

Sensur av hovedoppgaver

Høgskolen i Sørøst-Norge

Fakultet for teknologi og maritime fag



Prosjektnummer: **2016-04**

For studieåret: **2015/2016**

Emnekode: **SFHO3201**

Prosjektnavn:

Common Lever AutoTest.

Utført i samarbeid med:

Bacheloroppgaven er gitt av Kongsberg Maritime Merchant Marine Horten.

Ekstern veileder:

Magne Røed

Sammendrag:

Oppgaven er å utvikle en automatisk test av Common Lever parallelt med manuell testing. Common Lever er en ny hendel med berøringsskjerm som brukes til å styre hastighet og posisjon til thruster motorer på fartøy.

Stikkord:

- Automatisering
- Testing
- Utvikling

Tilgjengelig: DELVIS (AutoTest- og verifiseringskode er kun tilgjengelig hos KM)

Prosjekt deltagere og karakter:

Navn	Karakter
Thomas Gjerrud	
Kristoffer Stanger	
Nicolai Sætra	
Philip Vestlie	

Dato: 9. Juni 2016

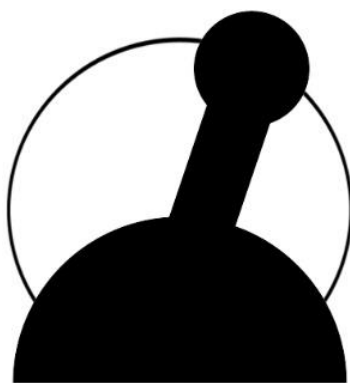
Antonio L. L. Ramos
Intern Veileder

Karoline Moholth
Intern Sensor

Håkon Gjone
Ekstern Sensor



Rapport



Common Lever Auto-Test

Bacheloroppgave for ingeniørstudie ved Høgskolen i Sørøst-Norge, avdeling Kongsberg, fakultet for teknologiske og maritime fag.

Dato: 23.05.2016

Oppdragsgiver: Kongsberg Maritime Merchant Marine Horten

Prosjektdeltagere: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Intern veileder: Antonio L.L. Ramos

Ekstern veileder: Magne Røed

Innholdsfortegnelse for rapporten

Forord	3
Bakgrunn for oppgaven.....	3
Prosjektet	4
Oppdragsgiver	5
Prosjektgruppen	6
Navigering i rapporten	7
Referanser i introduksjon	9
Prosjektplan.....	10
Kravspesifikasjon	39
Work Items	52
Testplan.....	115
Teknisk Dokumentasjon.....	126
Master Produkt Test Dokument	154
Etteranalyse	179
Vedlegg.....	189

Forord

Vi i bachelorgruppen ønsker å takke vår oppdragsgiver Kongsberg Maritime Merchant Marine Horten for muligheten til å bidra i deres prosjekt. Det har vært meget spennende og lærerikt å ta del i et reelt utviklingsprosjekt hvor oppgaven har vært testing og feilsøking på et nyutviklet produkt. Vi vil spesielt takke Magne Røed for strålende veiledning og oppfølging, samt Erik Saly for å dele sine erfaringer med automatisering av test. I tillegg retter vi en stor takk til alle andre ansatte hos KM som har bidratt med sin kompetanse for å hjelpe oss gjennom prosjektperioden.

Videre ønsker vi å takke veileder hos HSN Antonio L.L. Ramos, samt sensor hos HSN, Karoline Moholth for ideer, veiledning og tilbakemeldinger på den akademiske delen av prosjektet.

Bakgrunn for oppgaven

Common Lever er et samarbeidsprosjekt mellom KM Merchant Marine Horten og KM Carpus i Kongsberg for å utvikle en felles hendel for styring av thruster motorer som kan benyttes i begge avdelingenes systemer. Dagens løsning benytter forskjellige hendler til samme oppgave for K-Thrust 600 som KM Horten bruker og K-Thrust 720 som KM Kongsberg benytter. K-Thrust er KM sitt uavhengige thruster kontroll system for elektrisk drevne thruster motorer, fremdrifts enheter og ror. [1] K-Thrust kan også inngå i K-Chief som er et modulbasert alarm og overvåkingssystem, med avansert prosesskontroll og strømstyring for skip. [2]

Prosjektgruppen er involvert i testing av prototypen på Common Lever som var klar på nyåret 2016. Før produktet eventuelt kan prøves ut på et pilotprosjekt, må det tilfredsstille strenge krav til funksjonalitet og stabilitet. Dersom utstyret feiler når det er installert og tatt i bruk kan det få katastrofale følger. Det er derfor viktig at alle mulige feil og potensielle feilkilder fjernes så tidlig som mulig. Dette gjøres ved både automatisert og manuell testing i utviklingsfasen. De automatiserte testene benyttes også for å sjekke at enhetene fungerer optimalt etter oppdatering av software, etter at produktet er tatt i bruk. Nøye testing før release gir naturligvis store besparelser kontra feilsøking på produktet etter release.

En av fordelene med denne prosjektgruppen er at vi har flere års arbeidserfaring fra elektrobransjen. Alle har også erfaring med feilsøking og testing på elektrosystemer, og har derfor god bakgrunn for å lettere forstå sammensatte og komplekse systemer. Det fine med denne oppgaven er at den inneholder et stort spekter med oppgaver fra oppkobling og tilpassing av hardware, til forståelse og koding av software.

Prosjektet

Common Lever AutoTest er et utviklingsprosjekt hvor vår oppgave er testing av et nyutviklet produkt. Som nevnt innledningsvis er det meget viktig at komplekse manøvreringssystemer for fartøyer blir grundig testet før de tas i bruk. Vi i gruppen er stolte av å ha bidratt i et unikt og reelt prosjekt hvor hovedfokus har vært testing og feilsøking. I dette avsnittet vil vi fremheve de viktigste momentene ved dette prosjektet.

Alle i gruppen har fagbrev fra før og har allerede jobbet flere år før ingeniørstudiet. Vi hadde en visjon om å følge normale arbeidsdager for å mest mulig tilnærme oss en vanlig arbeidshverdag. Dette har fungert veldig bra, både i forhold til planlegging av prosjektet og samarbeidet i gruppa. Alle har fått føle på en reell arbeidssituasjon i et viktig og stort prosjekt.

Vi har oppnådd god praktisk tilnærming til de tekniske fagene vi har hatt på Høgskolen. Programmering og mikrokontrollere gav et bra grunnlag for å benytte TestComplete til programmering av AutoTesten. Uten dette faget, ville vi brukt mye mer tid på å lære scriptspråket. Kunnskap fra både Analog Elektronikk og Instrumentering og Styring har vært meget nyttig for å kunne forstå mye av teknologien som ligger bak produktet vi har testet. I Systems Engineering trente vi på å styre et fiktivt prosjekt med fokus på livssyklusen i et system. Vi har nå fått prøve oss på et reelt prosjekt hvor man har sett viktigheten av å se på helheten og livssyklusen i systemet.

KM presiserer at de har sett verdien av å utføre automatisk og manuell testing parallelt etter å ha veiledet oss gjennom dette prosjektet. Vi har også fått gode tilbakemeldinger på å ha avdekket feil og mangler ved produktet på et tidlig stadie, slik at de har hatt mulighet til å gjøre endringer fortløpende.

Oppdragsgiver

Kongsberg Maritime leverer systemer for dynamisk posisjonering og navigering, maritime simulatorer, offshore leting, sikkerhetsstyring, lasthåndtering, marine automasjon og satellitt posisjonering. KM sine løsninger forbedrer effektivitet og sikkerhet i hele det maritime teknologispekteret, og de tilbyr nøkkelferdige ingeniørtjenester innen sektor for skipsbygging og offshore produksjon. Deres største markeder er industri innen offshore, skipsverft og energileting- og produksjon. [3]



Bachelorgruppen har oppgave for Kongsberg Maritime Merchant Marine Horten. Avdelingen har support på blant annet K-Chief, AutoChief og Thruster control systems. [4]



Prosjektgruppen

Prosjektgruppen består av Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra og Philip Vestlie. Samtlige studerer ved Høgskolen i Sør-Øst Norge, campus Kongsberg, avdeling for teknologiske og maritime fag. Alle går studieretning mekatronikk og kybernetikk, og har opptak via Y-veien. Det vil si at man har fått opptak til treårig ingeniørstudie på bakgrunn av fagbrev fremfor generell studiekompetanse. Philip, Nicolai og Thomas har fagbrev som elektriker gruppe L og Kristoffer har fagbrev som produksjonselektroniker.



Philip Vestlie

Philip_vestlie@hotmail.com

92019500

Gruppeleder



Thomas Gjerrud

thomasgjerrud@gmail.com

91605157

Dokumentasjonsansvarlig



Nicolai Sætra

Nicco88@live.com

97193835

Programmeringsansvarlig



Kristoffer Stanger

Kristoffer_stanger@hotmail.com

93095778

Testansvarlig

Navigering i rapporten

Bacheloroppgaven består av flere dokumenter satt sammen til en rapport. Alle dokumentene inneholder egne innholdsfortegnelser og referanser. Her følger en forklarende oversikt over de forskjellige dokumentene i ordnet rekkefølge.

- **Prosjektplan**

Her beskrives i hovedsak oppgaven mer detaljert, prosjektmodellen som er benyttet, samt en hovedplan for prosjektet. I tillegg finnes informasjon om medlemmene i prosjektgruppen, målsettinger for de involverte, avgrensninger og forutsetninger for prosjektet, samt en risikovurdering.

- **Kravspesifikasjon**

Denne beskriver de funksjonelle kravene for hendelen. Disse kravene danner grunnlaget for å kunne utføre de manuelle testene i prosjektet. Denne spesifikasjonen er basert på KM's FRS (Functional Requirement Specification) for hendelen.

- **Work Items**

Denne spesifikasjonen er en beskrivelse og kryssreferanseliste for test cases, user stories og tasks i prosjektet. Test casene representerer kravene til AutoTesten, og beskriver hva den skal kontrollere, samt akseptkriteriene for forventet resultat. User storiene er opprettet på bakgrunn av test casene som arbeidsoppgaver i prosjektet, og er videre delt opp i mindre oppgaver i form av tasks.

- **Testplan**

Testplanen beskriver hvordan vi planlegger å verifisere at arbeidet gjøres riktig. Det følger beskrivelse på arbeidsverktøyene TFS og MTM, og hvordan feil og mangler som oppdages skal rapporteres til KM.

- **Teknisk dokument**

Dette forklarer systemet hendelen skal integreres i som en helhet, og hensikten med testingen av systemet. Det følger beskrivelser av hendelen som en enhet, oppbygging av koden for AutoTesten og oppbygging av testtriggen.

- **Master Product Test dokument**

MPT dokumentet er et «hand-over» dokument til KM som i hovedsak er en testrapport på det utførte arbeidet med hendelen gjennom prosjektperioden. Dette er et krav fra KM og følger deres mal for testing av utviklingsprosjekter. Av denne grunn er dokumentet skrevet på engelsk i motsetning til resten av rapporten. MPT dokumentet fungerer også som en konklusjon for testplanen.

- **Etteranalyse**

Her følger en konklusjon på gjennomføringen av prosjektet, hvor det reflekteres over hvilke erfaringer og læringsutbytte gruppen har oppnådd med arbeidet. Etteranalysen inneholder også en sprintlogg basert på oppfølgingsdokumentene som er benyttet til planleggingen i prosjektet.

- **Vedlegg**

Alle vedlegg ligger helt til slutt i rapporten. Dette er dokumenter som timelister, oppfølgingsdokumenter, møtereferater og annen relevant planleggingsdokumentasjon. Koden for AutoTesten er kun vedlagt i utgaven tilgjengelig for sensor, og vil derfor være utilgjengelig i den publiserte versjonen av rapporten.

Referanser:

[1] K-Thrust:

<https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/0A0C3F74B421A7DEC1256A49002DA456?OpenDocument> (sist besøkt 13.05.2016)

[2] K-Chief:

<https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/15AD4F8D74DCCB4EC12570B600526F83?OpenDocument> (sist besøkt 13.05.2016)

[3] Om KM generelt:

<https://www.km.kongsberg.com/ks/web/nokbg0237.nsf/AllWeb/0B29D39758C415B0C1256DF700560834?OpenDocument> (sist besøkt 13.05.2016)

[4] Om KM Horten:

<https://www.km.kongsberg.com/ks/web/nobkj0407.nsf/AllWeb/BCE08F54094C1225C1256ADF00397390?OpenDocument> (sist besøkt 13.05.2016)



Prosjektplan

Versjon 3.0



Common Lever Auto-Test

Bacheloroppgave for ingeniørstudie ved Høgskolen i Sørøst-Norge, avdeling Kongsberg, fakultet for teknologiske og maritime fag.

Dato: 23.05.2016

Oppdragsgiver: Kongsberg Maritime Merchant Marine Horten

Prosjektdeltagere: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Intern veileder: Antonio L.L. Ramos

Ekstern veileder: Magne Røed

Dokumenthistorie

Versjon	Dato	Endringer	Ansvarlig
1.0	05.02.2016	Første versjon	TG, PV, KS
1.1	11.03.2016	Revidert teksten i punkt 1, 1.1, 8.1. Lagt til revidert aktivitetsnettverk (tabell 2). Lagt til kapittel 7.3-7.9	TG
2.0	11.03.2016	Andre versjon ferdig	TG
2.1	18.04.2016	Revidert teksten i punkt 1.1, 1.1.3, 2.2, 3, 4, 6, 6.1, 6.2, Omstrukturering av hele kapittel 7	TG
2.2	19.04.2016	Revidert teksten i punkt 3, 5. Lagt til punkt 5.3.1, 6.2.1, 6.2.2, Lagt til revidert aktivitetsnettverk (tabell 2).	TG
2.3	20.04.2016	Revidert teksten i punkt 7.1, 7.2, 7.3, 7.11. Oppdatert tabelloversikt, figurliste, innholdsfortegnelse og referanseliste.	TG
2.4	10.05.2016	Rettet skrivefeil i hele rapporten	TG
3.0	18.04.2016	Tredje versjon ferdig	TG

Innholdsfortegnelse

1	Introduksjon.....	16
1.1	Forklaring på hendeltyper og motorer	16
1.1.1	Hendel for Azimuth thruster	17
1.1.2	Hendel for enkel tunell thruster	17
1.1.3	Hendel for dobbel tunell thruster	17
2	Målsetting	17
2.1	HSN, Kongsberg	17
2.2	Kongsberg Maritime	17
2.3	Gruppen	18
3	Avgrensninger	18
4	Forutsetninger	18
5	Oppgavebeskrivelser	19
5.1	Hovedoppgave	19
5.2	Spesifisering av hovedoppgave	19
5.3	Mulig utvidelse av hovedoppgave (<i>utgå</i>)	20
5.4	Bonusoppgave (<i>utgå</i>)	20
5.5	Nødvendig utvidelse av oppgaven	20
6	Aktiviteter	21
6.1	Aktivitetsnettverk	21
6.2	Oppfølgingsdokumenter	23
6.2.1	Timelister	23
6.2.2	Planleggingsdokumenter	23
6.3	Arbeidsverktøyene MTM, TFS og TC	24
7	Hovedplan	25
7.1	Prosjektmodellen (SCRUM)	25
7.2	SCRUM teamet	26
7.3	Produkt backlog	27

7.4	Sprint backlog	28
7.5	Sprint planlegging og gjennomgang	28
7.6	Daglig stand-up møte	28
7.7	Prosjektmodellen (Unified Process)	29
7.8	Timefordeling	31
7.9	Endringer i prosjektet	32
7.10	Gantt diagram	32
7.11	Risikovurdering	32
8	Prosjektgruppen.....	35
8.1	Ansvarsfordeling	36
8.1.1	Gruppeleder	36
8.1.2	Programmeringsansvarlig	36
8.1.3	Dokumentasjonsansvarlig	36
8.1.4	Testansvarlig	36
8.2	Organisasjonskart	37
9	Referanser.....	37

Figurliste

Figur 1: SCRUM modell	26
Figur 2: Unified Process prosjektmodellen som først var planlagt benyttet.	30
Figur 3: Overgang fra UP til SCRUM	30
Figur 4: Timefordeling i prosjektet	31
Figur 5: Organisasjonskart	37

Tabelloversikt

Tabell 1: Forklaring av forkortelser	15
Tabell 2: Aktivitetsnettverk.....	22
Tabell 3: Eksempel på timeliste	23
Tabell 4: Eksempel på planlegging av aktiviteter i starten av sprint	24
Tabell 5: Eksempel på gjennomgang av aktiviteter ved sprint avslutning.....	24
Tabell 6. Gradering av risiko.....	33
Tabell 7: Gradering av sannsynlighet og hvilken innvirkning dette har	33
Tabell 8: Risiko Faktorer.....	33

Tabell 1: Forklaring av forkortelser

Forkortelse	Forklaring
HSN	Høgskolen i Sørøst-Norge, avd. Kongsberg, Fakultet for teknologi og maritime fag
KM Carpus	Kongsberg Maritime i Kongsberg
KM Merchant Marine	Kongsberg Maritime i Horten
HW	Hardware
SW	Software
TC	TestComplete
OS	Operator station
TFS	Team Foundation Server
MTM	Microsoft Test Manager
PMS	Power Management System
I/O	Inngang/Utgang
WS	Work Station
SCU	Segment Controller Unit
MEI	Main Engine Interface
WI	Work Items
US	User Story

1 Introduksjon

Denne bacheloroppgaven er gitt av Kongsberg Maritime i Horten avdeling for nyutvikling (KM Merchant Marine). Oppgaven er å lage en automatisert test for Common Lever parallelt med manuell testing. Common Lever er en ny hendel med berøringsskjerm som brukes til å styre pådrag og posisjon på thruster motorer på fartøy. Med fartøy menes for eksempel supply båter og flytende rigger som må kunne posisjoneres i åpent farvann.

I dag bruker KM Carpus og KM Merchant Marine forskjellige type hendel til hvert sitt system. Den nye Common Lever skal brukes i begge systemene, derav navnet som betyr felles hendel. [1]

Bachelorgruppens oppgave er å lage en automatisert test som sjekker alle funksjonene til den nye hendelen. Dette er funksjoner som motoriserte roterende og aksiale bevegelser på hendelen, samt lyssettinger, knappetrykk, lyd og feedback fra skjermen. Det er stadig oppdateringer på programvaren til styringssystemet som hendelen er koblet opp mot. Etter SW oppdatering er det derfor viktig å kontrollere at alle funksjonene til hendelen fortsatt fungerer og kommuniserer med resten av systemet. Den automatiserte testen for Common Lever skal integreres i KM sitt system.

Disse automatiserte testene utvikles for å effektivisere arbeidet med testingen av utstyret. Testene utføres mye raskere enn hva en person vil klare. Testene kjøres i gang på ettermiddagen og utføres i løpet av natten slik at man kan begynne med eventuell feilretting dagen etter. Man gjentar denne prosedyren inntil alle tester gir et tilfredsstillende resultat.

Studentprosjektet er en del av KM sitt Common Lever utviklingsprosjekt som hadde oppstart i august 2015 og skal avsluttes i juli 2016. Det er viktig å presisere at studentene utfører en reel jobb i utviklingsfasen til KM sitt prosjekt, og frigjør i så måte tiden til andre ansatte i KM.

1.1 Forklaring på hendeltyper og motorer

Vi skal i dette prosjektet forholde oss til tre hendeltyper. Det brukes forskjellige hendeltyper avhengig av hva slags motor de skal styre. Hendlene er typisk lokalisert i en konsoll kalt work station (WS) på broen på skipet. På broen kan det være flere WS med flere hendler: framover, akter, babord og styrbord bro vinge. Hendlene som monteres i disse konsollene har ekstern berøringsskjerm. Hendlene kan også monteres i armlenet KM Carpus sin K-Master stol, da med integrert skjerm. [1]

I avsnittene under følger en forklaring på de tre hendeltypene og tilhørende motorer som vil bli omtalt i rapporten.

1.1.1 Hendel for Azimuth thruster

Azimuth thruster er en 360 graders dreibar motor som brukes primært på fartøyer som har et behov for en nøyaktig posisjonering i alle typer vær. De må ha mulighet til å ligge inntil et annet fartøy og utføre operasjoner, eller senke objekter ned til havbunnen og plassere det på en gitt plass. Disse motorene er som oftest elektriske og kan brukes både til framdrift og posisjonering. De finnes også som retractable, det vil si at de kan trekkes opp i skroget på båten når de ikke er i bruk. Hendlene som styrer disse motorene har spaker for pådrag og er 360 grader dreibare.

1.1.2 Hendel for enkel tunell thruster

Tunell thruster er en posisjoneringsmotor montert i en tunell som går på tvers av skroget i fartøyet. Den er i motsetning til azimuth ikke dreibar og gir kun sideveis kraft til fartøyet. Hendel for enkel tunell thruster har derfor heller ingen dreieskive, den har kun en spak for pådrag mot styrbord eller babord side.

1.1.3 Hendel for dobbel tunell thruster

Denne hendelen styrer to tunell thrustere, og har da to spaker for pådrag. Skjermen vil da være todelt.

2 Målsetting

I dette kapittelet beskrives hvilke mål og forventninger Høgskolen, Kongsberg Maritime og gruppen har til Bachelorprosjektet.

2.1 HSN, Kongsberg

Høgskolens mål med Bachelorprosjektet er at studentene skal bruke opparbeidet kunnskap til å styre et realistisk prosjekt på et tverrfaglig nivå. Studentene må vise evne til å håndtere prosjektstyring, gruppearbeid og produsere teknisk og faglig god dokumentasjon av arbeidet.

2.2 Kongsberg Maritime

Målet til KM er at vi skal utvikle og levere ett ferdig produkt i løpet av prosjektperioden. Dette produktet er den automatiske testen av Common Lever. AutoTesten skal dokumenteres og utarbeides i programmet TestComplete etter KM sin standard. Ekstern veileder ønsker en tett oppfølging under prosjektperioden, slik at han har muligheten til å holde oversikt over hvordan prosjektet ligger an.

2.3 Gruppen

Gruppen har som mål å gjennomføre prosjektet i henhold til de forventninger og krav HSN og KM har satt. Vi planlegger å levere oversiktlig og konkret dokumentasjon slik at det blir enkelt for alle interessenter å følge prosjektet. Denne oppgaven gir oss muligheten til å få et innblikk i hvordan industrien opererer og tilnærme oss ingeniørenes arbeidsmetodikk. Fagfeltet oppgaven bygger på er systemtesting, som er en viktig del av ingeniørarbeidet. Det blir stadig mer fokus på automatisering av testing som i nær fremtid vil fremme et behov for flere testingeniører. Alle i gruppen har fagbrev og har derfor et forhold til arbeidslivet fra før. Gruppen ønsker å ta mest mulig lærdom av oppgaven for å stille sterkere når vi skal tilbake i jobb.

3 Avgrensninger

En automatisert test utvikles for å teste et helt system. Disse autotestene er ofte store, kompliserte og tar lang tid å lage. På grunn av begrenset tid for vår prosjektperiode, er det viktig at vi begrenser oppgaven og jevnlig kontrollerer rammene for prosjektet.

For å kontrollere disse rammene jobbes det etter en SCRUM modell med to ukers sprinter (iterasjoner) hvor det holdes planleggingsmøter i starten av hver sprint og evalueringsmøter ved slutten av hver sprint. I tillegg skal det settes av tid til inntil 30 minutters «stand-up» møte hver dag hvor ekstern veileder er tilstede.

Oppgaven er i utgangspunktet begrenset til å lage en autotest for Common Lever hendelrespons og knappetrykk samt feedback fra disse. Hendlene som skal testes er prototyper. For å lage autotester må man forstå hvordan hendlene fungerer, som vil innebære mye manuell testing. Den manuelle testingen vil naturligvis avdekke feil og mangler som gruppen må rapportere som «bugs» til de ansvarlige avdelingene i KM. Dette kan være mekaniske feil eller programfeil. Hendlene skal kobles opp mot et real-time system, som består av en OS, en SCU og to MEI, som beskrives nærmere i den tekniske dokumentasjonen. Alle enhetene skal kobles sammen og monteres i en work station konsoll.

4 Forutsetninger

For å gjennomføre oppgaven tildelt av KM, legges følgende forutsetninger til grunn:

- Tilgang til KM sitt interne nettverk.
- Egne PC-er til alle i gruppen fra KM.
- Kursing i arbeidsverktøyene TFS og MTM.
- Kursing i oppbygging av script i TC.
- Tilgang til KM sine lokaler.
- Ekstern veileder fra KM.
- Ekstern sensor fra KM.

- Hardware levert av KM (Hendler, OS og nødvendige moduler).

Forutsetninger for gruppen:

- Strukturert oppbygging av dokumentasjon.
- Følge KM sin programmeringsstandard.
- Bruke biblioteker for programmeringskommandoer.
- Utviklingstiden skal ikke overstige 5 måneder.
- Tilfredsstille alle parter.

5 Oppgavebeskrivelser

I dette kapittelet følger en beskrivelse av hovedoppgaven som KM ønsker utført. I tillegg beskrives det en mulig utvidelse dersom hovedoppgaven blir for liten, samt en bonusoppgave som kan utføres dersom det blir nok tid til dette.

Kommentar til endring i prosjektet:

Siden dette er et utviklingsprosjekt, har forutsetningene for oppgaven endret seg etter hvert som prosjektet har gått framover. Dette har medført at kapittel 5.2 punkt 2 og 4 utgår. Kapittel 5.3 og 5.4 utgår i sin helhet. Kapittel 5.5 er lagt til som en nødvendig utvidelse av oppgaven for å kunne utføre det som er spesifisert i kapittel 5.2. Se kapittel 7.9 for utfyllende informasjon om endringer i prosjektet. Endringene i oppgavebeskrivelsen er gjort i samsvar med KM og er dokumentert i møtereferat fra 19.04.2016.

5.1 Hovedoppgave

Hovedoppgaven er å programmere en automatisert test for en Azimuth hendel. I tillegg til hendelbevegelsene, er det integrert en touchskjerm med mange funksjoner og knappetrykk.

Ønsket resultat:

- Automatisert test av Azimuth hendel med alle bevegelser og knappetrykk.
- Kjøre Azimuth hendel i et system, der den automatiserte testen sjekker hendelbevegelse opp mot thruster respons.

5.2 Spesifisering av hovedoppgave

Common Lever AutoTest:

- Få oversikt over alle mulige knappetrykk og funksjoner i Common Lever.
- Lage en matrise over alle mulige kombinasjoner av hendler og konfigurasjoner.
(Utgår)
- Lage en automatisert test som kjører et utvalg av disse kombinasjonene av funksjoner i henhold til Kongsberg Maritimes automatiseringsstandard.

- Bygge en virtuell eller elektrisk thruster som kan brukes til å verifisere at den automatiske testen virker. (*Utgår*)

5.3 Mulig utvidelse av hovedoppgave (*utgår*)

Dersom hovedoppgaven blir for liten kan den utvides til å teste en hendel for tunell thruster.

Ønsket resultat:

- Automatisert test av enkel tunell thruster hendel med alle bevegelser og mulige knappetrykk.
- Kjøring av enkel tunell thruster hendel i et system der den automatiserte testen sjekker hendelbevegelse opp mot thruster respons.
- Automatisert test av dobbel tunell thruster hendel med alle bevegelser og mulige knappetrykk.
- Kjøring av dobbel tunell thruster hendel i et system der den automatiserte testen sjekker hendelbevegelse opp mot thruster respons.

5.4 Bonusoppgave (*utgår*)

Mekanisk thruster som kan kobles på Kongsberg Maritime sin IO-modul, og simulere closed loop på både thrust og aksial bevegelse på Azimuth lever.

5.5 Nødvendig utvidelse av oppgaven

Manuelt teste hendelen parallelt med produsering av den automatiske testen til Common Lever Azimuth hendelen:

- Koble opp realtime-system slik at feedback, database og konfigurasjon kommer fra OS
- Manuell produkt test av systemet:
 - Kommunikasjon mellom hendel, SCU og MEI.
 - Lasting av software.
 - Testing av konfigurasjon.
 - Alle feil som oppdages ved manuell testing rapporteres som «bugs» til de ansvarlige personene slik at det blir rettet opp underveis.

Bygge en brokonsoll med hendel og OS integrert:

- Skjære ut i monteringsplater for hendel og indikatorer.
- Montere skjerm og pc til OS.
- Montere DPU'er på monteringsplater i konsoll.
- Trekke nye ledninger slik at det ser presentabelt ut.

6 Aktiviteter

I startfasen av prosjektet utarbeides det en grovfordeling av de aktiviteter som foreløpig er kjent i prosjektet. Frem mot første presentasjon vil disse aktivitetene bli fremstilt i et gantt-diagram som vil vise milepæler og relasjonene mellom de forskjellige aktivitetene. I tillegg blir det opprettet en oversikt over alle aktivitetene i et dokument kalt aktivitetsnettverk. Dette forklares nærmere i kapittel 6.1.

Etter første presentasjon skal det benyttes Microsoft Test Manager (MTM) og Microsoft Visual Studio Team Foundation Server (TFS) for å holde oversikt over work items (WI). Programmene krever opplæring og tas derfor i bruk etter første presentasjon. TFS, MTM og TestComplete (TC) forklares nærmere i kapittel 6.3 og i dokumentet testplan.

6.1 Aktivitetsnettverk

Aktivitetene er delt opp i hoved- og underaktiviteter. Hovedaktiviteter er navngitt med stor bokstav A og tall fortløpende på formen A1-XXX. For eksempel heter hovedaktivitetene «Administrativt» A1 og «Opplæring» A2. Videre er det satt av 3 plasser til nummerering av underaktiviteter. På denne måten er aktivitetsnettverket utvidbart for både hoved- og underaktivitetene. Det å ha god struktur på aktivitetene er meget viktig i forhold til tidsstyring av prosjektet.

Aktivitetsnettverket skal brukes til timeføring og planlegging av sprintene. Dette gjøres ved at man skriver timer hver dag på hvilke aktiviteter man har jobbet på og linker de mot de aktuelle WI i MTM. Ved starten av hver sprint estimeres det hvor mye tid man tror går med på de ulike aktivitetene og lagrer dette i et sprint planleggings dokument. Ved slutten av sprinten går man igjennom timene som er ført og sammenligner med det estimerte timeantallet. På denne måten kan man forbedre og prioritere mer riktig i neste sprint.

Tabell 2 viser aktivitetsnettverket.

Tabell 2: Aktivitetsnettverk

Aktivitetsnummer	Aktivitet
A1-XXX	Administrativt
A1-001	Daglig SCRUM møte
A1-002	Sprint planlegging og gjennomgang
A1-003	Sprint oppfølgingsdokumentasjon
A1-004	Generelt møte
A2-XXX	Opplæring / rigg
A2-001	Opplæring software
A2-002	Opplæring / rigg hardware
A3-XXX	Dokumentasjon
A3-001	Prosjektplan
A3-002	User story (krav)
A3-003	Testplan
A3-004	Hjemmeside
A3-005	Opprette maler
A3-006	Rapport
A4-XXX	Produkt
A4-001	AutoTest scripting
A4-002	Kode til verifisering
A5-XXX	Testing / verifisering
A5-001	Teste script
A5-002	Teste hendel
A5-003	Oppkobling av utstyr
A6-XXX	Presentasjon
A6-001	Produsere PowerPoint
A6-002	Forberedelse
A6-003	Holde presentasjon

6.2 Oppfølgingsdokumenter

Oppfølgingsdokumentene brukes for å holde veiledere oppdatert om fremdriften i prosjektet. I tillegg skal de brukes for å lage en sprintlogg til etteranalysedokumentet som forklarer gjennomføringen av prosjektet i sluttrapporten.

6.2.1 Timelister

I tabell 3 vises et eksempel på en timeliste ført i prosjektet. Den inneholder kolonner for aktivitetene med nummer som er hentet fra aktivitetsnettverket. Timelistene linkes også mot det aktuelle work item for aktiviteten. Malen for timelistene er laget i Microsoft XL regneark og summerer timene, samt henter ut aktivitetsnavnet automatisk ved å plote inn aktivitetsnummeret. Timelister for hele prosjektet ligger under vedlegg 3 til slutt i rapporten.

Tabell 3: Eksempel på timeliste

Timeliste										
Navn:				Uke nr: 15			Dato	Fra: 11.04.2016		
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237					1,5			1,5
A4-001	Auto-Test scripting	62067			4,5					4,5
A3-006	Rapport	63881	7,5	7,5						15
A4-001	Auto-Test scripting	62068			3					3
A4-001	Auto-Test scripting	63991				7,5	1			8,5
A1-004	Generelt Møte	62361					3			3
A5-005	Oppkobling av utstyr	64062					2			2
										0
Sum pr dag			8	8	8	8	8	0	0	40

6.2.2 Planleggingsdokumenter

I tabell 4 vises et eksempel på hvordan skjema for planlegging av sprint fylles ut. Under sprintplanleggingsmøtet estimeres hvor mye total tid det skal brukes på de ulike aktivitetene, som fylles ut i kolonnen for estimert tid. Ved avslutning av sprinten fylles det inn timer fra alle timelistene slik at man kan sammenligne totalt benyttede timer mot det estimerte. Tabell 5 viser et eksempel på dette. Planleggingsdokumentene finnes som vedlegg 4 til slutt i rapporten

Tabell 4: Eksempel på planlegging av aktiviteter i starten av sprint

Planlagte aktiviteter for Sprint 5 (uke 10,11)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 07.03.2016 til: 20.03.2016					Avvik:
				PV	TG	KS	NS	Sum:	
A1-001	Daglig SCRUM møte	62279	12					0	12
A1-002	Sprint Planlegging og Gjennomgang	62237	4					0	4
A1-003	Sprint Oppfølgingsdokumentasjon	62238	4					0	4
A6-001	Produsere Powerpoint	62224	24					0	24
A6-002	Forberedelse Presentasjon	62225	34					0	34
A6-003	Holde Presentasjon	62236	8					0	8
A3-006	Rapport	62360	38					0	38
A4-001	Auto-Test scripting	62240, 62280	50					0	50
A1-xxx	Omvisning Antonio hos KM	62360	6					0	6
								0	0
Sum			180	0	0	0	0		
Total tid								0	

Tabell 5: Eksempel på gjennomgang av aktiviteter ved sprint avslutning

Utførte aktiviteter for Sprint 5 (uke 10,11)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 07.03.2016 til: 20.03.2016					Avvik:
				PV	TG	KS	NS	Sum:	
A1-001	Daglig SCRUM møte	62279	12	3	3	3	3	12	0
A1-002	Sprint Planlegging og Gjennomgang	62237	4	4	3	0,5	1,5	9	-5
A1-003	Sprint Oppfølgingsdokumentasjon	62238	4		2		1	3	1
A6-001	Produsere Powerpoint	62224	24	13	8	20,5	10,5	52	-28
A6-002	Forberedelse Presentasjon	62225	34	18,5	16	11	19	64,5	-30,5
A6-003	Holde Presentasjon	62236	8	1	1		1	3	5
A3-006	Rapport	62360	38		8,5			8,5	29,5
A4-001	Auto-Test scripting	62240, 62280	50	5,5	3,5			9	41
A1-xxx	Omvisning Antonio hos KM	62360	6					0	6
A5-002	Teste Hendel	62810				6,5	4,5	11	-11
Sum			180	45	45	41,5	40,5		
Total tid								172	

6.3 Arbeidsverktøyene MTM, TFS og TC

TFS er et serverprogram KM bruker, hvor prosjektene lagres og oppdateres. Work items (WI) opprettes og lagres i TFS i form av user stories (US) og test cases som kan brytes ned i mindre oppgaver (tasks). Alle endringer i scriptet for AutoTesten i TestComplete sjekkes inn i TFS på den aktuelle US og blir liggende som et changeset. TFS og MTM forklares nærmere i dokumentet testplan, og TestComplete som verktøy forklares nærmere i det tekniske dokumentet. WI i TFS er linket mot MTM og har funksjoner som storyboard, backlog og testplan, som er kjente arbeidsverktøy i SCRUM. TFS gir alle i gruppen god oversikt over alle WI i prosjektet. WI opprettes med unik identifikasjon, ansvarlig person og tilstand på oppgaven. Man oppnår på denne måten full sporbarhet mot hvem og når de ble opprettet, og tilstander som tilsier om de er opprettet, aktive, løst, avsluttet eller klare for utgivelse. WI kan linkes mot hverandre i et hierarki eller flat struktur. De kan også linkes mot testresultatene og SW koden, som blir mest aktuelt i dette prosjektet.

7 Hovedplan

I følgende underkapitler forklares modellen som styrer prosjektet, hvordan vi planlegger å utnytte den tilgjengelige tiden og hvilke risikoer det er tatt høyde for.

Kommentar til prosjektmodellen:

Gruppen byttet prosjektmodell underveis i prosjektet. SCRUM som hovedsakelig er benyttet forklares først i kapittel 7.1 til 7.6. Unified Process som først var tenkt benyttet forklares i kapittel 7.7.

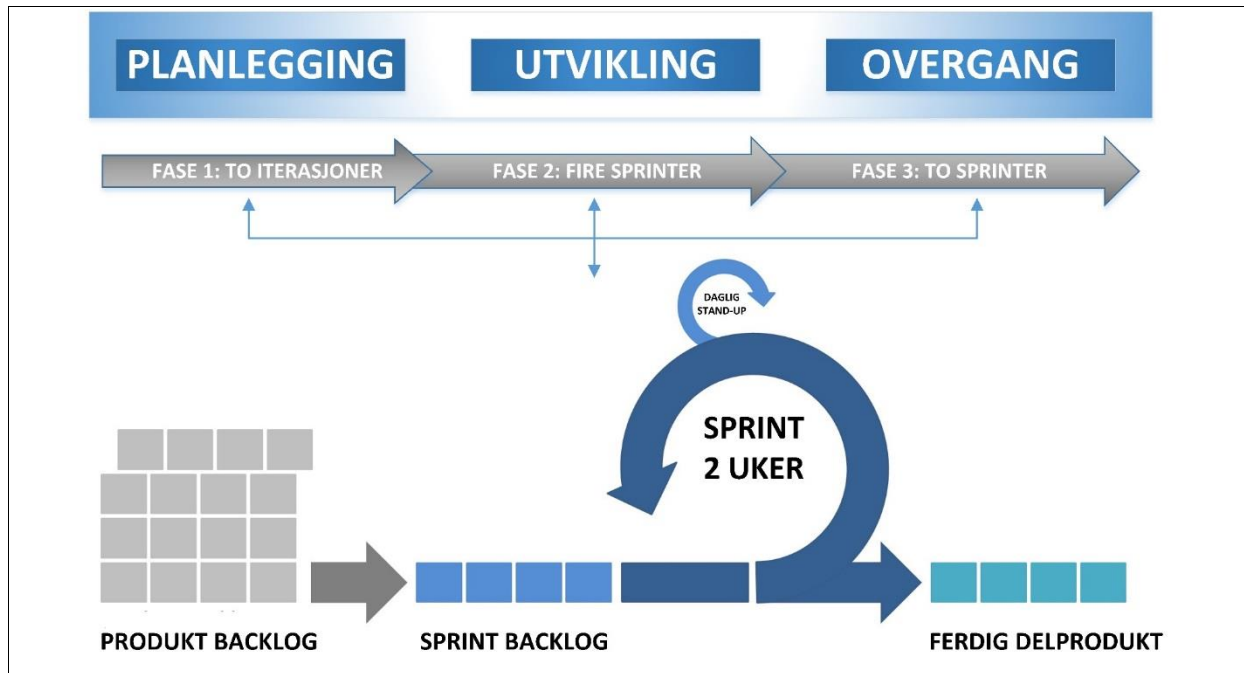
7.1 Prosjektmodellen (SCRUM)

Endring av prosjektmodell:

Før første presentasjon utarbeidet prosjektgruppen en iterativ prosjektmodell basert på Unified Process (UP) med tre faser, fem disipliner og to-ukers iterasjoner. (Dette forklares grundigere i kapittel 7.7). På dette tidlige stadiet i prosjektet så vi for oss en ren software oppgave, og av denne grunn virket det mest fornuftig å følge en slik modell. Etter hvert som vi fikk satt oss bedre inn i datasystemet til KM forstod vi at dette var bygget på SCRUM. Hovedsakelig av denne grunn fant prosjektgruppen det fornuftig å gå over til å benytte SCRUM fullt ut som arbeidsmetodikk. I tillegg er prosjekter basert på testing uforutsigbare og avhengige av tett samarbeid med et utviklingsteam i forhold til både software og hardware. Dette gjorde prosjektet vårt mer utsatt for endringer enn vi først hadde sett for oss. SCRUM var derfor en bedre egnet metode å jobbe etter enn UP som prosjektmodellen vår egentlig var bygget på. Prosjektgruppen ønsket å tilfredsstille både KM og HSN på best mulig måte, og ville ved å bytte arbeidsmetodikk få noe bedre tid til å arbeide med produktet, i tillegg til å tilfredsstille HSN sine krav til dokumentasjon.

