Rapport

*Fellesprosjekt 2015: Kalender*

Dette dokumentet er et resultat av arbeidet vårt i faget TDT4140 dette semesteret, og fellesprosjektet mellom fagene TDT4140 og TDT4145 som har gått over de siste fire ukene. Her finnes dokumentasjon til systemet vi har laget, og mye rundt selve prosessen

Buadu, Lange, Lundeby, Opheim, Tufteland

20.03.2015

Rapport

Fellesprosjekt 2015: Kalender

Innhold

[Beskrivelse av systemet og problemet 2](#_Toc412802297)

[Beskrivelse av utviklingsmetodikk 2](#_Toc412802298)

[Krav 2](#_Toc412802299)

[Use cases 2](#_Toc412802300)

[Aktivitetsdiagram 2](#_Toc412802301)

[Klassediagram 2](#_Toc412802302)

[Testing 2](#_Toc412802303)

[Riskikoanalyse 2](#_Toc412802304)

[Change management 2](#_Toc412802305)

[TIdsestimat og tid brukt 3](#_Toc412802306)

[Gjenstående arbeid 3](#_Toc412802307)

[Refleksjon om prosjektet 3](#_Toc412802308)

[Kommentarer til gjennomføring av faget 3](#_Toc412802309)

# Beskrivelse av systemet og problemet

# Beskrivelse av utviklingsmetodikk

Scrum er en metode som beskriver et prosjektledelserammeverk. Den bygger på at man har en serie med «sprints» som er fiksert på perioder der en økning i systemet er utviklet. Planlegging er gjort ved at man sjekker loggen og velger de høyest prioriterte gjenstående oppgaver til en sprint.

Scrum tas i bruk når man har en smidig utviklingsprosess. Og deles som sagt opp i sprinter. Lengden på en sprint vil variere fra prosjekt til prosjekt ut i fra størrelsen på prosjektet og organiseringen av teamet, men standarden er to uker lange sprinter.

Scrum er velutviklet og er i stor grad den utviklingsprosessen som brukes ute i arbeidslivet. Det at scrum er mye brukt tyder jo på at det funker og at det kanskje er fordelaktig framfor ante metoder. Det finnes flere fordeler og ulemper ved scrum. Vi har listet opp noen av dem.

* I smidige metoder som Scrum er det lettere å tilpasse seg endrede krav fra kunden. Et prinsipp for smidige metoder er å forvente forandringer i krav. I skiftende businessmiljøer skjer endringer raskt, og det er viktig å være i stand til å takle endrede krav.
* I Scrum leveres programvaren i flere inkrementer. Dermed får kunden mulighet til å prøve ut programvaren, og gi tilbakemelding på hva som fungerer og hva som ikke gjør det.
* Scrum er generelt mer kostnadseffektivt enn f.eks. vannfallsmodellen, grunnet fokuset på å redusere "overhead" i prosessen. Eksempelvis er det lite fokus på lage omfattende dokumentasjon.
* I Scrum er utviklingsprosessen i stor grad gjennomsiktig - alle teammedlemmene vet gjennom hele prosessen hva som skjer. Samtidig er hele teamet aktivt med i beslutningsprosesser. På denne måten føler utviklere et større ansvar for programvare som lages.
* Samtidig krever Scrum mer av hvert enkelt teammedlem. Det har blitt hevdet at utviklere må ha høyere faglig nivå i smidige metoder enn i plandreven utvikling. De må også ha en personlighet som passer til den intensive arbeidsmetoden, og være villige til å involvere seg i prosjektet. De må også ha anledning til å møte opp til de fastsatte møtene.
* Scrum krever samtidig at kunden har en representant som er tilgjengelig gjennom hele prosessen. I praksis er ikke dette alltid like enkelt å få til.
* Det mindre fokuset på dokumentasjon i Scrum kan gjøre vedlikeholdsarbeid vanskeligere.

Ulike teammedlemmer har ulike roller i scrum-teamet. Her har vi listet opp tre av de g hva deres oppgave er.

* Product owner: Skal være stemmen til kunden, og hele tiden passe på at utviklingen tilfredstiller kundens krav best mulig. Bør ikke kombineres med “Scrum master”
* Development team: Dette er utviklerene som gjør den faktiske jobben.
* Scrum master: Skal hindre at utviklingsgruppen blir distrahert fra andre ting, arrangere daglige møter, opprettholde Scrum-reglene.

