

# RAPPORT DEL1

## - SkyNet -

---



### Hensikt

I sprint 1, 2 og 3 har vi implementert de mest grunnleggende User Storiene. Med grunnleggende mener vi mindre User Stories som fungerer som byggeklosser før større oppgaver i systemet. For eksempel har vi implementert storiene legg til kunde og legg til komponent som er nødvendig for å kunne opprette og behandle en arbeidsordre. Vi mener nå at disse grunnleggende User Storiene er på plass og fungerer som de skal, og har derfor valgt å kalle denne perioden for del 1. Dette dokumentet er opprettet for å skrive ned erfaringer vi har gjort oss underveis i prosessen, og vil bli brukt av oss internt i utarbeidelsen av endelig dokumentasjon.

### Fullførte User Stories

Følgende user stories har blitt implementert i del 1:

- Innlogging
- Utlogging
- Søk kunde
- Legg til kunde
- Oppdater kunde
- Søk komponent
- Legg til komponent
- Oppdater komponent
- Legg til produsent
- Legg til modell
- Lett til størrelse

### Erfaringer med utviklingsverktøy og rammeverk

Nå har vi fått jobbet en del med verktøyene, og kan oppsummere våre erfaringer med de så langt:

**Visual Studio 2008:** Dette er vi svært fornøyd med, og har ingenting å utsette. Intuitivt og lett å bruke hvis man har erfaring med bruk av IDE fra før. God debug-funksjonalitet.

**NUnit:** Vi kjenner ikke til andre rammeverk for test-drevet utvikling fra før, så har lite sammenligningsgrunnlag. Litt knotete å få integrert NUnit i byggeserveren vår (TeamCity), mens integrasjonen i Visual Studio gikk automatisk.

**Tortoise SVN:** Dette er den underliggende SVN-klienten som er integrert i Windows utforsker. Denne har fungert best av alle SVN-klientene vi har prøvd, men lar seg dessverre ikke integrere i Visual Studio. Vi har derfor eksperimentert med flere klienter.

**AnkhSVN:** I tillegg til Tortoise valgte vi å bruke AnkhSVN for å integrere SVN i Visual Studio. Vi erfarte imidlertid synkroniseringsproblemer ved bruk av denne klienten, og valgte derfor å avinstallere AnkhSVN og heller prøve ut Visual SVN da vi leste at denne var bygget utenpå Tortoise-klienten.

**Visual SVN:** Er ikke freeware, så vi hadde bare en prøveperiode på 30 dager. Fungerte bra inntil prøveperioden utløp, og vi har fått testet ut og stiftet verdifull kjennskap til denne SVN-klienten. Vi står derfor igjen med Tortoise SVN i Windows utforsker.

**Rhino Mocks:** Ikke brukt enda. Har blitt nedprioritert ettersom vi føler vi har nok å gjøre med den øvrig teknologien vi har valgt og at vi ikke ser det helt store behovet for å bruke mocking objekter.

**Resharper:** Her er det blandede meninger i gruppen. Mens Tore og Øyvind brukte ReSharper til prøveperioden utløp avinstallerte Israr dette etter en ukes utprøving. Resharper har enda ikke klart å samle seg til å gi oss studentlisens, så vi må dessverre droppe dette for resten av prosjektet.

**Windows Server 2008:** Har fungert feilfritt for oss. På denne kjører vi Team City, SQL server og IIS.

**Microsoft SQL Server 2008:** Har også fungert feilfritt. Litt problemer med å replisere testdatabasen til produksjonsdatabasen, men det har alltid løst seg.

**TeamCity:** Dette er byggeserveren som kjører sørger for å kjøre alle testene våre hver gang vi committer nye endringer i koden. Vanskelig å sette opp, lite dokumentasjon, men Israr fikk det til til slutt. Sjekkes med jevne mellomrom for å se at koden ikke feiler.

**16Bugs:** Ikke tatt i bruk enda, da vi ikke har introdusert dette verktøyet for Skydesign enda. Blir introdusert for dem etter hvert når vi har mer funksjonalitet som de kan teste.

**BeanStalk:** Dette er SVN-serveren vi bruker. Dette er en onlinetjeneste vi benytter oss av da vi ikke ville bruke tid på å sette opp egen SVN-server. Tjenesten har kjørt stabilt hele tiden, og som en trivelig bonus så økte beanstalk lagringskapasiteten for gratisbrukere fra 20 til 100MB i løpet av prosjektperioden.

**LINQ:** Veldig kjekt med objektorientert databasespråk. Raskere å kode enn SQL-setninger, og gir renere og kortere kode. Har fungert feilfritt fra første dag.