SCRUM:

Det er nødvendig med noen tilpasninger for å benytte SCRUM i dette studentprosjektet. For å kunne forholde oss til sluttdato har vi beholdt de tre fasene i UP modellen. Vi beholder også to ukers iterasjoner, men de vil nå kalles sprint. Det er viktig å planlegge nøye før hver sprint begynner og gjennomgå status på slutten av hver sprint. Alle i gruppen må ha et klart forhold til hvor mye man kan klare å gjennomføre i løpet av en sprint. SCRUM krever også selvstendig jobbing og initiativ fra alle deltagere i teamet. Det er vanlig å benytte storypoints for å estimere hvor lang tid man vil bruke på de forskjellige oppgavene i løpet av en sprint. I dette prosjektet har gruppen valgt å heller estimere i forhold til arbeidstimer. Figur 1 viser hvordan vi planlegger å benytte SCRUM modellen.



Figur 1: SCRUM modell

7.2 SCRUM teamet

Studentprosjektet er en del av KM sitt Common Lever utviklingsprosjekt. Studentene i prosjektgruppen utfører mye av testingen i prosjektet, og samarbeider tett i et SCRUM team. Selv om mange ansatte i KM også jobber etter en SCRUM arbeidsmetode, definerer vi for enkelhets skyld ikke de som en del av vårt SCRUM team i studentprosjektet, med unntak av følgende personer som vi hovedsakelig forholder oss til:

- **Produkteier** for Common Lever i forhold til studentprosjektet: Magne Røed. Veileder og kontaktperson for prosjektgruppa. Oppretter og prioriterer test caser.
- **Support:** Erik Saly. Support i forhold til arbeidsverktøyene Microsoft Test Manager, TestComplete, og Team Foundation Server.
- **Prosjektgruppa (Utviklere):** Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie. Utvikling av autotest, rapportering av «bugs», oppkobling, rigg og test av hardware.

7.3 Produkt backlog

Produkt backloggen (PB) er en liste over alt som skal gjøres i prosjektet. Listen blir sortert etter prioritet på oppgavene og oppdatert etter hvert som prosjektet går. PB dokumentet i SCRUM kan sammenlignes med kravspesifikasjonen i en klassisk prosjektstyringsplan og er den eneste kilden til arbeid. Alle idéer, funksjoner, produktforbedringer, dokumentasjonskrav etc. som utviklerne skal jobbe med hentes fra PB. Hvert element i PB kalles user story (US) og inneholder hovedsakelig en beskrivelse, et tidsestimat, prioritet og ansvarlig teammedlem. Siden Common Lever er under utvikling vil PB være forholdsvis kort i begynnelsen av prosjektet, men bli lengre og mer konkret etter hvert som tiden går. [3]

Utviklerne i teamet må opprette US ut fra test cases som produkteier produserer. Det vil også opprettes US for alle oppgaver som er rettet mot den administrative delen av prosjektet. Test casene inneholder en fremgangsmåte for hvordan de individuelle autotestene skal bygges opp. Det er test casene som vil utgjøre selve kravene i dette prosjektet, da det er disse som definerer hvordan autotestene skal utføre kommandoer. For at en US skal kunne defineres som ferdig, må den tilfredsstillende det som er definert i den tilhørende test casen, og alle underoppgaver (tasks) må være ferdig.

US opprettes som work items med unik ID i Team Foundation Server og vil da utgjøre PB som en liste. Bruk av US er en måte å synliggjøre behovet for kravet. Det er vanlig i SCRUM å opprette US på følgende måte:

As a <type of user> I want <some goal> so that <some reason>

Et eksempel:

As a test engineer I want to set different palettes so that I can test that it functions properly

Siden det er hovedsakelig testingeniører som skal bruke produktet (autotesten), blir denne måten å ordlegge seg på unødvendig og derfor vil US i dette prosjektet stort sett fremstå som vanlige konkrete setninger.

Hvis en US opprettes for administrative oppgaver kan den for eksempel se slik ut:

«Create the first Presentation of the Bachelor Project»

En US for produktet kan se slik ut:

«Test Common Lever rotation command»

Det er viktig at US er klare for å inngå i kommende sprint, hvilket betyr at de må være godt definert, ikke for store og forstått av alle utviklerne. En egen aktivitet for dette er vedlikehold av PB hvor produkteier vil ha ansvaret for prioriteten på US, mens det er utviklerne som oppretter og endrer US. [3]

7.4 Sprint backlog

Før hver sprint velger utviklerne de US man forventer å ferdigstille i løpet av sprinten. Det er de valgte US som utgjør sprint backloggen. Utviklerne må bestemme en definisjon for «ferdig» for hver US. [3] US i dette prosjektet kan defineres som ferdig når kriteriene i test casen er oppfylt, og alle tasks tilhørende US er utført. Dersom en US ikke blir ferdig i løpet av sprinten, vil status fortsatt være aktiv og den må overføres til neste sprint.

7.5 Sprint planlegging og gjennomgang

Det settes av en time til møte for sprint planlegging før oppstart, samt en time til gjennomgang ved avslutning av sprinten. Planleggingsmøtet går ut på å sette opp sprint backlog og estimere hvor lang tid som skal brukes på de forskjellige oppgavene. I gjennomgangsmøtet skal det diskuteres hvordan sprinten har gått, hva som kan forbedres og hvordan man skal takle de utfordringene som dukker opp. På slutten av hver sprint opprettes oppfølgingsdokumenter for å informere veilederne om fremdriften i prosjektet. Oppfølgingsdokumentet gjelder for 2 uker av gangen, og inneholder timeestimatet fra planleggingsmøtet, oversikt over faktisk forbrukt tid og status på fremdrift. Dokumentet har egen kolonne for work items, slik at man kan spore hvilken US eller task det er jobbet med. Første Sprint oppfølging vil gjelde for uke 8 og 9.

7.6 Daglig stand-up møte

Det holdes daglige møter på inntil 30 minutter for å informere innad i teamet om status fra forrige dag og hva man planlegger å jobbe med til neste dag. Dette er viktig for å diskutere utfordringer og framdrift, i tillegg til å motivere hverandre videre i prosjektet.

7.7 Prosjektmodellen (Unified Process)

Kommentar: Her beskrives prosjektmodellen slik den først var tenkt.

Prosjektet skal følge en evolusjonær prosjektmodell bestående av tre hovedfaser og fem disipliner. Modellen er inspirert av måten Unified process (UP) er bygget opp, men tilrettelagt denne bacheloroppgaven. UP er en modell spesielt egnet for software prosjekter. Det unike med denne type modell er at alle disiplinene strekker seg over alle fasene, men med varierende grad av prioritet. For eksempel vil man jobbe med kravspesifikasjonen gjennom hele prosjektet, men det vil ha mye høyere prioritet i begynnelsen enn mot slutten. [3]

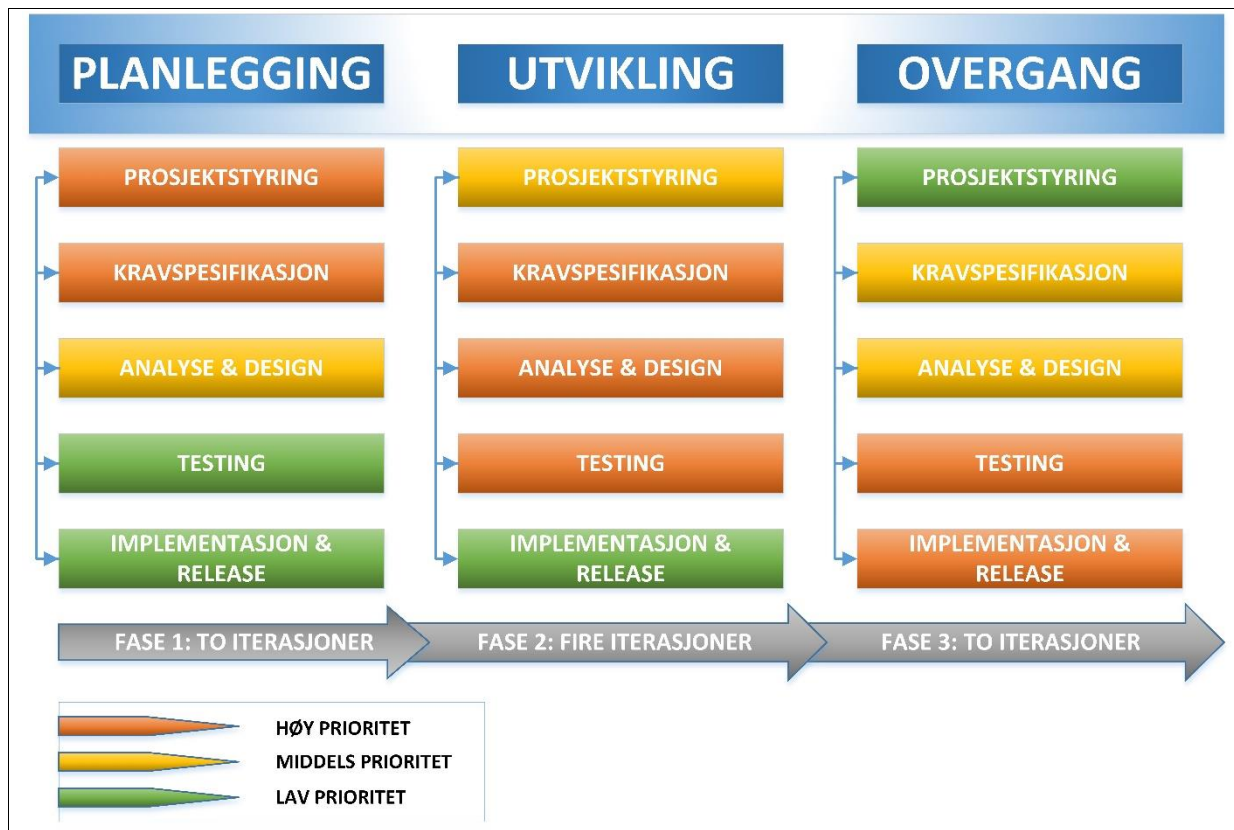
Hver fase inneholder flere iterasjoner hvor det vil være varierende fokus på de fem disiplinene. Vi ser for oss at hver iterasjon varer i to uker. Den første fasen er planleggingsfasen. Dette blir den korteste fasen med to iterasjoner. Fase to er utviklingsfasen med fire iterasjoner, hvor utforming og oppbygging av produktet prioriteres. Den tredje fasen er overgangsfasen med to iterasjoner, hvor gruppen gjør seg ferdig med oppgaven så langt som mulig, slik at KM kan overta resultatet. Det er også hentet inn elementer fra SCRUM i vår arbeidsmetodikk for å forholde oss til KM sin måte å jobbe på. De dagene vi arbeider hos KM holder vi blant annet daglige «standup-møter» på 15 minutter for å informere hverandre om hva som er gjort dagen før, og hva som planlegges neste dag. Bruk av User Story i testfasen og to ukers iterasjoner er elementer som følger av KM sin arbeidsmetodikk.

Disiplinene er som følger:

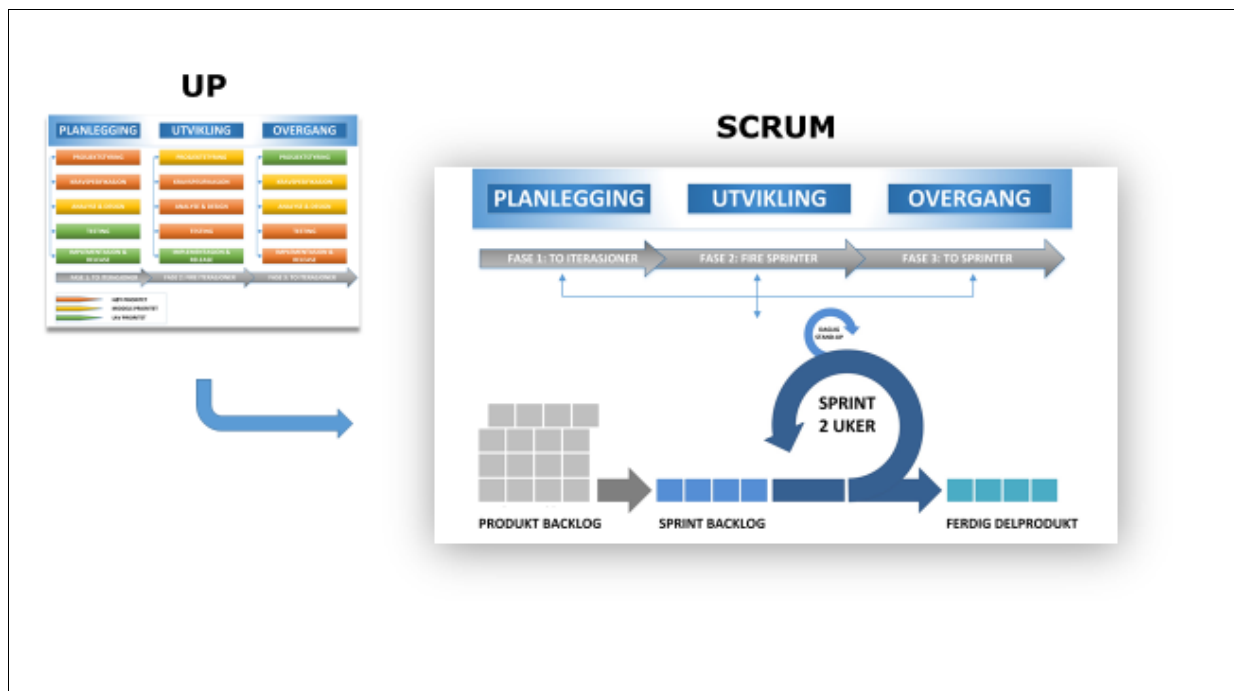
- Prosjektstyring (PS)
- Krav (K)
- Analyse og design (A/D)
- Test (T)
- Implementasjon og Release (I/R)

Under prosjektstyring ligger oppgaver som er administrative og nødvendige for at gruppen og prosjektet skal fungere optimalt. Krav er knyttet hovedsakelig mot arbeid med kravspesifikasjon. Under analyse og design følger oppgaver i forhold utvikling av alle funksjonene i selve produktet. Det blir noe opplæring som også faller naturlig under denne disiplinen. Test følger analyse og design tett, da alle funksjoner skal testes hver for seg før de kan implementeres.

Figur 2 viser UP prosjektmodellen. Figur 3 viser overgangen fra UP til SCRUM



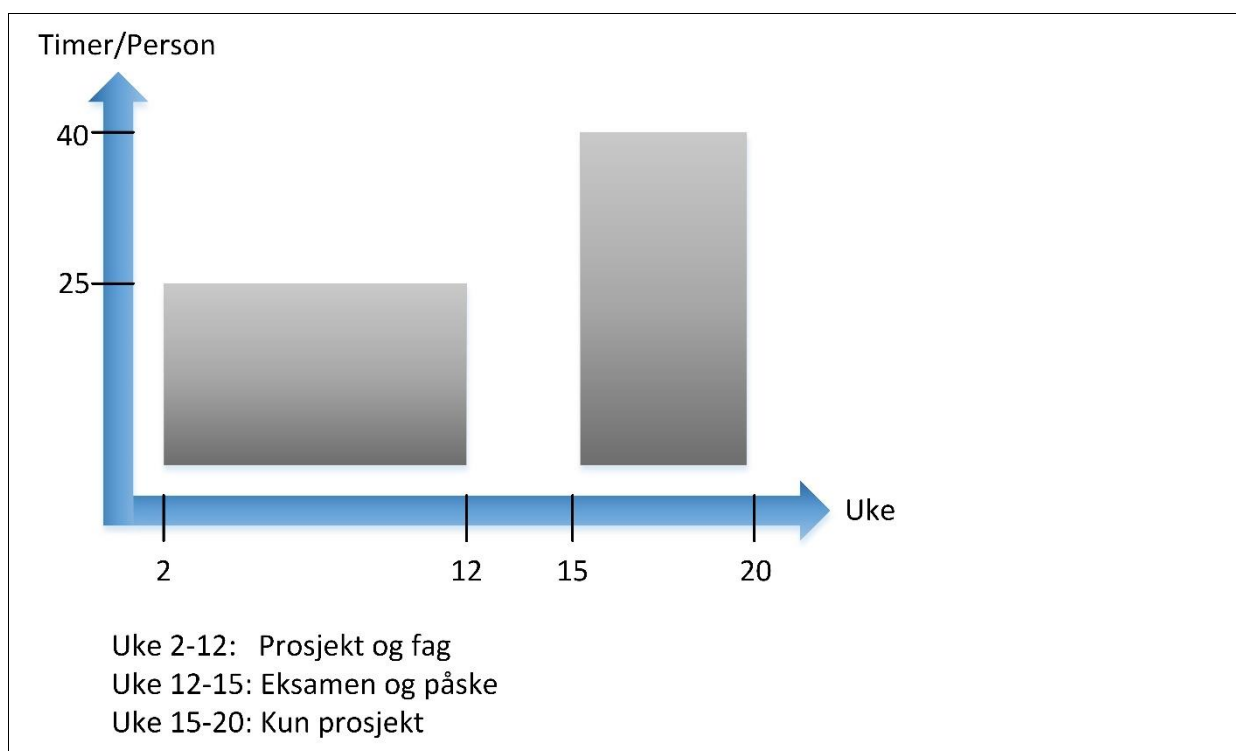
Figur 2: Unified Process prosjektmodellen som først var planlagt benyttet.



Figur 3: Overgang fra UP til SCRUM

7.8 Timefordeling

Det er beregnet at hver prosjektdeltager skal bruke omtrent 600 timer på bacheloroppgaven. I denne gruppen er det enighet om å forholde seg mest mulig til arbeid med oppgaven på dagtid. Frem til påske er det satt av åtte timer, tre dager i uken til bacheloroppgaven, da det i denne perioden kjøres et fag som krever to fulle ukedager. I uken som følger etter påske er det satt av tid til eksamen i dette faget, som da naturligvis må prioriteres fremfor bacheloroppgaven. Etter eksamen blir det satt av åtte timer, fem dager i uken frem til innlevering av prosjektet. Vi ønsker å forholde oss mest til arbeid på dagtid, da dette gjenspeiler vanlige arbeidsdager, og vil fungere best ved arbeid hos oppdragsgiver. Dersom det blir behov er alle i gruppen innstilt på å jobbe noe ekstra, enten på kveldstid eller i helgene. Figur 4 viser en fremstilling av timefordelingen.



Figur 4: Timefordeling i prosjektet

7.9 Endringer i prosjektet

Etter som prosjektet har gått videre, har det kommet noen utfordringer som har ført til endringer i arbeidet. Hovedfokuset i prosjektet er ikke lenger basert på ren scripting av autotest, da Common Lever prosjektet til KM ikke har kommet så langt som planlagt. Dette har medført at vi har kommet inn i prosjektet i en tidligere fase enn planlagt, og har derfor jobbet med oppgaver mer rettet mot nyutvikling og hardware, samt testing underveis.

Det er flere faktorer som spiller inn for at en autotest skal fungere. Den er blant annet avhengig av en fungerende database for at vi skal kunne koble opp simulerte innganger på I/O modulene. Man kan for eksempel simulere at man starter en thruster og motta feedback på at den er i drift. Dette er noe vi har hatt mer fokus på og som bidrar til at utviklingsavdelingen får rettet opp feil.

Scripting er fortsatt en stor del av oppgaven, ettersom vi scripter alle funksjoner fortløpende når de blir rettet opp av utviklingsavdelingen. Selve bevegelsen på hendelen, som tar for seg rotasjon og framdrift er ikke avhengig av databasen så dette er funksjoner vi har fokusert på å gjøre ferdig først.

7.10 Gantt diagram

Det er opprettet et gantt diagram for å holde oversikt over framdriften i prosjektperioden. Diagrammet viser til sammen fire milepæler som i dette tilfellet er tre presentasjoner og innlevering av dokumentasjon på slutten av prosjektet. Vi har i tillegg fordelt de kjente oppgavene i forhold til den tilgjengelige tiden i prosjektperioden. Gantt diagrammet vil bli tilpasset etter hvert som vi får bedre oversikt over oppgavene for å forholde oss til gitte tidsfrister. Diagrammet ligger som vedlegg til slutt i rapporten.

7.11 Risikovurdering

Denne risikoanalysen ser på de faktorer som i løpet av prosjektperioden kan ha innvirkning på hvordan oppgaven løses, eller hvordan sluttresultatet blir.

I tabell 6 er det beskrevet en gradering av hvor alvorlig en risiko er. Det er også tatt med i beregningen hvor sannsynlig det er at den enkelte risiko inntreffer. Dette kommer fram av tabell 7.

Tabell 6. Gradering av risiko

Gradering av Risiko	
1	Lav
2	Middels
3	Høy

Tabell 7: Gradering av sannsynlighet og hvilken innvirkning dette har

Sannsynlighet	Innvirkning
Svært liten	Prosjekt stoppes fram til løsning er funnet
Kan forekomme	Aktivitet revurderes, prosjekt fortsetter
Stor mulighet	Tiltak blir iverksatt, oppgave fortsetter

Tabell 8 lister opp de forskjellige faktorene vi ser på som relevante i vårt prosjekt. Her er det i tillegg til gradering av alvorlighet og sannsynlighet, skrevet opp hvilke tiltak som må gjøres hvis en av risikoene inntreffer.

Tabell 8: Risiko Faktorer

Risiko nr.	Beskrivelse	Gradering	Sannsynlighet	Tiltak
1	Sykdom som medfører lengere fravær	1	Kan forekomme	Fordele oppgavene til de andre gruppemedlemmene
2	Sen info fra KM	2	Kan forekomme	Holde en god dialog slik at vi hurtig får informasjon når den er klar

3	Utsatt opplæring	2	Kan forekomme	Følge prosjektmodellen slik at arbeid som kan utføres uten opplæring kan gjennomføres
4	Frafall i gruppe	3	Svært liten	Se hvor vi er i prosjektet og fordele gjenstående oppgaver etter det. I verste fall må man vurdere å begrense/redefinere oppgaven.
5	Nedbemannning ved bedrift	3	Svært liten	Hvis dette er personer som er knyttet til prosjektet, må det så raskt som mulig utnevnes en ny person som overtar ansvaret
6	Feil i design av produktet	2	Kan forekomme	Følge guidelines for TC og analysere hva som gikk galt
7	Feil estimering av tidsforbruk	2	Kan forekomme	Analysere og revurdere slik at man får en ny oversikt over hvor man står i prosjektet
8	Tap av arbeid på grunn av SW/HW	3	Svært liten	Daglig back-up av dokumenter
9	Lekkasje av konfidensielt materialet	3	Svært liten	Konfidensiell dokumentasjon skal kun lagres på KM sine databaser
10	Feilaktig avgrensning av oppgave	2	Kan forekomme	Gå tilbake og se hva som er etterspurt. Analysere og rådføre seg med veiledere.

8 Prosjektgruppen

Prosjektgruppen består av Philip Vestlie, Kristoffer Stanger, Nicolai Sætra og Thomas Gjerrud. Alle gruppemedlemmene studerer ved Høgskolen i sør-øst Norge, campus Kongsberg, avdeling for teknologiske og maritime fag. Samtlige går studieretning mekatronikk og kybernetikk, og har opptak via Y-veien. Det vil si at man har fått opptak til treårig ingeniørstudie på bakgrunn av fagbrev fremfor generell studiekompetanse. Philip, Nicolai og Thomas har fagbrev som elektriker gruppe L og Kristoffer har fagbrev som produksjonselektroniker. Alle i gruppen er nå i det avsluttende semesteret hvor hovedoppgaven skal skrives for å oppnå Bachelorgrad.



Philip Vestlie

Philip_vestlie@hotmail.com

92019500

Gruppeleder



Thomas Gjerrud

thomasgjerrud@gmail.com

91605157

Dokumentasjonsansvarlig



Nicolai Sætra

Nicco88@live.com

97193835

Programmeringsansvarlig



Kristoffer Stanger

Kristoffer_stanger@hotmail.com

93095778

Testansvarlig

8.1 Ansvarsfordeling

Alle i gruppen er tildelt et eget ansvarsområde. Det er viktig at alle får testet seg i realistiske roller i løpet av prosjektperioden. Alle må derfor involvere seg aktivt i de andre oppgavene i tillegg til å følge opp sitt ansvarsområde. Følgende ansvarsområder er tildelt:

- Philip Vestlie: Gruppeleder
- Nicolai Sætra: Programmering
- Thomas Gjerrud: Dokumentasjon
- Kristoffer Stanger: Testing

8.1.1 Gruppeleder

Gruppeleders hovedansvar er å delegere oppgaver innad i gruppen ved behov, samt sørge for at fremdriftsplanen og viktige tidsfrister i prosjektet følges. I tillegg må gruppeleder ta ansvar for god kommunikasjon mot skolen og oppdragsgiver.

8.1.2 Programmeringsansvarlig

Programmeringsansvarlig tar ansvar for at oppbygging av software koding blir utført i henhold til oppdragsgivers retningslinjer, samt kvalitetssikre at dette fungerer i henhold til kravspesifikasjon.

8.1.3 Dokumentasjonsansvarlig

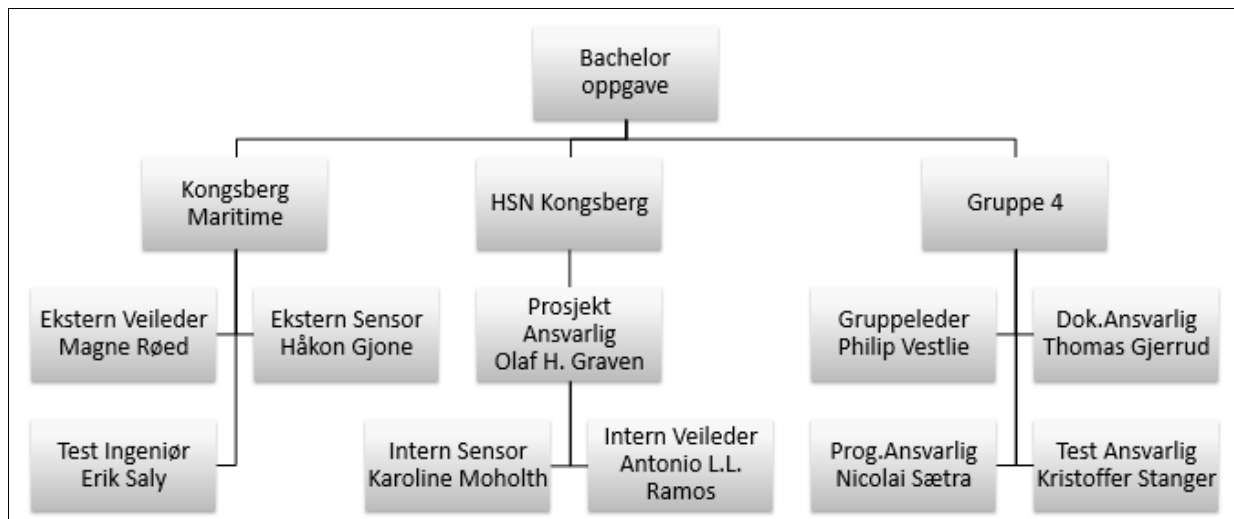
Dokumentasjonsansvarlig sørger for kvalitetssikring av dokumentasjonen til selve hovedoppgaven. Dette innebærer å sikre at dokumentasjonen blir produsert i henhold til skolens retningslinjer og at det blir brukt samme oppsett på dokumentene. I tillegg må han ta ansvar for å lagre og organisere dokumentasjonen på en ryddig og fornuftig måte.

8.1.4 Testansvarlig

Testansvarlig har hovedansvar for å overvåke testprosessen i løpet av prosjektet. Oppgavene vil være å kvalitetssikre og strukturere testene slik at produktet blir bygget etter kravspesifikasjonen.

8.2 Organisasjonskart

Figur 5 viser organisasjonskartet for de involverte i bacheloroppgaven.



Figur 5: Organisasjonskart

9 Referanser

- [1] Kongsberg Maritime. (2013). 372160B Component Requirement Specification
- [2] Scrumalliance, om scrum: <https://www.scrumalliance.org/why-scrum/scrum-guide>
(Sist besøkt 18.05.2016)
- [3] Wikipedia, om Unified Process: https://en.wikipedia.org/wiki/Unified_Process
(Sist besøkt 20.04.2016)



Kravspesifikasjon

Versjon 3.0



Common Lever Auto-Test

Bacheloroppgave for ingeniørstudie ved Høgskolen i Sørøst-Norge, avdeling Kongsberg, fakultet for teknologiske og maritime fag.

Dato: 23.05.2016

Oppdragsgiver: Kongsberg Maritime Merchant Marine Horten

Prosjektdeltagere: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Intern veileder: Antonio L.L. Ramos

Ekstern veileder: Magne Røed

Dokumenthistorie

Versjon	Dato	Endringer	Ansvarlig
1.0	05.02.2016	Første versjon	NS
1.1	09.05.2016	Endret innledningen til dokumentet	NS
2.0	12.05.2016	Andre versjon	TG, KS
2.1	13.05.2016	Oppdatert tabell 1 definisjoner Kap 2.5: <ul style="list-style-type: none"> Fjernet siste kolonne i tabell 6 Lagt til tekst om funksjonelle krav Kap 2.8: <ul style="list-style-type: none"> Krav K6-002 er fjernet Kap 2.10: <ul style="list-style-type: none"> Fjernet hele K8-1xx, kun i bruk hos Kongsberg Maritime avd. Carpus 	NS
3.0	13.05.2016	Tredje versjon	NS

Innholdsfortegnelse

1	Innledning.....	41
2	Krav.....	42
2.1	Kravtyper	42
2.2	Kravprioritet.....	42
2.3	Rammekrav	43
2.4	Maskin og programvarekrav	43
2.5	Funksjonelle Krav	44
2.6	GUI-Krav	45
2.7	Dimmekrav	48
2.8	Knappekrav	49
2.9	Høytalerkrav	50
2.10	CAN-krav	51
2.11	Andre krav	51

Tabelloversikt

Tabell 1. Definisjoner.....	41
Tabell 2. Kravtyper	42
Tabell 3. Kravprioritet.....	42
Tabell 4. Rammekrav	43
Tabell 5. Maskin og programvarekrav	43
Tabell 6. Funksjonelle Krav	44
Tabell 7. GUI-Krav	45
Tabell 8. Dimmekrav.....	48
Tabell 9. Knappekrav	49
Tabell 10. Høytalerkrav	50
Tabell 11. CAN-krav	51
Tabell 12. Andre krav.....	51

Tabell 1. Definisjoner

Begrep	Forklaring
KM	Kongsberg Maritime
CL	Common Lever
HMI	Human machine interface
CAN	Controller Area Network
CANopen	Communication protocol using CAN as physical layer
Vert	Beskrivelse for å identifisere Hoved-enhet på CAN nettverk

1 Innledning

Kravspesifikasjon brukes til å spesifisere hvilke betingelser og krav systemet må følge for at produktet skal fungere som oppdragsgiver ønsker. Hensikten med dette dokumentet er å sørge for at bachelorgruppen og oppdragsgiver (KM) har en felles forståelse av oppgaven og at det fungerer som en form for kvalitetssikring av sluttproduktet.

Kravspesifikasjonen ble i begynnelsen av prosjektperioden utarbeidet slik at man hadde oversikt over kravene til hendelen. Dette er da trukket ut fra KM sitt "Functional Requirements specification K-Thrust 720 HMI devices, Lever and command panel".

Dette dokumentet danner grunnlaget for manuell testing og videre arbeid i prosjektet.

2 Krav

En kravspesifikasjon er avgjørende for utforming, design, funksjonalitet og ytelse av produktet. Derfor er dette dokumentet avgjørende for hvor vellykket produktet blir.

Denne kravspesifikasjonen er laget i samarbeid med KM-Horten og er utledet ut i fra dokumentasjon som er blitt overlevert fra de.

2.1 Kravtyper

Tabell 2. Kravtyper

Type	Beskrivelse
Rammekrav	Krav som må oppfylles for at oppgaven skal ses på som gjennomført.
Maskin og programvarekrav	Maskin- og programvarekrav for Auto-Testen.
Funksjonelle krav	Hva testen skal gjøre.
GUI krav	Hva GUI skal gjøre.
Andre krav	Andre krav som må oppfylles.

2.2 Kravprioritet

Tabell 3. Kravprioritet

Prioritet	Beskrivelse
A	Krav som må oppfylles.
B	Krav som er viktige og bør være på plass.
C	Krav som kun oppfylles dersom det er tilstrekkelig tid til det.

2.3 Rammekrav

Tabell 4. Rammekrav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:
K1-XXX			Rammekrav
K1-001	A	KM	Produktet skal være en automatisk test av hendelrespons og feedback for CL.
K1-002	A	KM	Produktet skal teste knappetrykk og feedback for hendel

2.4 Maskin og programvarekrav

Tabell 5. Maskin og programvarekrav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:
K2-XXX		KM	Maskin og programvarekrav
K2-001	A	KM	Script skal lages i TestComplete
K2-002	A	KM	Testen skal bygges opp i MTM
K2-003	A	KM	TFS skal brukes for sporing av aktiviteter og oppgaver
K2-004	A	KM	Testen skal brukes på KM sine maskiner med KM sitt OS

2.5 Funksjonelle Krav

Dette er krav til hendelen for å kunne lage den automatiske testen. Disse kravene er utarbeidet av bachelorgruppen i samarbeid med KM.

Tabell 6. Funksjonelle Krav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:
K3-XXX			Funksjonelle krav
K3-0xx		KM	Generelt
K3-001	A	KM	Hendelen skal kunne settes i en bestemt posisjon
K3-002	A	KM	Hendelen skal gi tilbakemelding om hvilken posisjon den står i
K3-003	A	KM	Hendelen skal kunne settes i en bestemt vinkel
K3-004	A	KM	Hendelen skal gi feedback på hvilken vinkel den står i
K3-005	A	KM	Skjermknapper skal gi tilbakemelding når de blir trykket
K3-006	A	KM	Lys i hendelen skal kunne dimmes
K3-007	A	KM	Lys i skjerm skal kunne dimmes
K3-008	A	KM	Høytaler skal kunne gi alarmer
K3-009	A	KM	Søyleindikasjon skal vise pådrag
K3-010	A	KM	Indikator skal vise vinkel på thruster

2.6 GUI-Krav

Tabell 7. GUI-Krav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:	Referansenr. til KM HMI-Krav
K4-XXX		KM	GUI-Krav	
K4-001		KM	Generelt	
K4-001	A	KM	HMI-enhet skal ha en skjerm-låsfunksjon	4.01.205.0
K4-002	A	KM	Skjerm-låsfunksjonen skal etter en forhåndsinnstilt tid resettes	4.01.205.1
K4-003	A	KM	HMI-enhet skal tilby 3 paletter for dag, skumring, nattbruk	4.01.215.0
K4-004	A	KM	Tilgang til servicemeny skal være passordbeskyttet	4.22.410.0
K4-1xx		KM	Knappetrykk	
K4-101	A	KM	Bakgrunnen skal ha mulighet til å blinke mellom to farger	4.01.15.1
K4-102	A	KM	Rammen rundt knappen skal ha mulighet til å blinke mellom 2 farger	4.01.20.1
K4-103	A	KM	Knappen skal indikere når en forespørsel er sendt	4.01.25.0
K4-104	A	KM	Knappen skal indikere at den venter på bekreftelse fra vert	4.01.25.0
K4-2xx		KM	Indikator	

K4-201	A	KM	Bakgrunnen skal ha mulighet til å blinke mellom 2 farger	4.01.60.1
K4-202	A	KM	Rammen skal ha muligheten til å blinke mellom 2 farger	4.01.65.1
K4-3xx		KM	Søyleindikasjon	
K4-301	A	KM	Rammen skal ha mulighet til å blinke mellom 2 farger	4.01.80.1
K4-302	A	KM	Numerisk indikasjon skal indikere forskjellige enheter/elementer	4.01.85.1
K4-303	A	KM	Elementene skal ha en konfigurerbar maksimumsverdi	4.01.90.0
K4-304	A	KM	Elementene skal ha en konfigurerbar minimumsverdi	4.01.95.0
K4-305	A	KM	Elementene skal ha en konfigurerbar nullverdi	4.01.100.0
K4-306	A	KM	Elementene skal ha en indikasjon av settpunktverdiene.	4.01.105.0
K4-307	A	KM	Synligheten av Settpunktverdiene skal være mulig å skru av/på	4.01.105.0
K4-308	A	KM	Elementene skal støtte knappetrykk funksjonalitet. Når trykket skal man måtte bekrefte	4.01.120.0
K4-4xx		KM	Alarm	
K4-401	A	KM	Alarmenter skal indikeres med en enkel bokstav	4.01.135.0
K4-402	A	KM	Bakgrunnen skal ha mulighet til å blinke mellom 2 farger	4.01.145.1

K4-5xx		KM	Service	
K4-501	A	KM	HMI-enheten skal ved forespørsel vise en servicemeny	4.01.200.0
K4-502	A	KM	Servicemenyen skal være beskyttet fra uautorisert bruk	4.01.200.1
K4-503	A	KM	Servicemenyen skal etter en forhåndsinnstilt tid stenge om det ikke har vært brukeraktivitet	4.01.200.2
K4-504	A	KM	Servicemenyen skal støtte node id innstillinger for hoved- og backuphendel	4.01.200.3

2.7 Dimmekrav

Tabell 8. Dimmekrav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:	Referanse til KM HMI-Krav
K5-XXX			Dimmekrav	
K5-001	A	KM	HMI-enheten skal lese signalene fra lyssensor	4.01.155.0
K5-002	A	KM	Operatørinnstillinger skal støtte lokalt dimmemodus hvor dimmenivå er basert kun på lokal dimmesensor	4.01.202.3
K5-003	A	KM	HMI-enhet skal motta referanseverdi for dimmenivå via CAN	4.01.210.0
K5-004	A	KM	HMI-enhet skal basert på referansenivået for dimming kalkulere LCD-bakgrunnslysnivå	4.01.210.1
K5-005	A	KM	HMI-enhet skal basert på referansenivået for dimming kalkulere bakgrunnslysnivå på knapper	4.01.210.2
K5-006	A	KM	HMI-enhet skal basert på referansenivået for dimming kalkulere lysnivået på knappeindikasjonen	4.01.210.3
K5-007	A	KM	Hendel-enhet skal basert på referansenivået for dimming kalkulere lysnivået på hendelen	4.01.210.4

2.8 Knappekrav

Tabell 9. Knappekrav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:	Referanse til KM HMI-Krav
K6-XXX			Knappekrav	
K6-0xx		KM	Main	
K6-001	A	KM	Hendel hoved-enhet skal lese signalet fra nødstoppekappen	4.01.265.0
K6-002				
K6-003	A	KM	HMI-enheten skal lese signalene fra fysiske knappetrykk	4.01.160.0
K6-004	A	KM	Knappeindikasjon skal ha 4 blinketilstander: Av, sakte blink, rask blinking, av	4.01.165.1
K6-005	A	KM	Knappeindikasjon skal gi 2 fargestatuser: rød, grønn	4.01.165.2

2.9 Høytalerkrav

Tabell 10. Høytalerkrav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:	Referanse til KM HMI- Krav
K7-XXX			Høytalerkrav	
K7-001	A	KM	Høytalerkontrolleren skal støtte alarmlyd	4.01.175.0
K7-002	A	KM	Høytalerkontrolleren skal støtte varslingslyd	4.01.176.0
K7-003	A	KM	Varslingslyd skal automatisk resettes etter en forhåndsinnstilt tid	4.01.176.2
K7-004	A	KM	Høytalerkontrolleren skal støtte kommandooverføringslyd	4.01.180.0
K7-005	A	KM	Høytalerkontrolleren skal støtte kommandobyttelyd	4.01.185.0
K7-006	A	KM	Kommandooverføringslyd skal resettes automatisk etter en forhåndsinnstilt tid	4.01.185.2

2.10 CAN-krav

Tabell 11. CAN-krav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:	Referanse til KM HMI- Krav
K8-XXX			CAN-krav	
K8-0xx		KM	Hoved	
K8-001	A	KM	Hovedhendel-enhet skal lese posisjonen til hendelen og sende verdien til vert via CANopen	4.01.245.0

2.11 Andre krav

Tabell 12. Andre krav

Kravnummer:	Prioritet:	Utsteder:	Beskrivelse:
K9-XXX			Andre krav
K9-001	A	KM	Konfidensiell informasjon skal kun lagres på KM sine servere
K9-002	A	KM	Konfidensiell informasjon skal kun åpnes på KM sine pc'er



Work Items

Versjon 3.0



Common Lever Auto-Test

Bacheloroppgave for ingeniørstudie ved Høgskolen i Sørøst-Norge, avdeling Kongsberg, fakultet for teknologiske og maritime fag.

Dato: 23.05.2016

Oppdragsgiver: Kongsberg Maritime Merchant Marine Horten

Prosjektdeltagere: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Intern veileder: Antonio L.L. Ramos

Ekstern veileder: Magne Røed

Dokumenthistorie

Versjon	Dato	Endringer	Ansvarlig
1.0	06.02.2016	Første versjon	KS
1.1	12.05.2016	Rettet skrivefeil	TG, KS
2.0	13.05.2016	Andre versjon	TG, KS
2.1	14.05.2016	Lagt til resterende caser	KS
3.0	16.05.2016	Tredje versjon	KS

Sammendrag

Dette dokumentet inneholder tabeller med kryssreferanser mellom alle work items (WI) som er opprettet i Team Foundation Server, både for automatisk test, manuell test og administrative oppgaver. Innholdsfortegnelsen er bygd opp med tanke på god oversikt over WI ID.