Vi er 5 stk i vårt team. Daniel, Joanna, Stephanie, Synne og Tormod. Vi har selv listet opp våre styrker og svakheter og slik ble listen seende ut.

|  |  |  |
| --- | --- | --- |
| Navn | Styrker | Svakheter |
| Synne | Jobber effektiv, flink til å holde oversikt og samarbeide og formidle | Utålmodig |
| Stephanie | Jobber effektivt | Liker å ha kontroll |
| Joanna | Vant med å jobbe i grupper, liker å få ting gjort | Utålmodig, vant til å ha all kontroll selv |
| Daniel | Systematisk  Utholdende | Detaljert |
| Tormod | Effektiv | Utsetter ting |

# Krav

# Use cases

# Aktivitetsdiagram

# Klassediagram

# Testing

# Riskikoanalyse

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Sannsynlighet | Konsekvens | Løsning |
| Ett medlem må pga. sykdom eller lignende være borte fra prosjektet i store deler av tiden | Liten til middels | Noe av arbeidet kan bli utsatt. Flere av de andre teammedlemmene kan komme til å måtte arbeide mer for å klare å bli ferdig. I verste fall har personen som blir borte kompetanse som de andre ikke har, og det kan bli umulig å gjøre oppgaven før | Scrum master fordeler arbeidet til noen andre dersom det er mulig. Evt. må vi få hjelp fra eksterne personer til å løse oppgaven. |
| Programmet, og backup av dette blir slettet, eller korrupt. | Liten | Vi taper mye tid, og må kanskje droppe noen av funksjonalitetene for å klare å levere innen fristen. Hvis dette skulle skje vil det gå litt fortere å gjøre det på nytt, siden vi da vet hvordan vi skal gjøre det. | Man starter på nytt, eller fra siste versjon. |
| Det viser seg at vi ikke kan løse oppgaven på den fremgangsmøten vi har valgt. | Liten til middels | Endel av arbeidet vi har gjort må forkastes, og vi taper endel tid. | Man skjekker hva som mangler for at oppgaven kan løses, og legger til disse, evt. bytter det ut med det som viser seg å være unødvendige deler |
| Databaseoppsettet vi laget initielt viser seg å være for lite kompleks for å løse oppgaven, eller unødvendig kompleks for å løse oppgaven. | Middels | Dette vil gå ut over de funksjonelle kravene, og det vil ta endel tid å endre på dette. | Lage en ny database med de nødvendige relasjonstabellene, og oppdatere koden til denne. |
| Vi underestimerer størrelse og kompleksitet på programvare, og hvor lang tid vi bruker på det | Middels | Vi bruker mer tid enn vi antok på enkelte oppgaver, som igjen vil gå utover andre oppgaver som er avhengige. Dette kan i verste fall gjøre at vi ikke blir ferdige. | Vi må kanskje planlegge å legge ned ekstra tid i arbeidet, og vi bør se på tidsplanen for når oppgavene skal bli ferdige og mest sannsynlig gjøre endringer. Kanskje må vi droppe mindre viktig funksjonalitet. |
| Ikke god nok opplæring/kompetanse i programmer/ språk/ verktøy vi bruker | Middels til høy | Arbeidet vil ta mer tid fordi vi vil gjøre endel feil. Disse feilene kan være vanskelige å oppdage og rette opp. | Vi spør noen som kan det bedre enn oss om hjelp, og justerer litt på forventningene våre. Kanskje vi også må kutte ut mindre viktig funksjonalitet for å klare å bli ferdige med de viktigste. |
| Folk yter ikke så mye som forventet | Middels | Vi vil merke at noen oppgaver ikke blir ferdige i tide, og det vil utsette andre oppgaver. Dette vil føre til ekstra belastning på de andre teammedlemmene. | Scrum master setter seg ned og snakker med aktulle person. Hvis det ikke kommer noe ut av dette må arbeidet delegeres til en annen person. |
| Medlemmer av teamet blir uenige eller begynner å krangle | Middels | Dårlig arbeidsmiljø og samarbeid innad i teamet som vil gjøre til at arbeidet tar lengre tid. Det kan også bli lite samsvar mellom de forskjellige delene av systemene, som kan gjøre at det ikke fungerer. | Scrum master planlegger et møte for å løse konfliktene. Hvis den ikke blir løst gjennom dialog mellom de involverte medlemmene bestemmer Scrum master hva som bør gjøres. |

# Change management

# TIdsestimat og tid brukt

# Gjenstående arbeid

# Refleksjon om prosjektet

# Kommentarer til gjennomføring av faget

Appendix