**ASP.NET MVC:** Vi valgte å gå bort fra MVC-strukturen allerede i sprint 2 da vi følte at det tok for lang tid å utvikle modulene i forhold til hvor god tid vi har til disposisjon. At rammeverket avsluttet betafasen og gikk inn i release candidate-fasen underveis i prosessen gjorde også at mye av det vi hadde lært måtte læres på nytt ettersom det var flere endringer fra beta til RC-utgaven. At også kom en RC2 siste uken vi jobbet med MVC gjorde også at vi besluttet å ikke bruke mer tid på et uferdig rammeverk. Vi valgte derfor å gå for en ren ASP.NET-side uten MVC-rammeverket rundt. Dette økte utviklingsfarten betraktelig.

## Erfaringer med SCRUM

Etter tre sprinter er det på tide med en oppsummering av våre erfaringer med SCRUM-metodikken:

**User Stories:** Disse har vi vært flinke til å bruke i del 1, men ser at det kan bli vanskeligere å formulere gode User Stories i det videre arbeidet. Dette fordi User Storiene blir mer og mer kompliserte, samt at både vi og Skydesign ikke vet 100% hvordan ønsket gang i systemet er før vi har fått testet en løsning og gitt tilbakemeldinger på denne. Vi kommer allikevel til å prøve å jobbe etter dette prinsippet i den resterende delen av prosjektet, ettersom dette er en vesentlig del av SCRUM-metodikken og dermed en viktig erfaring å få med seg.

**Poker planning:** Positive erfaringer. Tvinger frem en dialog mellom oss hvis estimatene spriker, og fungerer som en forberedende aktivitet foran hver sprint der alle blir får et innblikk i hva som kreves i sprinten. Hvorvidt vi har truffet på estimatene vi har enes om har imidlertid variert. Treffsikkerhet krever imidlertid erfaring, noe det tar tid å opparbeide seg.

**Burndown charts:** En grei måte å synliggjøre hvordan vi ligger an i forhold til sprinten. Virker også som dokumentasjon på hvordan de forskjellige sprintene har gått. Motiverende når vi ligger foran skjema. Går raskt å lage og oppdatere.

**Parprogrammering:** I sprint 2 prøvde vi oss på simultanprogrammering der vi jobbet på hvert vårt område av koden. Vi erfarte imidlertid at det var for mange avhengigheter så tidlig i prosjektet og at vi ble sittende å vente på at den andre skal bli ferdig med sin del for å kunne fortsette. Vi opplevde også en del problemer med synkronisering og merging opp mot SVN-serveren. Dette, samt at vi alle følte at vi laget smartere kode når vi jobbet sammen gjorde at vi besluttet å jobbe videre med parprogrammering. Dette gjør også at vi alle har oversikt over, og forståelse for koden. Parprogrammering samsvarer også med smidige utviklingsprinsipper.

**Daily SCRUM:** Det å ha et kort møte på starten av hver arbeidsdag har faktisk gjort at vi kommer raskere i gang med arbeidet. På disse møtene tar vi opp hva som ble gjort dagen før, hva som skal gjøres i dag, og hvilke problemer vi trenger hjelp til å overkomme, hvor lenge skal vi jobbe i dag, osv. Veldig nyttig, og veldig trivelig. Forsterker team-følelsen, og bedrer kommunikasjonen i gruppen.

**Sprint demoer:** Disse møtene har gitt oss anledning til å demonstrere for Skydesign hva vi har laget så langt, og få deres vurdering av systemet hyppig. På disse møtene har vi også bestemt oss for hvilke User Stories som skal utvikles i neste sprint. Skydesign har fått muligheten for å omformulere ønsker og User Storer, noe som gjør at vi har fått en jevnlig korrigerende av kurs slik at vi ikke utvikler for mye på en eventuell blindvei før vi må snu. Sprintdemoene har fungert som en god kommunikasjonskanal mellom oss og Skydesign.

En ting vi har savnet med SCRUM er mer rom for en grundig planlegging av databasestrukturen. Alt (inkludert opprettelse av databasetabeller) skal deles inn i User Stories, og dermed utføres gradvis i prosessen. Vi har prøvd så godt vi kan å holde oss til dette prinsippet.

## Andre erfaringer

Ting som ikke passer inn under de andre overskriftene:

**Arbeidstid:** Vi har vært flinke til å møte til avtalt tid pluss minus (som oftest pluss) 10 minutter. Faste arbeidsdager gjør at vi kan planlegge fritiden til ikke å kolliderer med skolen og vise versa.

**Pauser:** Hyppige (men korte) pauser gir bedre konsentrasjon og bedre humør. Dessverre er vi for dårlige til å benytte oss av dette :P