Dokumentet består av to hovedpunkter hvor det første beskriver WI for automatisk test og administrative oppgaver. Den andre delen beskriver WI for manuell test.

AutoTesten som er vår hovedprioritet kommer først og er delt inn i test case oversikt, user story oversikt og task oversikt. Disse omhandler automatisering av funksjoner i Common Lever og gjøres ved scripting. Det er test casene som representerer KM sine krav til AutoTesten. User stories er overordnede arbeidsoppgaver, disse er igjen delt opp i flere tasker.

De manuelle testene går hovedsakelig ut på å sjekke koblinger til Power og CAN linjer, samt lasting av software og kommunikasjon.

I dette dokumentet vil også den gjennomgående sporbarheten mellom WI komme godt frem.

Innholdsfortegnelse

1	AutoTest og adm. work items	61
1.1	Test case oversikt	61
1.1.1	TC 61672	61
1.1.2	TC 61898	62
1.1.3	TC 61899	62
1.1.4	TC 61900	63
1.1.5	TC 61901	63
1.1.6	TC 61902	64
1.1.7	TC 61904	65
1.1.8	TC 61905	65
1.1.9	TC 61909	66
1.1.10	TC 61910	66
1.1.11	TC 61913	67
1.1.12	TC 61914	67
1.1.13	TC 61916	68
1.1.14	TC 61919	69
1.1.15	TC 61921	70
1.1.16	TC 61923	70
1.1.17	TC 61924	71
1.1.18	TC 62079	72
1.1.19	TC 62083	73
1.1.20	TC 64131	74
1.1.21	TC 64149	74
1.1.22	Test Case 64903	75
1.2	User story oversikt	76
1.2.1	US 62060	76
1.2.2	US 62061	76
1.2.3	US 62062	76
1.2.4	US 62063	76
1.2.5	US 62064	77
1.2.6	US 62065	77

1.2.7	US 62066	77
1.2.8	US 62067	77
1.2.9	US 62068	78
1.2.10	US 62069	78
1.2.11	US 62070	78
1.2.12	US 62071	79
1.2.13	US 62171	79
1.2.14	US 62202	80
1.2.15	US 60790	80
1.2.16	US 61330	80
1.2.17	US 61988	80
1.2.18	US 62810	81
1.2.19	US 63960	81
1.2.20	US 63990	81
1.2.21	US 64457	82
1.2.22	US 64554	82
1.2.23	US 64904	82
1.3	Task oversikt	83
1.3.1	Task 62153	83
1.3.2	Task 62220	83
1.3.3	Task 62223	83
1.3.4	Task 62224	83
1.3.5	Task 62225	83
1.3.6	Task 62226	84
1.3.7	Task 62227	84
1.3.8	Task 62228	84
1.3.9	Task 62236	84
1.3.10	Task 62237	84
1.3.11	Task 62238	84
1.3.12	Task 62239	85
1.3.13	Task 62240	85
1.3.14	Task 62279	85
1.3.15	Task 62280	85

1.3.16	Task 62281.....	85
1.3.17	Task 62360.....	85
1.3.18	Task 62361.....	86
1.3.19	Task 62372.....	86
1.3.20	Task 62812.....	86
1.3.21	Task 63855.....	86
1.3.22	Task 63856.....	86
1.3.23	Task 63857.....	86
1.3.24	Task 63858.....	87
1.3.25	Task 63878.....	87
1.3.26	Task 63881.....	87
1.3.27	Task 63890.....	87
1.3.28	Task 63892.....	87
1.3.29	Task 63941.....	87
1.3.30	Task 63942.....	88
1.3.31	Task 63943.....	88
1.3.32	Task 63945.....	88
1.3.33	Task 63946.....	88
1.3.34	Task 63947.....	88
1.3.35	Task 63961.....	88
1.3.36	Task 63964.....	89
1.3.37	Task 63991.....	89
1.3.38	Task 64007.....	89
1.3.39	Task 64008.....	89
1.3.40	Task 64012.....	89
1.3.41	Task 64062.....	89
1.3.42	Task 64063.....	90
1.3.43	Task 64136.....	90
1.3.44	Task 64165.....	90
1.3.45	Task 64181.....	90
1.3.46	Task 64182.....	90
1.3.47	Task 64183.....	90
1.3.48	Task 64210.....	91

1.3.49	Task 64222.....	91
1.3.50	Task 64223.....	91
1.3.51	Task 64224.....	91
1.3.52	Task 64225.....	91
1.3.53	Task 64226.....	91
1.3.54	Task 64243.....	92
1.3.55	Task 64244.....	92
1.3.56	Task 64250.....	92
1.3.57	Task 64296.....	92
1.3.58	Task 64301.....	92
1.3.59	Task 64305.....	92
1.3.60	Task 64310.....	93
1.3.61	Task 64311.....	93
1.3.62	Task 64313.....	93
1.3.63	Task 64401.....	93
1.3.64	Task 64458.....	93
1.3.65	Task 64459.....	93
1.3.66	Task 64487.....	94
1.3.67	Task 64488.....	94
1.3.68	Task 64555.....	94
1.3.69	Task 64602.....	94
1.3.70	Task 64604.....	94
1.3.71	Task 64617.....	94
1.3.72	Task 64728.....	95
1.3.73	Task 64756.....	95
1.3.74	Task 64758.....	95
1.3.75	Task 64759.....	95
1.3.76	Task 64790.....	95
1.3.77	Task 64888.....	95
1.3.78	Task 64890.....	96
1.3.79	Task 64891.....	96
1.3.80	Task 64892.....	96
1.3.81	Task 64905.....	96

1.3.82	Task 64906.....	96
1.3.83	Task 64907.....	96
1.3.84	Task 64908.....	97
1.3.85	Task 64909.....	97
1.3.86	Task 64911.....	97
2	Manuelle work items	98
2.1	Test case oversikt.....	98
2.1.1	TC 61877	98
2.1.2	TC 61883	99
2.1.3	TC 61885	99
2.1.4	TC 61886	100
2.1.5	TC 61887	100
2.1.6	TC 61889	101
2.1.7	TC 61891	101
2.1.8	TC 61927	102
2.1.9	TC 61930	102
2.1.10	TC 61931	102
2.1.11	TC 61933	103
2.1.12	TC 61934	103
2.1.13	TC 63825	103
2.1.14	TC 64466	104
2.1.15	TC 64506	104
2.1.16	TC 64509	105
2.1.17	TC 64515	106
2.1.18	TC 64558	107
2.1.19	TC 64571	107
2.1.20	TC 64601	108
2.1.21	TC 64603	109
2.1.22	TC 64606	110
2.1.23	TC 64607	111
2.1.24	TC 64610	112
2.1.25	TC 64613	113
2.1.26	TC 64614	113

2.1.27	TC 64615	114
2.1.28	TC 64618	114

Tabell med forkortelser

Forkortelse	Forklaring
HW	Hardware
SW	Software
TFS	Team Foundation Server
SCU	Segment Controller Unit
WI	Work Items
US	User Story

1 AutoTest og adm. work items

Her følger tabeller over work items for den automatiske testen og de prosjekterelaterte oppgavene.

Første del inneholder test cases, andre del inneholder user stories, og tredje del inneholder tasks.

1.1 Test case oversikt

Det er test casene som representerer KM sine krav til AutoTesten. User stories er overordnede arbeidsoppgaver, disse er igjen delt opp i flere tasker.

1.1.1 TC 61672

ID	TEST CASE	LINKS	
		User Story	Task
61672	Thrust set point test		
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Give Lever command full speed ahead		1. Full ahead position	
2. Give lever command full speed astern		2. Full astern position	
3. Give lever stop command		3. Stop position	

1.1.2 TC 61898

ID	TEST CASE	LINKS		
61898	Command Lever Screen lock function	User Story	Task	Bug
		62060		64273
AKSEPTKRITERIER				
Handling		Forventet resultat		
<div>1. Activate screen lock function</div> <div>2. Push a button on the Common lever panel</div> <div>3. Try to push all buttons on the panel</div> <div>4. Deactivate the screen lock function</div> <div>5. Activate a function on the panel</div>		<div>1. After a preset time screen lock will be activated</div> <div>2. Nothing will happen, screen lock symbol will be visible</div> <div>3. Nothing will happen, screen lock symbol will be visible</div> <div>4. The screen will now be possible to push</div> <div>5. The activated button will react and command will be sent</div>		

1.1.3 TC 61899

ID	TEST CASE	LINKS		
61899	Common lever Day pallet	User Story	Task	Bug
		62061		
AKSEPTKRITERIER				
Handling		Forventet resultat		
<div>1. Activate Day pallet</div> <div>2.</div> <div>3.</div> <div>4. Activate a button on the panel</div> <div>5. Activate a slide bar</div> <div>6. Go in to all sub menus</div>		<div>1. The panel will change to Day pallet</div> <div>2. Verify that all buttons and functions on the common lever change to Day pallet</div> <div>3. Backlight on the lever will adjust to Day pallet</div> <div>4. Button will be activated and command will be sent</div> <div>5. Slider will be activated and command will be sent accordingly</div> <div>6. All menus will be opened in Day pallet</div>		

1.1.4 TC 61900

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61900	Common lever Dusk pallet	62062		
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Activate Day pallet 2. 3. Activate a button on the panel 4. Activate a slide bar 5. Go in to all sub menus		1. The pallet will change to Dusk pallet 2. Verify that all buttons and functions on the common lever change to Dusk pallet 3. Backlight on the lever will adjust to Dusk pallet 4. Button will be activated and command will be sent 5. Slider will be activated and command will be sent accordingly. 6. All menus will be opened in Dusk pallet.		

1.1.5 TC 61901

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61901	Common lever Night pallet	62063		
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Activate night pallet 2. 3. Activate a button on the panel 4. Activate a slide bar 5. Go in to all sub menus		1. The panel will change to Night pallet 2. Verify that all buttons and functions on the common lever change to night pallet 3. Backlight on the lever will adjust to Night pallet 4. Button will be activated and command will be sent 5. Slider will be activated and command will be sent 6. All menus will be opened in Night pallet		

1.1.6 TC 61902

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61902	Common lever password protection	62064		
AKSEPTKRITERIER				
Handling		Forventet resultat		
<ol style="list-style-type: none"> Go to Home page on common lever Activate Service menu Type wrong password Activate Service Menu Type correct nr 1 password Verify that all menus that should be available on this level are there, and are accessible Try one application/menu that is restricted Go to home page and do the same test with password nr2 		<ol style="list-style-type: none"> Home page will appear Password protection will be activated Indication of wrong password will be given Home page will appear after wrong password indication Password protection will be activated Indication of correct password will be given Service menu will appear All menus and functions are accessible Access denied Same result as described for password nr1 		

1.1.7 TC 61904

ID	TEST CASE	LINKS		
61904	Common lever button background color change	User Story	Task	Bug
				64276
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Activate button 3. Simulate background change feedback		1. Home page will appear 2. Button will be activated and command will be sent 3. The background on the button will change from one color to another in a blinking sequence		

1.1.8 TC 61905

ID	TEST CASE	LINKS		
61905	Common lever blinking button frame	User Story	Task	Bug
				64277
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to home page 2. Activate a button 3. Simulate blinking frame feedback		1. Home page will appear 2. Button will be activated and command will be sent 3. The button frame will change color from one color to another in a blinking sequence		

1.1.9 TC 61909

ID	TEST CASE	LINKS		
61909	Common lever pushed button indication	User Story	Task	Bug
		62065		64278
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Activate button 3. Simulate pushed button indication		1. Home page will appear 2. Button will be activated and command will be sent 3. The button will indicate that it is pushed		

1.1.10 TC 61910

ID	TEST CASE	LINKS		
61910	Common lever button in waiting for confirmation state	User Story	Task	Bug
				64168 64297
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Activate button 3. Simulate button in waiting for confirmation state		1. Home page will appear 2. Button will be activated and command will be sent 3. The button will indicate a waiting for confirmation state		

1.1.11 TC 61913

ID	TEST CASE	LINKS		
61913	Common lever Service menu returning to Home page	User Story	Task	Bug
		62066		64299
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Activate Service menu 3. Wait the preset time		1. Home page will appear 2. Service menu will appear 3. Service menu disappear and Home page appear		

1.1.12 TC 61914

ID	TEST CASE	LINKS		
61914	Common lever light sensor	User Story	Task	Bug
		62067		64520
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Go to Service menu 3. Go to light sensor page 4. Adjust light to the sensor		1. Home page will appear 2. Service menu will appear 3. Light sensor value is shown 4. Verify that the sensor correspond to the change in light		

1.1.13 TC 61916

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61916	Common lever Dimming from internal sensor	62068		
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Activate Day pallet 3. Simulate high light sensor value 4. 5. Simulate low sensor value 6. Activate Dusk pallet 7. Simulate high light sensor value 8. 9. Simulate low light sensor value 10. 11. Activate Night pallet 12. Simulate high light sensor value 13. 14. Simulate low light sensor value 15. 16.		1. Home page appear 2. Day pallet activated 3. Verify that the background light on the panel adjust correspondingly 4. Verify that the light in lever adjust correspondingly 5. Verify that the background light on the panel adjust correspondingly 6. Verify that the light in the lever adjust correspondingly 7. Dusk pallet activated 8. Verify that the background light on the panel adjust correspondingly 9. Verify that the light in lever adjust correspondingly 10. Verify that the background light on the panel adjust correspondingly 11. Verify that the light in lever adjust correspondingly 12. Night pallet activated 13. Verify that the background light on the panel adjust correspondingly 14. Verify that the light in lever adjust correspondingly 15. Verify that the background light on the panel adjust correspondingly 16. Verify that the light in lever adjust correspondingly		

1.1.14 TC 61919

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61919	Common lever Dimming from external sensor	62069		
AKSEPTKRITERIER				
Handling		Forventet resultat		
<ol style="list-style-type: none"> Go to Home page Go to Day pallet Activate external dimming and pallet change Send Day pallet and low dimming Send Day pallet and high dimming Send Dusk pallet and low dimming Send Dusk pallet and high dimming Send Night pallet and low dimming Send Night pallet an high dimming Deactivate external dimming 		<ol style="list-style-type: none"> Home page will appear Day pallet will appear The panel and lever will react accordingly The panel and lever will react accordingly The panel and lever will react accordingly The panel and lever will react accordingly The panel and lever will react accordingly The panel and lever will react accordingly The panel and dimming will adjust accordingly to the internal light sensor 		

1.1.15 TC 61921

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61921	Common lever button blinking speed			64276
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Activate a button 3. Simulate slow blinking feedback 4. Simulate fast blinking feedback 5. Simulate on feedback 6. Simulate off feedback		1. Home page will appear 2. The command is sent 3. 4. 5. 6.		

1.1.16 TC 61923

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61923	Common lever button color			
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Simulate green button on indication 2. Simulate red button indication		1. Button indication will change to green 2. Button indication will change to red		

1.1.17 TC 61924

ID	TEST CASE	LINKS		
		User Story	Task	Bug
61924	Common lever sound			
AKSEPTKRITERIER				
Handling		Forventet resultat		
1. Go to Home page 2. Activate alarm sound 3. Activate an alert sound 4. Wait the dedicated time before the alert sound will turn itself automatically off 5. Activate commando transfer sound 6. Wait the dedicated time before the commando transfer sound will turn itself automatically off 7. Activate all the sounds available in the lever 8.		1. Home page will appear 2. Alarm sound will be activated 3. Alert sound will be activated 4. 5. Alert sound will be turned off 6. Commando transfer sound will be activated 7. Commando transfer sound will be turned off 8. All sounds will be activated accordingly		

1.1.18 TC 62079

ID	TEST CASE	LINKS		
		User Story	Task	Bug
62079	Common lever pitch command	61988		
AKSEPTKRITERIER				
Handling		Forventet resultat		
<ol style="list-style-type: none"> 1. Give 20% ahead command (from 0 to 20%) 2. Give 20% ahead command (from 50% to 20%) 3. Give 30% ahead command (from 0 to 30%) 4. Give 30% ahead command (from 50 to 30%) 5. Give 50% ahead command (from 0 to 50%) 6. Give 50% ahead command (from 100 to 20%) 7. Give 70% ahead command (from 0 to 70%) 8. Give 70% ahead command (from 100 to 70%) 9. Give 100% ahead command (from 0 to 100%) 10. Repeat the above test with astern commands 		<ol style="list-style-type: none"> 1. The lever will give 20% command 2. The lever will give 20% command 3. The lever will give 30% command 4. The lever will give 30% command 5. The lever will give 50% command 6. The lever will give 50% command 7. The lever will give 70% command 8. The lever will give 70% command 9. The lever will give 100% command 10. 		

1.1.19 TC 62083

ID	TEST CASE	LINKS		
		User Story	Task	Bug
62083	Common lever rotation command	61330		
AKSEPTKRITERIER				
Handling		Forventet resultat		
<ol style="list-style-type: none"> 1. Give 0 degree command 2. Give 10 degree command 3. Give 25 degree command 4. Give 45 degree command 5. Give 70 degree command 6. Give 90 degree command 7. Give 120 degree command 8. Give 160 degree command 9. Give 180 degree command 10. Give 200 degree command 11. Give 270 degree command 12. Give 300 degree command 13. Give 350 degree command 14. Give 359 degree command 15. Give 0 degree command 16. Go from 0 degree command to 179 degree command via 90 degree 17. Go from 179 degree command to 0 degree command via 90 degree 18. Go from 180 degree command to 359 degree command via 270 degree 19. Go from 359 degree command to 180 degree command via 270 degree 20. Go from 359 degree command to 2 degree command via 0 degree 21. Go from 2 degree command to 358 degree command via 0 degree 		<ol style="list-style-type: none"> 1. The lever will give 0 degree command 2. The lever will give 10 degree command 3. The lever will give 25 degree command 4. The lever will give 45 degree command 5. The lever will give 70 degree command 6. The lever will give 90 degree command 7. The lever will give 120 degree command 8. The lever will give 160 degree command 9. The lever will give 180 degree command 10. The lever will give 200 degree command 11. The lever will give 270 degree command 12. The lever will give 300 degree command 13. The lever will give 350 degree command 14. The lever will give 359 degree command 15. The lever will give 0 degree command 16. Verify that it actually goes from 0 to 179 degree via 90 degree 17. Verify that it actually goes from 179 to 0 degree via 90 degree 18. Verify that it actually goes from 180 to 359 degree via 270 degree 19. Verify that it actually goes from 359 to 180 degree via 270 degree 20. Verify that it actually goes from 359 to 2 degree via 0 degree 21. Verify that it actually goes from 2 to 358 degree via 0 degree 		

1.1.20 TC 64131

ID	TEST CASE	LINKS		
64131	Common lever Start/Stop of Thruster/Servo pump	User Story	Task	Bug
		64457		
AKSEPTKRITERIER				
Handling		Forventet resultat		
<div>1. Go to Home page</div> <div>2. Verify if the thruster is started or stopped, and make sure that the thruster I stopped and ready for start</div> <div>3. Start thruster and pump</div> <div>4. Verify that all status fields have changed to correct state</div> <div>5. Stop thruster and pump</div> <div>6. Verify that all status fields have changed to correct state</div>		<div>1. Home page will appear</div> <div>2. Thruster stopped, and ready to start</div> <div>3. Thruster is starting, and running</div> <div>4. All states correct</div> <div>5. Thruster stopped and pump stopped</div> <div>6. All states correct</div>		

1.1.21 TC 64149

ID	TEST CASE	LINKS		
64149	Common lever status after reboot	User Story	Task	Bug
		64554		64295
AKSEPTKRITERIER				
Handling		Forventet resultat		
<div>1. Send a reset command to Common lever</div> <div>2. Verify that the lever starts up correctly, all values will be available on the lever</div> <div>3. Verify that light an pallet starts up correctly</div>		<div>1. The lever will shut down and reboot</div> <div>2. All values OK</div> <div>3. Starts up with correct pallet and dimming. (Have to be night mode with no flashing of any light)</div>		

1.1.22 Test Case 64903

ID	TEST CASE	LINKS		
		User Story	Task	Bug
64903	Common Lever take command	64904		
AKSEPTKRITERIER				
Handling		Forventet resultat		
4. Send a reset command to Common lever 5. Verify that the lever starts up correctly, all values will be available on the lever 6. Verify that light an pallet starts up correctly		4. The lever will shut down and reboot 5. All values OK 6. Starts up with correct pallet and dimming. (Have to be night mode with no flashing of any light)		

1.2 User story oversikt

User stories er laget på bakgrunn av test casene. Dette er laget for at vi skal kunne ha flere tasker under som representerer arbeidsoppgaver. I mange tilfeller blir det flere arbeidsoppgaver per user story. I tabellene i dette kapittelet er det linket både til en test case og til en eller flere tasker.

1.2.1 US 62060

ID	USER STORY	LINKS	
62060	Automate Test Case 61898 to make sure that the screen lock function is set after a preset time	Test Case	Task
		61898	63878

1.2.2 US 62061

ID	USER STORY	LINKS	
62061	Automate Test Case 61899 to make sure that day palette function works	Test Case	Task
		61899	63855 64225

1.2.3 US 62062

ID	USER STORY	LINKS	
62062	Automate Test Case 61900 to make sure that the dusk palette function works	Test Case	Task
		61900	63857 64311

1.2.4 US 62063

ID	USER STORY	LINKS	
62063	Automate Test Case 61901 to make sure that the night palette function works	Test Case	Task
		61901	63858 64310

1.2.5 US 62064

ID	USER STORY	LINKS	
		Test Case	Task
62064	Automate Test Case 61902 to make sure that the password protection works when someone tries to enter the service menu	61902	63856 64181 64182

1.2.6 US 62065

ID	USER STORY	LINKS	
		Test Case	Task
62065	Automate Test Case 61909 to make sure that the feedback of a pushed button replies as it should	61909	63890 64313

1.2.7 US 62066

ID	USER STORY	LINKS	
		Test Case	Task
62066	Automate Test Case 61913 to make sure that the return to home page function work as it should	61913	64243

1.2.8 US 62067

ID	USER STORY	LINKS	
		Test Case	Task
62067	Automate Test Case 61914 to make sure that the ambient light sensor reacts to changes in light	61914	64012 64224

1.2.9 US 62068

ID	USER STORY	LINKS	
		Test Case	Task
62068	Automate Test Case 61916 to check the dimming values of the light and the screen when different palettes are active	61916	63945
			63946
			64222

1.2.10 US 62069

ID	USER STORY	LINKS	
		Test Case	Task
62069	Automate Test Case 61919 to check the dimming values of the light and the screen when different palettes are active	61919	63947
			64223

1.2.11 US 62070

ID	USER STORY	LINKS	
		Test Case	Task
62070	Work related to presentations		62224
			62225
			62236

1.2.12 US 62071

ID	USER STORY	LINKS	
		Test Case	Task
62071	Work related to the project report and documentation		62226
			62227
			62228
			62281
			62360
			63881
			63892
			63941
			63942
			63943
			64136
			64183
			64210
			64226
			64296
			64401
			64728
			64756
			64758
			64790

1.2.13 US 62171

ID	USER STORY	LINKS	
		Test Case	Task
62171	Work related to software training and hardware rigging		62220
			62223

1.2.14 US 62202

ID	USER STORY	LINKS	
		Test Case	Task
62202	Work related to administrative tasks		62237 62238 62279 62361 64165 64617 64759

1.2.15 US 60790

ID	USER STORY	LINKS	
		Test Case	Task
60790	Get the lever to move from full speed ahead to full speed astern and back to zero		62153

1.2.16 US 61330

ID	USER STORY	LINKS	
		Test Case	Task
61330	Automate Test Case 62083 to make sure that the lever rotates to different set points	62083 64603	62239 62240 64604

1.2.17 US 61988

ID	USER STORY	LINKS	
		Test Case	Task
61988	Automate Test Case 62079 to make sure that the pitch lever goes to different setpoints	62079 64601	62280 62372 64602

1.2.18 US 62810

ID	USER STORY	LINKS	
		Test Case	Task
62810	Test of Lever		62812 64063

1.2.19 US 63960

ID	USER STORY	LINKS	
		Test Case	Task
63960	Production of test equipment		63961 63964 64007 64008 64062 64301 64305

1.2.20 US 63990

ID	USER STORY	LINKS	
		Test Case	Task
63990	Make sure that the script has a good setup		63991 64244 64250 64488

1.2.21 US 64457

ID	USER STORY	LINKS	
		Test Case	Task
64457	Test Common Lever Start/Stop of Thruster and Servo pump	64131	64458 64459 64487

1.2.22 US 64554

ID	USER STORY	LINKS	
		Test Case	Task
64554	Check TCL status after reboot	64149	64555

1.2.23 US 64904

ID	USER STORY	LINKS	
		Test Case	Task
64904	Common Lever take command	64903	64892 64905

1.3 Task oversikt

Denne oversikten viser tabeller over alle enkeltoppgaver som tilhører user storiene som er listet over. Det er flere tasker som er linket opp til samme US og det er derfor delt opp på denne måten.

1.3.1 Task 62153

ID	TASK	LINKS
62153	Check movement	User Story
		60790

1.3.2 Task 62220

ID	TASK	LINKS
62220	Software training	User Story
		62171

1.3.3 Task 62223

ID	TASK	LINKS
62223	Hardware connections	User Story
		62171

1.3.4 Task 62224

ID	TASK	LINKS
62224	Produce power point	User Story
		62070

1.3.5 Task 62225

ID	TASK	LINKS
62225	Preparation for presentations	User Story
		62070

1.3.6 Task 62226

ID	TASK	LINKS
62226	Project plan	User Story 62071

1.3.7 Task 62227

ID	TASK	LINKS
62227	Test plan	User Story 62071

1.3.8 Task 62228

ID	TASK	LINKS
62228	Create Homepage	User Story 62071

1.3.9 Task 62236

ID	TASK	LINKS
62236	Presentation 2	User Story 62070

1.3.10 Task 62237

ID	TASK	LINKS
62237	Sprint planning/retrospective	User Story 62202

1.3.11 Task 62238

ID	TASK	LINKS
62238	Sprint Review	User Story 62202

1.3.12 Task 62239

ID	TASK	LINKS
62239	Verification of 62240	User Story 61330

1.3.13 Task 62240

ID	TASK	LINKS
62240	Autotest Azimuth angle scripting	User Story 61330

1.3.14 Task 62279

ID	TASK	LINKS
62279	Daily scrum Meeting	User Story 62202

1.3.15 Task 62280

ID	TASK	LINKS
62280	Autotest Azimuth pitch scripting	User Story 61988

1.3.16 Task 62281

ID	TASK	LINKS
62281	User Stories	User Story 62071

1.3.17 Task 62360

ID	TASK	LINKS
62360	Report	User Story 62071

1.3.18 Task 62361

ID	TASK	LINKS
62361	Tour at KM with supervisor from HSN	User Story 62202

1.3.19 Task 62372

ID	TASK	LINKS
62372	Verification of 62280	User Story 61988

1.3.20 Task 62812

ID	TASK	LINKS
62812	Test of lever functionality	User Story 62810

1.3.21 Task 63855

ID	TASK	LINKS
63855	Go from main page to palette menu and change to Day palette	User Story 62061

1.3.22 Task 63856

ID	TASK	LINKS
63856	Test if lever accepts wrong password	User Story 62064

1.3.23 Task 63857

ID	TASK	LINKS
63857	Go form main page to palette menu and change to Dusk palette	User Story 62062

1.3.24 Task 63858

ID	TASK	LINKS
63858	Go from main page to palette menu and change to Night palette	User Story 62063

1.3.25 Task 63878

ID	TASK	LINKS
63878	Automate screen lock	User Story 62060

1.3.26 Task 63881

ID	TASK	LINKS
63881	Project retrospective document	User Story 62071

1.3.27 Task 63890

ID	TASK	LINKS
63890	Automate Test Case 61909	User Story 62065

1.3.28 Task 63892

ID	TASK	LINKS
63892	Create Bug reports	User Story 62071

1.3.29 Task 63941

ID	TASK	LINKS
63941	Write list of User Stories, Test Cases and Tasks	User Story 62071

1.3.30 Task 63942

ID	TASK	LINKS
63942	Common lever technical documentation	User Story 62071

1.3.31 Task 63943

ID	TASK	LINKS
63943	SCU/MEI technical documentation	User Story 62071

1.3.32 Task 63945

ID	TASK	LINKS
63945	Make verifying function for Dimming mode	User Story 62068

1.3.33 Task 63946

ID	TASK	LINKS
63946	Automate Test Case 61916	User Story 62068

1.3.34 Task 63947

ID	TASK	LINKS
63947	Automate Test Case 61919	User Story 62069

1.3.35 Task 63961

ID	TASK	LINKS
63961	Cut out in mounting plate for DPU's	User Story 63960

1.3.36 Task 63964

ID	TASK	LINKS
63964	Make mounting plate for DPU's	User Story 63960

1.3.37 Task 63991

ID	TASK	LINKS
63991	General script editing	User Story 63990

1.3.38 Task 64007

ID	TASK	LINKS
64007	Wiring test equipment for bridge console	User Story 63960

1.3.39 Task 64008

ID	TASK	LINKS
64008	Wiring test equipment for bridge console	User Story 63960

1.3.40 Task 64012

ID	TASK	LINKS
64012	Automate Test Case 61914	User Story 62067

1.3.41 Task 64062

ID	TASK	LINKS
64062	Wiring test equipment for bridge console	User Story 63960

1.3.42 Task 64063

ID	TASK	LINKS
64063	Testing lever after rewiring for bridge console	User Story 62810

1.3.43 Task 64136

ID	TASK	LINKS
64136	Technical documentation for Verification, manual testing, and Thruster functionality	User Story 62071

1.3.44 Task 64165

ID	TASK	LINKS
64165	General meeting	User Story 62202

1.3.45 Task 64181

ID	TASK	LINKS
64181	Automate Test Case 61902	User Story 62064

1.3.46 Task 64182

ID	TASK	LINKS
64182	Make verification function for Test Case 61902	User Story 62064

1.3.47 Task 64183

ID	TASK	LINKS
64183	Requirements specification document	User Story 62071

1.3.48 Task 64210

ID	TASK	LINKS
64210	Technical document	User Story
		62071

1.3.49 Task 64222

ID	TASK	LINKS
64222	Verification for Test Case 61916	User Story
		62068

1.3.50 Task 64223

ID	TASK	LINKS
64223	Verification for Test Case 61919	User Story
		62069

1.3.51 Task 64224

ID	TASK	LINKS
64224	Verification for Test Case 61914	User Story
		62067

1.3.52 Task 64225

ID	TASK	LINKS
64225	Verification for Test Case 61899	User Story
		62061

1.3.53 Task 64226

ID	TASK	LINKS
64226	Making of bridge console written documentation	User Story
		62071

1.3.54 Task 64243

ID	TASK	LINKS
64243	Check that the "gotomain" function works as it should	User Story 62066

1.3.55 Task 64244

ID	TASK	LINKS
64244	General script editing	User Story 63990

1.3.56 Task 64250

ID	TASK	LINKS
64250	Structure the CommonLeverMain tab	User Story 63990

1.3.57 Task 64296

ID	TASK	LINKS
64296	Script technical documentation	User Story 62071

1.3.58 Task 64301

ID	TASK	LINKS
64301	Troubleshooting on test equipment	User Story 63960

1.3.59 Task 64305

ID	TASK	LINKS
64305	Making mounting plate	User Story 63960

1.3.60 Task 64310

ID	TASK	LINKS
64310	Verification for Test Case 61901	User Story 62063

1.3.61 Task 64311

ID	TASK	LINKS
64311	Verification for Test Case 61900	User Story 62062

1.3.62 Task 64313

ID	TASK	LINKS
64313	Verification for Test Case 61909	User Story 62065

1.3.63 Task 64401

ID	TASK	LINKS
64401	Master Product Test Document	User Story 62071

1.3.64 Task 64458

ID	TASK	LINKS
64458	Automate Test Case 64131	User Story 64457

1.3.65 Task 64459

ID	TASK	LINKS
64459	Verification for Test Case 64131	User Story 64457

1.3.66 Task 64487

ID	TASK	LINKS
64487	Automate Test Case 64131	User Story 64457

1.3.67 Task 64488

ID	TASK	LINKS
64488	Edited IO Interface	User Story 63990

1.3.68 Task 64555

ID	TASK	LINKS
64555	Automate Test Case 64149	User Story 64554

1.3.69 Task 64602

ID	TASK	LINKS
64602	Autotest TCL pitch scripting	User Story 61988

1.3.70 Task 64604

ID	TASK	LINKS
64604	Autotest TCL angle scripting	User Story 61330

1.3.71 Task 64617

ID	TASK	LINKS
64617	Sprint Review Documentation	User Story 62202

1.3.72 Task 64728

ID	TASK	LINKS
64728	Introduction document for the project report	User Story 62071

1.3.73 Task 64756

ID	TASK	LINKS
64756	Documentation strategy	User Story 62071

1.3.74 Task 64758

ID	TASK	LINKS
64758	Introduction document for the project report	User Story 62071

1.3.75 Task 64759

ID	TASK	LINKS
64759	Last presentation invitations	User Story 62202

1.3.76 Task 64790

ID	TASK	LINKS
64790	Product description for HSN	User Story 62071

1.3.77 Task 64888

ID	TASK	LINKS
64888	Editing Homepage for final release	User Story 62071

1.3.78 Task 64890

ID	TASK	LINKS
64890	General editing of report text	User Story 62071

1.3.79 Task 64891

ID	TASK	LINKS
64891	Make expo poster	User Story 62071

1.3.80 Task 64892

ID	TASK	LINKS
64892	Test of AutoTest script	User Story 64904

1.3.81 Task 64905

ID	TASK	LINKS
64905	Make a function I TestComplete that makes the Lever take command before testing	User Story 64904

1.3.82 Task 64906

ID	TASK	LINKS
64906	Fixed Check function for Azimuth	User Story 61330

1.3.83 Task 64907

ID	TASK	LINKS
64907	Fixed Check function for Pitch	User Story 61988

1.3.84 Task 64908

ID	TASK	LINKS
64908	Test AutoTest script	User Story
		64554

1.3.85 Task 64909

ID	TASK	LINKS
64909	Make final presentation	User Story
		62202

1.3.86 Task 64911

ID	TASK	LINKS
64911	Prepare for final presentation	User Story
		62202

2 Manuelle work items

Her følger tabeller over work items for den manuelle testingen. De manuelle test casene har ikke egne user stories og task som de automatiske. De er følger KM sitt prosjekt, og var nødvendig å utføre for å danne en forståelse av hvordan hendelen fungerer.

2.1 Test case oversikt

2.1.1 TC 61877

ID	TEST CASE	LINKS	
61877	Common lever SW download test	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Set Node ID with DIM		1.	
2. Download Common Lever boot SW with DIM		2. Downloaded OK	
3. Download Common Lever basis SW with DIM		3. Downloaded OK	
4. Download configuration SW with DIM		4. Downloaded OK	
5. Download Common Lever basis SW with RioLoad		5.	
6. Do all step once more, but take power on the module half way through operation		6.	
7. Try to load an empty module straight from the production line		7.	
8. Try to replace module on a running system		8.	
9. Do a upload of the configuration from the Lever		9. Same configuration as sent down	

2.1.2 TC 61883

ID	TEST CASE	LINKS	
61883	Common lever Installation test	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Connect power on the Common Lever 2. Connect Common Lever on Can A 3. Connect Common Lever on Can B 4. Connect "Emergency Stop" button, and push it		1. Verify that power on the system does not do a significant drop in power 2. Verify that no other components in the system are affected 3. Verify that no other components in the system are affected 4.	

2.1.3 TC 61885

ID	TEST CASE	LINKS	
61885	Common lever DIM	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Verify that "select all" works 2. Verify that "Select all of this DPU type" works 3. Verify Upload Configuration" works 4. Verify that "Debug Report" works 5. Verify that "Reload boot software" works 6. Verify that "Download specific version" works 7. Verify that "Clear configuration" works 8. Verify that "Reset Node" works 9. Verify that "Reset and Don't start application" works 10. Verify that "Start application" works 11. Verify that "Set Node ID" works			

2.1.4 TC 61886

ID	TEST CASE	LINKS	
61886	Common lever long term test	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Apply constant power (24V) for at least a month and verify that the module still works 2. Trend the Lever signal for 2-4 days, and see that there are no dips, or interference to the speed set during that time. 3. Monitor the emergency stop signal for 2-4 days and see that there are no dips, or interference to the signal 4. Apply static electricity and interference to the connectors on the module and verify that nothing happens 			

2.1.5 TC 61887

ID	TEST CASE	LINKS	
61887	Common lever calibration test	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Calibrate the Lever 2. Take power on the module, start it up again 3. Load new database in the Lever 4. Load new DPU SW in the module 5. Load new boot SW in the module 6. 		<ol style="list-style-type: none"> 1. 2. The Lever is still calibrated 3. The Lever is still calibrated 4. The Lever is still calibrated 5. The Lever is still calibrated 6. The Lever will go back to factory calibration 	

2.1.6 TC 61889

ID	TEST CASE	LINKS	
61889	Common lever parameter change test	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Change parameter in the Common Lever 2. Change many parameters over time, so that the SCU needs to do a upload		1. Parameters are saved OK 2. Parameters are saved OK	

2.1.7 TC 61891

ID	TEST CASE	LINKS	
61891	Common lever communication load test	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Apply load on CAN A and verify that everything still works 2. Apply load on CAN B and verify that everything still works 3. Apply load on CAN A and B and verify that everything still works		1. Everything still works 2. Everything still works 3. Everything still works	

2.1.8 TC 61927

ID	TEST CASE	LINKS	
61927	Common lever 24V connected to the CAN plugs	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Connect 24V to CAN A plug		1. CAN does not work, but comes up again as soon as CAN is connected	
2. Connect 24V to CAN B plug		2. CAN does not work, but comes up again as soon as CAN is connected	

2.1.9 TC 61930

ID	TEST CASE	LINKS	
61930	Common lever reversed power	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Connect +24V on 0V connector and 0V on the 24V connector		1. The module don't start, but comes up again as soon as power is reversed back to normal	

2.1.10 TC 61931

ID	TEST CASE	LINKS	
61931	Common lever short circuit CAN A	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Common Lever short circuit CAN A		1. CAN goes down, but comes up again as soon as the short is removed	

2.1.11 TC 61933

ID	TEST CASE	LINKS	
61933	Common lever short circuit CAN B	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Common Lever short circuit CAN A		1. CAN goes down, but comes up again as soon as the short is removed	

2.1.12 TC 61934

ID	TEST CASE	LINKS	
61934	Common lever short circuit power	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Common Lever short circuit power on the module		1. The module goes down, but comes up again as soon as the short is removed	

2.1.13 TC 63825

ID	TEST CASE	LINKS	
63825	Common Lever new configuration of SCU	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	

2.1.14 TC 64466

ID	TEST CASE	LINKS	
64466	Common Lever Image download	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Take a new Lever from warehouse 2. Connect power 3. Insert micro SD card in the Lever 4. Verify that image download successfully 5. Remove the micro SD card and reboot the Lever 6. Do the same test again, but remove the micro SD card half way through image download 7. Reboot the lever 		<ol style="list-style-type: none"> 1. 2. Lever power up 3. The Lever instantly recognize the SD card and begin to download a new image 4. All OK 5. The lever starts up with the new image. All OK 6. Error occur 7. The Lever will start up with old image. All OK 	

2.1.15 TC 64506

ID	TEST CASE	LINKS	
64506	Common Lever Set Node ID using DIM	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Insert a new Lever from the warehouse to a running system 2. Do a node scan using DIM 3. Set node ID on the lever using DIM 4. Load SW on the Lever and verify that all basic functions work as normal 		<ol style="list-style-type: none"> 1. System running as normal, and no effect on the system 2. The Lever will be visible on DIM with Node ID 4 3. The Lever with Node ID 4 will disappear, and the new Node ID will go from "Not Found" to "Booting". Then it will go to "No Application" 4. All OK 	

2.1.16 TC 64509

ID	TEST CASE	LINKS	
		User Story	Task
64509	Common Lever set Node ID using the panel on the Lever		
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Insert a new Lever from the warehouse to a running system 2. Power up the module 3. Set Node ID on the Lever using the panel 4. Load SW on the Lever and verify that all basic functions work as normal		1. System running as normal, and no effect on the system 2. The Lever starts, and no effects to the running system 3. The Lever goes from Node ID 4 to the new Node ID, and the state will go from "Not Found" to "Booting". Then it will go to "No Application". No effect to the running system 4. The Lever starts, and no effects to the running system	

2.1.17 TC 64515

ID	TEST CASE	LINKS	
		User Story	Task
64515	Common Lever Service page		
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Push the sprocket on the screens top right corner 2. Insert correct password for the highest level of access 3. Go into calibration menu 4. Calibrate the Lever 5. Take power off/on 6. Go back to main service menu 7. Go to Node ID page 8. Change the Node ID 9. Take power off/on 10. Go back to main service menu 11. Go to "info" page 12. Go to "unit state page" 13. Go to main service page 14. Push Lamp test on the Lever 15. Activate Lamp from operator station (e.g. K-Chief) 16. Push Sound test 17. Do a upload of the configuration 		<ol style="list-style-type: none"> 1. A password site will appear 2. The service page will appear 3. Calibration menu will appear 4. Verify that the lever get the new calibration 5. Verify that the Lever starts as it should, and that the new calibration is saved 6. The service page will appear 7. Verify that you can read Node ID on the Lever 8. Verify that Node ID is changed 9. Verify that Lever starts as it should, and that the new Node ID is saved 10. The service page will appear 11. Check Serial no, SW version, Image version and boot SW version 12. Check run hrs, temp info, CP load, current and ambient light 13. The service page will appear 14. All lamps in ABE11 and KBE15 are lit, and the light in LTU11 barrel do light sequence on all states 15. All lamps in ABE11 and KBE15 are lit, and the light in LTU11 barrel do light sequence on all states 16. The sound will buzz as high as class rules requires 17. Verify that the same OD as you sent to the lever is possible to upload 	

2.1.18 TC 64558

ID	TEST CASE	LINKS	
		User Story	Task
64558	Common Lever light sensor		
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> Go to home page Go to service menu Go to light sensor page Adjust light to the sensor Connect a CP12, AIPC13 to the system and verify that the light density correspond with the Common Lever Adjust the pallet, light and input sensor for light Connect the thruster system to an external system, and verify that it is possible to control dimming on the Levers/system externally 		<ol style="list-style-type: none"> Home page appear Service menu appear Light sensor is shown Verify that the sensor correspond to the change in light Verify that the light density is the same as all KM products connected to the same system as the Common Lever Verify that the light density is the same as all KM products connected to the same system as the Common Lever All OK 	

2.1.19 TC 64571

ID	TEST CASE	LINKS	
		User Story	Task
64571	Common Lever Emergency stop button status		
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> Thruster running and bridge control Check light in Emergency stop button Push Emergency stop Push Emergency stop once more 		<ol style="list-style-type: none"> All OK Green light The thruster stops, and alarm is given. Light in the button turns to red Green light again in the button 	

2.1.20 TC 64601

ID	TEST CASE	LINKS	
		User Story	Task
64601	Common Lever pitch command		
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Give 20% ahead command (from 0 to 20%) 2. Give 20% ahead command (from 50% to 20%) 3. Give 30% ahead command (from 0 to 30%) 4. Give 30% ahead command (from 50 to 30%) 5. Give 50% ahead command (from 0 to 50%) 6. Give 50% ahead command (from 100 to 20%) 7. Give 70% ahead command (from 0 to 70%) 8. Give 70% ahead command (from 100 to 70%) 9. Give 100% ahead command (from 0 to 100%) 10. Repeat the above test with astern commands 		<ol style="list-style-type: none"> 1. The lever will give 20% command 2. The lever will give 20% command 3. The lever will give 30% command 4. The lever will give 30% command 5. The lever will give 50% command 6. The lever will give 50% command 7. The lever will give 70% command 8. The lever will give 70% command 9. The lever will give 100% command 10. 	

