

INGENIØRHØJSKOLEN ÅRHUS

ELEKTRO-INGENIØR LINIEN

SEMESTERPROJEKT E4PRJ4

Bias Reducing Operating System

Skrevet af:

Kategorina GLUD

Studienummer: 11102

Johnny KRISTENSEN

Studienummer: 10734

Rasmus LUND-JENSEN

Studienummer: 11111

Mick HOLMARK

Studienummer: 11065

Jacob ROESEN

Studienummer: 10095

Vejleder:

Carl Jakobsen



10. december 2012

Indholdsfortegnelse

Kapitel 1	Abstract	3
Kapitel 2	Indledning	5
Kapitel 3	Opgaveformulering	7
Kapitel 4	Systembeskrivelse	9
Kapitel 5	Kravspecifikation	11
Kapitel 6	Afgrænsning	13
Kapitel 7	Projektbeskrivelse	15
7.1	Projektgennemførelse	15
7.1.1	Rollefordelinger	15
7.2	Metoder	15
7.2.1	SCRUM	16
7.2.2	V-model	17
7.3	Analyse	17
Kapitel 8	Systemarkitektur	19
8.1	Systemarkitektur	19
8.1.1	Formål	19
8.2	Design og Implementering	19
8.3	Resultater	19
8.4	Opnåede erfaringer	19
Kapitel 9	Konklusion	21
Kapitel 10	Referencer	23
10.1	Artefakter	23
10.1.1	Kravspecifikation	23
10.1.2	Accepttestspecifikation	23
10.1.3	Systemarkitektur	23

10.1.4	Integrationstestspecifikation	23
10.1.5	Detaljeret design	23
10.1.6	Enhedstestspecifikation	23
10.2	Hjemmesider	24
10.3	Liste over bilag på CD	24
10.3.1	Kode	24
10.3.2	Dokumentation	24
10.3.3	Datablade	24
10.3.4	Billeder	24

Indledning 2

Opgaveformulering 3

Når man laster eller lossrer et containerskib bruges der ofte mange ressourcer på at kontrollere at skibet ikke får slagside.¹ Skibets ansvarshavende officer står på broen under hele lastningen/losningen og holder øje med at det bliver gjort ordentligt. Denne opgave vil vi gerne gøre nemmere. Med et system der automatisk sørger for at skibet altid er i vatter skal kaptajnen kunne interagere med systemet hvis systemet kommer med en alarm. Kaptajnen kan manuelt vælge at flytte ballast til den ene side, hvis han ved at der kommer en række tunge containere, der skal stå i modsatte side. For at sikre at alle skibe i havnen er i vatter, sender systemet statusbeskeder til en database på havnekontoret. Kontoret kan derfor sende bemanning eller tage kontakt til skibet, der har en alarm.

¹FiXme Note: Vi skal enten have en kilde på det her eller også skal det fjernes!

Systembeskrivelse 4

BROS er et sikkerhedssystem til skibe. Systemet tages i brug ved lastning eller losning. Her er det systemets opgave at sørge for at skibet ikke får slagsside - heraf navnet: Bias Reducing Operating System (Slagsidereducerende Operativt System). I systemet er der indbygget en hældningssensor og to vandballasttanke - en i hver side af skibet. På baggrund af målinger fra hældningssensoren vil indholdet af tankene blive justeret således at der korrigeres for en slagsside af skibet.

Hele systemet styres fra Skibsførens kontor hvor et grafisk brugerinterface er installeret. Her kan der aflæses skibets hældning af skibet, vandindholdet af tankene og statusmeldinger for systemet. Som udgangspunkt vil systemet automatisk opretholde en hældning på nul grader, men hvis man ønsker det kan man her manuelt give skibet en mindre slagsside. Dette kan gøres for at imødekomme en større slagsside til modsatte side påført af en forestående ændring i skibets last.

For at indsætte et ekstra sikkerhedselement vil systemet under hele processen løbende sende værdier for systemet til en ekstern database. Dermed kan en repræsentant fra terminalen følge skibets status.

