.
3. Oppfølger - sjekker om estimater stemmer og tar logg for utvikl.
4. } utviklere som velger,
5. } estimerer, og fullfører oppgaver (tasks). Disse skal også utføre utvikle og utføre erke-tester. Disse kan også være organisert som par programmerer (sammen m. 2. og 3.) og gjøre refactoring (omstrukturering av



Emnekode : 15-402
Kandidatnr. : 3308
Dato : 26-11-2012
Ark nr. : 12 av 12

programmet.

1. (Skrinnmaster) bør ideelt være level 3, minst 2

2. og 3. bør være level 2

4 og 5. bør være minst level 1A, helst 2.

Det vil i det coset som her er beskrevet være nødvendig (samsynligvis) med kontakt med - og kanskje innleie av - en representant for ERP eller CRM system-leverandøren som kan bistå utviklene med integrasjon. Hovedkontakter vil være produsentens og skrinmesteren, men det er ikke utelukkende at det vil være nødvendig å samlokalisere 3P system-representanten med vårt utvikles team i perioder.

Dette inkluderer noe av hvor enkelt og velstiltent 3 part-systemets programmerbare grensesnitt (API'er) er.