2.1.21 TC 64603

ID	TEST CASE	LINKS	
		User Story	Task
64603	Common Lever rotation command		
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Give 0 degree command 2. Give 10 degree command 3. Give 25 degree command 4. Give 45 degree command 5. Give 70 degree command 6. Give 90 degree command 7. Give 120 degree command 8. Give 160 degree command 9. Give 180 degree command 10. Give 200 degree command 11. Give 270 degree command 12. Give 300 degree command 13. Give 350 degree command 14. Give 359 degree command 15. Give 0 degree command 16. Go from 0 degree command to 179 degree command via 90 degree 17. Go from 179 degree command to 0 degree command via 90 degree 18. Go from 180 degree command to 359 degree command via 270 degree 19. Go from 359 degree command to 180 degree command via 270 degree 20. Go from 359 degree command to 2 degree command via 0 degree 21. Go from 2 degree command to 358 degree command via 0 degree 		<ol style="list-style-type: none"> 1. The lever will give 0 degree command 2. The lever will give 10 degree command 3. The lever will give 25 degree command 4. The lever will give 45 degree command 5. The lever will give 70 degree command 6. The lever will give 90 degree command 7. The lever will give 120 degree command 8. The lever will give 160 degree command 9. The lever will give 180 degree command 10. The lever will give 200 degree command 11. The lever will give 270 degree command 12. The lever will give 300 degree command 13. The lever will give 350 degree command 14. The lever will give 359 degree command 15. The lever will give 0 degree command 16. Verify that it actually goes from 0 to 179 degree via 90 degree 17. Verify that it actually goes from 179 to 0 degree via 90 degree 18. Verify that it actually goes from 180 to 359 degree via 270 degree 19. Verify that it actually goes from 359 to 180 degree via 270 degree 20. Verify that it actually goes from 359 to 2 degree via 0 degree 21. Verify that it actually goes from 2 to 358 degree via 0 degree 	

2.1.22 TC 64606

ID	TEST CASE	LINKS	
		User Story	Task
64606	Common Lever screen lock function		
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Activate screen lock function 2. Push a button on the Common Lever panel 3. Try to push all buttons on the panel 4. Deactivate the screen lock function 5. Activate a function on the panel 		<ol style="list-style-type: none"> 1. After a preset time screen lock will be activated 2. Nothing will happen, screen lock symbol will be visible 3. Nothing will happen, screen lock symbol will be visible 4. The screen will now be possible to push 5. The activated button will react and command will be sent 	

2.1.23 TC 64607

ID	TEST CASE	LINKS	
		User Story	Task
64607	Common Lever Azimuth rose		
AKSEPTKRITERIER			
Handling		Forventet resultat	
<ol style="list-style-type: none"> 1. Thruster stopped and bridge control 2. Start thruster 3. Verify that the azimuth rose angle handles both 0-360 and +- 180 degree input on both set point and feedback 4. Verify that the value of the azimuth rose follows the process value 5. Verify that the background color changes according to state (pink/red/transparent) 6. Verify that the set point and feedback color are according to document 324268 		<ol style="list-style-type: none"> 1. Verify that the azimuth rose have grey outline when the thruster is stopped 2. Verify that the azimuth rose have green outline when the thruster is running 3. Correct values regardless of configuration type 4. Value = process value 5. Transparent = Normal Pink = Invalid Red = Alarm 6. Colors OK 	

2.1.24 TC 64610

ID	TEST CASE	LINKS	
		User Story	Task
64610	Common lever dynamic ctrl pos text field		
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Thruster running, and bridge control		1. Text and color in command button: Bridge = "In Command" and green filled box. Text and color in command button: Aft Bridge = "Not in command" and grey filled box	
2. Take command on aft bridge		2. Yellow button on both bridge and aft bridge with "command transfer" text	
3. Command on aft bridge		3. Text and color in command button: Aft Bridge = "In Command" and green filled box. Text and color in command button: Bridge = "Not in command" and grey filled box	
4. Command local		4. Text and color on all positions: Color = red, text = "Local" 5. Ctrl pos configured: Red fill, text Local when local ctrl. Green fill, text acc to position when in command. Yellow fill with current ctrl pos when not in command and not local	

2.1.25 TC 64613

ID	TEST CASE	LINKS	
64613	Common Lever alarm symbol [A] in top left corner	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Thruster running and bridge control 2. Activate an alarm, and a warning on the system		1. All OK 2. If no alarm and no warning: text "A" on grey filled box, new alarm text "A" on red flashing box acked, not returned "A" on red box, new warning no alarm text "W" on yellow flashing box, acked warning no alarm text "W" on yellow box, acked warning and acked alarm text "A" on red filled box, new warning and acked alarm "W" on yellow filled box	

2.1.26 TC 64614

ID	TEST CASE	LINKS	
64614	Common Lever Dynamic running/limiter text field	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Thruster running in normal mode 2. Thruster stopped 3. Thruster running with limiter		1. Running on green filled box (digital input) 2. Stopped in grey filled box 3. Yellow box, text from limiter active (tag type 88.5)	

2.1.27 TC 64615

ID	TEST CASE	LINKS	
64615	Command Lever audio support	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Thruster running and bridge control 2. Activate an alarm		1. All OK 2. The speaker releases a sound. The sound will be active as long as the alarm	

2.1.28 TC 64618

ID	TEST CASE	LINKS	
64618	Common Lever Ready for DP/JS text field	User Story	Task
AKSEPTKRITERIER			
Handling		Forventet resultat	
1. Thruster running and bridge control 2. System ready for DP 3. System ready for JS 4. System ready for DP/JS		1. All OK 2. Green text "Ready for DP" when ready for DP and not ready for JS 3. Green text "Ready for JS" when ready for JS and not ready for DP 4. Green text "Ready for DP/JS" when ready for DP and ready for JS	



Test plan

Versjon 3.0



Common Lever Auto-Test

Bacheloroppgave for ingeniørstudie ved Høgskolen i Sørøst-Norge, avdeling Kongsberg, fakultet for teknologiske og maritime fag.

Dato: 23.05.2016

Oppdragsgiver: Kongsberg Maritime Merchant Marine Horten

Prosjektdeltagere: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Intern veileder: Antonio L.L. Ramos

Ekstern veileder: Magne Røed

Dokumenthistorie

Versjon	Dato	Endringer	Ansvarlig
1.0	06.02.2016	Første versjon	KS
1.1	10.03.2016	Gjort endringer på bakgrunn av ny info ang test	KS
2.0	11.03.2016	Andre versjon	KS
2.1	18.04.2016	Lagt til beskrivelse av Bugs, (2.1.1).	KS
2.2	11.05.2016	Endret kapittel 1, 2, og lagt til kapittel 3 og 4	TG
3.0	13.05.2016	Tredje versjon	TG

Innholdsfortegnelse

1	Introduksjon.....	119
2	Verifisering av AutoTesten	119
3	Programmer	120
3.1	Visual Studio Team Foundation Server.....	120
3.1.1	TFS PC-versjon	120
3.1.2	TFS nettbasert versjon	121
3.2	TestComplete	122
3.3	TFS linket mot Microsoft Test Manager	124
4	Bugs	125
5	Referanser:	125

Figurliste

Figur 1: Test plan struktur KM	119
Figur 2: Liste over US i TFS	120
Figur 3: Utdrag av board I TFS	121
Figur 4: Utdrag av backlog I TFS	122
Figur 5: Sammenligning av script i changeset	123
Figur 6: Liste med changeset.....	123
Figur 7: Test suite i TFS.....	124
Figur 8: Utdrag av liste over bugs I TFS.....	125

Tabelloversikt

Tabell 1: Forklaring på begreper.....	118
Tabell 2: Beskrivelse av forkortelser.....	118

Tabell 1: Forklaring på begreper

Begreper	Forklaring
User Story	Beskrivelse av produktet som lages (Common Lever Autotest)
Test Case	Opprettes og signeres til en ansvarlig tester
Suites	Inneholder flere test caser som har en sammenheng med hverandre
Sprint	To ukers periode det jobbes med spesifikke oppgaver
Bugs	Bugs lages hvis man oppdager noe uventet ved gjennomkjøring av en test
Work Items	Felles betegnelse for arbeidsoppgave

Tabell 2: Beskrivelse av forkortelser

Forkortelser	Forklaring
TFS	Visual Studio Team Foundation Server
MTM	Microsoft Test Manager
US	User story
TC	Test case
WI	Work item
KM	Kongsberg Maritime

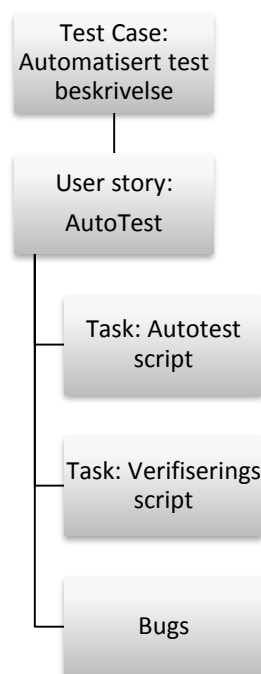
1 Introduksjon

Testplanen beskriver hvordan bachelorgruppen planlegger å verifisere at arbeidet gjøres riktig. For å verifisere dette benyttes Kongsberg Maritime (KM) sitt system for å dokumentere og strukturere testene i prosjektet. Programmene som KM benytter er Team Foundation Server (TFS), Microsoft Test Manager (MTM) og TestComplete. Gruppen skal forholde seg til TFS for styring av prosjektet og TestComplete til scripting av AutoTesten, mens MTM i mindre grad benyttes av studentene. Det følger beskrivelse på disse programmene og hvordan feil og mangler som oppdages skal rapporteres til KM.

2 Verifisering av AutoTesten

KM oppretter test cases (TC) i programmet MTM som beskriver steg for steg hva de enkeltstående automatiserte testene skal gjøre. Alle de automatiserte TC utgjør AutoTesten, og det må verifiseres at AutoTesten fungerer slik den er designet.

Gruppen oppretter user stories (US) i programmet TFS for hver test case som skal automatiseres. US deles opp i tasks for autotestscript og verifiseringsscript. Verifiseringsscriptet er likt bygget opp som autotestscriptet, men med andre verdier for å se at AutoTesten feiler når den skal. Dersom det oppdages feil i testingen skal dette rapporteres som bugs i TFS. Figur 1 viser sammenhengen mellom TC, US, task og bugs.



Figur 1: Test plan struktur KM

3 Programmer

Her forklares sammenhengen mellom de tre programmene nevnt innledningsvis som benyttes for testing, scripting og prosjektstyring.

3.1 Visual Studio Team Foundation Server

TFS er et Microsoft program som gjør det mulig å samkjøre forskjellige software prosjekter. Programmet inneholder en rekke verktøy som gjør det enkelt å jobbe i samme software script. [1] Programmet er tilgjengelig i både pc-versjon og som nettbasert.

3.1.1 TFS PC-versjon

I TFS har alle i teamet tilgang til å lese og endre alle work items som TC, US, tasks og bugs i prosjektet.

En US blir tildelt (assigned) den personen som har ansvar for at hele US fullføres. Taskene blir tildelt den personen som skal utføre den spesifikke oppgaven tasken omhandler. Figur 2 viser et utdrag av en spørring (query) for US listet opp med ID, WI type, tittel, tilstand og hvem som er ansvarlig.

Query Results: 56 items found (1 currently selected).

ID	Work Item Type	Title	Assigned ...	State
62063	User Story	Test Common Lever night palette		Not Started
62064	User Story	Check Common Lever password protection		Not Started
62065	User Story	Check pushed button indication on Common Lever		Not Started
62066	User Story	Check returning to home meny function		Not Started
62067	User Story	Check Common Lever light sensor		Not Started
62068	User Story	Test dimming from internal sensor Common Lever		Not Started
62069	User Story	Test dimming from external sensor Common Lever		Not Started
62070	User Story	Prepare for project presentations	Kristoffer S...	Active
62071	User Story	Produce documentation for the project report	Thomas Gj...	Active
61988	User Story	Test Common Lever pitch command	Thomas Gj...	Active
62171	User Story	Training and rigging equipment	Philip Hem...	Active
62202	User Story	Administrative tasks	Philip Hem...	Active
60790	User Story	MoveLever	Nicolai Sæ...	Resolved
61330	User Story	Test Common Lever rotation command	Nicolai Sæ...	Active

Figur 2: Liste over US i TFS

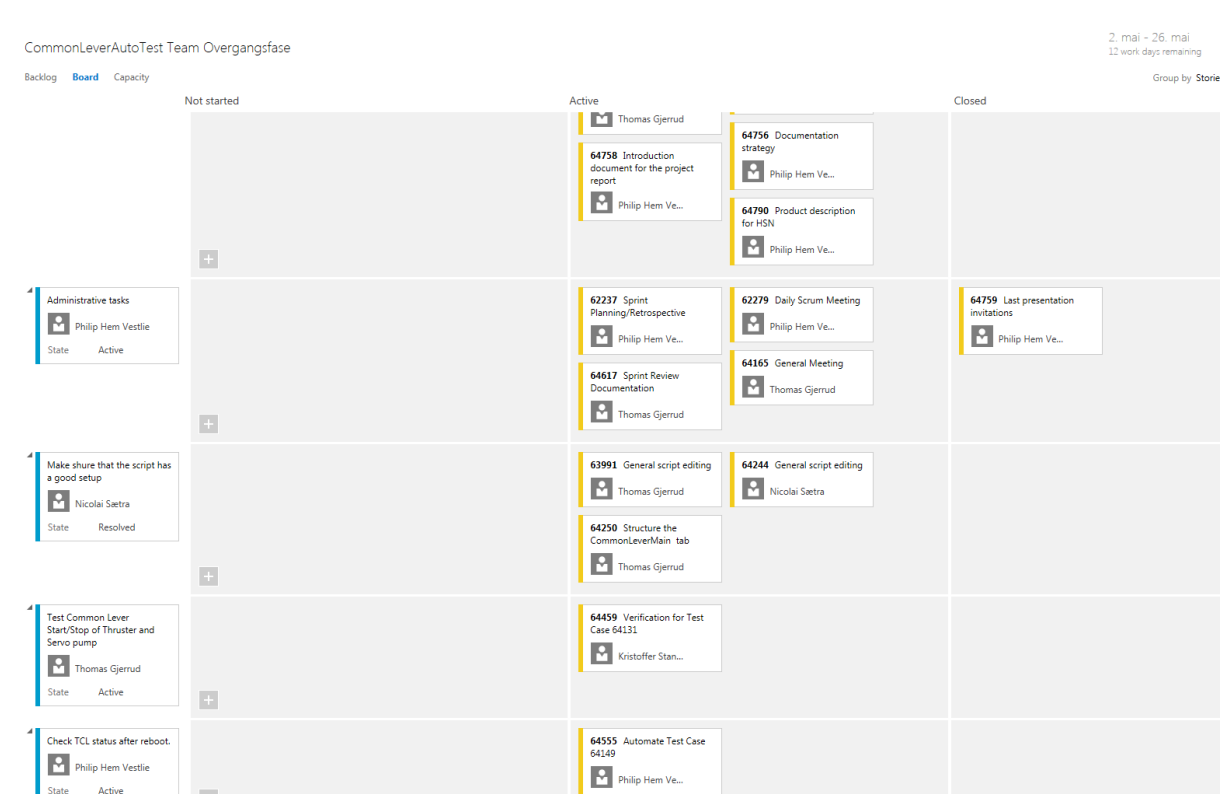
3.1.2 TFS nettbasert versjon

Den nettbaserte versjonen tilbyr blant annet funksjoner som «backlog» og «board», hvor man enkelt kan holde oversikt framdriften i prosjektet.

US og tasks kan ha tre forskjellige tilstander i dette prosjektet:

- Ikke startet (Opprettet men ikke begynt på)
- Aktiv (Startet på og inngår i en sprint)
- Avsluttet (Ferdig i dette prosjektet)

Figur 3 viser et utklipp av board med US og tasks, figur 4 viser et utdrag av backloggen i prosjektet.



Figur 3: Utdrag av board i TFS

Stories

Backlog Board

New

🔍

📄

Create query

Column options

✉

Type

User Story

✕

Title

As a <type of user> I want <some goal> so that <some reason>

Add

Order	Work Item Type	Title	State	Iteration Path	Assigned To	Changed By
1	User Story	MoveLever	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Kristoffer Stanger
2	User Story	▶ Test Common Lever rotation command	Active	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Build C20
3	User Story	▶ Test Common Lever day palette	Active	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Build C20
+ 4	User Story	▶ Check Common Lever screen lock function	Not Started	CommonLeverAutoTest\Core\Utv...		Thomas Gjerrud
5	User Story	▶ Test Common Lever pitch command	Active	CommonLeverAutoTest\Core\Utv...	Thomas Gjerrud	Build C20
6	User Story	▶ Test Common Lever dusk palette	Active	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Kristoffer Stanger
7	User Story	▶ Test Common Lever night palette	Active	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Kristoffer Stanger
8	User Story	▶ Check Common Lever password protection	Active	CommonLeverAutoTest\Core\Utv...	Philip Hem Vestlie	Build C20
9	User Story	▶ Check pushed button indication on Common Lever	Not Started	CommonLeverAutoTest\Core\Utv...		Thomas Gjerrud
10	User Story	▶ Check returning to home meny function	Active	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Nicolai Sætra
11	User Story	▶ Check Common Lever light sensor	Active	CommonLeverAutoTest\Core\Utv...	Thomas Gjerrud	Thomas Gjerrud
12	User Story	▶ Training and rigging equipment	Active	CommonLeverAutoTest\Core\Utv...	Philip Hem Vestlie	Thomas Gjerrud
13	User Story	▶ Test dimming from internal sensor Common Lever	Active	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Thomas Gjerrud
14	User Story	▶ Test dimming from external sensor Common Lever	Active	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Thomas Gjerrud
15	User Story	▶ Prepare for project presentations	Active	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Thomas Gjerrud
16	User Story	▶ Produce documentation for the project report	Active	CommonLeverAutoTest\Core\Ove...	Thomas Gjerrud	Thomas Gjerrud
17	User Story	▶ Administrative tasks	Active	CommonLeverAutoTest\Core\Ove...	Philip Hem Vestlie	Thomas Gjerrud
18	User Story	▶ Test of Lever	Active	CommonLeverAutoTest\Core\Utv...	Philip Hem Vestlie	Kristoffer Stanger
19	Bug	Configuration	Resolved	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Kristoffer Stanger
20	Bug	Take command	Resolved	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Kristoffer Stanger
21	Bug	"Sound off" button	Resolved	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Kristoffer Stanger
22	Bug	Host communication	Resolved	CommonLeverAutoTest\Core\Utv...	Kristoffer Stanger	Philip Hem Vestlie
23	Bug	SCU Crash	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Nicolai Sætra
24	Bug	TCL handle config fail	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Nicolai Sætra
25	Bug	TCL dimming slider adjust	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Nicolai Sætra
26	Bug	TCL GUI-page status	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Nicolai Sætra
27	Bug	TCL GUI-page change	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Kristoffer Stanger
28	Bug	TCL Ambient light manipulation	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Nicolai Sætra
29	Bug	TCL Touch lock	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Thomas Gjerrud
30	Bug	TCL aft.bridge landscape option	Resolved	CommonLeverAutoTest\Core\Utv...	Nicolai Sætra	Kristoffer Stanger

Figur 4: Utdrag av backlog i TFS

3.2 TestComplete

AutoTesten scriptes i programmet TestComplete av SmartBear som er beskrevet i kapittel 5 i den tekniske dokumentasjonen. Når scriptet er endret eller ferdigstilt sjekkes det inn mot TFS og linkes mot den aktuelle US. Alle endringer blir på denne måten sporbart i en changeset liste tilgjengelig via US. Som vist i figur 5, kan man ved hjelp av changeset se hvilke endringer som er gjort i de forskjellige versjonene av scriptet.

```

230 EnterButton();
231 Delay(500);
232 VerifyPassword("Invalid\r\npassword");
233 //Delay(500);
234 //GoToMain();
235 //Delay(200);
236 //PasswordMenuButton();
237 //Delay(200);
238 //Button1();
239 //Button2();
240 //Button3();
241 //Button4();
242 //var value = TypedPassword(); //dette lagrer passord til bruk i verifis
243 //EnterButton();
244 //VerifyPasswordProtection(value);
245 //DisableSpeakerButton();
246
247 //VersionInfoButton();
248
249 //CloseVersionPageButton();
250 //DeviceStatusButton();
251
252 //CloseDevicePageButton();
253 //HardwareTestButton();
254
255 //CloseHardwarePageButton();
256 //GoToMainButton();

```

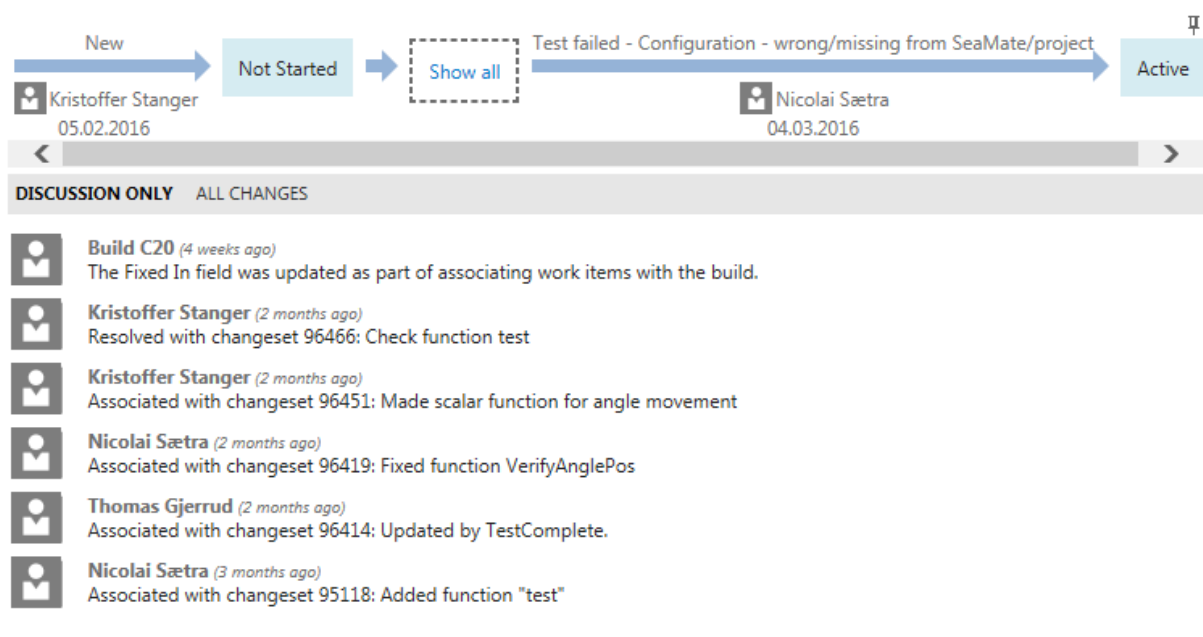
```

230 EnterButton();
231 Delay(500);
232 VerifyPassword("Invalid\r\npassword");
233 Delay(500);
234 GoToMain();
235 Delay(200);
236 PasswordMenuButton();
237 Delay(200);
238 Button1();
239 Button2();
240 Button3();
241 Button4();
242 var value = TypedPassword(); //dette lagrer passord til
243 EnterButton();
244 VerifyPasswordProtection(value);

```

Figur 5: Sammenligning av script i changeset

Figur 6 viser hvordan changeset fremkommer som en liste under fanen «detaljer» i US.



New → Not Started → Show all → Test failed - Configuration - wrong/missing from SeaMate/project → Active

Kristoffer Stanger 05.02.2016 Nicolai Sætra 04.03.2016

< >

DISCUSSION ONLY ALL CHANGES

- Build C20** (4 weeks ago)
The Fixed In field was updated as part of associating work items with the build.
- Kristoffer Stanger** (2 months ago)
Resolved with changeset 96466: Check function test
- Kristoffer Stanger** (2 months ago)
Associated with changeset 96451: Made scalar function for angle movement
- Nicolai Sætra** (2 months ago)
Associated with changeset 96419: Fixed function VerifyAnglePos
- Thomas Gjerrud** (2 months ago)
Associated with changeset 96414: Updated by TestComplete.
- Nicolai Sætra** (3 months ago)
Associated with changeset 95118: Added function "test"

Figur 6: Liste med changeset

3.3 TFS linket mot Microsoft Test Manager

TFS kan linkes mot programmet MTM og har en funksjon som de automatiserte testene kan kjøres fra. KM benytter MTM til å sammenstille de og kjøre de automatiske testene. Gruppen har kun mulighet til å kjøre AutoTesten fra TestComplete, men skal sette opp scriptet på en slik måte at det er enkelt for KM å implementere og kjøre testen i MTM. Gruppen benytter derfor ikke MTM direkte, men TC er opprettet i MTM som er linket mot TFS, hvor TC hentes fra. Figur 7 viser test suiteen i TFS hvor test casene kan kjøres fra under forutsetning at de er linket mot MTM.

Test suite: Common Lever AutoTest (Suite ID: 61667)

Tests Charts

Run

Outcome	ID	Title	Configurat...	Tester
Active	61898	Common Lever Screen lock function	Windows...	
Active	61899	Common lever Day palette	Windows...	
Active	61900	Common lever dusk palette	Windows...	
Active	61901	Common lever night palette	Windows...	
Active	61902	Common Lever password protection	Windows...	
Active	61904	Common Lever button bacground color change	Windows...	
Active	61905	Common Lever blinking button frame	Windows...	
Active	61909	Common Lever pushed button indication	Windows...	
Active	61910	Common Lever button in waiting for confirmation state	Windows...	
Active	61913	Common Lever Service menu returning to Home page	Windows...	
Active	61914	Common Lever Light sensor	Windows...	
Active	61916	Common Lever Dimming from internal sensor	Windows...	
Active	61919	Common Lever Dimming from external signal	Windows...	
Active	61921	Common Lever button blinking speed	Windows...	
Active	61923	Common Lever button color	Windows...	
Active	61924	Common Lever sound	Windows...	
Active	62079	Common lever pitch command	Windows...	
Active	62083	Common Lever rotation Command	Windows...	
Active	64131	Common Lever Start/Stop of Thruster/Servo pump	Windows...	
Active	64149	Common Lever status after reboot	Windows...	

Figur 7: Test suite i TFS

4 Bugs

Når det oppdages feil eller mangler ved testing skal det opprettes bugs i TFS. Bugs er i likhet med US også et WI i TFS og er søkbart enten på ID eller ved å opprette en spørring i tilknytning til prosjektet det jobbes i.

En bug kan inneholde alt fra hardware til software feil, og skal rapporteres til rette vedkommende slik at det blir rettet opp. Når bugen er opprettet settes status som «active». Dersom feilen blir utbedret, endres status til «resolved». I tilfeller hvor bugen som er rapportert ikke er feil eller lar seg utbedre, settes status til «closed»

CommonLeverAutoTest [Results]					
Save Results Save Query Open in Microsoft Office Edit Query Column Options					
Query Results: 157 items found (1 currently selected). The query has been modified.					
ID	Work Item Type	Title	Assigned ...	State	
63876	Bug	SCU Crash	Nicolai Sæ...	Resolved	
63998	Bug	The standalone TCL touch screen are to small	Nicolai Sæ...	Closed	
63999	Bug	Dimming the light with sensor fails	Nicolai Sæ...	Closed	
64013	Bug	TCL play alarm sound	Nicolai Sæ...	Closed	
64134	Bug	TCL password menu	Nicolai Sæ...	Resolved	
64202	Bug	Take command pitch	Philip Hem...	Resolved	
64168	Bug	When pressing button on lever	Kristoffer S...	Active	
64142	Bug	Parameters for indicators	Kristoffer S...	Resolved	
64246	Bug	Text in GUI	Kristoffer S...	Closed	
64273	Bug	TCL touchlock	Thomas Gj...	Closed	
64276	Bug	Buttons in main page flashing mode	Thomas Gj...	Closed	
64277	Bug	Flashing frame feedback	Thomas Gj...	Closed	
64278	Bug	Read button status	Thomas Gj...	Resolved	
64295	Bug	Reboot command	Philip Hem...	Resolved	
64297	Bug	Button confirmation state function	Philip Hem...	Active	
64299	Bug	Return to homepage after a preset time	Philip Hem...	Closed	
64372	Bug	Full memory in lever while running Autotest	Philip Hem...	Resolved	
64424	Bug	Sending command to Common lever pitch	Kristoffer S...	Active	
64483	Bug	While running Autotest TCL gives Application error	Philip Hem...	Resolved	
64572	Bug	Motor controller pitch	Kristoffer S...	Active	
64576	Bug	Motor controller Azimuth	Kristoffer S...	Active	
64520	Bug	Internal TCL light sensors gives different value in the same light conditions	Philip Hem...	Closed	

Figur 8: Utdrag av liste over bugs i TFS

5 Referanser:

[1] Om TFS online: <https://www.visualstudio.com/en-us/products/tfs-overview-vs.aspx>
(sist besøkt 11.05.2016)



Teknisk dokument

Versjon 1.0



Common Lever Auto-Test

Bacheloroppgave for ingeniørstudie ved Høgskolen i Sørøst-Norge, avdeling Kongsberg, fakultet for teknologiske og maritime fag.

Dato: 23.05.2016

Oppdragsgiver: Kongsberg Maritime Merchant Marine Horten

Prosjektdeltagere: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Intern veileder: Antonio L.L. Ramos

Ekstern veileder: Magne Røed

Dokumenthistorie

Versjon	Dato	Endringer	Ansvarlig
1.0	18.05.2016	Første versjon	PV

Innholdsfortegnelse

1	Introduksjon.....	132
2	Automatisert og manuell testing i system	132
2.1	Automatisert testing	132
2.2	Manuell testing	133
2.3	Komponenter og feedback	134
2.4	Verifisering av AutoTest	136
3	Realtimesystemet	136
3.1	K-chief 600	136
3.2	Operator Station (OS)	137
3.3	Distributed Processing Units (DPU)	137
3.4	Segment Controller Unit (SCU)	137
3.5	Main Engine Interface (MEI)	138
4	Hendlene	139
4.1	Eksisterende hendler	139
4.2	Nye hendelen, Common Lever	139
4.3	Common Lever azimuth typer	140
4.4	Programmer til kommunikasjon mot hendlene.....	140
4.5	Hendel hardware	141
4.6	Hendel software	141
4.7	Kommunikasjon med hendlene	142
5	Script av AutoTest	143
5.1	TestComplete	143
5.2	JScript kodespråk	143
5.3	Oppbygging og struktur av koden	144
5.4	AutoTesten.....	145
6	Testtrigg	146
6.1	Hardware sammenstilling	146

6.1.1	Skjermen	147
6.1.2	Montasjeplate 1	147
6.1.3	Montasjeplate 2	147
6.1.4	Montasjeplate for hendler og indikatorer	148
6.1.5	Mekaniske tegninger for komponenter	149
6.1.6	Oppkobling av utstyr	151
6.1.7	Verktøy	152
6.1.8	Koblingsdiagram	152
7	Referanser	153

Figurliste

Figur 1: Rele brukt til start/stop/running signal	135
Figur 2: Lokalt og Globalt CAN nettverk	138
Figur 3: MEI.....	139
Figur 4: Hendelkomponenter.....	141
Figur 5: Kommunikasjon mellom hendel og PC.....	142
Figur 6: Mappestruktur i TestComplete	144
Figur 7: Brokonsoll som er brukt til testtrigg.....	146
Figur 8: OS skjerm	147
Figur 9: Montasjeplate 1	147
Figur 10: Montasjeplate 2	147
Figur 11: Azimuth CANInstrument cutout	149
Figur 12: Pitch CAN Instrument cutout	150
Figur 13: Common Lever cutout.....	150
Figur 14: Common Lever ekstern skjerm utkapp.....	151
Figur 15: CAN koblingsskjema.....	152

Tabelloversikt

Tabell 1: Forklaring av forkortelser	131
Tabell 2: I/O Liste over koblede kanaler på DPU	135

Tabell 1: Forklaring av forkortelser

Forkortelse	Forklaring
HSN	Høgskolen i Sørøst-Norge, avd. Kongsberg, Fakultet for teknologi og maritime fag
KM Carpus	Kongsberg Maritime i Kongsberg
KM Merchant Marine	Kongsberg Maritime i Horten
HW	Hardware
SW	Software
TC	Test case
OS	Operator station
TFS	Team Foundation Server
MTM	Microsoft Test Manager
PMS	Power Management System
I/O	Inngang/Utgang
WS	Work Station
SCU	Segment Controller Unit
DPU	Distributed Processing Units
MEI	Main Engine Interface
WI	Work Items
US	User Story
HP	Hydraulikkpumpe

1 Introduksjon

Dette dokumentet tar for seg det tekniske produktet i oppgaven. Dokumentet inneholder utdypinger og forklaringer på de tekniske bidragene fra medlemmene i bachelorgruppen.

Først forklares hva automatisert og manuell testing går ut på og hvordan det er gjennomført i prosjektet, med komponenter og feedback. Det tar også for seg hvordan AutoTesten er verifisert som testverktøy. Videre forklares realtimesystemet som hendlene får feedback fra, med hva det består av og hvordan systemet virker. Deretter følger utdyping og forklaring på hendlene. Her forklares de testede hendelstypene, deres oppbygging og virkemåte. Så forklares det hvordan AutoTesten er utviklet. Til sist beskrives testtriggen som hendlene og realtimesystemet er montert i, for å simulere det miljøet de skal operere i.

Hensikten med dokumentet er å gi en forståelse av AutoTesten, med hvordan system hendlene er testet i, hvordan testingen er gjennomført og hvordan hendlene virker.

2 Automatisert og manuell testing i system

Dette kapitlet tar for seg hva automatisert og manuell testing er. Det er forklart hvordan vi har gjennomført disse prosessene, med komponenter og feedback fra systemet. Kapitlet tar også for seg hvordan vi har verifisert AutoTesten.

2.1 Automatisert testing

Hensikten med automatisering av test er å effektivisere og forbedre testprosessen til et produkt. En automatisk test kan teste mye på kort tid i forhold til hva en person kan klare. I tillegg gir automatisk test mulighet til å avdekke mer dyptliggende feil, mangler og svakheter i hardware (HW) og software (SW) til hendlene, noe som er tilnærmet umulig med manuell testing.

Automatisering av test utføres ved at test caser (TC) er gitt som krav fra KM, som brukes for å bestemme kriteriene til gjennomføring av den automatiserte testen. TC inneholder også akseptkriteriene for hva den automatiske testen skal godkjenne.

Den automatiske testen er produsert i TestComplete. Her automatiseres hele testprosessen, med SW kode som gjør alle de manuelle operasjonene automatisk. For å utvikle den automatiserte testen er funksjonene først kontrollert manuelt på hendlene. Deretter er Putty brukt som terminal (se kapittel 4.4), for å se hvilke kommandoer som er tilgjengelig i testgrensesnittet i SW til hendlene. AutoTesten blir så bygd opp av funksjoner i TestComplete med tilgjengelige kommandoer i hendlene. En nærmere beskrivelse av TestComplete er beskrevet i kapittel 5.1.

Vi har utviklet den automatiske testen i utviklingsfasen av prosjektet til KM. Det har gitt oss muligheten til å gi utviklerne tilbakemeldinger på hvordan SW fungerer mot hendlene underveis. Det har gjort det mulig å kjøre automatisk test på hendlene samtidig som en SW-utvikler også er koblet til hendlene, og kan observere hvordan hendlene oppfører seg under testing i det øyeblikket feilen inntreffer. Dette samarbeidet har vist seg å være meget nyttig for KM slik at de kan levere et feilfritt og gjennomprøvd produkt.

2.2 Manuell testing

I et utviklingsprosjekt er det ønskelig å finne de feil og mangler produktet måtte ha. Manuell testing gjøres på prototyper for å avdekke svakheter i utviklingsfasen. KM oppretter TC på hvordan disse prototypene skal testes og hvilket resultat som er ventet. Er testene underkjent, blir de rapportert som bugs i Team Foundation Server (TFS).

Manuell testing består av lasting av konfigurasjon, DPU software og skifte av Node ID. Det sjekkes også at all funksjonalitet fungerer. Når alle nødvendige filer er lastet til systemet og ser ut til å fungere som det skal, må det testes at alle knapper og funksjoner i hendelen fungerer som spesifisert i test casene. Den manuelle testingen medfører rapportering av feil og mangler av produktet til SW og HW utviklerne, og er tidkrevende.

Vanligvis utfører KM manuell testing på produkter i utviklingsfasen, og automatisert test utvikles etter at produktene er tatt i bruk. I Common Lever prosjektet har derimot manuell testing og automatisering av TC blitt gjort parallelt. KM presiserer at de har sett verdien av å utføre manuell og automatisk testing parallelt i utviklingsfasen, etter å ha veiledet oss gjennom dette prosjektet. KM har sett at ved å begynne automatiseringen i denne fasen, oppdages mer dyptliggende feil kontra hva manuell testing klarer å avdekke.

2.3 Komponenter og feedback

For å simulere feedbacker fra thrustersystemet har vi benyttet releer for å simulere start og stopp på thruster motorer og pumper, brytere for å simulere digitale innganger, potensiometere for å simulere analoge innganger og utganger.

Når hendlene er koblet i et system, krever de at bestemte I/O er koblet til på MEI som closed loop:

- RPM potentiometer
- Load potentiometer
- Vinkel feedback
- Pitch feedback
- Local/Remote brytere for thruster og hydraulikkpumpe (HP).
- Start thruster
- Stopp thruster
- Running signal thruster
- Start HP
- Stopp HP
- Running signal HP

Systemet består i tillegg til dette av interlokker. Interlokker er betingelser som må tilfredsstilles for at funksjoner i hendelen skal fungere.

- Det skal ikke være mulig å starte/stoppe thruster fra en hendel som ikke er i kommando.
- Ved kommando skifte mellom hendler, skal den «nye» hendelen finne samme posisjon som kommando hendelen.
- Når thruster skal startes må HP starte først.