Kravspecifikation 5

Afgrænsning 6

Projektbeskrivelse 7

7.1 Projektgennemførelse

Projektet er udført af en gruppe på fem personer. Gruppen er en fortsættelse fra et tidligere projektforsøg. Dette har givet en fordel i kommunikation og samarbejde. Gruppen valgte fra start at dele ansvar ud til personer med interesse for ansvarsområdet. Gruppen har dog stadigvæk, som følge af det velfungerende samarbejde, i store træk været fælles om opgaverne i projektet. Tidsplanen for projektet blev udarbejdet i den første fase og er blevet overvåget siden. Gruppen var opmærksom på at være realistisk frem for optimistisk. Det har derfor været nødvendigt senere hen at revurdere tidsplanen i forhold til længden af faserne. Projektets overordnede tidsplan for faserne og de eksterne milestones ligger som bilag.

7.1.1 Rollefordelinger

Projektleder:	Jacob Roesen
Projektkoordinator:	Nicolai Glud Jacob Roesen
Scrummaster:	Johnny Kristensen

Tabel 7.1. Tabel over rollefordelinger

Vi har valgt at lave roller ud fra vores udviklingsmetode, SCRUM, der er beskrevet senere. Projektlederen har haft som ansvar at strukturere arbejds og scrummøder. Projektkoordinatorens ansvar har ligget i at planlægge møder og bestille lokaler. Scrummasteren er ansvarlig for udviklingsplatformen, SCRUM, og sørge for at metoden anvendes mest optimalt.

7.2 Metoder

En kort præsentation af de to mest dominerende arbejdsmetoder der er anvendt.

I dette projekt er der anvendt metoder indlært gennem et tidligere projekt. Værktøjerne er de værktøjer som gruppen føler sig trygge ved og som gruppen føler bidrager mest til processen.

7.2.1 SCRUM

I gruppens implementering af SCRUM startes der med at lave en produktbacklog, som er den kunden ser. Derefter planlægges det første sprint. Et sprint spænder over 2 uger. Når et sprint starter bliver opgaver overført fra backloggen til sprintet. Når nye opgaver bliver sat på sprintet bliver opgaven vurderet for hvor stort et omfang den har. Derefter diskuteres der hvilke opgaver de forskellige dele af gruppen skal lave. En gang om ugen laver man et SCRUM-Meeting. Her bliver fulgt op på opgaver lavet i løbet af ugen samt tilføjelse af nye opgaver. Alle de opgaver der ikke er færdige, når sprintet er slut, overføres til næste sprint.

I projektet er der i alt 7 sprint. På *Figur 7.1* vises det 2. sprint.

Sprint nr:	Scrummaster:	JK	Koordinator:	NG		
2	Projektleder:	JR				
Overordnet opgave:	Opgaver:	Arbejdsvægtning:	Resource #1	Resource #2	Process (% done):	Kommentar:
	Videreført: Jura	3	MH	RLJ	100%	
	Videreført: Snak m. Arne vedr. sensor	1	NG	JK	100%	NG, JK
Videreført Accepttest		2	RLJ	NG	100%	NG, JR, JK, MH
Systemarkitektur						
	Systemkomponenter - Enheder	2	JR	RLJ	80%	JR, JK
	Komponentvalg	3	ALLE		100%	RLJ: er afsnittet skrevet og læst af alle?
	Strukturel systemarkitektur	4	MH	JK	60%	MH: Tilrettelse af tegning, tror vi skal sætte os ned os diskutere, flere ting
	Behavior systemarkitektur	5	JK	MH	60%	JK, NG
	HW Systemarkitektur	0	JR	JK	20%	
	Grænseflader / interface	3	NG	JR	80%	NG, JK
	Forside	1	RLJ		25%	
Teknologiundersøgelse		5	ALLE		100%	
Integrationstest		3	NG		80%	NG, JK
Total:		32				
				Procentvis færdig:	81.25 %	

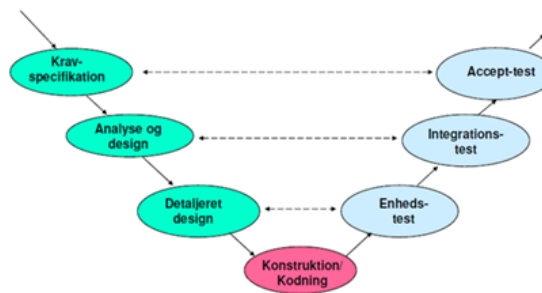
Figur 7.1. Sprint 2

Sprintet er afsluttet og man kan se hvordan nogle opgaver er færdige og hvordan resten skal overføres til næste sprint. Billedet illustrerer også hvordan planlægningen er opbygget. Forklaring af statusfarver er vist på *Figur 7.2*

