

SWT - Mandatory Assignment 3 : Integration Test

Gruppe 16

November 29, 2019

Navn	Studieretning	Student Number
Sivert Sømmer Sagmo	IKT	201608544
Glenn Laursen	IKT	201703930
Saeed Soltani	IKT	201710716

Github Repository : https://github.com/glennlaursen/SWT_Assignment3_GR16/

Jenkins Unit Test : http://ci3.ase.au.dk:8080/job/SWT_E2019_16_MicroWave/

Jenkins Integration Test : http://ci3.ase.au.dk:8080/job/SWT_E2019_16_MicroWave_Integration/

Contents

1	Indledning	2
2	Dependency Tree	3
3	Dependency Plan	4
3.1	Valg for Integrations plan	4
4	Fejl rettelser	5
4.1	Powertube	5
4.2	Timer	5
4.3	Unit Test PowerTube og Timer	5
4.4	Diagrammer	5
5	Konklusion	6

1 Indledning

I denne journal har vi fået udleveret et Mikrobølgeovn system. Journalen omhandler den tredje obligatoriske aflevering, hvor der er fokus på at integrere Mikrobølgeovn. Dvs vi skal lave integrationstest for det udlevert kode. I koden var der også lavet fejl bevidst som vi skulle finde og rette. Disse test er også blevet Integrationstestet gennem jenkins.