Potentiometre er koblet til for å simulere 4-20mA signal for turtall og belastning på motoren. Når hendlene styrer azimuth (rotasjon) eller pitch (pådrag) på thrusteren, sendes det 4-20mA signaler til en lokalt plassert motorkontroller. Denne sender feedback til MEI modulen og gir beskjed om den faktiske posisjonen til thrusteren. Signalet som sendes fra hendlene, har vi simulert ved å koble direkte fra utgang til inngang på MEI. I et full skala system ville thrusteren brukt tid på å oppnå ønsket posisjon, det har vi løst med å sette en forsinkelse på signalet i OS. Hensikten med forsinkelsen er at feedbacken fra den simulerte thrusteren skal være mest mulig realistisk.

MEI modulene har en inngang hver som forteller systemet om thruster eller hydraulikkpumpe blir styrt lokalt. «Local» er definert som standard ved manglende signal. Brytere er koblet på disse inngangene for å gi systemet et remote signal og gi tilgang til å starte/stoppe thrusteren fra hendelen.

Releer er benyttet for å indikere start/stopp og running signal, til både thruster og hydraulikkpumpe. Disse er nødvendig for å simulere at motor ogpumpe har startet.
For å inidikere start/stopp og running for thruster er det benyttet tre stykk releer som en holdekrets (figur 1).
For hydraulikkpumpe er det benyttet et pumpekort fra KM, med samme funksjon som releene til thruster.



Figur 1: Rele brukt til start/stop/running signal

Liste over kanaler som er koblet på MEI 0520 og 0521[1]:

Tabell 2: I/O Liste over koblete kanaler på DPU

DPU Type: MEI_XL				
DPU Nr	TagName	TagDescription	SignalType	Channel
0520	T3 START	T3 VFD START	Digital Output	1
0520	T3 STOP	T3 VFD STOP	Digital Output	2
0520	T3 R SETP	T3 RPM CMD TO VFD	4-20mA Out	3
0520	T3 RUN	T3 MOTOR RUNNING	Digital Input	16
0520	T3 VFD LOC	VFD REMOTE CONTROL	Digital Input	19
0520	T3 R FEEDB	T3 MOTOR RPM	4-20mA	29
0520	T3 LOAD	T3 MOTOR LOAD	4-20mA	30
0520	T3 A FEEDB2	T3 ANGLE FEEDBACK 2	4-20mA	31
0521	T3 A SETP	T3 STEERING COMMAND	4-20mA out	3
0521	T3 A PMP STA	T3 HYDRAULIC PUMP START	Digital Output	6
0521	T3 A PMP STO	T3 HYDRAULIC PUMP STOP	Digital Output	7
0521	T3 A PMP LOC	T3 HYDRAULIC PUMP REMOTE	Digital Input	22
0521	T3 A PMP RUN	T3 HYDRAULIC PUMP RUNNING	Digital Input	23
0521	T3 R FEEDB2	T3 MOTOR RPM	4-20mA	30
0521	T3 A FEEDB1	T3 ANGLE FEEDBACK 1	4-20mA	31

2.4 Verifisering av AutoTest

Verifisering er en meget viktig del av vår kvalitetssikring av arbeidet med AutoTesten. Uten dette kan vi ikke med sikkerhet si at AutoTesten gjør som den skal. Når vi kjører AutoTesten til Common Lever, sender vi kommandoer, leser av statuser og verdier på feedback fra systemet. I AutoTesten er TC delt opp i funksjoner, hvor kommandoer blir sendt og feedbacker lest. TestComplete lager en logg, der det blir indikert om testen er underkjent eller godkjent ut ifra akseptkriteriene fra TC. For at vi skal kunne godkjenne resultatet til AutoTesten må den verifiseres, slik at vi kan bevise at AutoTesten feiler ved en reell feil i feedbacken fra hendelen.

VerificationFunctions:

For å verifisere AutoTest av Common Lever, har vi utviklet et eget script for denne oppgaven. Her har vi manipulert verdiene som sendes til hendlene slik at vi simulerer en feil for å se at testen feiler innenfor de krav og retningslinjene som er gitt fra KM. Ved å sende en annen verdi enn den verdien som AutoTesten forventer kan vi se at testen feiler og at marginer som er satt i koden fungerer som de skal. Scriptet til verifisering av AutoTesten ligger som vedlegg 2 i rapporten.

3 Realtimesystemet

Dette kapitlet tar for seg thruster-realtimesystemet som hendlene er koblet mot, med forklaring på hva systemet består av og hvilke komponenter som er benyttet.

3.1 K-chief 600

K-chief 600 er et automatisert system for marine fartøyer som krever standardisert alarm-, overvåking- og kontrollsystem. Systemet er en videreutvikling av K-chief 500. K-chief 600 kan bestå av alt fra enkle alarmsystemer til avanserte kontroll- og overvåkningssystemer. Det er modulært og kan konfigureres fra 16 til 20 000 Kanaler. [2]

Typiske delsystemer er:

- Hjelpeskontroll
- Strømstyring
- Fremdriftskontroll
- Ballastautomasjon
- Lastkontroll og overvåkning
- HVAC kontroll (Klimakontroll)
- Brann

Hendlene er en del av et fremdriftskontrollsystem, og skal kunne integreres i K-chief 600 og overvåkes derifra med feedbacker og alarmer.

3.2 Operator Station (OS)

Operatørstasjon er en PC og et grensesnitt mellom systemet og bruker. Den brukes for å motta alarmer, overvåking og kontroll av systemene. Her vises interaktive prosessdiagrammer og logger kan lagres. Operatørstasjonen kommuniserer med DPU og kan forandre variabler og sette parametere for inn- og utganger. Det er lagd en mimic for en mer visuell fremstilling av hva som blir overvåket og feedbackene i systemet. [2]

3.3 Distributed Processing Units (DPU)

DPU er et samlebegrep for de forskjellige modulene KM bruker. Dette er I/O (input/output) moduler med tag-basert prosesseringsevne som er meget solide og stabile. KM har flere typer DPU, men i Common Lever prosjektet forholder vi oss til følgende typer DPU:

- Segment Controller Unit (SCU)
- Main Engine Interface (MEI)

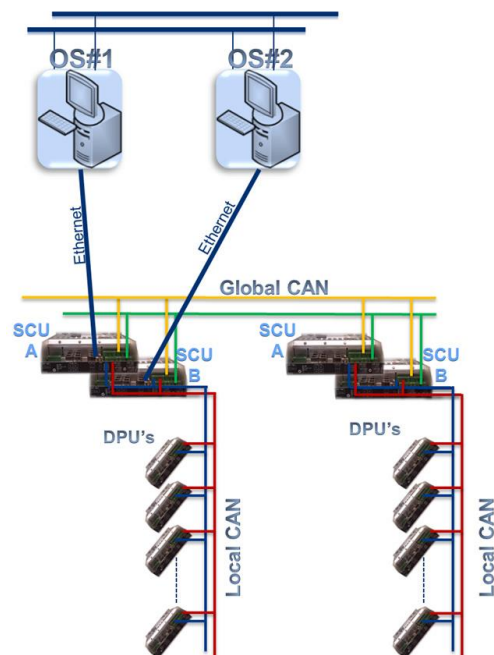
3.4 Segment Controller Unit (SCU)

Dette er en modul som kan styre et lokalt Controller Area Network (CAN) segment. Flere DPU kan kobles sammen og kommunisere på dette lokale CAN segmentet. Flere segmenter kan kobles sammen på et globalt CAN nettverk. Operator Station (OS) kommuniserer med SCU via standard Ethernet. Systemet gjøres redundant ved å koble to SCU i parallell, hvor den ene er en master SCU og den andre er i stand-by. Begge tilkobles både det lokale og det globale CAN nettverket. Det er master SCU som er ansvarlig for det lokale CAN nettverket og kommunikasjon mot OS. Dersom master SCU feiler, vil back-up SCU overta styringen. [3]

Se figur 2 for generell oppkobling av systemet.

SCU har blant annet følgende funksjoner:

- Routing av real-time data og service data
- Alarmsystem med tidsstempling
- Programmerbar Logisk Kontroller
- Forhåndsprogrammert logikk og fjernstyring av skipets tilleggsutstyr



Figur 2: Lokalt og Globalt CAN nettverk

3.5 Main Engine Interface (MEI)

Dette er en motorstyringsmodul med flere typer digitale og analoge innganger og utganger (figur 3). Alle inngangene kan brukes til å aktivere utgangene, hvorpå verdiene rapporteres via CAN. Modulen har dobbel CAN-bus og LED indikasjon for strøm, status, watchdog og CAN kommunikasjon. [4]

- Kanal 1 og 2 er releutganger med maks induktiv last på 3A ved 230V.
- Kanal 3 og 4 er analoge utganger med 4 - 20 mA med maks last på 550Ω ved 10V.
- Kanal 5 til 14 er solenoid utganger med 18 - 32V med maks 500mA. Disse utgangene kommer rett fra strømforsyningen.
- Kanal 15 til 28 er digitale innganger.
- Kanal 29 til 36 kan velges som digitale eller analoge innganger. 4 – 20 mA.
- Kanal 34 til 36 kan velges som digitale eller potensiometer innganger.

MEI har blant annet følgende funksjoner:

- Kanalene som nevnt ovenfor
- Linjesjekkfunksjon på 14 kanaler
- Alarm og overvåkning på alle kanaler
- Tidsstempling på alarmer og hendelser
- Automatisk håndtering av feil på CAN-bus



Figur 3: MEI

4 Hendlene

Dette kapitlet beskriver azimuth hendlene som vi har utviklet AutoTesten for. Det forklarer hva de består av og hvordan de kommuniserer med realtimesystemet.

4.1 Eksisterende hendler

Dagens løsning har forskjellige hendler til samme oppgave for K-Thrust 600 som KM Horten bruker og K-Thrust 720 som KM Kongsberg benytter.

Azimuth hendelen LF70 i K-Thrust 600 er større og krever mere fysisk plass.

I LF70 sitter kun motorer og sensorer på hendelen, kretskort og logikk ligger i en egen enhet som er koblet til hendelen som igjen er koblet til SCU. LF70 hendelen er et tredjeparts produkt som kjøpes inn av KM. [5] [6]

4.2 Nye hendelen, Common Lever

Common Lever er et samarbeidsprosjekt mellom KM Horten og Kongsberg for å standardisere hendelen slik at den kan benyttes i både K-Thrust 600 og K-Thrust 720. De nye hendlene har ikke logikken innebygd som i LF70. De benytter en ny teknologi hvor logikken ligger i SCU og MEI som blir kommunisert til hendlene via CAN. Common Lever har kretskortene innebygd. Den er utviklet av og produsert for KM, er besparende i innkjøp og gir bedre produktkontroll. [7]

4.3 Common Lever azimuth typer

Det er to forskjellige azimuth hendeltyper. Den ene har integrert berøringsskjerm, og den andre har ekstern berøringsskjerm. Begge to har samme størrelse, men den eksterne skjermen er da ikke medregnet.

Azimuth leveren består av to børsteløse DC-motorer og to absolutt encodere for å måle posisjon på motoren til rotasjonen (azimuth) og vinkel til motoren for pådrag(thrust/pitch). Den ene DC-motoren sitter i boksen sammen med kretskortene for å styre azimuth bevegelsen (360 grader) til hendelen og den andre DC-motoren sitter i selve hendelenheten for å styre thrust bevegelsen til hendelen. Absolutt encoderne sitter da sammen med hver sin DC-motor. Det er dobbelt sett med sensorer. [8]

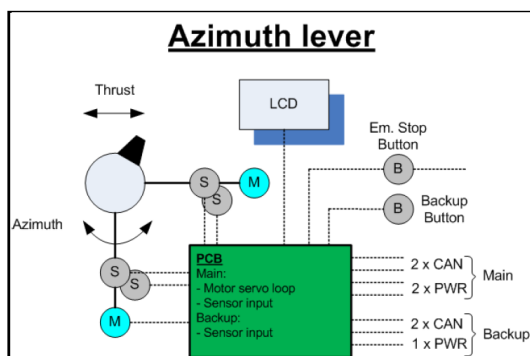
4.4 Programmer til kommunikasjon mot hendlene

- Putty benyttes som en terminal for å kunne sende kommandoer og lese ut av hendelen hvilke funksjoner som ligger tilgjengelig. Ved å koble opp hendelen mot PC og kjøre Putty, vil hendelen svare «Test server ready», som en velkomstmelding for å fortelle at kommunikasjonen mellom hendelen og Putty er opprettet. Vi kan da sende «gui objects» for å spørre hvilke funksjoner som er tilgjengelig i det bildet berøringsskjermen står i. [9]
- Testprogrammet TCL HW TEST er et program som benyttes av utviklere og service ingeniører for å teste HW og SW på hendelen. Det ga oss mulighet til å bruke hendelen og teste funksjoner uten å ha databasen i SCU. Konfigurasjon, I/O og standard verdier er kodet inn. [10]
- Realtimesystemet som komplett thrustersystem består av SCU som har logikken og databasen til hendelen, MEI med analoge og digitale inn- og utganger. Dette brukes for å simulere feedbacker fra systemet. OS brukes for å lese av feedbacker og for å kunne sette parametere for inn- og utganger. Realtimesystemet er dypere forklart i kapittel 2.2.

4.5 Hendel hardware

Hendlene består av HW som er beskrevet i denne listen og illustrert med figur 4. [8]

- Motor i hendelen:
Børsteløs DC-motor
- Posisjons sensor:
Absolutt encoder
- Nødstopp:
Egen sløyfe dirket til motoren for å kutte motordriften i nødstilfeller (hard wired).
- Backup-knapp:
Deaktiverer motorstyringen hvis noe er galt og da vil backup unit ta over som leser sensoren fra en annen giver. Dette benytter KM Kongsberg seg av på grunn av høyere klassekrav til deres bruk.
- LED
Indikerer forover (ahead), bakover (astern) og nullpunkt på hendelen.
- Lys sensor:
Måler lysmengden for å vurdere hvilken modus skjermen skal stå i; day, dusk eller night.
- LCD touch screen. Gir visuell fremstilling settpunkter, feedbacker og alarmer.
- Kretskort.
Hovedkort: Dette kortet benytter både KM Horten og Kongsberg seg av.
Backup-kort: Dette benytter KM Kongsberg seg av på grunn av høyere klassekrav til deres bruk.



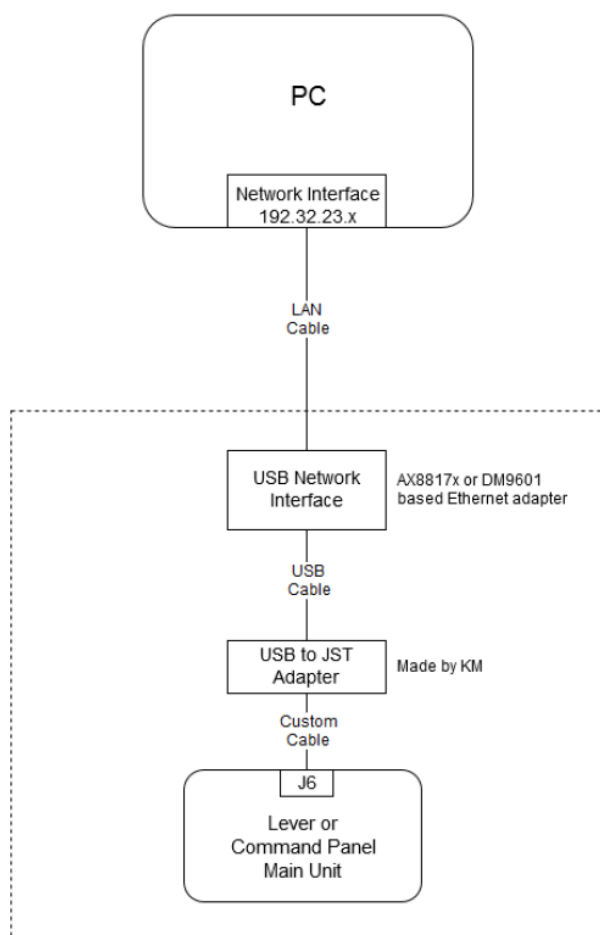
Figur 4: Hendelkomponenter

4.6 Hendel software

Består av en Linux-kjerne sammen med spesiallaget programvare som er utviklet av KM. Hendlene består av tre applikasjoner; TclManager, TclGui og TclServer. Testgrensesnittet vi har benyttet for å utvikle AutoTesten er tilgjengelig som kommandoer i TclServer. [10]

4.7 Kommunikasjon med hendlene

Kommunikasjon mellom PC og hendlene er som beskrevet i figur 5. Hendlene kommuniserer med andre moduler på CAN. [10]



Figur 5: Kommunikasjon mellom hendel og PC

5 Script av AutoTest

Dette kapitlet tar for seg hvordan vi har utviklet AutoTesten til Common Lever. Her forklares programmet vi har produsert AutoTesten i, hvordan vi har bygd opp og strukturert koden og AutoTesten som produkt.

5.1 TestComplete

TestComplete er ett verktøy laget av Smartbear for testing av software for forskjellige plattformer. Programmet støtter forskjellige typer kodespråk: Python, VBScript, JScript, DelphiScript, C++Script, og C#Script [11]. Fordelen med at TestComplete støtter flere forskjellige kodespråk, er at de fleste softwareutviklere og de som er kjent med datakoding fra før kan et av disse språkene.

TestComplete kan brukes sammen med programmer som f.eks. Team Foundation Server (TFS), slik at man hele tiden har mulighet for å lagre endringer som er gjort. I tillegg til dette, har man mulighet for å se endringer i koden og eksakt hvor de er gjort. En slik sporbarhet og oversikt er nyttig i ett prosjekt hvor det jobber flere utviklere.

Vi har brukt TestComplete sammen med TFS under utviklingen av AutoTesten. Som ett krav fra KM har vi brukt Jscript som scriptspråk[12].

5.2 JScript kodespråk

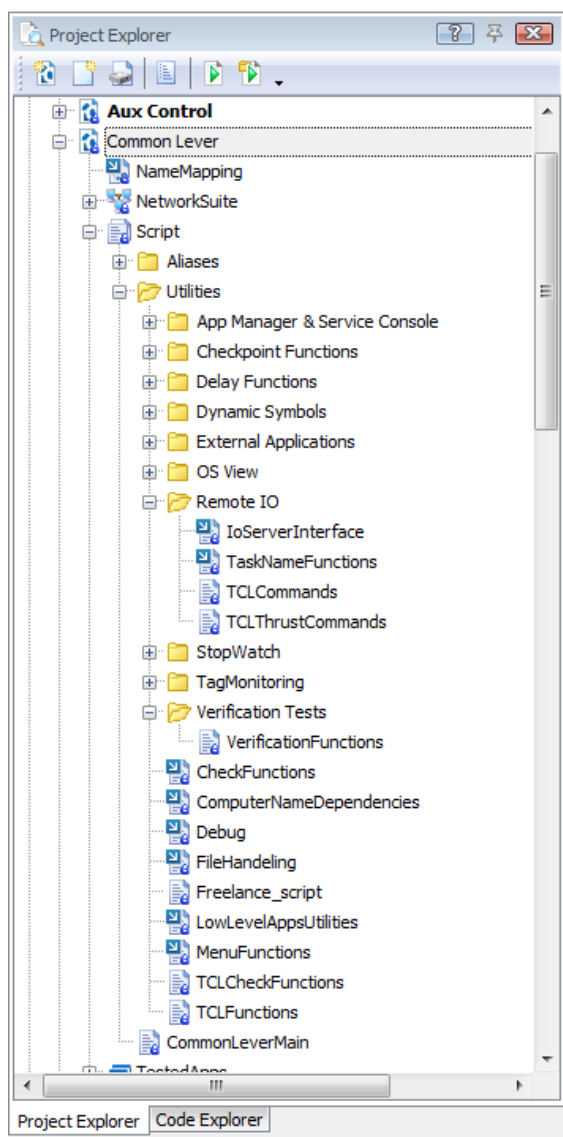
Jscript er utviklet av Microsoft og er identisk med kodespråket JavaScript som er utviklet av Sun Microsystems. Grunnen til at Microsoft valgte å kalle det Jscript, var for å slippe trademark problemer[13].

Jscript er et «High-level» programmeringsspråk, noe som gjør at man kan lage programmer som mer eller mindre uavhengige av hva slags datamaskin de skal brukes på. Det blir sett på som «High-level» språk fordi de er nærmere det menneskelige språk en maskinspråk[14].

Det er flere fordeler med bruk av Jscript, det er forståelig for oss som brukere, samt at det er oversiktlig å lese/skrive funksjoner/variabler. Ved definisjon av variabler i Jscript, trenger man ikke å definere om variablene skal være integer, number, string etc. det gjenkjennes automatisk av programmet[15].

5.3 Oppbygging og struktur av koden

Mappestrukturen som brukes er gitt av KM, for å holde koden til forskjellige prosjekter adskilt. Det er en projektsuite som er en hovedmappe, under denne ligger de forskjellige prosjektene. I undermappene er det delt opp i flere mapper som inneholder scriptfiler. På denne måten er det lett og holde oversikt over prosjekter og hva de inneholder av kode. En mer visuell oversikt over strukturoppbyggingen kan man se i figuren under (figur 6).



Figur 6: Mappestruktur i TestComplete

5.4 AutoTesten

Koden er utviklet etter KM sin standard for scripting. [12] Dette er et krav fra KM for å holde strukturen oversiktlig, og for at prosjektet skal ha likt oppsett som andre prosjekter. Vi har laget fem hovedscript som inneholder all koden, disse kan vi bruke sammen ved å hente de opp som «biblioteker» i de forskjellige scriptene. Scriptene av AutoTesten er samlet i et dokument og ligger som vedlegg 1 i rapporten.

CommonLeverMain:

Dette scriptet er den overordnede AutoTesten, her blir alle automatiserte test caser hentet opp og lagt sammen. Ved å bygge det opp på denne måten, er det enkelt å legge til eller fjerne test caser ettersom hva som skal testes. Test case funksjonene er hentet opp fra TCLFunctions og TCLThrustCommands scriptene. For å kjøre GUI test er man nødt til å slå av skjermlåsen, den funksjonen er hentet opp fra TCLCommands scriptet.

TCLThrustCommands:

Dette scriptet inneholder de automatiserte test casene for bevegelse av hendelen. Funksjonene til test casene er bygget opp ved at de bruker posisjons- og sjekkfunksjoner som sender til, og mottar verdier fra hendelen. Posisjons- og sjekkfunksjonene er hentet opp fra TCLCheckFunctions og TCLCommands scriptene.

TCLFunctions:

Dette scriptet inneholder de automatiserte test casene for GUI, belysning, sensorer og feedback. De automatiserte test casene er satt sammen som funksjoner av andre funksjoner fra TCLCheckFunctions og TCLCommands, slik at de tilfredsstiller kravene som KM har satt.

TCLCheckFunctions:

Dette scriptet inneholder funksjoner som de automatiserte test casene trenger for å kunne sjekke/bekreftede at ønskede posisjoner og verdier er oppnådd, slik som beskrevet i akseptkriteriene for test casene. Dette scriptet bruker kommandoer fra TCLCommands scriptet.

TCLCommands:

Dette scriptet inneholder alt av kommandoer og knappetrykk hendelen støtter, og som er nødvendig for å utvikle AutoTesten. Det er bygget opp slik at vi enkelt kan finne de kommandoene som er nødvendig til test casene vi skal lage funksjoner til. For hver GUI side berøringsskjermen til hendelen har, er det laget oppdelinger slik at de kommandoene som er tilgjengelig på hver side ligger samlet i scriptet.

IoServerInterface:

Dette scriptet er utviklet av KM, og er laget for kommunikasjon mellom TestComplete og produkter som skal testes. AutoTesten benytter scriptet som et bibliotek for kommunikasjon med hendelen i alle scriptene vi har utviklet.

Denne scriptfilen inneholder koding som KM ikke ønsker å publisere, derfor uteblir denne fra all dokumentasjon.

6 Testrigg

Når et produkt skal testes mot utvikling hos KM, lages det en testrigg. Denne riggen inneholder alt utstyr nødvendig for at produktet skal kunne testes i sin helhet. I prosjektet Common Lever har vi jobbet med prototyper av hendlene. Når testriggen skulle bygges ble det tatt hensyn til at ved installasjon på båt vil utstyret bli utsatt for et annet miljø enn i en testlab. Dette tatt i betraktning, bygde vi en brokonsoll for å gjennomføre testene av hendlene i. Vi har benyttet KM sin egen K-bridge deep line console (figur 7). Dette var med på å avdekke feil på HW og utformingen til hendlene. Med denne type testrigg har vi også testet at temperaturen ikke blir høyere enn hendelen tåler i det miljøet den skal stå i.



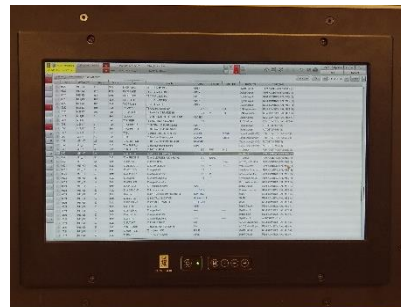
Figur 7: Brokonsoll som er brukt til testrigg

6.1 Hardware sammenstilling

K-bridge deep line console som vi har benyttet som testrigg for å montere hendlene og realtimesystemet i, er satt sammen og beskrevet i dette kapitlet.

6.1.1 Skjermen

Skjermen er montert i K-bride deep line console. Den er skrudd i en 5mm aluminiumsplate med 6mm stjerneskruer, som er festet til konsollen (figur 8).



Figur 8: OS skjerm

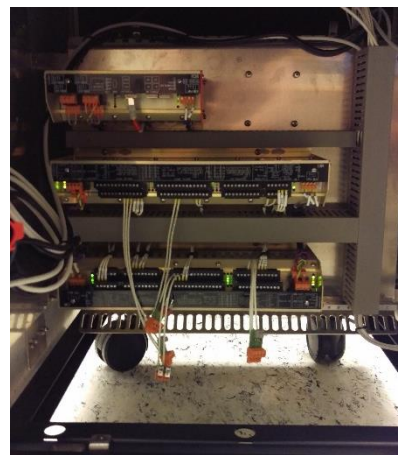
6.1.2 Montasjeplate 1

Denne platen sitter montert på innsiden av døren i front.

Platen inneholder:

- SCU
- 2stk MEI

Utstyret er festet i en 3mm aluminiums plate med M5 stjerneskruer. Kabelkanalene er festet med forsenket M5 stjerne skrue med et plastlokk over for å forhindre vibrasjonsskader (figur 9).



Figur 9: Montasjeplate 1

6.1.3 Montasjeplate 2

Denne platen er montert inni brokonsollen, vendt mot baksiden.

Platen inneholder:

- ABB C6 sikring
- Power, Quint-PS/1A/24DC/20
- Weidmüller ZDU 2.5-2/3AN
- Weidmüller ZPE 2.5-2/3AN
- Phoenix Contact EMG 90-25D/LA 16A 250V≈
- Switch 6xRJ45 & 2xSC (MOXA EDS)
- 6 stk Weidmüller PXS 35 releer

Utstyret er festet i en 3mm aluminiums plate med M5 stjerneskruer. Kabelkanalene er festet med forsenket M5 stjerne skrue med et plastlokk over for å forhindre vibrasjonsskader (figur 10).



Figur 10: Montasjeplate 2

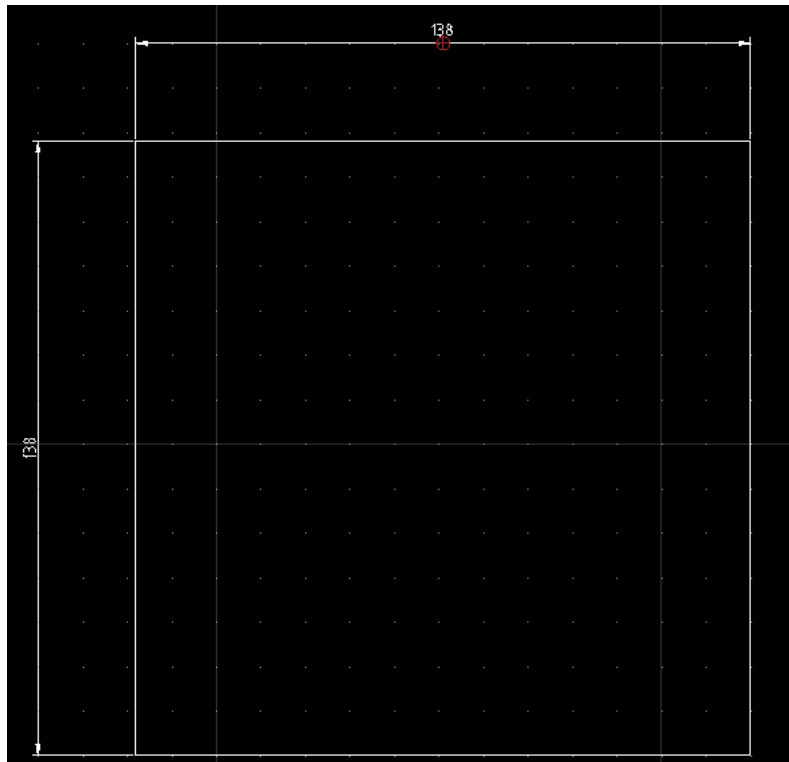
6.1.4 Montasjeplate for hendler og indikatorer

På denne platen så er hendlene montert jevnt med overflaten, mens indikatorene er montert på oversiden. Platen inneholder:

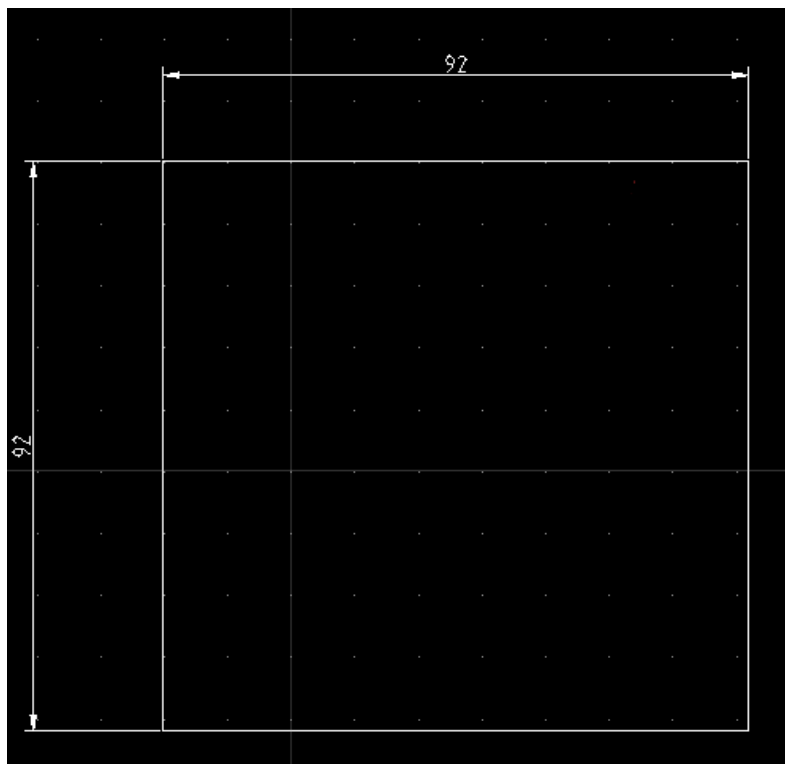
- Common Lever AZ15 (hendel med integrert skjerm)
- Common Lever AZ18 (hendel med eksternskjerm)
- Deif XL144 Dual CAN open (Azimuth CAN instrument)
- Deif XL96 Dual CAN open (Pitch CAN instrument)

6.1.5 Mekaniske tegninger for komponenter

Dette kapitlet viser de tegningene som er brukt for å kappe ut riktige hull til komponentene sitter i konsollen (figur 11, 12, 13, 14).

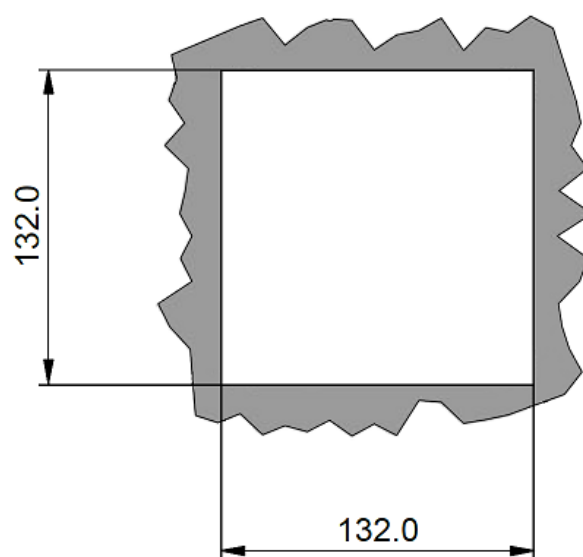


Figur 11: Azimuth CANInstrument cutout

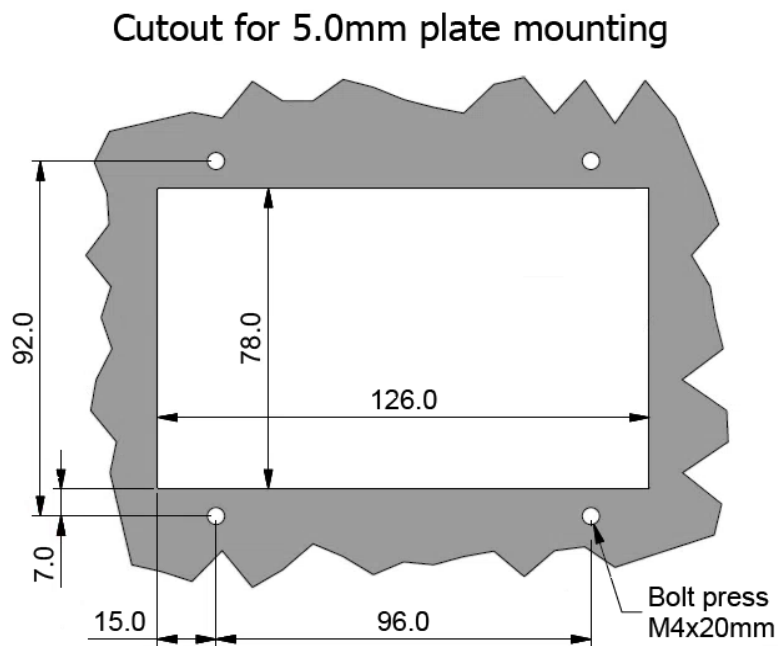


Figur 12: Pitch CAN Instrument cutout

Cutout for plate mounting



Figur 13: Common Lever cutout



Figur 14: Common Lever ekstern skjerm utkapp

6.1.6 Oppkobling av utstyr

Komponentene i konsollen bruker et 24V DC strømforsyning, PC og skjerm bruker 230V. Under så er det listet opp en liste med hva som blir brukt når man skal kunne koble opp power og kommunikasjon mellom DPU.

Liste over kabel og annet materiell som er brukt til oppkobling:

- Niter
- 2*0.5mm² skjermet (CAN)
- 2*1mm uskjermet (Power)
- Strips
- Krympestrømpe

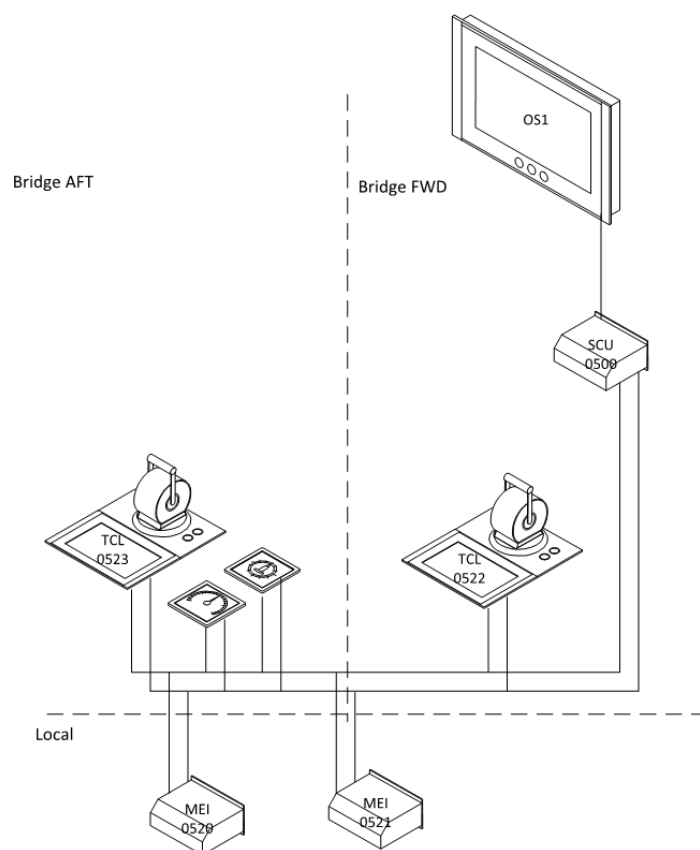
6.1.7 Verktøy

Dette er en liste med verktøyet og utstyr som ble brukt for å gjennomføre koblingsarbeidet som er gjort på montasjeplatene i testriggen:

- Avisoleringstang
- Nitetang
- Kniv
- Avbiter
- 2.5mm flat skrutrekker

6.1.8 Koblingsdiagram

Her beskrives hvordan CAN er koblet mellom realtimesystemet og hendlene i testriggen (figur 15).



Figur 15: CAN koblingsskjema

7 Referanser

Referanser merket med Kongsberg Maritime er tilgjengelige på server hos KM ([\\delta\04311\10 for testerne\15 Bachelor oppgaver\2016 - Common Lever AutoTest](#)) ved å ta kontakt med Mange Røed (e-post: magne.roed@km.kongsberg.com).

- [1] Kongsberg Maritime, Connection Arrangement Document 25.02.2016
- [2] Kongsberg Maritime, 334609D K-Chief 600
- [3] Kongsberg Maritime, 349386A_ SCU datasheet_web
- [4] Kongsberg Maritime, 330153-A_print Main engine interface unit - MEI datasheet
- [5] Kongsberg Maritime, 334024C KThrust datasheet - web
- [6] Kongsberg Maritime, LF 70 lilaas
- [7] Kongsberg Maritime, 409639B Remote Test Interface
- [8] Kongsberg Maritime, 372160B Component Requirement Specification
- [9] Putty: <http://the.earth.li/~sgtatham/putty/0.67/puttydoc.txt> (sist besøkt 14.05.2016)
- [10] Kongsberg Maritime, 402995P3 Software Design Description
- [11] TestComplete, <https://smartbear.com/product/testcomplete/overview/> (sist besøkt 25.04.2016)
- [12] Kongsberg Maritime, [xxxxxxA Guidelines And Rules for using TestComplete0](#)
- [13] Jscript, <https://en.wikipedia.org/wiki/JScript> (sist besøkt 16.05.2016)
- [14] High-level språk: http://www.webopedia.com/TERM/H/high_level_language.html (sist besøkt 16.05.2016)
- [15] Jscript <https://msdn.microsoft.com/en-us/library/hbxc2t98.aspx> (sist besøkt 16.05.2016)



KONGSBERG

Master Product Test Document

Common Lever AutoTest Bachelor Project

KM-MM, R&D

Revision – B

Revisions

Rev.	Written by		Checked by		Approved by	
	Date	Sign.	Date	Sign.	Date	Sign.
A	25.04.2016	TG				
B	12.05.2016	NS	13.05.2016	PV		
C						
D						

Document history

Issue	Section	Description of change
A		Initial Version
04.05.16	1.2, 3.1, 3.2, 3.3, 3.4 3.5, 6.1.1, 6.2.	General editing and updating of the text
B		Updated Version
12.05.16	4, 6.3	General editing and updating of the text

The information contained in this document may be subject to change at a later date (due, for instance, to availability of components). Notice will be given only in case such a change is deemed to be of any consequence for customers. Kongsberg Maritime AS shall not be liable for incidental or consequential damages in connection with the furnishing, performance, or use of this document.

© 2009 Kongsberg Maritime AS. All rights reserved. No part of this work covered by the copyright hereon may be reproduced or otherwise copied without prior permission from Kongsberg Maritime AS.

Table of contents

1	ABOUT THIS DOCUMENT	158
1.1	Scope of document	158
1.2	Abbreviations	158
1.3	About the test process and this document	159
1.3.1	Life cycle planning	159
2	REFERENCES.....	160
2.1	Editable documents	160
2.2	Reference documents.....	160
2.3	Key people.....	161
3	TEST PLAN.....	162
3.1	Scope of test within lifecycle stages.....	162
3.2	Requirements before testing	162
3.3	Safety	163
3.4	Technical approach for verification.....	163
3.4.1	Test design.....	163
3.4.2	Test setup.....	163
3.4.3	Test environment	163
3.4.4	Choices made.....	163
3.5	Skills and competence	164
3.6	Work plan	164
4	TEST PROCEDURES	165
5	APPROVALS/CERTIFICATION.....	166
6	TEST REPORT	167
6.1	Test Setup	167
6.1.1	Physical.....	167
6.1.2	Software.....	168
6.1.3	Database.....	168
6.2	Test Log.....	169
6.2.1	Completed Test cases for AT	169
6.2.2	Remaining Test cases for AT	171

6.2.3	Test cases for manual tests	172
6.2.4	Bugs	173
6.3	Evaluation	178

1 ABOUT THIS DOCUMENT

1.1 Scope of document

This document is the master of all test documents for the Common Lever AutoTest Student Project in the spring of 2016. All documentation and necessary information for test of this lever should be found in or via this document.