0%	Ikke startet
20%	Påbegyndt
40%	Godt i gang
60%	Umiddelbart færdigt. Mangler 2. opinion
80%	Mangler gennemlæsning af alle
100%	Gennemlæst af alle

Figur 7.2. Forklaring af sprint points

7.2.2 V-model



Figur 7.3. V modellen

Vi har valgt at anvende V modellen som udviklingsmodel. Dette muliggør iterative processer hvilket er optimalt for vores udviklingstil.

7.3 Analyse

Systemarkitektur 8

8.1 Systemarkitektur

Dette afsnit beskriver systemarkitekturen for projektet "BROS" som formuleret i projektbeskrivelsen og specificeret i kravspecifikationen. Afsnittet indeholder beskrivelse af systemkomponenter, systemarkitektur, SW-komponenter, HW-komponenter og interfaces, i den givende rækkefølge.

8.1.1 Formål

Formålet med dokumentet er:

- At nedbryde systemet i overordnede HW- og SW-komponenter, baseret på kravene specificeret i kravspecifikationen.
- At fastlægge grænsefladen mellem systemets overordnede komponenter.
- At identificere arbejdsopgaver for projektets design- og implementeringsfase

8.2 Design og Implementering

8.3 Resultater

8.4 Opnåede erfaringer

Konklusion 9

Referencer 10

10.1 Artefakter

10.1.1 Kravspecifikation

Kravspecifikationsdokumentet er udarbejdet i begyndelsen af projektet og omfatter beskrivelse af Use Cases, ikke funktionelle krav samt kvalitetsfaktorer. Den fuldstændige kravspecifikation kan ses i bilag (Kravspecifikation.pdf)

10.1.2 Accepttestspecifikation

Accepttestspecifikationsdokumentet beskriver de tests der skal laves for at undersøge om de ønskede krav er opfyldt. Den fuldstændige accepttestspecifikation kan ses i bilag (Accepttest.pdf).

10.1.3 Systemarkitektur

Systemarkitektur dokumentet beskriver systemets HW/SW opbygning og grænseflader. Den fuldstændige systemarkitektur kan ses i bilag (Systemarkitektur.pdf).

10.1.4 Integrationstestspecifikation

Integrationstestspecifikation beskriver de test der skal laves for at undersøge hvorledes de forskellige komponenter kan kommunikere. Den fuldstændige Integrationstest kan ses i bilag (Integrationstest.pdf).

10.1.5 Detaljeret design

Det detaljerede design dokument beskriver hvordan HW/SW er designet og hvordan systemets komponenter fungerer. Det fuldstændige Detaljeret design dokument kan ses i bilag (Detaljeret Hardware design.pdf og Detaljeret Software design.pdf).

10.1.6 Enhedstestspecifikation

Enhedstestspecifikation beskriver de tests der skal laves for at undersøge om de forskellige stubbe af systemet fungerer hensigtsmæssigt. Den fuldstændige enhedstestspecifikation kan ses i bilag (Enhedstest.pdf).

10.2 Hjemmesider

<http://www.docs.google.com> <http://office.microsoft.com/en-us/visio/> <http://www.maplesoft.com/>
<http://www.ni.com/multisim/>

10.3 Liste over bilag på CD

Komponentliste.pdf SCRUM.xls Logbog.pdf

10.3.1 Kode

KI

hest

SM

hestning

VBTE

honning

Server

10.3.2 Dokumentation

Accepttest.pdf Arkitektur.pdf Detaljeret _hardware _design.pdf Detaljeret _software _design.pdf
Enhedstest.pdf Integrationstest.pdf Kravspecifikation.pdf

10.3.3 Datablade

PSoC Kionix KXSC7 datasheet (Accelerometer) ST3232 OSV! HESTE

10.3.4 Billeder

hvis vi har billeder af vores produkt!