

Projektmål

Erik Österberg, Lucas Arnström, Marcus Fröling,
Mikaela Eriksson, Tobias Andersson, Tony Wang

January 13, 2015

Contents

| | | |
|----------|--|----------|
| 1 | Y63 - Testdriven utveckling | 1 |
| 2 | Y64 - Tillämpa Scrum eller Kanban | 1 |
| 3 | Y65 - Kodstandard | 1 |
| 4 | Y66 - kodgranskning | 2 |

1 Y63 - Testdriven utveckling

Vi har inte tillämpat ren testdriven utveckling då projektuppgiften varit svår att förstå till en början. När vi väl kom igång och fick fram lite kod gick det lättare att tillämpa testdriven utveckling på de kommande delarna av projektet. Det som fungerat bra med testdriven utveckling är att man lättare hittar buggar i koden eftersom man använder sig av offensiv programmering. Detta har bidragit till att kodningen flytit på bättre och gjort att vi fått fram en säkrare kod.

Dåliga aspekter med testdriven utveckling anser vi är att i början är det svårt att veta exakt vad man skall testa, det framgår bättre efter man satt sig in i koden. Med testdriven utveckling blir koden lätt rörig då man ofta gör småändringar i koden för att det bättre skall passa de tester man skrivit.

2 Y64 - Tillämpa Scrum eller Kanban

Vi har valt att använda oss av Scrum till Trello där vi har uppskattat hur lång tid de olika delarna av projektet samt i efterhand lagt till hur lång tid det egentligen tog. Vi anser att Scrum och Trello var rörigt och onödigt då vi ständigt satt i helgrupp och kunde diskutera med varandra direkt. Dock tror vi att Trello och Scrum kan fungera som ett bra medel vid större projekt i större grupper samt vid bättre grundkoll av verktygen. Vänligen se delen om gruppreflekktion där vi diskuterar dett yttligare.

3 Y65 - Kodstandard

Kodstandarden för oss har varit ganska självklar då vi alla gått samma väg och lärt oss på samma sätt från början. Den kodstandard vi valde påverkade oss inte under själva utvecklingen av koden då vi alla hade samma kodstandard i ryggmärgen och det flöt på utan problem. På grund av detta blev läsbarheten tydlig. När det uppstod saknader i kodstandard så anpassade vi oss till varandra för att göra koden konsekvent. Vi valde dock att göra en textfil med kodstandarderna i fall en osäkerhet skulle uppstå. Se bifog fil.

4 Y66 - kodgranskning

Alla i gruppen var med och deltog när vi valde att analysera modulerna heap.c och stringutil.c. Vi valde dessa eftersom heap.c är en viktig modul för att gcen ska fungera i helhet och för att stringutil.c är den modul vi haft mest problem med. Vi anser att vi valde rätt koder att analysera då de innehöll mest olikheter och detta tog oss ca en timme totalt. Något vi upptäckte som är värt att nämna är att vid granskning av stringutil upptäckte vi att en del av koden ej blivit inkluderat från tidigare skriven kod och detta medförde en diskussion om varför detta hade blivit gjort.