2 Dependency Tree

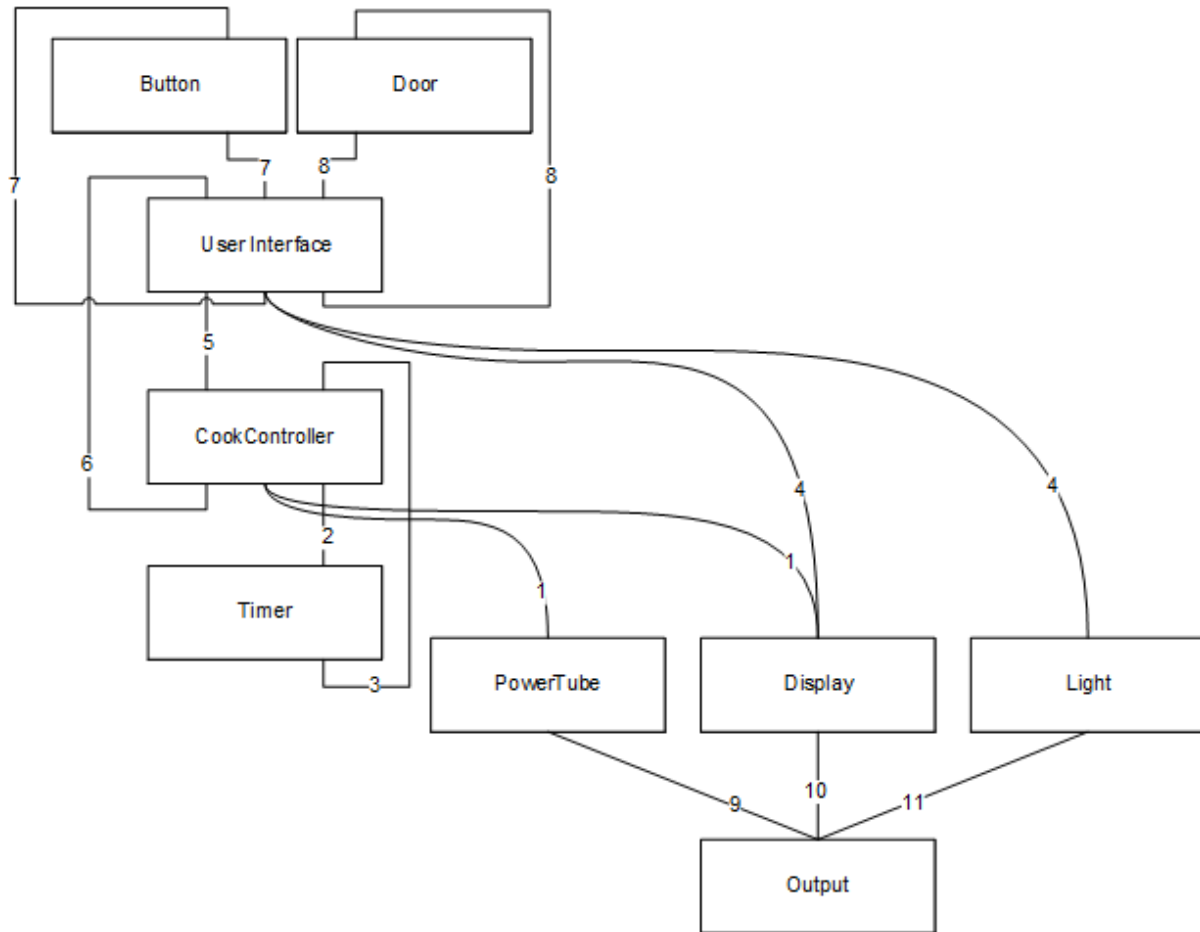


Figure 1: Dependency Tree for Microwave program, with step numbers

3 Dependency Plan

S T E P S	#	Output	Light	Display	PowerTube	Timer	CookController	User Interface	Button	Door
	1	S		X	X	S	T			
	2	S		X	X	X	T			
	3	S		X	S	T	X			
	4	S	X	X				T		
	5				S		X	T	S	
	6	S	S		X		T	X	X	
	7	S	X	X	X	X	X	X	T	X
	8	S	S	S			S	X	S	T
	9	X			T					
	10	X		T						
	11	X	T							

Figure 2: Integrationsplan. Nummerne tilsvare de i figure 1. T er unit under test, s er moduler der er fakes og x er moduler der er inkluderet i testen.

3.1 Valg for Integrations plan

hovedpointen i bottom up metoden er at man kan integrations teste systemet ved start af de komponenter med færrest afhængigheder. dvs komponenter med færrest afhængigheder testes først. Når disse komponenter er blevet testet og godkendt så kan man rykker videre til de næste komponenter indtil hele systemet er blevet testet. Fordelen ved bottom op er at vi kan hurtigt går i gang med at integerer en smule af systemet. der kan integeres parallelt hvis projektet er stort. I andre ord, så kan der være flere udvikler der arbejder samtidigt på systemets test. Da de ikke er afhængig af de andre udviklers integrationstest. Set ud fra vores afhængighedstræ er der en del komponenter der afhænger af andre komponenter i afhængigheds træet. derfor tænker vi at det vil være oplagt at anvende bottom op metoden.

4 Fejl rettelser

4.1 Powertube

i funktionen TurnOn i klassen PowerTube var fejlen at i stedet for enheden var i Watt så var enheden i procent. Valideringen i if sætningen blev ændret fra 50 til 700. hvor der originalt stod 1 til 100. OutputLine der skal procent ændres til Watt

4.2 Timer

i klassen Timer og UserInterface der er enhederne ikke konsekvent. hele tiden så vi har ændret i Timer klassen i funktionen OnTimerEvent hvor TimeRemaining er ændret til 1 hvor den var 1000 før

4.3 Unit Test PowerTube og Timer

vi har rettet de fejl der hører til under de fejl vi fandt i de tilhørende boundary klasser. f.eks. i Unit testen TurnOn_HighPower_ThrowsException ændrede vi power fra 101 til 701.

4.4 Diagrammer

fra Set time til Cooking i state diagrammet så er der ikke staten Turn On light når man cooker. Men det er der i sekvens diagrammet.

5 Konklusion

Først fik vi rettet de bevidste fejl i det udleveret kode. Vi fik lavet Integrationstest på de forskellige moduler. Først skulle vi sikre os på hvordan vi skulle integrere Mikrobølgeovn da der var mange forskellige integrations pattern. Vi endte med Bottom-up da vi synes det passede os bedst. Derudover har vi også kørt integrations testne på Jenkins.