Common Lever AutoTest is a bachelor project by four students from HSN Kongsberg. The project started in January 2016 and ended in May 2016. The main task was to test the prototype of the Common Lever. Testing included making an automated test combined with manually testing, to verify that it meets the specifications.

The master product test document will cover the following topics:

Test plan, test procedure and test report.

1.2 Abbreviations

KM	Kongsberg Maritime
SW	Software
HW	Hardware
HMI	Human Machine Interface
GUI	Graphical User Interface
LAN	Local Area Network
MEI	Main Engine Interface
DPU	Distributed Processing Unit
SCU	Segment Controller Unit
TCL	Common Lever
MTM	Microsoft Test Manager
TFS	Team Foundation Server
OS	Operator Station
FRS	Functional Requirement Specification
FDD	Functional Design Description
AT	Automated Tests
WI	Work Item

1.3 About the test process and this document

This is the master document which has all necessary information about the testing of this product. This document must be evaluated / updated each time a change is made to the test part of the project/product. All references to test procedures shall be updated against product revisions. All test documents that are used must be linked from this document in the reference chapter.

1.3.1 Life cycle planning

This document is to be used within the stages development, utilization and maintenance.

This document is to be used as a test plan for these stages. The outputs of the development stage are both a verification of the product and a set of standard test to validate and insure quality through its lifecycle.

This document will be used as template for project and products in development phase and also to some extent for project or products that shall be release tested during the utilization and maintenance phase.

2 REFERENCES

2.1 Editable documents

Reference to automatic test program and remote test interface

Name	Description	Document number	Version
Link to script	\\delta\04311\10 For testerne\15 Bachelor oppgaver\2016 - Common Lever AutoTest		
Remote Test Interface	K-Thrust 600/720 HMI devices	409639	B

2.2 Reference documents

Documents for information and reference only

Name	Description	Document number	Version
Product Test Strategy	KM-MM R&D test strategy	361061	A
Product management	Main Process	MPD 0001	
Product development	Main Process	MPD 0002	
Functional Requirement Specification	K-Thrust 600	388014	A
Functional Design Description	K-Thrust 720, HMI devices specification	390570	P7

2.3 Key people

Name	Function	Department	Contact
Ørjan Sti	Product owner		
Sven Arild Sande	Project manager		
Pål Kristian Halle	Developer	R&D	
Kjell Stene	Developer	R&D	
Trond Inge Henriksen	Developer	R&D	
Magne Røed	Tester	TEST	
Erik Saly	Tester	TEST	
Thomas Gjerrud	Student HSN	TEST	
Kristoffer Stanger	Student HSN	TEST	
Nicolai Sætra	Student HSN	TEST	
Philip Vestlie	Student HSN	TEST	

3 TEST PLAN

3.1 Scope of test within lifecycle stages

The levers are delivered as prototypes when the Test Department and the students are involved. The students will develop an automated test for the levers, and simultaneously do manual testing. Errors and deficiencies found should be reported as bugs for the Development Department to correct. Testing conducted by the students undergoes the following stages:

1. HW and SW test. In this stage the prototype will be tested to see if it fulfils the specification, and that it responds to the commands in the document HMI Devices – Lever Remote Test Interface.
2. Testing the lever in a real-time feedback system.
3. Simulator and automation. It is a requirement that all new product should be adapted for automation and automated test.

The test results and automated test will be delivered to KM at the end of the student project for further development and implementation in a full system test.

The final test will be with class and product owners. When this is passed the lever is ready for mass production and the delivery system.

3.2 Requirements before testing

It is a goal for the Test Department to automate as many tests as possible.

To achieve this goal, the new SW and HW has to be adapted to take commandos from automation equipment/software.

I the case of Common Lever the requirements are that the lever movement (azimuth/thrust) and GUI buttons must be possible to remote control from TestComplete and/or National Instruments modules.

In addition, the students need the following hardware:

- OS
- SCU (Segment Controller Unit)
- x2 MEI (Main Engine Interface)
- Various switches, relays and feedback simulators
- Test console
- Computers with necessary licenses for TestComplete, MTM and TFS

3.3 Safety

Testing is performed based on specifications contained in the following documents:

- FRS (Functional Requirement Specification), K-Thrust 720 HMI devices, Lever and command panel
- FDD (Functional Design Description), K-Thrust 720, HMI devices specification
- Guidelines and Rules for use of MTM (Microsoft Test Manager)

3.4 Technical approach for verification

3.4.1 Test design

The design of the test will follow the FRS as rule in this project.

In addition, all old tests made regarding the manoeuvring system will be reused and updated. Commando Transfer system is also a vital part of the lever, and will be a big part of the tests.

3.4.2 Test setup

The main setup for the student project will be azimuth thruster system including one lever with external display and one lever with internal display. The levers, modules, instruments and OS will be mounted in a test rig that is similar to a workstation bridge console used on vessels. The automated test will be run from a PC with TestComplete installed, connected to the levers directly via LAN.

3.4.3 Test environment

The test is to be performed at room ambient temperature in a normal office environment.

3.4.4 Choices made

It was taken several decisions regarding TCL without the Test Department present in the early development stage. Some of these choices have probably resulted in some delays in the TCL project. The student project was mainly meant as a SW task focusing on creating automated tests with some possible extensions. Since the TCL project had not come as far as scheduled by the time the students started their project, they got more involved in HW and development than planned. Because of this, the students have gained a much better understanding of a larger part of the system.

3.5 Skills and competence

The students should have knowledge about Thruster and Auto Chief Systems, and how a maritime engine is working. This knowledge will be provided by KM in form of internal training in these systems.

To make the automated test in TestComplete requires knowledge about scripting, and the students have basic knowledge from Arduino course at school. Arduino script knowledge comes in handy when they need to learn KM's script standard Jscip (same as JavaScript).

There are a lot of wiring and connections to get the system to work. All of the students have years of competence for doing this job.

The students will specialize in four fields that are relevant to the TCL project:

- Testing (Verification of AT (automated test), manual testing and thruster behaviour)
- K-Chief 600 (As a system, operating OS and DPU)
- Common Lever (401-module / new technology)
- Scripting AT (Structure and language)

3.6 Work plan

The work plan for the project can be found at KM's testservers in Horten via this link:

<\\delta\04311\10 for testerne\15 Bachelor oppgaver\2016 - Common Lever AutoTest>

4 TEST PROCEDURES

The MTM test plan has to be developed when the lever is finished in the development stage and ready for testing.

This link enters the bachelor groups project management for the common lever project. This contains the user stories, tasks and test cases we have used.

http://ptfs:8080/tfs/MM-RD/CommonLeverAutoTest/_backlogs#level=Stories&showParents=false&a=backlog

5 APPROVALS/CERTIFICATION

This part covers the next stage of the project after the students are finished. KM will take care of this part

6 TEST REPORT

6.1 Test Setup

All the hardware was mounted in a bridge work station console with an integrated operator station. The purpose is to run a real-time system in a presentable test rig, which can also be used at the final presentation.

TestComplete by SmartBear was used to design and run the AT.

6.1.1 Physical

Mounting plate 1:

- 230V Primary power in
- 24V Power supply
- Terminals for 24V and 230V
- Two double sockets
- Network switch
- Relays for engine and pump simulation

Mounting plate 2:

- SCU (Segment Controller Unit)
- x2 MEI (Main Engine Interface)

Mounting plate for levers and indicators:

- Lever W.Gessmann Prototype AZ15 with internal screen
- Lever W.Gessmann Prototype AZ18 with external screen
- DEIF XL96 Pitch indicator
- DEIF XL144 Rotation indicator

The OS is mounted in the front door of the console.

6.1.2 Software

The software for the project can be found at KM's testservers in Horten via this link:

<\\delta\04311\10 for testerne\15 Bachelor oppgaver\2016 - Common Lever AutoTest>

The following software is updated throughout the project:

Software for lever:

TCL_0.00.01-r97429.DPU	4/01/2016
TCL_0.00.01-r.97747DPU	4/08/2016
TCL_0.00.01-r.97762DPU	4/08/2016
TCL_0.00.01-r.97810DPU	4/11/2016
TCL_0.00.01-r.98225DPU	4/19/2016
TCL_0.00.01-r.98409DPU	4/22/2016
TCL_0.00.01-r.98439DPU	4/25/2016
TCL_0.00.01-r.98955DPU	5/09/2016

Software for the modules:

8.14w01-r97560	4/4/2016
8.14w01-r97726	4/7/2016
8.14w01-r97811	4/11/2016
8.14w01-r97828	4/11/2016
8.14w01-r97217	4/19/2016
8.14w01-r98803	5/04/2016

6.1.3 Database

The database for the project can be found at KM's testservers in Horten via this link:

<\\delta\04311\10 for testerne\15 Bachelor oppgaver\2016 - Common Lever AutoTest>

6.2 Test Log

The test log will describe the test cases with results and current state for both automated and manual tests, as well as the bugs that has been discovered and reported. The test log for Common Lever AutoTest are now stored locally on the computers which have conducted the test. This log will later be stored as results in each test case on KM's servers, when they perform a fully finished automated test. Manual test logs will also be stored on KM's servers when the product are ready for final testing.

6.2.1 Completed Test cases for AT

This chapter describes the test cases for the automated tests that has been completed as far as possible within the time limit of the student project.

6.2.1.1 TCL pitch command (TC 62079)

Make an AT to verify that the pitch knob on the lever goes to different settings on command. The AT worked as described in test case 62079 on the first SW version for the lever. With updated SW installed, the verification part of the AT stopped working, and had to be redesigned. After redesign, the test works as described by the test case. Scripted by Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra and Philip Vestlie.

6.2.1.2 TCL rotation command (TC 62083)

Make an AT to verify that the rotation on the lever goes to different settings on command. The AT worked as described in test case 62083 on the first SW version for the lever. With updated SW installed, the verification part of the AT stopped working, and had to be redesigned. After redesign, the test works as described by the test case. Scripted by Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra and Philip Vestlie.

6.2.1.3 TCL Start/Stop of Thruster/Servo pump (TC 64131)

Make an AT to check that the lever sends the right signals to start and stop thruster and pump. It shall also verify that the buttons in GUI changes state when pressed. The AT works as described by the test case 64131. Scripted by Thomas Gjerrud and Nicolai Sætra.

6.2.1.4 TCL status after reboot (TC 64149)

Make an AT to see that the lever starts up correctly with all values equal to what they were before the reboot occurred. Works as described by test case 64149. Scripted by Philip Vestlie.

6.2.1.5 TCL day palette (TC 61899)

Make an AT to verify that the day palette function works as intended. The AT works as described in test case 61899. Scripted by Nicolai Sætra.

6.2.1.6 TCL dusk palette (TC 61900)

Make an AT to verify that the dusk palette function works as intended. The AT works as described in test case 61900. Scripted by Nicolai Sætra.

6.2.1.7 TCL night palette (TC 61901)

Make an AT to verify that the dusk palette function works as intended. The AT works as described in test case 61901. Scripted by Kristoffer Stanger.

6.2.1.8 TCL password protection (TC 61902)

Make an AT to verify that the password protection function works as intended. The AT is described by test case 61901. The AT was affected by bug ID 63901 to work as intended. The bug is closed, the AT is redesigned and works as intended. Scripted by Kristoffer Stanger and Nicolai Sætra.

6.2.1.9 TCL light sensor (TC 61914)

Make an AT to verify that the internal light sensor is dimming the display and LED's according to test case 61914. The AT works as described. Scripted by Thomas Gjerrud.

6.2.1.10 TCL dimming from internal sensor (TC 61916)

Make an AT to verify that the internal light sensor is responding according to test case 61916. The AT is affected by bug ID 63999, but works as intended. Scripted by Nicolai Sætra.

6.2.1.11 TCL dimming from external sensor (TC 61919)

Make an AT test to verify that the external light sensor input is responding according to test case 61919. The AT is finished as far as possible and seems to work, but cannot be fully tested without an external sensor input. Affected by bug ID 63999. Scripted by Nicolai Sætra.

6.2.1.12 TCL thrust set point test (TC 61672)

Make an automated test to verify that the thrust set point is responding according to test case 61672. Works as intended. Scripted by Philip Vestlie.

6.2.1.13 TCL take command (TC 64903)

Check if the lever are in command before running AutoTest according to test case 64903. Works as intended. Scripted by Nicolai Sætra.

6.2.2 Remaining Test cases for AT

The table lists the test cases for the automated tests that have not been started. This is because of the time limit of the student project, and the circumstances described in 3.4.4.

ID	WI Type	Title	State
61898	Test Case	Common Lever Screen lock function	Design
61904	Test Case	Common Lever button background colour change	Design
61905	Test Case	Common Lever blinking button frame	Design
61909	Test Case	Common Lever pushed button indication	Design
61910	Test Case	Common Lever button in waiting for confirmation state	Design
61913	Test Case	Common Lever Service menu returning to Home page	Design
61921	Test Case	Common Lever button blinking speed	Design
61923	Test Case	Common Lever button colour	Design
61924	Test Case	Common Lever sound	Design

6.2.3 Test cases for manual tests

This table lists the manual test cases done in order to start with the automated testing.

ID	WI Type	Title	State
11676	Test Case	HW Connections	Ready
11678	Test Case	Mimic Layout	Ready
11681	Test Case	SW Interface to DP	Ready
11683	Test Case	Ready for DP	Ready
11684	Test Case	Communication Fault	Ready
11686	Test Case	SW Interface to C-Joy	Ready
11687	Test Case	HMI interface PMS-DP	Ready
15017	Test Case	Load AIPC8 with CP8.exe	Ready
61877	Test Case	Common Lever SW download test	Ready
61883	Test Case	Common Lever Installation test	Ready
61885	Test Case	Common lever DIM test	Ready
61886	Test Case	Common lever long-term test	Ready
61887	Test Case	Common lever calibration test	Ready
61889	Test Case	Common Lever parameter change test	Ready
61891	Test Case	Common Lever communication load test	Ready
61927	Test Case	Common Lever 24V connected to to the CAN plugs	Ready
61930	Test Case	Common Lever reversed power	Ready
61931	Test Case	Common Lever Short circuit CAN A	Ready
61933	Test Case	Common Lever short circuit CAN B	Ready
61934	Test Case	Common Lever short circuit power	Ready
63825	Test Case	Common Lever New Configuration of SCU	Ready

6.2.4 Bugs

This table lists the faults and defects that have been discovered during the testing phase. Everything has been reported as bugs regardless of whether there is mechanical failure or software error. The state of the bug is either active, closed or resolved. If the state is active, work is still in progress. If it is closed, it is no longer valid for the student project. Resolved state means that there have been implemented a solution for the bug.

If the bug was found during AT, it is linked to its appropriate WI in TFS.

ID	WI	Title	Reported by	State
64372	Bug	Full memory in lever while running Autotest	Philip Hem Vestlie	Resolved
		While running Autotest, the memory in the lever gets full after 256 connections		
64273	Bug	TCL touch lock	Thomas Gjerrud	Closed
		Need a function to activate the touch lock		
64276	Bug	Buttons in main page flashing mode	Thomas Gjerrud	Closed
		Need a function to set the buttons in main page in flashing sequence with different colours		
64277	Bug	Flashing frame feedback	Thomas Gjerrud	Closed
		Need a function to simulate flashing frame on the buttons in main page		
64278	Bug	Read button status	Thomas Gjerrud	Resolved
		Need a function to read the status on buttons		
64295	Bug	Reboot command	Philip Hem Vestlie	Resolved
		Need a command to shut down and reboot common lever		
64424	Bug	Sending command to Common lever pitch	Kristoffer Stanger	Active

		When sending a position command, the lever will find position and then randomly go to some other position afterwards.		
64297	Bug	Button confirmation state function	Philip Hem Vestlie	Active
		Need a function that will set the button in confirmation state when pressed (for command buttons in main window)		
64299	Bug	Return to homepage after a pre-set time	Philip Hem Vestlie	Closed
		When entering service menu, the GUI will not return to homepage after pre-set time.		
64483	Bug	While running Autotest TCL gives Application error	Philip Hem Vestlie	Resolved
		While running Autotest of Pitch commands, TCL gave Application error, and connection was lost.		
64520	Bug	Internal TCL light sensors gives different value in the same light conditions	Philip Hem Vestlie	Closed
		Ambient light value on test levers are different in the same light conditions		
64572	Bug	Motor controller pitch	Kristoffer Stanger	Active
		When giving a pitch command, the lever does not always manage to reach the designated position. It stops before it should. It fails randomly for each test run.		
64576	Bug	Motor controller Azimuth	Kristoffer Stanger	Active
		When giving an Azimuth command, the lever doesn't always manage to reach the designated position. It stops before it should. It fails randomly for each test run.		
63880	Bug	TCL handle configuration fail	Nicolai Sætra	Resolved
		TCL handle couldn't get configuration from SCU. GUI prints out: Config fail A0_21.		

63998	Bug	The standalone TCL touch screen are to small	Nicolai Sætra	Closed
		The standalone touch screen for TCL are not thick enough, when mounted in the bridge console it does not lay flush with the console mounting plate.		
63999	Bug	Dimming the light with sensor fails	Nicolai Sætra	Closed
		When we try to dim the light with the ambient light manipulation for auto-test, it only works when the dimming mode is set to Auto. The percentage of dimming are equal to the lux manipulation in the range: 0-100lux = 0-100%light, the scaling must be wrong. All other dimming modes does not work with manipulation of light sensor.		
64013	Bug	TCL play alarm sound	Nicolai Sætra	Closed
		Need a function to start audio sound, so that we can test and verify that alarm sound works.		
64134	Bug	TCL password menu	Nicolai Sætra	Resolved
		Need two different passwords, to check that the "Administrator" password has full access in the service menu and the "User" password has restricted access in the menu.		
64246	Bug	Text in GUI	Kristoffer Stanger	Closed
		The load bar has "Angle" as notation, should have been "Load". It doesn't show the notation of the value.		
64202	Bug	Take command pitch	Philip Hem Vestlie	Resolved
		When command is taken from one lever to another, the pitch does not take the position of the lever in command.		
64142	Bug	Parameters for indicators	Kristoffer Stanger	Resolved
		The parameters that are set for the indicators to show right value, resets after blackout on OS		

64168	Bug	When pressing take command on lever	Kristoffer Stanger	Active
		When a person presses a button on the lever, the confirm button press does not appear.		
63876	Bug	SCU Crash	Nicolai Sætra	Resolved
		The SCU crashes when the database for TCL is loaded. SCU reset itself continuously.		
63896	Bug	TCL dimming slider adjust	Nicolai Sætra	Resolved
		Need a function for adjusting the dimming slider for Autotest.		
63897	Bug	TCL GUI-page status	Nicolai Sætra	Resolved
		Need a function for checking what page we are in. Some examples are: Service page, Homepage etc.		
63898	Bug	TCL GUI-page change	Nicolai Sætra	Resolved
		Need a function for setting Homepage in GUI.		
63899	Bug	TCL Ambient light manipulation	Nicolai Sætra	Resolved
		Need a function for manipulation of the ambient light.		
63900	Bug	TCL Touch lock	Nicolai Sætra	Resolved
		Need a function to check touch lock status (enabled/disabled) Need a function to disable touch lock		
63901	Bug	TCL return to Homepage after invalid password are entered	Nicolai Sætra	Closed
		GUI-page does not return to Homepage after an invalid password are entered.		
63913	Bug	TCL aft.bridge landscape option	Nicolai Sætra	Resolved

		Aft.bridge screen needs to be i landscape mode. Now it shows GUI in portrait mode.		
63833	Bug	Configuration	Kristoffer Stanger	Resolved
		When installing software on SCU and configuration on DPU's, the SCU could not handle loading configuration on two TCL at the same time.		
63836	Bug	Take command	Kristoffer Stanger	Resolved
		When command is taken from aft bridge to fwd bridge the lever doesn't compare to the other in the transfer process.		
63841	Bug	Sounder alarm	Kristoffer Stanger	Closed
		When TCL restarts the Alarm starts and has to be shut off manually every time.		
63844	Bug	Wrong text in GUI	Kristoffer Stanger	Closed
		The load bar below azimuthroze is named angle instead of load		
63846	Bug	"Sound off" button	Kristoffer Stanger	Resolved
		When the sound off button is pressed the frame still blink red instead of going to a stationary red light.		
63847	Bug	Emergency button light	Kristoffer Stanger	Closed
		When emergency button is pressed the light changes from red to blue. Should be green		
63852	Bug	Host communication	Kristoffer Stanger	Resolved
		Host comm. shows an error on the lever		

6.3 Evaluation

The automated test we have made includes testing of important functions. Some of this functions are pitch, rotation, password protection, start/stop pumps and set different palettes. The AutoTest have already payed of by finding a critical SW fail which wouldn't be found by manual testing. This fail resulted in application error on the lever, and could potentially have cost tens of thousands NOK if it had happened on a sailing ship.

The AutoTest have also found some less critical fail and error with the lever, but these are also important to fix so that KM can deliver a flawless product.

The initial goal was to complete a fully automated test for both azimuth and tunnel thruster levers. Due to delays in KM's common lever project, the bachelor project has some changes from its origin. This has resulted in more need for manual testing and troubleshooting, to make sure that the lever has the functions and support needed to make the automated test.

At this point in the project, all the material available to us has been processed to make an automated test.

The automated test and manual test results will be handed over to the Test Department at the end of the project, so that it can be implemented and further developed to their testing procedures.

There are some things that need extra attention to make sure they are as intended (see 6.2.4 Bugs for full list):

The pitch and azimuth commands should support degrees, but it only supports steps at this point. (software developer)

The hardware light and touchscreen backlight are after reboot 70-75% this has to be changed to night palette as standard. (Software developer)

The loose touchscreen has a hardware fail that needs to be fixed, the screen are 2mm to low for flush mounting in the bridge console. (Software developer)

The hardware buttons for Backup and Emergency Stop may need to be rotated so that the text are in the right way (are now sideways). (Hardware developer)



Etteranalyse

Versjon 1.0



Common Lever Auto-Test

Bacheloroppgave for ingeniørstudie ved Høgskolen i Sørøst-Norge, avdeling Kongsberg, fakultet for teknologiske og maritime fag.

Dato: 23.05.2016

Oppdragsgiver: Kongsberg Maritime Merchant Marine Horten

Prosjektdeltagere: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Intern veileder: Antonio L.L. Ramos

Ekstern veileder: Magne Røed

Dokumenthistorie

Versjon	Dato	Endringer	Ansvarlig
1.0	13.05.2016	Første versjon	TG

Innholdsfortegnelse

1	Innledning.....	182
2	Etteranalyse	182
3	Sprintlogg	183
3.1	Iterasjon 1	183
3.2	Iterasjon 2	184
3.3	Iterasjon 3	184
3.4	Sprint 4.....	185
3.5	Sprint 5.....	185
3.6	Sprint 6.....	186
3.7	Sprint 7.....	187
3.8	Sprint 8.....	187
3.9	Sprint 9.....	188

1 Innledning

Dette dokumentet inneholder en kollektiv evaluering av bachelorprosjektet i sin helhet. Dokumentet består av en etteranalyse og en sprintlogg. I etteranalysen reflekteres det over det arbeidet som er utført, og de erfaringene og læringsutbyttet studentene har fått ut av arbeidet. Sprintloggen er basert på oppfølgingsdokumentene som er ført gjennom hele prosjektperioden, og beskriver grundig fremdriften i prosjektet.

2 Etteranalyse

Common Lever AutoTest har vært et meget lærerikt og spennende prosjekt for alle i gruppen. Bachelorprosjektet var en del av KM sitt Common Lever utviklingsprosjekt, og vi har fått prøve oss i en reell arbeidssituasjon hos oppdragsgiver.

Vi er stolte og takknemlige for å ha fått bidra i utviklingsfasen på et så relevant og komplekst system som Common Lever. Ved å jobbe med prosjektet har vi fått føle på mange av aspektene som følger i en ingeniørs arbeidshverdag. Dette er alt fra stor mestringsfølelse ved å få produktet til å fungere, til frustrasjon når prosjektet ikke går som planlagt.

Denne prosjektgruppen ble satt sammen på bakgrunn av at alle har flere års arbeidserfaring fra elektrobransjen, og kjenner hverandres arbeidsmetoder og holdninger. Alle har også erfaring med feilsøking og testing på elektrosystemer, og har derfor god bakgrunn for å lettere forstå sammensatte og komplekse systemer. Dette dannet et glimrende utgangspunkt for det gode samarbeidet vi har hatt i denne gruppen. Gruppen fikk tilgang til KM sine lokaler i Horten med eget kontorlokale. Vi har hele tiden hatt et tett samarbeid innad i gruppen, og alle har bidratt like mye i prosjektet. Alle har tatt fullt ansvar for sitt ansvarsområde, i tillegg til å bidra aktivt i de andre arbeidsoppgavene.

Vi har oppnådd god praktisk tilnærming til de tekniske fagene vi har hatt på Høgskolen. Programmering og mikrokontrollere gav et bra grunnlag for å benytte TestComplete til programmering av AutoTesten. Uten dette faget, ville vi brukt mye mer tid på å lære scriptspråket. Kunnskap fra fagene Analog elektronikk og Instrumentering og styring har vært meget nyttig for å kunne forstå mye av teknologien som ligger bak produktet og systemet vi har testet. I faget Systems engineering trente vi på å styre et fiktivt prosjekt med fokus på livssyklusen i et system. Vi har nå fått prøve oss på å styre et reelt prosjekt hvor man har sett viktigheten av å tenke helhet og livssyklus i et system. Prosjektet inneholdt et stort spekter med oppgaver som har gitt oss god forståelse av alt fra oppkobling og tilpassing av hardware, til forståelse og koding av software.

Prosjektet var i utgangspunktet definert som en oppgave med hovedfokus på å lage AutoTest for azimuth og tunell hendler med mulighet for praktiske utvidelser. Endringene som har oppstått underveis i prosjektet har i tillegg gitt oss en mye bedre forståelse av det systemet hendlene skal implementeres i. Vi har derfor fått mye bedre læringsutbytte av hele oppgaven enn det vi så for oss i begynnelsen.

Det har vært lærerikt og nyttig å få benytte de programmene KM benytter til prosjektplanlegging og scripting. Vi hadde utfordringer med å sette opp riktig prosjektmodell i begynnelsen, og valgte å bytte til SCRUM underveis. Dette var for å tilpasse oss KM sine arbeidsmetoder, samt TFS (Team Foundation Server) som er basert på SCRUM arbeidsmetodikk. TFS passet veldig bra til styring av prosjektet og til den prosjektplanen vi har jobbet etter.

I følge prosjektplanen ble det satt av tid til prosjektet som skulle gjenspeile normale arbeidsdager. Dette var åtte timers arbeidsdager tre dager i uken de ti første ukene, og fem dager i uken de resterende ukene etter påske og eksamen. Til sammen skulle dette være tilstrekkelig for å oppfylle de antall timer det er normalt å bruke på et bachelorprosjekt. Timebruken totalt i prosjektet har avviket veldig lite fra det som ble planlagt, og dette er vi naturligvis godt fornøyd med.

KM har gitt uttrykk for at de er meget fornøyd med det vi har klart å oppnå, i tillegg til vår kommunikasjon mot de ansatte hos KM som har vært involvert i prosjektet. Vi har hatt et tett samarbeid med vår eksterne veileder i tillegg til de som jobber med utvikling av SW til hendlene. Gruppen har holdt daglige stand-up møter med ekstern veileder for tett oppfølging av prosjektet. Dette har gitt bra utbytte for begge parter, med tanke på at det kreves god toveis kommunikasjon i et utviklingsprosjekt. Ekstern veileder har påpekt at daglige stand-up møter er noe de heretter vil benytte på fremtidige studentprosjekter.

3 Sprintlogg

Dette er en gjennomgang av iterasjonene/sprintene i prosjektet. Teksten er basert på oppfølgingsdokumentene som er ført gjennom hele prosjektperioden. Oppfølgingsdokumentene finnes som vedlegg til rapporten. Hensikten med oppfølgingsdokumentene er å ha en logg over det arbeidet som er utført i prosjektperioden. Oppfølgingsdokumentene er også til sendt intern veileder etter hver iterasjon/sprint for å holde han oppdatert om fremdriften i prosjektet. Grunnen til at det er en blanding av iterasjoner og sprinter, er at det ble byttet prosjektmodell fra Unified Process til SCRUM etter at prosjektet hadde startet. Dette er grundigere forklart i dokumentet prosjektplan.

3.1 Iterasjon 1

Vi bestemte tidlig å benytte en iterativ prosjektmodell. Uke 2 og 3 ble vår første iterasjon hvor fokuset var å sette seg inn i hva oppgaven innebar og etablere kontorplass hos oppdragsgiver. Det ble forespeilet en til to ukers opplæring i systemene hos oppdragsgiver. På grunn av utfordringer med tilgang til datasystemene ble mye av opplæringen utsatt. Mye av tiden gikk isteden med til å utvikle kravspesifikasjon, prosjektplan og testplan. Det ble også brukt tid på å lage systemer og maler for oppfølgingsdokumenter, timeføring og aktivitetsplan. KM jobber etter SCRUM arbeidsmetodikk og derfor ønsket ekstern veileder å holde korte daglige stand-up møter i tillegg til ukentlige planleggingsmøter med prosjektgruppen.

3.2 Iterasjon 2

I iterasjon 2 som bestod av uke 4 og 5 fortsatte arbeidet med prosjektplanen, kravspesifikasjonen og testplanen som måtte være ferdig til første presentasjon. Vi så at tiden gikk meget fort og første presentasjon var planlagt i uke 6. Vi hadde litt problemer med å bestemme oss for prosjektmodell, men etter mye diskusjon innad i gruppen og råd fra intern veileder og sensor valgte vi å benytte en iterativ prosjektmodell basert på Unified Process. Grunnen til at vi valgte denne modellen var at vi så for oss oppgaven som en ren software oppgave, noe denne modellen er egnet for. Det ble arbeidet videre med å sette opp en tidsplan i form av gantt, og systemene for timeføring, oppfølging samt aktivitetsnettverket ble videreutviklet. Det var fortsatt problemer med tilgang til arbeidsverktøyene TestComplete, Microsoft Test Manager (MTM) og Team Foundation Server (TFS). På grunn av dette valgte prosjektgruppen å utsette opplæringen i arbeidsverktøyene og heller fokusere fullt på dokumentasjonen som skulle leveres i forbindelse med første presentasjon.

3.3 Iterasjon 3

Iterasjon 3 var i uke 6 og 7. Torsdag uke 6 var satt av til første presentasjon, slik at den uken gikk hovedsakelig med til forberedelse av presentasjonen. Fremføringen av presentasjonen gikk veldig bra og vi fikk gode tilbakemeldinger både på ryddig dokumentasjon og fremføring. Sensor satte spørsmål med hvorfor vi hadde valgt en prosjektmodell basert på Unified Process, når alt lå til rette for bruk av SCRUM i oppdragsgivers datasystem. På bakgrunn av dette valgte vi nå å benytte SCRUM som prosjektmodell. Dersom vi hadde hatt mer tid til opplæring i datasystemet til oppdragsgiver i iterasjon 1, ville vi sannsynligvis sett at SCRUM ville vært et åpenbart valg av prosjektmodell på et tidligere tidspunkt. Mye av planleggingen og de opprinnelige fasene fra den første modellen ble beholdt, slik at det å bytte modell har kun vært positivt for prosjektet.

I uke 6 ble det opprettet tilgang til skoleserver for hjemmeside til bacheloroppgavene. Det gikk naturligvis med mange timer til å utvikle hjemmesiden, da ingen i gruppen hadde grunnlag for å gjøre dette.

Uke 7 var starten på utviklingsfasen i prosjektet. Den første azimuth hendelprototypen med integrert skjerm var nå klar, og det ble igangsatt opplæring i TestComplete for scripting av AutoTesten. Det ble også arbeidet med tilpasning av prosjektplanen i forhold til SCRUM modellen. Scripting og oppdatering av prosjektplanen var planlagt å ta med over i sprint 4. På grunn av SCRUM omdøpes nå iterasjonene til sprint, men med fortsettende nummerering.

3.4 Sprint 4

Sprint 4 var i uke 8 og 9. Vi hadde nå fått nødvendige tilganger til oppdragsgivers datasystem (TFS, MTM og TestComplete). Det gikk med en del tid til å lære oss disse systemene. Vår eksterne veileder Magne Røed opprettet tidlig i uke 8 de første test casene slik at vi kunne begynne på AutoTesten (AT).

For å simulere et realtime system og designe AT, ble det koblet opp en Operator Station (OS), en Segment Controller Unit (SCU), to Main Engine Interface (MEI) samt en prototype av Common Lever (TCL). Vi oppdaget raskt at det var problemer med databasen for å få TCL til å spille mot real-time systemet. En foreløpig løsning for å komme i gang med selve scriptingen av AT, ble å kjøre TCL via et testprogram (TclHW test) med et USB til CAN adapter fra PC.

De første test casene var hovedsakelig for GUI (touch-skjerm funksjoner). Disse ble det vanskelig å lage AT for på dette tidspunktet, på grunn av det vi trodde var mangelfull kommunikasjon mot hendelen ved scripting. Magne Røed opprettet så to nye test caser for hendel bevegelse (rotasjon og pådrag) som vi skulle utvikle. Disse var ikke påvirket av problemene med databasen. Det ble utført mye manuell testing av hendelen for å forstå virkemåten og hvilke kommandoer man skulle benytte.

Da oppgaven ble planlagt høsten 2015, var det meningen at real-time systemet og den manuelle testingen skulle være klart til januar, slik at vi kunne begynne rett på utvikling av AT. KM hadde i januar ikke kommet så langt i utviklingen av prosjektet som planlagt. Dette påvirket i grunn prosjektet vårt positivt, siden vi i større grad ble en del av nyutviklingen av funksjonalitet og hardware, og fikk mulighet til å se systemet mer i sin helhet.

Vi anså sprint 4 som en prøvesprint, for å se hvor lang tid det ville gå med til scripting og utfordringer med programmene vi brukte. I sprint 5 planla vi å jobbe fullt etter SCRUM metodikken.

3.5 Sprint 5

Denne sprinten var i uke 10 og 11 med andre presentasjon i uke 11.

Oppfølgingsdokumentasjon med timeestimering var nå blitt bra innarbeidet. Vi planla 66 timer til forberedelse og fremføring av presentasjonen, 38 timer til dokumentasjon, og 50 timer til AT scripting. Ved avslutningen av sprint 5 hadde vi brukt 120 timer på alt i forbindelse med presentasjonen, 8,5 timer på dokumentasjon og 9 timer på scripting. Dette viser at vi ønsket å jobbe omtrent like mye med produktet som med presentasjonen, men endte opp med å bruke nesten all tiden på presentasjonen. Vi jobbet med å gjøre ferdig AT for rotasjon og pådrag i løpet av sprint 5. Vi fikk en ny versjon av Remote Test Interface dokumentet som omhandlet hvilke funksjoner vi kunne sette, spørre etter og få svar på i testprogrammet til hendelen. Det ble jobbet med å se hvordan dette fungerte og hvilke user stories vi da kunne jobbe videre med.

For å komme videre med scriptingen av test casene for GUI, presset Magne Røed på for å få mer framdrift på prosjektet internt i KM. Vedvarende tekniske og software relaterte utfordringer med hendelen gjorde at vi måtte ta stilling til om vi skulle planlegge en reserveplan for studentprosjektet. Den foreløpige reserveplanen var å sette opp en fysisk modell av en thruster-motor som kunne kontrolleres fra OS, som i utgangspunktet var satt som et C krav i prosjektet vårt. I samarbeid med oppdragsgiver besluttet vi at i starten av sprint 6 (uke 14) måtte vi ta stilling til om vi skulle iverksette reserveplanen for prosjektet.

Før påske satte KM mer ressurser på å få ordnet opp i de tekniske utfordringene. Det så da positivt ut for å fortsette med den opprinnelige prosjektplanen. Vi avtalte å holde et statusmøte med KM i begynnelsen av sprint 6 for å avklare planen videre.

På grunn av de generelle utfordringene og endringene som oppsto, oppnådde vi en bedre oversikt og forståelse av hele systemet enn det vi kanskje ellers kunne fått. Dette var veldig positivt og ikke minst lærerikt.

Andre presentasjon ble gjennomført og vi fikk gode konstruktive tilbakemeldinger på denne. Intern sensor etterlyste mer teknisk dokumentasjon med referanser, og gjorde oppmerksom på at hver av oss må ha et eget teknisk felt å fordype oss i. Ellers var begge sensorene godt fornøyd med hvordan presentasjonen ble gjennomført.

3.6 Sprint 6

Uke 12, 13 og de to første dagene i uke 14 ble holdt av til påske og forberedelse til eksamen i faget vi hadde ved siden av bacheloroppgaven. Sprint 6 begynte derfor onsdag uke 14 og varte ut uke 15. Siden det andre faget nå var avsluttet, skulle det nå arbeides med prosjektet 5 dager i uken.

I denne sprinten fikk vi en ny azimuth hendelprototype med ekstern skjerm. Det ble bygget en testtrigg til å installere alt utstyret i, og det ble arbeidet mye med å sette opp en god struktur på AutoTesten. Magne kjørte opplæring i K-Chief 600. Alle i gruppen fikk da tilgang til et fullt operativt testanlegg for Wind Innovation som var koblet opp med alle moduler nødvendig for å simulere motor- og alarmstyring av hele skipet. Opplæringen gikk ut på manøvrering i OS, og hvordan man kjører en FAT (Factory Acceptance Test) på anlegget.

Tidligere i prosjektet var modulene, OS og hendelen stående på en pult på kontoret. Dette var greit for testing av hardware tidlig i prosjektet, men vi ønsket å bygge en testtrigg å plassere alle enhetene i. Vi fikk benytte en brokonsoll for skip til dette formålet. Resultatet ble veldig bra med to hendler for å simulere forover og akter på brua. Det ble også montert inn instrumenter som viser feedback på vinkel og pådrag for en azimuth thruster. Hensikten med brokonsollen var å kunne vise fram hendlene og demonstrere AutoTesten på siste presentasjon.

I begynnelsen av sprinten hadde det oppstått nye utfordringer med databasen til hendelen. KM's ingeniører hadde nå fullt fokus på å rette opp feilene, og i løpet av et par dager var databasen operativ.

Vi avtalte et møte med KM på mandag uke 15 for å diskutere videre framdrift i prosjektet, og hvordan vi skulle fordele hvert vårt tekniske bidrag i bacheloroppgaven. Siden det nå så ut til at vi hadde stabil database for hendelen, valgte vi å fortsette planen for utviklingen av AutoTesten. De tekniske bidragene ble fordelt i fire kategorier og bør presenteres i følgende rekkefølge:

1. Automatisert testing i system (med komponenter/feedback og verifisering)
2. Realtimesystemet (K-Chief, OS, DPU)
3. Hendel (Eksisterende/ny type, kommunikasjon, HW, SW)
4. Script av AutoTest (Testprogram, kodespråk, struktur)

3.7 Sprint 7

Sprint 7 var den første med 10 fulle dager hos KM, og var i uke 16 og 17. Dette var den siste sprinten i utviklingsfasen, og fokuset var å ferdigstille så mye som mulig av AutoTesten og den tekniske delen av oppgaven. Det ble ferdigstilt 11 test caser som kunne kjøres sammenhengende i TestComplete. Vi fikk bygd opp et oversiktlig oppsett på scriptet til AutoTesten som det skal være enkelt for KM å bygge videre på når vi overleverer produktet. Begge hendlene i testkonsollen fungerte nå bra med både transfer og slavemodus. Vi fikk tilgang til å lage mimic på OS, det vil si at man kan sette opp et grafisk bilde på skjermen på OS som viser blant annet pådrag og rotasjon. KM var meget behjelpelige med å følge opp de bugs som måtte rettes for at vi skulle komme videre med AutoTesten.

Det ble holdt et møte med KM 19.04.16 for å redefinere bacheloroppgaven på bakgrunn av at forutsetningene for oppgaven har endret seg underveis. Punktene i hovedoppgaven består, mens de mulige utvidelsene og bonusoppgaven utgår. Det ble innført et nytt punkt med beskrivelse av nødvendig utvidelse av oppgaven. Dette er alle oppgaver som har vært nødvendig å utføre på forhånd for å komme videre med å produsere AutoTesten. Grundigere forklaring finnes i prosjektplanen under kapittel 5, samt i møtetreferat fra 29.04.16.

Planleggingen og gjennomføringen av sprinten gikk veldig bra med lite avvik i forhold til estimerte og brukte timer. Tilbakemeldingene på produktet var så langt meget bra fra KM sin side. Det ble også produsert mye teknisk dokumentasjon til rapporten i denne sprinten.

Det var lenge usikkerhet i hvorvidt vi fikk lov å vise hendlene på siste presentasjon, da det ikke var planlagt å lansere de før i august. Under forutsetning at det ble lukket visning, og at det ikke ble tatt bilder på presentasjonen, kunne vi ha en uoffisiell visning av Common Lever.

3.8 Sprint 8

Sprint 8 var starten på overgangsfasen uke 18 og 19. I denne sprinten jobbet vi med å ferdigstille dokumentasjonen. I tillegg til dette, ble det laget en expo plakat, begynt på powerpoint presentasjon og oppdatert hjemmesiden.

Vi hadde problemer med AutoTesten i forhold til hendel bevegelsene. Problemet var at CAN sendte beskjeder syklisk til hendelen, slik at den forstyrret testen vår. Dette ble rettet opp, i tillegg til at det ble gjort endringer i skjermlås funksjonen. Skjermlås funksjonen har tidligere ikke fungert når man har sendt kommandoer via TestComplete. Koden ble derfor endret slik at vi hadde mulighet til å skru av denne funksjonen ved kjøring av AutoTest.

Databasen ble oppdatert samtidig som software på hendelen, og dette resulterte i at mimicen som ble laget til OS ikke fungerte. Her ble det gjort en ny jobb for å få den til å virke, slik at vi har en presentabel visning av feedback til siste presentasjon.

3.9 Sprint 9

Dette var den avsluttende sprinten, med levering av oppgaven og fremføring av den siste presentasjonen. Det ble satt av fire dager til å ferdigstille presentasjonen og dokumentasjonen i uke 20. Mandag uke 21 var frist for innlevering av dokumentasjon, og siste presentasjon var torsdag samme uke.

Vedlegg

1. AutoTest-script
2. Verifikasjons-script
3. Timelister
4. Oppfølgingsdokumenter
5. Møtereferater
6. Intern samarbeidskontrakt for bachelorgruppen
7. Samarbeidskontrakt mellom HSN og KM
8. Tidsplan
9. Gantt diagram

Vedlegg 1 og 2 er kun vedlagt i utgaven tilgjengelig for sensor, og vil derfor være utilgjengelig i den publiserte versjonen av rapporten.

Iterasjon 1

Timeliste									
Navn: Thomas Gjerrud			Uke nr: 2		Dato	Fra: 11.01.16		Til: 17.01.16	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	Scrummøte			1	0,5	0,5			2
	Tidsplan			1					1
	Opplæring			3	3	2			8
	Kravspek				2				2
	Organisering			2		2			4
	Dokumentmal			0,5	2	0,5			3
	Omvisning					2			2
	Rapport			1	1,5	1,5			4
									0
Sum pr dag		0	0	8,5	9	8,5	0	0	26

Timeliste									
Navn: Kristoffer Stanger			Uke nr: 2		Dato	Fra: 11.01.16		Til: 17.01.16	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	ScrumMøter			0,5	0,5	1			2
	Tidsplan			1					1
	Opplæring			1	3	4			8
	Timelister			1					1
	Organisering			1	2	1			4
	Dokumentmaler				1,5	1,5			3
	Org. Kart			1					1
	Omvisning			1	1				2
	Bakgrunnsteori			2	1	1			4
Sum pr dag		0	0	8,5	9	8,5	0	0	26

Timeliste									
Navn:			Uke nr:		Dato	Fra: xx.xx.2016		Til: xx.xx.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	Scrummøte			1	0,5	0,5			2
	Tidsplan			1	2				3
	Opplæring			3	3	2			8
	Kravspek			3	2	3,5			8,5
	Organisering				0,5				0,5
	Omvisning					2			2
									0
									0
									0
Sum pr dag		0	0	8	8	8	0	0	24

Timeliste									
Navn: Philip Vestlie			Uke nr: 2		Dato	Fra: 11.01.2016		Til: 17.01.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	ScrumMøter			1	0,5	0,5			2
	Tidsplan			2	1				3
	Opplæring			3	3	2			8
	Organisering			2	2	1			5
	Dokumentmaler			0,5	2	0,5			3
	Omvisning					2			2
	Bakgrunnsteori				0,5	2,5			3
									0
									0
Sum pr dag		0	0	8,5	9	8,5	0	0	26

Timeliste									
Navn: Thomas Gjerrud			Uke nr: 3		Dato	Fra: 18.01.16		Til: 22.01.16	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	Opplæring			3					3
	Prosjektplan			4	6,5	6,5			17
	Scrummøte			0,5	0,5	0,5			1,5
	Testplan				0,5				0,5
	Veiledermøte		1						1
	Oppfølg.dok					0,5			0,5
									0
									0
									0
Sum pr dag		0	1	7,5	7,5	7,5	0	0	23,5

Timeliste									
Navn: Kristoffer Stanger			Uke nr: 3		Dato	Fra: 18.01.16		Til: 22.01.16	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	Målsetting			1,5	1				2,5
	Avgrensninger			1	1				2
	forutsetninger			1	1				2
	Opplæring			3	1				4
	Scrummøte			1	0,5				1,5
	Prosjektplan				3	5,5			8,5
	Logo					2			2
									0
									0
Sum pr dag		0	0	7,5	7,5	7,5	0	0	22,5

Timeliste									
Navn: Nicolai Sætra			Uke nr: 3		Dato	Fra: 18.01.16		Til: 22.01.16	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	Opplæring	0	0	3	1				4
	Kravspek			3,5	5	5			13,5
	Scrum møte			1	0,5	0,5			2
	Gantt-skjema				0,5				0,5
	Administrativt				0,5	2			2,5
									0
									0
									0
									0
Sum pr dag		0	0	7,5	7,5	7,5	0	0	22,5

Timeliste									
Navn: Philip Vestlie			Uke nr: 3		Dato	Fra: 18.01.16		Til: 24.01.16	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
	Opplæring			3	2				5
	SCRUM			1	1	1			3
	Referat			1					1
	Kontrakt			1	2				3
	prosjektplan			1,5	1,50				3
	admin				1	3,5			4,5
	kravspek					3			3
									0
									0
Sum pr dag		0	0	7,5	7,5	7,5	0	0	22,5

Iterasjon 2

Timeliste									
Navn:			Uke nr: 4		Dato	Fra: 25.01.2016		Til: 29.01.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A3-001	Prosjektplan			2,5	5	5,5			13
A3-005	Oppr. Maler			2					2
A3-009	Adm.av data			3					3
A1-009	Veiledning				2	2			4
A6-003	Int. forb. Pres.				0,5				0,5
									0
									0
									0
									0
Sum pr dag		0	0	7,5	7,5	7,5	0	0	22,5

Timeliste									
Navn: Kristoffer Stanger			Uke nr: 4		Dato	Fra: 25.01.2016		Til: 31.01.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A3-001	Prosjektplan			5,5	7,5	1,5			14,5
A1-002	Internt Veiledermøte	1							1
									0
A3-003	Testplan					3		2	5
A1-010	Logo			2					2
									0
									0
									0
									0
Sum pr dag		1	0	7,5	7,5	4,5	0	2	22,5

Timeliste									
Navn:Nicolai Sætra			Uke nr: 4		Dato	Fra: 25.01.2016		Til: 29.01.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A3-002	Kravspesifikasjon			4	4	2			10
A1-001	Daglig Scrummøte			0,5	0,5	0,5			1,5
A1-002	Internt Veiledermøte	1			1	1			3
A3-001	Prosjektplan			1	2	2			5
A1-009	Veiledning/diskusjon			2	2	2			6
									0
									0
									0
									0
Sum pr dag		1	0	7,5	9,5	7,5	0	0	25,5

Timeliste									
Navn: Philip			Uke nr: 4		Dato	Fra: 25.01.2016		Til: 31.01.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-002	Møte med veilder	1							1
A1-009	Veiledning/ Diskusjon			1	2	2			5
A6-003	Intern forberedelse				2				2
A3-001	Prosjektplan(gantt/modell)				3,5	5,5			9
A1-010	Utforming			1					1
A3-002	Kravspesifikasjon			5,5					5,5
									0
									0
									0
Sum pr dag		1	0	7,5	7,5	7,5	0	0	23,5

Timeliste									
Navn: Thomas Gjerrud			Uke nr: 5		Dato	Fra: 01.02.2016		Til: 06.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Internt Veiledermøte			1					1
A3-001	Prosjektplan			4,5	7,5	4,5	3		19,5
A1-009	Veiledning/diskusjon			2		1			3
A6-003	Intern Forberedelse					2	2		4
									0
									0
									0
									0
Sum pr dag		0,5	0,5	8	8	8	5	0	30

Timeliste									
Navn: Kristoffer Stanger			Uke nr: 5		Dato	Fra: 01.02.2016		Til: 07.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A3-003	Testplan		2,5	6,5	2	3			14
A6-001	Produsere Powerpoint				5,5				5,5
A6-003	Intern Forberedelse					4,5	5		9,5
A1-002	Internt Veiledermøte			1					1
									0
									0
									0
									0
									0
Sum pr dag		0	2,5	7,5	7,5	7,5	5	0	30

Timeliste									
Navn: Nicolai Sætra			Uke nr: 5		Dato	Fra: 01.02.2016		Til: 07.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Internt Veiledermøte			1					1
A1-003	Eksternt Veiledermøte								0
A3-001	Prosjektplan			2		0,5			2,5
A6-001	Produsere Powerpoint				6	4			10
A6-003	Intern Forberedelse					2	5		7
A3-002	Kravspesifikasjon			2,5	1,5	1			5
A1-009	Veiledning/diskusjon			2					2
									0
Sum pr dag		0,5	0,5	8	8	8	5	0	30

Timeliste									
Navn: Philip Vestlie			Uke nr: 5		Dato	Fra: 01.02.2016		Til: 07.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Internt Veiledermøte			1					1
A1-005	Møtereferat Intern Veiledermøte				1				1
A3-001	Prosjektplan			2,5	6,5	1			10
A3-003	Testplan			1		3			4
A6-001	Produsere Powerpoint			1		3,5			4,5
A6-002	Administrativ Forberedelse (invitasjoner etc)			2					2
A1-008	Oppfølgingsdokument							1	1
A6-003	Intern Forberedelse							4	4
Sum pr dag		0,5	0,5	8	8	8	5	0	30

Iterasjon 3

Timeliste									
Navn: Thomas Gjerrud			Uke nr: 6		Dato	Fra: 08.02.2016		Til: 14.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte	0,5	0,5	0,5	0,5				2
A6-001	Produsere Powerpoint			6					6
A6-003	Intern Forberedelse	2	2	1,5	3				8,5
A6-004	Holde Presentasjon				1				1
									0
									0
									0
									0
									0
Sum pr dag		2,5	2,5	8	4,5	0	0	0	17,5

Timeliste									
Navn: Kristoffer Stanger			Uke nr: 6		Dato	Fra: 08.02.2016		Til: 14.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte	0,5	0,5	0,5	0,5				2
A6-001	Produsere Powerpoint			6					6
A6-003	Intern Forberedelse	2	2	1,5	3				8,5
A6-004	Holde Presentasjon				1				1
									0
									0
									0
									0
									0
Sum pr dag		2,5	2,5	8	4,5	0	0	0	17,5

Timeliste									
Navn: Nicolai Sætra			Uke nr: 6		Dato	Fra: 08.02.2016		Til: 14.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte	0,5	0,5	0,5	0,5				2
A1-002	Internt Veiledermøte								0
A1-003	Eksternt Veiledermøte								0
A3-001	Prosjektplan								0
A6-001	Produsere Powerpoint			6					6
A6-003	Intern Forberedelse	2	2	1,5	3				8,5
A3-002	Kravspesifikasjon								0
A1-010	Utforming (Logo, Skilt etc.)						4	10	14
A6-004	Holde Presentasjon				1				1
Sum pr dag		2,5	2,5	8	4,5	0	4	10	31,5

Timeliste									
Navn: Philip Vestlie			Uke nr: 6		Dato	Fra: 08.02.2016		Til: 14.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte	0,5	0,5	0,5	0,5				2
A6-001	Produsere Powerpoint			6					6
A6-003	Intern Forberedelse	2	2	1,5	3				8,5
A6-004	Holde Presentasjon				1				1
A1-008	Oppfølgingsdokument					2			2
									0
									0
									0
									0
Sum pr dag		2,5	2,5	8	4,5	2	0	0	19,5

Timeliste									
Navn: Thomas Gjerrud			Uke nr: 7		Dato	Fra: 15.02.2016		Til: 21.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte			0,5	0,5	0,5			1,5
A2-001	Microsoft Test Manager (MTM)				1	2			3
A2-002	Visual Studio Team Foundation Server (TFS)				1	2			3
A2-003	Test Complete (TC)				4	2,5			6,5
A5-002	Teste Hendel			1,5	1,5	1			4
A5-003	Oppkobling av Utstyr			6					6
									0
									0
									0
Sum pr dag		0	0	8	8	8	0	0	24

Timeliste									
Navn: Kristoffer Stanger			Uke nr: 7		Dato	Fra: 15.02.2016		Til: 21.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte			0,5	0,5	0,5			1,5
A2-001	Microsoft Test Manager (MTM)				1	2			3
A2-002	Visual Studio Team Foundation Server (TFS)				1	2			3
A2-003	Test Complete (TC)				4	2,5			6,5
A5-002	Teste Hendel			1,5	1,5	1			4
A5-003	Oppkobling av Utstyr			6					6
									0
									0
									0
Sum pr dag		0	0	8	8	8	0	0	24

Timeliste									
Navn: Nicolai Sætra			Uke nr: 7		Dato	Fra: 15.02.2016		Til: 21.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte			0,5	0,5	0,5			1,5
A2-001	Microsoft Test Manager (MTM)				1	2			3
A2-002	Visual Studio Team Foundation Server (TFS)				1	2			3
A2-003	Test Complete (TC)				4	2,5			6,5
A5-002	Teste Hendel			1,5	1,5	1			4
A5-003	Oppkobling av Utstyr			6					6
									0
									0
									0
Sum pr dag		0	0	8	8	8	0	0	24

Timeliste									
Navn: Philip Vestlie			Uke nr: 7		Dato	Fra: 15.02.2016		Til: 21.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig Scrummøte			0,5	0,5	0,5			1,5
A2-001	Microsoft Test Manager (MTM)				1	2			3
A2-002	Visual Studio Team Foundation Server (TFS)				1	2			3
A2-003	Test Complete (TC)				4	2,5			6,5
A5-002	Teste Hendel			1,5	1,5	1			4
A5-003	Oppkobling av Utstyr			6					6
									0
									0
									0
Sum pr dag		0	0	8	8	8	0	0	24

Sprint 4

Timeliste									
Navn: Thomas Gjerrud			Uke nr: 8		Dato	Fra: 22.02.2016		Til: 28.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang			3					3
A1-003	Sprint Oppfølgingsdokumentasjon			4					4
A3-001	Prosjektplan				3	6,5			9,5
A4-001	Auto-Test scripting				4				4
									0
									0
									0
									0
Sum pr dag		0	0	7,5	7,5	7	0	0	22

Timeliste									
Navn: Kristoffer Stanger			Uke nr: 8		Dato	Fra: 22.02.2016		Til: 28.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
				Syk	Syk	Syk			0
									0
									0
									0
									0
									0
									0
									0
									0
Sum pr dag		0	0	0	0	0	0	0	0

Timeliste									
Navn: Nicolai Sætra			Uke nr: 8		Dato	Fra: 22.02.2016		Til: 28.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang			1					1
A4-001	Auto-Test scripting			4	3	2,5			9,5
A3-002	User Story (Krav)				2	4			6
A2-001	Opplæring Software			2	2				4
									0
									0
									0
									0
Sum pr dag		0	0	7,5	7,5	7	0	0	22

Timeliste									
Navn: Philip Vestlie			Uke nr: 8		Dato	Fra: 22.02.2016		Til: 28.02.2016	
Aktivitets nr.	Aktivitet	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang			1					1
A3-001	Prosjektplan			6	2	6,5			14,5
A4-001	Auto-Test scripting				5				5
									0
									0
									0
									0
									0
Sum pr dag		0	0	7,5	7,5	7	0	0	22

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 9		Dato	Fra: 29.02.2016		Til: 04.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	1		0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237			2		3			5
A4-001	Auto-Test scripting	62280			4					4
A5-003	Oppkobling av Utstyr	62223				1,5				1,5
A3-002	User Story (Krav)	62281			1,5	6	2,5			10
A6-002	Forberedelse	62225					2			2
										0
										0
										0
Sum pr dag			1	0	8	8	8	0	0	25

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 9		Dato	Fra: 29.03.2016		Til: 06.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	1		0,5	0,5	0,5			2,5
A4-001	Auto-Test scripting	62280			5,5	7,5	7,5			20,5
A4-002	Kode til verifisering	62239			2					2
										0
										0
										0
										0
										0
Sum pr dag			1	0	8	8	8	0	0	25

Timeliste										
Navn: Nicolai Sætra				Uke nr: 9		Dato	Fra: 29.02.2016		Til: 06.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	1		0,5	0,5	0,5			2,5
A4-001	Auto-Test scripting	62240			3,5	4,5	6,5			14,5
A4-001	Auto-Test scripting	62280			4					4
A2-001	Opplæring Software	62220				3	1			4
										0
										0
										0
										0
Sum pr dag			1	0	8	8	8	0	0	25

Timeliste										
Navn: Philip Vestlie				Uke nr: 9		Dato	Fra: 29.01.2016		Til: 06.02.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	1		0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237					3			3
A1-003	Sprint Oppfølgingsdokumentasjon	62238			7,5					7,5
A5-003	Oppkobling av Utstyr	62223				3				3
A6-002	Forberedelse	62225					3,5			3,5
A3-002	User Story (Krav)	62281				2	1			3
A2-001	Opplæring Software	62220				2,5				2,5
										0
										0
Sum pr dag			1	0	8	8	8	0	0	25

Sprint 5

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 10		Dato	Fra: 07.03.2016		Til: 13.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237			2					2
A6-002	Forberedelse Presentasjon	62225			1		3			4
A6-001	Produsere Powerpoint	62224			3,5	3,5				7
A3-001	Prosjektplan	62226				4	4,5			8,5
										0
										0
										0
										0
Sum pr dag			0	0	7	8	8	0	0	23

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 10		Dato	Fra: 07.03.2016		Til: 13.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237			0,5					0,5
A6-001	Produsere Powerpoint	62224			5	3,5	1			9,5
A6-002	Forberedelse	62225			1	4	2			7
										0
										0
										0
										0
										0
Sum pr dag			0	0	7	8	3,5	0	0	18,5

Timeliste										
Navn: Nicolai Sætra				Uke nr: 10		Dato	Fra: 07.03.2016		Til: 13.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237			0,5					0,5
A6-001	Produsere Powerpoint	62224			5	3,5	1			9,5
A6-002	Forberedelse	62225			1	4	2			7
										0
										0
										0
										0
										0
Sum pr dag			0	0	7	8	3,5	0	0	18,5

Timeliste										
Navn: Philip Vestlie				Uke nr: 10		Dato	Fra: 07.03.2016		Til: 13.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237			3					3
A1-003	Sprint Oppfølgningsdokumentasjon	62238								0
A6-001	Produsere Powerpoint	62224			3,5	7,5				11
A6-002	Forberedelse	62225					7,5			7,5
										0
										0
										0
										0
Sum pr dag			0	0	7	8	8	0	0	23

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 11		Dato	Fra: 14.03.2016		Til: 20.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237					1			1
A6-002	Forberedelse Presentasjon	62225			6,5	5,5				12
A6-001	Produsere Powerpoint	62224			1					1
A6-003	Holde Presentasjon	62236				1				1
A1-003	Sprint Oppfølgingsdokumentasjon	62238					2			2
A4-001	Auto-Test scripting						3,5			3,5
										0
										0
										0
Sum pr dag			0	0	8	7	7	0	0	22

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 11		Dato	Fra: 14.03.2016		Til: 20.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A5-002	Teste Hendel	62810					6,5			6,5
A6-001	Produsere Powerpoint	62224			7,5	3,5				11
A6-002	Forberedelse	62225				4				4
										0
										0
										0
										0
										0
Sum pr dag			0	0	8	8	7	0	0	23

Timeliste										
Navn: Nicolai Sætra				Uke nr: 11		Dato	Fra: 14.03.2016		Til: 20.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomg	62237					1			1
A1-003	Sprint Oppfølgingsdokumentasjon						1			1
A6-002	Forberedelse	62225			6,5	5,5				12
A6-003	Holde Presentasjon	62236				1				1
A6-001	Produsere Powerpoint	62224			1					1
A5-002	Teste Hendel						4,5			4,5
										0
										0
Sum pr dag			0	0	8	7	7	0	0	22

Timeliste										
Navn: Philip Vestlie				Uke nr: 11		Dato	Fra: 14.03.2016		Til: 20.03.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237					1			1
A6-001	Produsere Powerpoint	62224			2					2
A6-002	Forberedelse	62225			5,5	5,5				11
A6-003	Holde Presentasjon	62236				1				1
A4-001	Auto-Test scripting	61330					5,5			5,5
										0
										0
										0
Sum pr dag			0	0	8	7	7	0	0	22

Sprint 6

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 14		Dato	Fra: 04.04.2016		Til: 10.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237				1	1			2
A2-001	Opplæring Software	62220				6,5				6,5
A3-006	Rapport	63943			3,5		5,5			9
A5-001	Teste Script	62220			2					2
A5-002	Teste Hendel	62223			2					2
										0
										0
										0
Sum pr dag			0	0	8	8	7	0	0	23

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 14		Dato	Fra: 04.04.2016		Til: 10.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237				1	1			2
A2-001	Opplæring Software	62220				6,5				6,5
A3-006	Rapport	63943			3,5		5,5			9
A5-001	Teste Script	62220			2					2
A5-002	Teste Hendel	62223			2					2
										0
										0
										0
Sum pr dag			0	0	8	8	7	0	0	23

Timeliste										
Navn: Nicolai Sætra				Uke nr: 14		Dato	Fra: 04.04.2016		Til: 10.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5	0,5			1,5
A1-002	Sprint Planlegging og Gjennomgang	62237				1	1			2
A2-001	Opplæring Software	62220				6,5				6,5
A3-006	Rapport	63943			3,5		5,5			9
A5-001	Teste Script	62220			2					2
A5-002	Teste Hendel	62223			2					2
										0
										0
										0
Sum pr dag			0	0	8	8	7	0	0	23

Timeliste										
Navn: Philip Vestlie				Uke nr: 14		Dato	Fra: 04.04.2016		Til: 10.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279			0,5	0,5				1
A1-002	Sprint Planlegging og Gjennomgang	62237				1				1
A2-001	Opplæring Software	62220				6,5				6,5
A3-006	Rapport	63943			3,5					3,5
A5-001	Teste Script	62220			2					2
A5-002	Teste Hendel	62223			2					2
										0
										0
										0
Sum pr dag			0	0	8	8	0	0	0	16

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 15		Dato	Fra: 11.04.2016		Til: 17.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237					1,5			1,5
A4-001	Auto-Test scripting	62067			4,5					4,5
A3-006	Rapport	63881	7,5	7,5						15
A4-001	Auto-Test scripting	62068			3					3
A4-001	Auto-Test scripting	63991				7,5	1			8,5
A1-004	Generelt Møte	62361					3			3
A5-005	Oppkobling av utstyr	64062					2			2
										0
Sum pr dag			8	8	8	8	8	0	0	40

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 15		Dato	Fra: 11.04.2016		Til: 17.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomg	62237					1			1
A4-001	Auto-Test scripting	63858,63857,63855	6	5,5						11,5
A3-002	User Story (Krav)	63941		4						4
A5-001	Teste Script		5							5
A2-002	Opplæring/Rigg Hardware	63961,64008			9,5	9,5	3			22
A1-xxx	Omvisning med Antonio	62361					3			3
										0
										0
Sum pr dag			11,5	10	10	10	7,5	0	0	49

Timeliste										
Navn: Nicolai Sætra				Uke nr: 15		Dato	Fra: 11.04.2016		Til: 17.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomg	62237					1			1
A2-002	Opplæring/Rigg Hardware	63964			4	1	2			7
A3-006	Rapport	63892	3	2,5	2					7,5
A4-001	Auto-Test scripting	63947,63946,63945	4,5	5	1,5	6,5				17,5
A5-002	Teste Hendel	64063					1,5			1,5
A1-004	Generelt Møte	62361					3			3
										0
Sum pr dag			8	8	8	8	8	0	0	40

Timeliste										
Navn: Philip Vestlie				Uke nr: 15		Dato	Fra: 11.04.2016		Til: 17.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237					1,5			1,5
A3-006	Rapport	63942	5,5	7,5	5,5					18,5
A4-001	Auto-Test scripting	62068			2					2
A5-002	Teste Hendel	62812	2				1			3
A1-004	Generelt Møte	62361					3			3
A5-005	Oppkobling av utstyr	64007				7,5	2			9,5
Sum pr dag			8	8	8	8	8	0	0	40

Sprint 7

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 16		Dato	Fra: 18.04.2016		Til: 24.04.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging	62237	2							2
A1-004	Generelt Møte	64165		1						1
A3-001	Prosjektplan	62226	5,5	6,5	7,5					19,5
A4-001	Auto-Test scripting	64250				3				3
A5-001	Teste Script	64273,64276,64277				4,5	5,5			10
										0
										0
										0
Sum pr dag			8	8	8	8	6	0	0	38

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 16		Dato	Fra: 18.04.2016		Til: 24.05.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomg	62237	1							1
A3-006	Rapport	64136	6,5	5	7,5	8,5	5			32,5
A4-002	Kode til verifisering			2,5			1,5			4
										0
										0
										0
										0
										0
Sum pr dag			8	8	8	9	7	0	0	40

Timeliste										
Navn: Nicolai Sætra				Uke nr: 16		Dato	Fra: 18.04.2016		Til: 01.05.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237	1							1
A3-006	Rapport	64296	2,5				5,5			8
A1-004	Generelt møte	64165		3						3
A4-001	Auto-Test scripting	64181, 64182	4	3,5	7,5	5,5				20,5
A5-002	Teste Hendel	64063		1						1
Sum pr dag			8	8	8	6	6	0	0	36

Timeliste										
Navn: Philip Vestlie				Uke nr: 16		Dato	Fra: 18.04.2016		Til: 01.05.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237	1							1
A1-003	Sprint Oppfølgingsdokumentasjon	62238	4							4
A3-006	Rapport	63942	2,5							2,5
A1-004	Generelt møte	64165		3						3
A3-006	Rapport	64210		4,5	6,5	7,5				18,5
A5-001	Teste Script	63892					3,5			
A5-002	Teste Hendel	62812			1		2			3
Sum pr dag			8	8	8	8	6	0	0	34,5

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 17		Dato	Fra: 25.04.2016		Til: 01.05.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Oppfølgingsmøte	62237								0
A3-003	Testplan	64401	5,5	7,5						13
A5-002	Teste Hendel	64372	2							2
A4-001	Auto-Test scripting	64458, 64488			7,5					7,5
A4-001	Auto-Test scripting	63991, 64250				7				7
A4-001	Auto-Test scripting	63990, 64149					7,5			7,5
										0
										0
Sum pr dag			8	8	8	7,5	8	0	0	39,5

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 17		Dato	Fra: 25.04.2016		Til: 01.05.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A3-006	Rapport	64136								0
A4-001	Auto-Test scripting		6	8,5	7,5	7,5	2,5			32
A4-002	Kode til verifisering		2,5				3			5,5
										0
										0
										0
										0
										0
Sum pr dag			9	9	8	8	6	0	0	40

Timeliste										
Navn: Nicolai Sætra				Uke nr: 17		Dato	Fra: 25.04.2016		Til: 01.05.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Oppfølgingsmøte	62237								0
A3-006	Rapport	64296	5,5							5,5
A5-002	Teste Hendel	64372	2							2
A4-001	Auto-Test scripting	64458, 64488			7,5					7,5
A4-001	Auto-Test scripting	63991, 64250				7				7
A4-001	Auto-Test scripting	63990, 64149					7,5			7,5
A4-001	Auto-Test scripting	64244		7,5						7,5
										0
Sum pr dag			8	8	8	7,5	8	0	0	39,5

Timeliste										
Navn: Philip Vestlie				Uke nr: 17			Dato	Fra: 25.04.2016		Til: 01.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A5-002	Teste Hendel	62812	4							4
A3-006	Rapport	63942	3,5							3,5
A3-006	Rapport	64210		7,5						7,5
A4-001	Auto-Test scripting	64602			7,5					7,5
A4-001	Auto-Test scripting	64604				7,5				7,5
A4-001	Auto-Test scripting	64555					7,5			7,5
										0
										0
Sum pr dag			8	8	8	8	8	0	0	40

Sprint 8

Timeliste										
Navn: Thomas Gjerrud				Uke nr: 18			Dato	Fra: 02.05.2016		Til: 08.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5					1,5
A1-002	Sprint Oppfølging og gjennomgang	62237	2							2
A1-003	Sprint Oppfølgingsdokumentasjon	64617	2,5							2,5
A3-006	Rapport	63881	3	5						8
A3-003	Testplan	64401		2,5	7,5					10
A3-006	Rapport	64728					6			6
										0
										0
										0
Sum pr dag			8	8	8	0	6	0	0	30

Timeliste										
Navn: Kristoffer Stanger				Uke nr: 18			Dato	Fra: 02.05.2016		Til: 08.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237	2							2
A3-006	Rapport	63941,64226,64136				5	7,5	7,5		20
A4-001	Auto-Test scripting	64906,64907	5,5	7,5						13
A4-002	Kode til verifisering	64310,64311,64313,64459				2,5				2,5
										0
										0
										0
										0
Sum pr dag			8	8	8	8	8	0	0	40

Timeliste										
Navn: Nicolai Sætra				Uke nr: 18			Dato	Fra: 02.05.2016		Til: 08.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5					1,5
A1-002	Sprint Oppfølging og gjennomgang	62237	2							2
A3-002	User Story (Krav)	64890	2,5							2,5
A3-006	Rapport	64296,6489	3	7,5	7,5	6	4			28
A3-004	Hjemmeside	64888				2	4			6
										0
										0
										0
										0
Sum pr dag			8	8	8	8	8	0	0	40

Timeliste											
Navn: Philip Vestlie				Uke nr: 18			Dato		Fra: 02.05.2016		Til: 08.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5	0,5			2,5
A1-002	Sprint Planlegging og Gjennomgang	62237	2								2
A4-001	Auto-Test scripting	64555	4								4
A1-xxx	Mail, invitasjoner, oppfølging av sensorer etc.	64759	1,5								1,5
A3-006	Rapport	64756		7,5							7,5
A5-002	Teste Hendel	64555				6					6
A3-006	Rapport	64210				2	7,5				9,5
A3-006	Rapport	64758						7,5			7,5
											0
Sum pr dag			8	8	8,5	8	8	0	0		40,5

Timeliste											
Navn: Thomas Gjerrud				Uke nr: 19			Dato		Fra: 09.05.2016		Til: 15.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5	0,5			3
A1-002	Sprint Oppfølging og gjennomgang	62237									0
A1-003	Sprint Oppfølgingsdokumentasjon	64617									0
A3-006	Rapport	63881,63941				2	1,5				3,5
A3-003	Testplan	62227			7,5	2					9,5
A3-006	Rapport	64728,62226	7,5	7,5		3,5	4				22,5
A3-006	Rapport	63881					2	4,5			6,5
											0
											0
Sum pr dag			8	8	8	8	8	5	0		45

Timeliste											
Navn: Kristoffer Stanger				Uke nr: 19			Dato		Fra: 09.05.2016		Til: 15.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5				2,5
A1-002	Sprint Planlegging og Gjennomgang		2								2
A3-006	Rapport	63941,64226,64136	5,5	7,5	7,5	7,5	7,5	5			40,5
											0
											0
											0
											0
											0
											0
Sum pr dag			8	8	8	8	8	5	0		45

Timeliste											
Navn: Nicolai Sætra				Uke nr: 19			Dato		Fra: 09.05.2016		Til: 15.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5						1,5
A1-002	Sprint Oppfølging og gjennomgang	62237	2								2
A3-004	Hjemmeside	64888	2								2
A3-006	Rapport	64890	3,5	5,5		5	5,5	5			24,5
A3-007	Expo-plakat	64891		2	7,5						9,5
A4-001	Auto-Test scripting	64244				2	2				4
A5-001	Teste Script	64892					0,5				0,5
A5-002	Teste Hendel					1					1
											0
Sum pr dag			8	8	8	8	8	5	0		45

Timeliste											
Navn: Philip Vestlie				Uke nr: 19			Dato	Fra: 09.05.2016			Til: 15.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5	0,5	0,5	0,5	0,5	0,5		3	
A1-002	Sprint Planlegging og Gjennomgang	62237						1		1	
A5-001	Teste Script	64908		2	3					5	
A3-006	Rapport	64210	7,5	3	4,5	5,5	5,5	2		28	
A3-006	Rapport	64790		2,5						2,5	
A3-006	Rapport	64756				2	2	1,5		5,5	
										0	
										0	
										0	
Sum pr dag			8	8	8	8	8	5	0	45	

Sprint 9

Timeliste											
Navn: Thomas Gjerrud				Uke nr: 20			Dato	Fra: 16.05.2016			Til: 22.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5		0,5	0,5	0,5			2	
A3-006	Rapport		5,5		11	11	7,5			35	
										0	
										0	
										0	
										0	
										0	
										0	
										0	
Sum pr dag			6	0	11,5	11,5	8	0	0	37	

Timeliste											
Navn: Kristoffer Stanger				Uke nr: 20			Dato	Fra: 16.05.2016			Til: 22.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5		0,5	0,5	0,5			2	
A1-002	Sprint Planlegging og Gjennomgang									0	
A3-006	Rapport	62202	5,5		10	11	7,5			34	
										0	
										0	
										0	
										0	
										0	
Sum pr dag			6	0	10,5	11,5	8	0	0	36	

Timeliste											
Navn: Nicolai Sætra				Uke nr: 20			Dato	Fra: 16.05.2016			Til: 22.05.2016
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet	
A1-001	Daglig SCRUM møte	62279	0,5		0,5	0,5	0,5			2	
A1-002	Sprint Oppfølging og gjennomgang	62237	2							2	
A3-004	Hjemmeside	64888								0	
A3-006	Rapport	64890	3,5		11	9	7,5			31	
A6-001	Produsere Powerpoint					2				2	
										0	
										0	
Sum pr dag			6	0	11,5	11,5	8	0	0	37	

Timeliste										
Navn: Philip Vestlie				Uke nr: 20		Dato	Fra: 16.05.2016		Til: 22.05.2016	
Aktivitets nr.	Aktivitet	Work Item ID	Man	Tir	Ons	Tor	Fre	Lør	Søn	Sum pr aktivitet
A1-001	Daglig SCRUM møte	62279	0,5		0,5	0,5	0,5			2
A1-002	Sprint Planlegging og Gjennomgang	62237	1							1
A3-006	Rapport	64756	2		3	3	1,5			9,5
A3-006	Rapport	64210	2,5		8	3				13,5
A6-001	Produsere Powerpoint	64909				5	4			9
A6-002	Forberedelse	64911					2			2
										0
										0
										0
Sum pr dag			6	0	11,5	11,5	8	0	0	37

Total timeberegning

	Iterasjon 1	Iterasjon 2	Iterasjon 3	Sprint 4	sprint 5	sprint 6	sprint 7	sprint 8	sprint 9	Totalt
Philip	48,5	53,5	43,5	47	45	56	77	85	64	519,5
Nicolai	48,5	53,5	55,5	47	40,5	63	75,5	87	64	534,5
Thomas	48,5	53,5	41,5	47	45	63	77,5	85	64	525
Kristoffer	48,5	53,5	41,5	25	41,5	72	80	80	64	506
Sum	194	214	182	188	172	254	310	337	256	2107

Utført Aktivitet											
Aktivitet Nummer:	Aktivitet:	Estimert tid:	Dato fra: til:					Total tid brukt:	Avvik:	Årsak:	Tiltak:
			PV	TG	KS	NS	Sum:				
	ScrumMøter		2	2	2	2	8				
	Tidsplan		3	1	1	1	6				
	OppfølgningsDok.					3	3				
	Opplæring		8	8	8	8	32				
	Kravspesifikasjon			2			2				
	Timelister				1		1				
	Organisering		5	4	4	4	17				
	Dokumentmaler		3	3	3	3	12				
	Organisasjons kart				1		1				
	Møter						0				
	Omvising		2	2	2	2	8				
	Bakgrunnsteori		3	3	4	3	13				
	Rapport			1			1				
							0				
							0				
							0				
							0				
							0				
							0				
							0				
		Sum	26	26	26	26					
Sum Akkumulert tid:							104				
Sum Forbrukt tid totalt:							0				

[illegible]

Vi er godt i gang med oppstart av prosjektet og har fått etablert oss hos oppdragsgiver, og delvis etablert oss på skolen. Vi har fått en oversikt over oppgaven slik at vi vet hva den går ut på. Møtestruktur med oppdragsgiver er avklart, med daglige Scrum møter og ukentlige planleggingsmøter. Det er beregnet en del tid til kursing og opplæring i oppdragsgivers system til uken.

Bachelorgruppe 4 - 2016 - Common Lever AutoTest

Utført Aktivitet											
Aktivitet Nummer:	Aktivitet:	Estimert tid:	Uke: 3 Dato fra: 20.01.2015 til: 22.01.2015					Total tid brukt:	Avvik:	Årsak:	Tiltak:
			PV	TG	KS	NS	Sum:				
	ScrumMøter		1,5	1,5	1,5	2	6,5				
	Tidsplan					0,5	0,5				
	OppfølgingsDo						0				
	Opplæring		5	3	4	4	16				
	Kravspesifikasjo		3			13,5	16,5				
	Timelister						0				
	Organisering						0				
	Dokumentmaler						0				
	Organisasjons ka						0				
	Møter		1,5	1,5			3				
	Omvisning						0				
	Bakgrunnsteori						0				
	Rapport						0				
	Prosjektplan		4	16	15		35				
	Testplan			0,5			0,5				
	administrativt		7,5			2	2,5	12			
								0			
								0			
								0			
								0			
		Sum	22,5	22,5	22,5	22,5					
Sum Akkumulert tid:								90			
Sum Forbrukt tid totalt:								0			

Denne uken har vi jobbet mest med utvikling av kravspesifikasjon, prosjektplan og testplan. Vi har utarbeidet en prosjektmodell. Det ble mindre opplæring enn forventet.

[illegible]

Utført Aktivitet											
Aktivitet Numme	Aktivitet:	Estimert tid	Dato fra: til:					Total tid brukt:	Avvik:	Årsak:	Tiltak:
			PV	TG	KS	NS	Sum:				
A3-001	Prosjektplan		9	13	14,5	5	41,5		-41,5		
A3-002	Kravspek		5,5			10	15,5		-15,5		
A3-005	Oppr. Maler			3			3		-3		
A1-009	Veiledning		5	4		6	15		-15		
A6-003	Int. Forb. Pres.		2	0,5			2,5		-2,5		
A3-009	Adm.av data			2			2		-2		
A1-001	Daglig scrummøte					1,5	1,5		-1,5		
A1-002	Internt Veilederm		1	1	1	3	6		-6		
A1-010	Utforming		1		2		3		-3		
A3-003	Testplan				6		6		-6		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
							0		0		
		Sum	23,5	23,5	23,5	25,5					
Sum Akkumulert tid:							96				
Sum Forbrukt tid totalt:							0				

Utførte aktiviteter for uke 5												
Aktivitet Nr:	Aktivitet:	Ansvarlig:	Estimert tid:	Dato fra: 01.02.2016 til: 07.02.2016					Total tid brukt:	Avvik:	Årsak:	Tiltak:
				PV	TG	KS	NS	Sum:				
A1-001	Daglig Scrummøte	Alle	2	2,5	2,5		2,5	7,5		-5,5		
A1-002	Intern veiledermøte	PV	4	1	1	1	1	4		0		
A1-005	Møtereferat intern	TG	1	1				1		0		
A3-001	Prosjektplan	TG	29	10	19,5		2,5	32		-3		
A3-002	Kravspek	NS	25				5	5		20		
A3-003	Testplan	KS	4	4		14		18		-14		
A6-001	Produsere PP	PV	30	4,5		5,5	10	20		10		
A6-002	Adm. Forberedelser	PV	1	2				2		-1		
A1-008	Oppfølgingsdokument	Alle	1	1				1		0		
A6-003	Intern Forberedelse			4	4	9,5	7	24,5		-24,5		
A1-009	Diskusjon/Veiledning				3		2	5		-5		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
		Sum	97	30	30	30	30					
Sum Akkumulert tid:								120				
Sum Forbrukt tid totalt:									0			

Fokuset denne uken har vært på prosjektplanen med prosjektmodell og gantt. Det har også blitt jobbet med kravspesifikasjonen og testplanen. Etter råd fra veileder valgte prosjektgruppen og dreie fokuset fra opplæring hos KM over på dokumentasjon som HSN krever til den første presentasjonen.

Aktivitetsnummer er også opprettet, slik at vi lettere skal kunne se hva prosjekttimene går til.

Dette er med på å gjøre det lettere for oss i prosjektgruppen og planlegge og estimere hva vi kommer til å bruke tid på og hva som tar lengere eventuelt kortere tid enn hva vi hadde kalkulert med.

Status:

Vi ser at det gikk med en del mer tid på scrum-møter, kravspesifikasjon og testplanen.

Det har også gått med tid til produksjon av den første presentasjonen og dokumentasjon som skal leveres inn i forkant av presentasjonen.

Det er planlagt at mye av prosjekttiden i uke 6 går med på forberedelse og gjennomføring av presentasjon 1.

Oppfølgningsdokument uke 7

Planlagte aktiviteter:

Planlagte aktiviteter for uke 7												
Aktivitet Nr:	Aktivitet:	Ansvarlig:	Estimert tid:	Dato fra: 15.02.2016 til: 21.02.2016				Sum:	Total tid brukt:	Avvik:	Årsak:	Tiltak:
				PV	TG	KS	NS					
A1-001	Daglig Scrummøte	PV	6					0		6		
A1-002	Internt Veiledermøte	TG	4					0		4		
A2-xxx	Opplæring	Alle	62					0		62		
A5-002	Teste Hendel	NS	12					0		12		
A5-003	Oppkobling av Utstyr	KS	12					0		12		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		
								0		0		

Status:

Uke 6 har gått med på å utarbeide den første presentasjonen av oppgaven og gjennomføring av denne. Vi fikk tilgang til server for å opprette hjemmeside til oppgaven, og det gikk med mange timer på å utvikle denne.

Uke 7 er starten på vår utviklingsfase i prosjektet og vi vil derfor ha en del opplæring hos KM og oppkobling og rigging av hendel og tilhørende utstyr.

Vi fikk også tilbakemelding på presentasjonen, om at vi bør revurdere modellen vår siden vi jobber opp mot KM som mer eller mindre jobber opp mot en SCRUM-modell. Dette for å spare tid og slippe å måtte dokumentere dobbelt opp.

Planlegging sprint 4 uke 8 og 9

Planlagte aktiviteter:

Planlagte aktiviteter for Sprint 4 (uke 8,9)									
Aktivitet Nr:	Aktivitet:	Estimert tid:	Dato fra: 22.02.2016 til: 06.03.2016				Sum:	Avvik:	Årsak:
			PV	TG	KS	NS			
A1-001	Daglig SCRUM møte	12					0	12	
A1-002	Sprint Planlegging og Gjennomgang	12					0	12	
A1-003	Sprint Oppfølgingsdokumentasjon	2					0	2	
A2-001	Opplæring Software	64					0	64	
A4-001	Auto-Test scripting	78					0	78	
A3-001	Prosjektplan	24					0	24	
							0	0	
							0	0	
							0	0	
							0	0	
Sum		192	0	0	0	0			
Total tid							0		

Status:

I denne sprinten er det planlagt å koble opp hendelen mot ROS, MEI'r og SCU.

Det er satt av tid til opplæring i TC og TFS.

Det er også planlagt at Magne oppretter Test Caser som vi skal utvikle som user stories.

KM har ikke kommet så langt i utviklingen av i prosjektet med hendelen og prosjektet vårt blir påvirket slik at vi i større grad blir en del av nyutviklingen av funksjonalitet og hardware.

Gjennomgang sprint 4 uke 8 og 9

Utførte aktiviteter:

Utførte aktiviteter for Sprint 4 (uke 8,9)									
Aktivitet Nr:	Aktivitet:	Estimert tid:	Dato fra: 22.02.2016 til: 06.03.2016				Sum:	Avvik:	Årsak:
			PV	TG	KS	NS			
A1-001	Daglig SCRUM møte	12	4	4	2,5	4	14,5	-2,5	
A1-002	Sprint Planlegging og Gjennomgang	12	4	8		1	13	-1	
A1-003	Sprint Oppfølgingsdokumentasjon	2	7,5	4			11,5	-9,5	
A2-001	Opplæring Software	64	2,5			8	10,5	53,5	
A4-001	Auto-Test scripting	78	5	8	20,5	28	61,5	16,5	
A3-001	Prosjektplan	24	14,5	9,5			24	0	
A4-002	Kode til verifisering				2		2	-2	
A3-002	User Story (Krav)		3	10		6	19	-19	
A5-003	Oppkobling av Utstyr		3	1,5			4,5	-4,5	
A6-002	Forberedelse		3,5	2			5,5	-5,5	
Sum		192	47	47	25	47			
Total tid							166		

Status:

Oppkobling av hendelen mot ROS, MEI'r og SCU er utført.

Det oppsto utfordringer med software mot hardware-komponentene, som har resultert i at mye tid har gått med til feilsøking og jobb mot utviklingsavdelingen.

Vi har fått tilgang til Test Caser og opprettet user stories. Oppdaget mangelfull kommunikasjon mot hendelen ved scripting. Dette ble tatt videre med utviklingsavdelingen og Magne opprettet to nye Test Caser for bevegelse og rotasjon, hvor kommunikasjon mot hendelen fungerer.

I godt samarbeid med oppdragsgiver ser vi at prosjektet går fra å være en ren scriptings-oppgave til å i større grad omhandle testing og nyutvikling av funksjonalitet og hardware.

Vi har fått alle tilganger til TFS, MTM og TC. Det er gått med tid til å sette oss inn i systemene og hvordan vi skal bruke disse.

Det er opprettet User Stories av Test Caser og User Stories er delt opp i flere mindre Tasker.

Vi anser sprint 4 som en prøvesprint, for å se hvor lang tid som går med til scripting og utfordringer med programmene vi bruker. I sprint 5 kjører vi fullt etter scrum metodikken.

Planlegging sprint 5 uke 10 og 11

Planlagte aktiviteter:

Planlagte aktiviteter for Sprint 5 (uke 10,11)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 07.03.2016 til: 13.03.2016				Sum:	Avvik:
				PV	TG	KS	NS		
A1-001	Daglig SCRUM møte	62279	12					0	12
A1-002	Sprint Planlegging og Gjennomgang	62237	4					0	4
A1-003	Sprint Oppfølgingsdokumentasjon	62238	4					0	4
A6-001	Produsere Powerpoint	62224	24					0	24
A6-002	Forberedelse Presentasjon	62225	34					0	34
A6-003	Holde Presentasjon	62236	8					0	8
A3-006	Rapport	62360	38					0	38
A4-001	Auto-Test scripting	62240, 62280	50					0	50
A1-xxx	Omvisning Antonio hos KM	62360	6					0	6
								0	0
Sum			180	0	0	0	0		
Total tid								0	

Status:

Denne sprinten består av en presentasjon Torsdag i uke 11.

Presentasjonen skal ha et teknisk fokus og gi en oversikt over hva som er gjort og hvordan prosjektet ligger an.

Det vil derfor bli fokusert på å produsere presentasjon og oppdatere dokumentasjon mot HSN.

De to User Stories med bevegelse og rotasjon tas med videre fra sprint 4 og vi beregner å fullføre de i denne sprinten. VI har fått en ny versjon av Remote Test Interface dokumentet som omhandler hvilke funksjoner vi kan sette, spørre etter og få svar på. Det vil da ligge en god del jobb i å se om dette fungerer som det skal og hvilke User Stories vi da kan jobbe videre med.

I samarbeid med oppdragsgiver har vi besluttet at i oppstarten av sprint 6 uke 14 må ta stilling til om vi skal iverksette reserveplanen, da vi ser at prosjektet får et større fokus på utvikling av funksjonalitet og hardware.

Reserveplanen er foreløpig å sette opp en fysisk modell av en thruster-motor som kan kontrolleres fra ROS, som i utgangspunktet er satt som C krav i prosjektet vårt.

Gjennomgang sprint 5, uke 10 og 11

Utførte aktiviteter:

Utførte aktiviteter for Sprint 5 (uke 10,11)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 07.03.2016 til: 20.03.2016					Avvik:
				PV	TG	KS	NS	Sum:	
A1-001	Daglig SCRUM møte	62279	12	3	3	3	3	12	0
A1-002	Sprint Planlegging og Gjennomgang	62237	4	4	3	0,5	1,5	9	-5
A1-003	Sprint Oppfølgingsdokumentasjon	62238	4		2		1	3	1
A6-001	Produsere Powerpoint	62224	24	13	8	20,5	10,5	52	-28
A6-002	Forberedelse Presentasjon	62225	34	18,5	16	11	19	64,5	-30,5
A6-003	Holde Presentasjon	62236	8	1	1		1	3	5
A3-006	Rapport	62360	38		8,5			8,5	29,5
A4-001	Auto-Test scripting	62240, 62280	50	5,5	3,5			9	41
A1-xxx	Omvisning Antonio hos KM	62360	6					0	6
A5-002	Teste Hendel	62810				6,5	4,5	11	-11
Sum			180	45	45	41,5	40,5		
Total tid								172	

Status:

Vi har produsert en powerpoint presentasjon med et teknisk fokus.

Presentasjonen er gjennomført og vi fikk gode konstruktive tilbakemeldinger på denne. Intern sensor etterlyser mer teknisk dokumentasjon med referanser, og hver av oss må ha et eget teknisk felt å fordype oss i. Eller var sensorene godt fornøyd med hvordan presentasjonen ble gjennomført.

Med tanke på de utfordringene vi har hatt med databasen, så har KM satt en del ressurser på å få ordnet opp i dette den siste tiden. Det har bedret seg såpass at vi regner med at alt skal være klart da vi begynner på sprint 6, men vi vil fortsatt holde ett møte for å sikre oss at dette er i orden til den tid.

På grunn av de utfordringene som har oppstått, så har vi ved å prøve å løse dette fått en bredere oversikt og forståelse av hele systemet enn det vi kanskje ellers ville hatt. Dette ser vi på som veldig positivt og ikke minst lærerikt.

Når vi ser på timer i forhold til avvik, så er det brukt mer tid på forberedelse til presentasjonen enn estimert og mindre tid på rapport. Ved gjennomgang i sprint så kommer det fram at dette er en kombinasjon av feil estimering og at deler av forbedrende arbeid burde vært på rapport.

Planlegging sprint 6 uke 14 og 15

Planlagte aktiviteter:

Planlagte aktiviteter for Sprint 6 (uke 14,15)											
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 07.03.2016 til: 20.03.2016				Sum:	Avvik:	Årsak:	Tiltak:
				PV	TG	KS	NS				
A1-001	Daglig SCRUM møte	62279	12					0	12		
A1-002	Sprint Planlegging og Gjennomgang	62237	4					0	4		
A1-003	Sprint Oppfølgingsdokumentasjon	62238	4					0	4		
A4-001	Auto-Test scripting		90					0	90		
A3-006	Rapport	62360	90					0	90		
A1-xxx	Omvisning Antonio hos KM	62360	16					0	16		
A2-001	Opplæring Software	62220	32					0	32		
A1-004			8					0	8		
								0	0		
								0	0		
Sum			256	0	0	0	0				
Total tid								0			

Status:

Sprint 6 er første sprinten etter eksamen, den består av 8 arbeidsdager til prosjektet.

Som en oppstart av sprint 6 skal vi ha ett avklaringsmøte med KM for å avklare om vi har databasen til hendelen og at systemet fungerer slik det skal for at vi skal kunne jobbe videre med prosjektet etter planen vår.

Det er satt av tid til opplæring i ROS for å få den informasjonen og kunnskapen vi trenger for å kunne koble til, forandre og feilsøke på testanlegget vi har satt sammen.

Det er også estimert en del timer til scripting og rapportskriving.

Scripting:

- Lage Autotest og verifikasjon for de funksjoner vi vet fungerer
- Gå igjennom alle ukjente funksjoner og se at vi kan sende, spørre og få svar

Rapport:

- Gå nærmere inn på den tekniske dokumentasjonen
- Hente informasjon på alle komponenter som brukes og skrive dette inn i vår rapport mal

Gjennomgang sprint 6 uke 14 og 15

Utførte aktiviteter:

Utførte aktiviteter for Sprint 6 (uke 14,15)										
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 07.03.2016 til: 20.03.2016					Avvik:	Årsak:
				PV	TG	KS	NS	Sum:		
A1-001	Daglig SCRUM møte		12	3,5	4	4	4	15,5	-3,5	
A1-002	Sprint Planlegging og Gjennomgang		4	2,5	3,5	3	3	12	-8	
A1-003	Sprint Oppfølgingsdokumentasjon		4					0	4	
A4-001	Auto-Test scripting		90	2	16	11,5	17,5	47	43	
A3-006	Rapport		90	22	24	9	16,5	71,5	18,5	
A1-xxx	Omvisning Antonio hos KM		16	3	3	3	3	12	4	
A2-001	Opplæring Software		32	6,5	6,5	6,5	6,5	26	6	
A1-004	Generelt Møte		8					0	8	
A5-001	Teste Script			2	2	7	2	13	-13	
A3-002	User Story (Krav)					4		4	-4	
A2-002	Opplæring/Rigg Hardware			9,5	2	22	7	40,5	-40,5	
A5-002	Teste Hendel			5	2	2	3,5	12,5	-12,5	
Sum			256	56	63	72	63			
								Total tid	254	

Status:

Uke 12, 13 og de to første dagene i uke 14 ble holdt av til påske og forberedelse til eksamen i faget vi hadde ved siden av bacheloroppgaven. Sprint 6 begynte derfor onsdag uke 14 og varte til og med uke 15. Siden det andre faget nå var avsluttet, skulle det nå arbeides med prosjektet 5 dager i uken.

Det har vært utfordringer med utvikling av databasen og vi har funnet svakheter og mangler som er blitt rapportert inn til KM.

Vi har hatt opplæring i K-chief 600. Alle i gruppen fikk da tilgang til et fullt operativt testanlegg for Wind Innovation som var koblet opp med alle moduler nødvendig for å simulere motorstyring og alarmstyring av hele skipet. Opplæringen gikk ut på manøvrering i OS, og hvordan man kjører en FAT (Factory Acceptance Test) på anlegget.

Vi hadde møte med KM mandag uke 15 for å diskutere videre framdrift i prosjektet, og hvordan vi skulle fordele hvert vårt tekniske bidrag i bacheloroppgaven. Siden det nå så ut til at vi hadde stabil database for hendelen, valgte vi å fortsette planen for utviklingen av AutoTestene. De tekniske bidragene ble fordelt i fire kategorier og bør presenteres i følgende rekkefølge:

1. Testing (verifisering, manuell testing og thruster)
2. K-Chief 600 (Systemet / manøvrering i OS/DPU)
3. Common Lever (401-modul / ny teknologi).
4. Scripting (Auto-test / kodespråk)

Oppgaven er blitt utvidet mye i forhold til utgangspunktet og oppgavebeskrivelsen må revideres.

Det er blitt brukt en god del tid på å sette sammen og koble opp realtime-systemet i et presentabelt kabinett.

Det er også jobbet med å strukturere oppsettet for scripting av AutoTesten.

Vi har hatt omvisning med Antonio hos KM i Horten. For å vise hva vi jobber med og hvordan vi jobber. Det er også med på å skape en felles forståelse av hva det er vi jobber med.

Planlegging sprint 7 uke 16 og 17

Planlagte aktiviteter:

Planlagte aktiviteter for Sprint 7 (uke 16,17)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 18.04.2016 til: 01.05.2016					Avvik:
				PV	TG	KS	NS	Sum:	
A1-001	Daglig SCRUM møte	62279	20					0	20
A1-002	Sprint Planlegging og Gjennomgang	62237	8					0	8
A1-003	Sprint Oppfølgingsdokumentasjon	62238	8					0	8
A3-006	Rapport	62071	98					0	98
A4-001	Auto-Test scripting		106					0	106
A4-002	Kode til verifisering		32					0	32
A5-001	Teste Script		16					0	16
A5-002	Teste Hendel		16					0	16
A5-003	Oppkobling av Utstyr		16					0	16
								0	0
Sum			320	0	0	0	0		
Total tid								0	

Status:

Etter planen er dette den siste sprinten i utviklingsfasen før vi går over i overgangsfasen hvor prosjektet skal avsluttetes og ferdigstilles, slik at KM kan ta prosjektet videre.

Vi i prosjektet har rapportert inn en liste med bugs (feil og mangler) til KM. Flere av feilene eller manglene er kritiske for at vi skal kunne få gjort ferdig flere av Test Casene fra KM.

Dette har vi en god dialog med KM om, og utviklingsavdelingen til KM er på saken.

Det planlegges i denne sprinten å i stor grad jobbe med AutoTest-scripting og med teknisk dokumentasjon til rapporten. Under AutoTest-scripting skal også koden struktureres og bygges opp på en oversiktlig og fornuftig måte. Noen av AutoTestene har også blitt påvirket av ny database til hendelen og det er også planlagt å bruke tid på å få disse til å virke som de skal.

Det er også planlagt å legge inn de manuelle TestCasene fra KM, slik at vi får dokumentert alle manuelle tester vi i prosjektet har gjennomført for KM, som en del av oppgaven.

Gjennomgang sprint 7 uke 16 og 17

Utførte aktiviteter:

Utførte aktiviteter for Sprint 7 (uke 16,17)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 18.04.2016 til: 01.05.2016					Avvik:
				PV	TG	KS	NS	Sum:	
A1-001	Daglig SCRUM møte	62279	20	5	5	5	5	20	0
A1-002	Sprint Planlegging og Gjennomgang	62237	8	1	2	1	1	5	3
A1-003	Sprint Oppfølgingsdokumentasjon	62238	8	4				4	4
A3-006	Rapport	62071	98	32	32,5	32,5	13,5	110,5	-12,5
A4-001	Auto-Test scripting		106	22,5	25	32	50	129,5	-23,5
A4-002	Kode til verifisering		32			9,5		9,5	22,5
A5-001	Teste Script		16	3,5	10			13,5	2,5
A5-002	Teste Hendel		16	6	2		3	11	5
A5-003	Oppkobling av Utstyr		16					0	16
A1-004	Generelt Møte			3	1		3	7	-7
Sum			320	77	77,5	80	75,5		
Total tid								310	

Status:

Sprint 7 var den siste sprinten i utviklingsfasen, og fokuset har vært å ferdigstille så mye som mulig av autotesten og det tekniske i oppgaven. Det har blitt ferdigstilt 11 test caser som kan kjøres sammenhengende i TestComplete. Det sitter to prototype hendler i testkonsollen som nå fungerer bra. Man kan ta kommando på hver av de, og den andre følger etter (slavemodus). Det er gitt tilgang til å lage mimic, det vil si at man kan sette opp et grafisk bilde på skjermen på OS som viser bl.a pådrag og rotasjon. KM har vært meget behjelpelige med å følge opp de bugs som måtte rettes for at vi skulle komme videre med autotesten.

Vi er godt fornøyd med planleggingen og gjennomføringen av sprinten, og føler at vi har klart å produsere et meget bra produkt. Som det går frem av tabellen over, ble det lite avvik, noe som betyr at vi stadig har blitt bedre på å planlegge sprintene, og dette har vært den beste hittil.

Det har blitt produsert en del teknisk dokumentasjon til rapporten, som det må fortsettes på i neste sprint.

Planlegging sprint 8 uke 18 og 19

Planlagte aktiviteter:

Planlagte aktiviteter for Sprint 8 (uke 18,19)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 02.05.2016 til: 15.05.2016					Avvik:
				PV	TG	KS	NS	Sum:	
A1-001	Daglig SCRUM møte		20					0	20
A1-002	Sprint Planlegging og Gjennomgang		12					0	12
A1-003	Sprint Oppfølgingsdokumentasjon		4					0	4
A3-002	User Story (Krav)		16					0	16
A3-003	Testplan		24					0	24
A3-004	Hjemmeside		8					0	8
A4-001	Auto-Test scripting		20					0	20
A4-002	Kode til verifisering		16					0	16
A3-006	Rapport		200					0	200
								0	0
Sum			320	0	0	0	0		
Total tid								0	

Status:

Dette er den første sprinten i overgangsfasen, som er den avsluttende fasen for prosjektet. Overgangsfasen består av to sprinter, hvor den første sprinten består av to uker og den siste er på en uke.

Utviklingen av produktet er nå hovedsakelig avsluttet, men vi setter av noen timer til de siste justeringer. Ellers vil det meste av tiden gå med til å skrive rapport på det vi har oppnådd med prosjektet.

Prosjektet ligger godt an i forhold til de fasene vi har planlagt med fra starten av.

Gjennomgang sprint 8 uke 18 og 19

Utførte aktiviteter:

Utførte aktiviteter for Sprint 8 (uke 18,19)									
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 02.05.2016 til: 15.05.2016					Avvik:
				PV	TG	KS	NS	Sum:	
A1-001	Daglig SCRUM møte		20	5,5	5,5	5,5	5,5	22	-2
A1-002	Sprint Planlegging og Gjennomgang		12	5	4	4	4	17	-5
A1-003	Sprint Oppfølgingsdokumentasjon		4					0	4
A3-002	User Story (Krav)		16				2,5	2,5	13,5
A3-003	Testplan		24		19,5			19,5	4,5
A3-004	Hjemmeside		8				8	8	0
A4-001	Auto-Test scripting		20	16,5		10	5,5	32	-12
A4-002	Kode til verifisering		16			3		3	13
A3-006	Rapport		200	58	56	57,5	52	223,5	-23,5
A3-007	Expo-plakat						9,5	9,5	-9,5
Sum			320	85	85	80	87		
								Total tid	337

Status:

Sprint 8 var starten på overgangsfasen uke 18 og 19. I denne sprinten jobbet vi med å ferdigstille dokumentasjonen. I tillegg til dette, ble det laget en expo plakat, begynt på powerpoint presentasjon og oppdatert hjemmesiden.

Vi hadde problemer med AutoTesten i forhold til hendel bevegelsene. Problemet var at CAN sendte beskjerer syklisk til hendelen, slik at den forstyrret testen vår. Dette ble rettet opp, i tillegg til at det ble gjort endringer i skjermlås funksjonen. Skjermlås funksjonen har tidligere ikke fungert når man har sendt kommandoer via TestComplete. Koden ble derfor endret slik at vi hadde mulighet til å skru av denne funksjonen ved kjøring av AutoTest.

Databasen ble oppdatert samtidig som software på hendelen, og dette resulterte i at mimicen som ble laget til OS ikke fungerte. Her ble det gjort en ny jobb for å få den til å virke, slik at vi har en presentabel visning av feedback til siste presentasjon.

Planlegging sprint 9 uke 20 og 21

Planlagte aktiviteter:

Planlagte aktiviteter for Sprint 9 (uke 20,21)										
Aktivitet Nr:	Aktivitet:	Work Item ID:	Estimert tid:	Dato fra: 16.05.2016 til: 26.05.2016					Avvik:	Årsak:
				PV	TG	KS	NS	Sum:		
A1-001	Daglig SCRUM møte		18					0	18	
A1-002	Sprint Planlegging og Gjennomgang		6					0	6	
A3-006	Rapport		32					0	32	
A6-001	Produsere Powerpoint		80					0	80	
A6-002	Forberedelse Presentasjon		88					0	88	
A6-003	Holde Presentasjon		32					0	32	
Sum			256	0	0	0	0			
Total tid								0		

Status:

Sprint 9 er den avsluttende sprinten i dette prosjektet. Vi skal i denne perioden fokusere på dokumentasjon, ferdigstilling og kvalitetssikring av rapporten. Det skal også brukes tid på produksjon av PowerPoint og forberedelser til framføringen. 23. Mai skal prosjektet leveres til høgskolen, 26 Mai skal vi presentere prosjektet.

Det vil ikke bli gjort en sprintgjennomgang etter denne perioden da prosjektet er ferdig for vår del. KM tar over og implementerer produktet vi har laget, i sitt komplette AutoTest system.

Møtereferat med intern veileder.

18.01.2016

Dato: 18/1-16

Tid: 15.30

Sted: Rom 2256 HSN Kongsberg

Tilstede: Philip Vestlie, Thomas Gjerrud, Nicolai Sætra, Kristoffer Stanger og internveileder Antonio L. L. Ramos.

Fraværende: -

Referent: Philip Vestlie

Godkjenning av innkalling:

Godkjent av alle.

Sak1:

Oppstart av prosjekt

Diskusjon:

- Oppfølging av veileder og rapportering.
- Forventninger til prosjektet og gjennomføring.

Konklusjon/avgjørelse:

- Ukentlig oppfølging med Mandag kl 14.30 som møtedag.
- Gruppen er meget motivert for å gjøre ett solid arbeid for å oppnå best mulig resultat. Gruppen er innstilt på å tilfredsstille alle involvertes krav til prosjektet.

Neste møte:

Dato: Mandag 25. Januar 2016

Sted: HSN Kongsberg, Rom 2256

Møtereferat Møte 2

Uke 4

Dato: 25.01.2016

Tid: 14.30

Sted: Rom 2256 HSN Kongsberg

Tilstede: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie

Fraværende: Antonio L. L. Ramos

Referent: Kristoffer Stanger

Godkjenning av innkalling:

Godkjent av alle.

Sak1:

Fremdrift av prosjektet

Diskusjon:

- Prosjektplan
 - Hva mangler?
- Prosjektmodell
 - Trenger den endringer?
- Kravspesifikasjon
 - Hva skal denne konkret inneholde?
- Testplan
 - Hvordan skal denne utarbeides?

Konklusjon/avgjørelse:

- Dokumentstruktur skal utarbeides og ferdigstilles.
- Prosjektmodellen må vurderes om tilbakemelding fra veileder er ønskelig.
- Det jobbes med utvikling av hendelen fra KM og informasjon mot krav er påbegynt. Kravspesifikasjonen skal begynnes på.
- Det skal skrives generelt om hvordan vi skal teste produktet vårt.
- Hovedfokuset skal ligge mot første presentasjon.

Neste møte:

Dato: 01.02.2016, kl. 15.00, Rom 2256 HSN Kongsberg

Møtereferat Møte 3

Uke 5

Dato: 04.02.2016

Sted: Rom 2256 HSN Kongsberg

Tilstede: Thomas Gjerrud, Kristoffer Stanger, Nicolai Sætra, Philip Vestlie og interveileder Antonio L. L. Ramos

Fraværende: -

Referent: Nicolai Sætra

Godkjenning av innkalling:

Godkjent av alle

Sak1:

Fremdrift av prosjektet

Diskusjon:

- Prosjektplan
 - Hva mangler?
- Prosjektmodell
 - Hvordan er oppsettet?
- Presentasjon1
- Tanken om strukturering og oppsett?

Konklusjon/avgjørelse:

- Prosjektplan ryddes opp i slik at det som er vesentlig å få med til første innlevering er med.
- Prosjektmodellen trenger en liten justering slik at man enkelt kan se at vi kan gå frem og tilbake under iterasjonene.
- Presentasjonen bør bygges opp slik at det er en god flyt og at folk forstår problemstillingen og hvordan vi planlegger å løse den.

Neste møte:

Dato: 08.02.2016, kl. 15.00, Rom 2256 HSN Kongsberg

Møtereferat Møte 4

Uke 6

Dato: 8/2-16

Tid: Kl 14.00

Sted: HSN Kongsberg, Rom 2256

Tilstede: Thomas Gjerrud, Nicolai Sætra, Kristoffer Stanger, Philip Vestlie

Fraværende: Antonio L. L. Ramos

Referent: Thomas Gjerrud

Godkjenning av innkalling:

Godkjent av alle

Sak1:

Presentasjon 1.

Diskusjon:

Innhold i presentasjonen:

- Er presentasjonen klar.
- Dokumentasjon som skal leveres.
- Digital tilgang.

Konklusjon/avgjørelse:

Presentasjonen trenger kun litt finpuss og gjennomgang av kvalitet.

Dokumentasjonen er lagret som versjon 1.0 som er den første som blir gitt ut.

Mappe med digital tilgang til prosjektet blir opprettet og sendt til sensorer og veiledere på mail.

Annet:

-Vi har fått tilgang til domene for opprettelse av hjemmeside. Dette har vært en svært tidkrevende oppgave å sette seg inn i og opprette. Mye prosjekttid har altså gått med til dette.

-Vi ser nå som prosjektet er i gang at det vil være mer tidseffektivt å kjøre ett veiledermøte per iterasjon, siden hver iterasjon består av to uker av tre dager.

Neste møte:

Dato: Uke 8

Møtereferat sprint 4**Uke 8 og 9****Dato:** 29/2-16**Tid:** 14.00**Sted:** HSN Kongsberg, rom 2256**Tilstede:** Thomas Gjerrud, Nicolai Sætra, Kristoffer Stanger, Philip Vestlie, Antonio L. L. Ramos**Fraværende:** -**Referent:** Philip Vestlie**Godkjenning av innkalling:**

Godkjent av alle

Sak1:

Fremdrift av prosjektet

Diskusjon:

Fremdriften i prosjektet:

- Etter gode tilbakemeldinger og god veiledning, ser vi i bachelorgruppen en helt klar fordel av å kjøre prosjektet etter en scrum modell. Mye av det vi får fra KM og måten de jobber på er etter Scrum.
- Denne fasen består av en presentasjon. Som bachelorgruppen har planlagt å ha 17/3-16
- Vi har også fått mulighet til å invitere intern veileder til KM for å se hvordan vi jobber der.

Konklusjon/avgjørelse:

- Scrum som arbeidsform ser ut til å være fornuftig for prosjektet. Og alle har satt seg godt inn i hvordan scrum fungerer.
- Presentasjon 2 blir 17/3-16 fra kl 13.00 med formøte, 13.30 presentasjon og ettermøte kl 14.00, invitasjon kommer.
- Antonio vil veldig gjerne se hvordan vi jobber og har satt av 10/3-16 uke 10, til å ta seg en tur til Horten.

Neste møte:**Dato:** Uke 10.

Møtereferat Teknisk med oppdragsgiver

Uke 15

Dato: 11.4.2016

Tid: 10.00 – 11.00

Sted: KM Horten

Tilstede: Magne Røed, Håkon Gjone, Thomas Gjerrud, Philip Vestlie, Kristoffer Stanger, Nicolai Sætra

Fraværende:

Referent: Nicolai Sætra

Godkjenning av innkalling:

Godkjent av alle

Sak1:

Teknisk bidrag

Diskusjon:

Som et krav fra HSN så må hver enkelt student ha en teknisk fordypning i oppgaven.

Konklusjon/avgjørelse:

Vi har kommet frem til at vi da deler opp i fire forskjellige tekniske kategorier, disse er som følger: 1. Test (verifisering, manuell testing og thruster)

2. K-Chief 600 (System/manøvrering i ROS/DPU'er)
3. Hendelen (401-modul/ny teknologi). Kjell og Pål sitter på info om dette.
4. Scripting (Auto-test/kodespråk)

Denne rekkefølgen bør benyttes ved siste presentasjon.

Sak2:

Oppdatering av oppgavebeskrivelse

Diskusjon:

Hvordan synliggjøre endringer i prosjektet

Konklusjon/avgjørelse: Dette er ett utviklingsprosjekt som hele tiden er under endring og derfor har det også blitt mer manuelltesting i tillegg til auto-testingen. Det har nå blitt kjørt manuelltesting og auto-testing på samme tid og dette har ikke KM gjort før. Vanligvis så er det manuelltesting som blir gjort først, for så å utvikle auto-testen. Ved å gjøre det på denne måten, så har det blitt avdekket en del feil med systemet som har måttet bli rettet opp i.

Møtereferat med oppdragsgiver.

Uke 16

Dato: 19.04.2016

Tid: 09.30-10.30

Sted: KM Horten

Tilstede: Magne Røed, Thomas Gjerrud, Nicolai Sætra, Kristoffer Stanger, Philip Vestlie

Fraværende: -

Referent: Philip Vestlie

Godkjenning av innkalling:

Godkjent av alle

Sak1:

Oppdatering av oppgavebeskrivelsen.

Diskusjon:

Common Lever prosjektet er et utviklingsprosjekt og hendelen er en prototype. På grunn av utfordringer hos KM i prosjektet, har bachelorgruppen kommet inn i prosjektet i en tidligere fase i forhold til hva som var tenkt. Dette har ført til at det er flere oppgaver som må utføres før arbeidet med å lage autotesten kan gjennomføres.

Konklusjon/avgjørelse:

Disse utvidelsene skrives inn som en del av oppgaven, og blir derfor arbeidsoppgaver som bachelorguppen må gjennomføre. Dette fører til at mulig utvidelse av oppgaven utgår, det samme gjør også bonusoppgaven, og at det blir lagt til nødvendig utvidelse av oppgaven. Dette blir nærmere beskrevet i prosjektplanen kapittel 5.

Samarbeidskontrakt

Kontraktens gyldighet

Fra 6.1.2016 til avslutning av prosjektet

Kontrakten omfatter følgende medlemmer

- Thomas Gjerrud
- Kristoffer Stanger
- Nicolai Sætra
- Philip Vestlie

Kommunikasjon

- Det skal holdes ukentlige møter med intern veileder
- Det skal holdes daglige SCRUM møter med ekstern veileder
- Kommunikasjon via lukket Facebookgruppe
- Dokumentlagring i Dropbox
- Backup av dokumenter utføres daglig på harddisk i tillegg til Dropbox

Beslutning

- Viktige avgjørelser skal avgjøres i felleskap
- Ved uenighet skal det holdes avstemming
- Ved vanskelige avgjørelser bør gruppen rådføre seg med veileder
- Ved uenighet har gruppeleder dobbel stemme

Deltagelse

- Arbeidsmengde må fordeles likt mellom gruppedeltakerne
- Alle skal delta på forelesninger i forbindelse med bacheloroppgaven
- Planlagt fravær skal jobbes inn
- Akutt sykdom meldes om så tidlig som mulig
- Oppmøte på presentasjoner er obligatorisk

Gruppens felles ansvar

- Alle i gruppen skal inkluderes og behandles med gjensidig respekt
- Alle er ansvarlig for å overholde tidsfrister og delta aktivt i arbeidet
- Ved gjentagende mangel på deltagelse i gruppearbeidet skal vedkommende og skolen varsles, og i verste fall kan vedkommende bli ekskludert fra gruppen

Leders ansvar

- Delegere oppgaver ved behov
- Innkalle til møter
- Utpeke møterreferent og møteleder
- Være bindeledd mellom bachelorgruppen, HSN og KM



Thomas Gjerrud



Kristoffer Stanger



Nicolai Sætra



Philip Vestlie



**STANDARDAVTALE FOR STUDENTENES ARBEID MED
BACHELOROPPGAVEN MED EKSTERNE OPPDRAGSGIVERE VED
HØGSKOLEN I SØRØST NORGE
FAKULTET FOR TEKNOLOGI OG MARITIME FAG
KONGSBERG INSTITUTE FOR INGENIØRFAG.**

**Høgskolen i Sørøst Norge
Fakultet for Teknologi og Maritime fag
Kongsberg Institutt for Ingeniørfag**

1. BAKGRUNN

Formålet med denne kontrakten er å formalisere forholdet mellom ekstern oppdragsgiver, Høgskolen i Sørøst Norge og studentene i arbeidet med hovedprosjektet for avgangstudentene ved Avdeling for Teknologi og Maritime fag – Kongsberg Institutt for Ingeniørfag ved Høgskolen i Sørøst Norge.

Denne avtale definerer plikter og rettigheter for partene i forbindelse med dette arbeidet.

Oppdragsgiver er kjent med og aksepterer fullt ut at dette arbeidet er utført som et ledd i en høgskoleingeniørutdanning og at Høgskolen i Sørøst Norge ikke kan garantere for arbeidets kvalitet eller påta seg noe økonomisk eller juridisk ansvar for det produktet som arbeidet fører til, eller bruken av produktet i noen sammenheng. Høgskolen kan heller ikke påta seg vedlikeholdsansvar for det produktet som utvikles i forbindelse med hovedprosjektet.

Studentene har ikke anledning til å motta lønn for arbeidet som utføres.

2. PARTER

Avtalen har tre parter:

- Avgangsstudentene i prosjektgruppen, nedenfor kalt Studentene.
 - Thomas Gjerrud
 - Kristoffer Stanger
 - Nicolai Sætra
 - Philip Vestlie
- Ekstern oppdragsgiver, nedenfor kalt oppdragsgiver.
 - Kongsberg Maritime Merchant Marine
- Høgskolen i Sørøst Norge, nedenfor kalt Høgskolen.

3. OPPDRAGSGIVERS PLIKTER

Oppdragsgiver skal oppnevne en av sine ansatte som ekstern veileder for studentene. Det forutsettes at veilederen gir studentene den nødvendige hjelp med å definere oppgaven samt skaffe nødvendig underlag for arbeidet og at veilederen har myndighet til å opptre som ansvarlig representant for oppdragsgiver i spørsmål som omfattes av denne kontrakten.

Under arbeidet er det viktig at studentene får god anledning til regelmessige samtaler med veilederen i prosjektperioden.

Videre kreves det at oppdragsgiver stiller kvalifisert ekstern sensor (mastergrad / siv.ing. eller tilsvarende innen fagområdet) til rådighet for prosjektet. Ekstern sensor skal medvirke ved evaluering av prosjektarbeidet. Ekstern sensor kan være den samme som veilederen.

Den eksterne veilederen skal godkjenne alle studentenes rapporter.

4. OPPDRAGSGIVERS RETTIGHETER

Oppdragsgiver har fulle rettigheter til å benytte resultatet av hovedprosjektet med mindre noe annet er angitt i vedlegget til denne kontrakten.

Ved bruk og eventuell publisering av resultatene av oppgaven skal det henvises til at arbeidet er et studentarbeid ved Høgskolen i Sørøst Norge, Fakultet for Teknologi og Maritimefag samt studentenes og veiledernes navn.

5. HØGSKOLENS PLIKTER

Ved starten av avgangsåret skal studentene deles opp i prosjektgrupper på 4-6 personer som sammen skal gjennomføre en større oppgave, et hovedprosjekt. Omfanget er 20 studiepoeng. Dette tilsvarer ca. 600 timers arbeid pr. student.

Det er ønskelig at studentene innhentet en egnet prosjektoppgave fra oppdragsgiver og står for hoveddelen av kommunikasjonen med oppdragsgiver. Studentene skal i starten på prosjektet produsere en kravspesifikasjon og en prosjektplan i forbindelse med den foreslåtte prosjektoppgaven og rapportere disse skriftlig.

Hovedprosjektet skal evalueres med vanlige karakterer. Evalueringen vil bli gjort på grunnlag av det utførte arbeidet og av en rekke dokumenter og muntlige fremføringer/utspøringer samt regelmessige møter. Karakterene settes på individuelt grunnlag av intern veileder, intern sensor og ekstern sensor.

Høgskolen stiller intern veileder og intern sensor til rådighet i forbindelse med dette arbeidet.

6. UTGIFTER

Det vil påløpe en del utgifter i forbindelse med arbeidet. Disse dekkes på følgende måte:

Lønn	Studentene skal ikke motta lønn for arbeidet med hovedoppgaven. Den eksterne veilederens og sensors lønn og utgifter dekkes fullt ut av oppdragsgiver. Den interne veilederens og sensors lønn og utgifter dekkes fullt ut av høgsolen.
Innkjøp av utstyr	Innkjøp av utstyr og bøker skal ordnes via oppdragsgiver. Innkjøp utstyr og bøker er oppdragsgiver sin eiendom etter prosjektperioden.
Andre utgifter inkludert reiser	Oppdragsgiver dekker studentenes direkte utgifter. Alle utgifter skal avtales og godkjennes på forhånd av oppdragsgiver.

7. UTSTYR OG KOMPONENTER

Høgsolen stiller sitt utstyr vederlagsfritt til disposisjon for studentene i den grad utstyret er egnet og tilgjengelig.

Utstyr som lånes ut fra oppdragsgiver skal fortrinnsvis monteres i høgsolens lokaler. Utstyret skal holdes forsikret av oppdragsgiver. Utgifter til vedlikehold av utstyret dekkes av oppdragsgiver.

Komponenter til utstyr som konstrueres og bygges under arbeidet med hovedprosjektet betales av oppdragsgiver.

Utlånt utstyr og innkjøpte komponenter skal dokumenteres i eget vedlegg.

8. OPPHAVSRETT

Studenten forbeholdes rettigheter etter Lov om arbeidstakeroppfinnelser, Lov om åndsverk og annen lovgivning der hvor dette er aktuelt. Høgsolen i Sørøst Norge beholder imidlertid alle rettigheter til bruk av eventuelle resultater, rådata og lignende til forskning, undervisning og annet som faller inn under Høgsolen sin primæroppgave, så fremt dette ikke åpenbart ville stride mot studentens rettigheter og interesser.

9. \ ENDRINGER

Alle endringer på denne standardavtalen skal dokumenteres i eget vedlegg.

10. VEDLEGGSLISTE

Dato:

1/2 - 2016

For Oppdragsgiver

Magne Ræed

For Høgskolen

Olav A. Grev

Studentene:

Thomas Gjerd, Nicolai Sætra, Kristoffer Stangor, Philip Veklie

Tidsplan

Fase:		Ukenr.:	Disp. dager:	Merknader:
Oppstartsfase	Iterasjon 1	2	3	
		3	3	
	Iterasjon 2	4	3	
		5	3	Leverer dokumentasjon til pres. 1
Utviklingsfase	Iterasjon 3	6	3	Presentasjon 1, 11/2-16
		7	3	
	Sprint 4	8	3	
		9	3	
	Sprint 5	10	3	Leverer dokumentasjon til pres. 2
		11	3	Presentasjon 2, 17/3-16
	Sprint 6	14	3	Statusmøte 6/4 på fremdrift med KM
		15	5	
	Sprint 7	16	5	
		17	5	
Overgangsfase	Sprint 8	18	5	
		19	5	
	Sprint 9	20	5	Hovedinnlevering
		21	2	Presentasjon 3, 26/5

[illegible]