



UPPSALA UNIVERSITET

Grupprefleksion

Projekt 2

IMPERATIV OCH OBJEKTORIENTERAD
PROGRAMMERINGS-METODIK

Elwira Johansson, KandDV
Love Nordling, KandDV
Manuel Alexan, KandDV
Pontus Eriksson, IT
Ricardo Danza, KandDV

January 11, 2018

Contents

1	Sidensvansarna	2
1.1	Group members	2
2	Kvantifiering	2
3	Process	3
3.1	Påbörjande	3
3.2	Implementation	3
4	Användning av verktyg	3
4.1	Discord	3
4.2	Github	4
4.3	Trello	4
5	Kommunikation, samarbete och koordination	4
6	Arbetsuppdelningsstruktur	4
6.1	Kodpar	4
6.2	Vad kunde varit bättre	5
7	Kvalitetssäkring	6
7.1	Kodstandard	6
7.2	Pull requests	6
7.3	Testning	6
7.4	Kvalitén på arbetet	7
8	Reflektion	7

1 Sidensvansarna

Gruppen Sidensvansarnas projektrapport för kursen IOOPM vid Uppsala Universitet hösten 2017.

1.1 Group members

Group members		
Name	Email	Active dates
Elwira Johansson	elwira.johansson.4420@student.uu.se	11/12/17-12/1/18
Love Nordling	förnamn.efternamn.1234@student.uu.se	11/12/17-12/1/18
Manuel Alexan	manuel.alexan.1927@student.uu.se	11/12/17-12/1/18
Pontus Eriksson	pontus.eriksson.4631@student.uu.se	11/12/17-12/1/18
Ricardo Danza	ricardo.danzamadera.8564@student.uu.se	11/12/17-12/1/18

2 Kvantifiering

Projektet startade den elfte december och på bara ca 1 månad till den tolfte januari så har vi producerat 1725 rader projektkod, ej inkluderat tester som blivit kvar. Vi har skrivit 561 rader testkod och 159 rader scriptkod. För att komma till den här punkten så har det gått åt 238 h. Vi har haft 150 commits och 28 pull requests dock bara 4 github issues.

3 Process

3.1 Påbörjande

Vi diskuterade en del i början av projektet om vilken process vi skulle använda, de flesta tyckte scrum var intressant men ingen hade riktigt koll på hur den fungerade. Så en i gruppen läste på om scrum och lärde ut den till gruppen.

Efter det så bestämde gruppen hur de ville implementera processen, i vårt fall så hade vi inte en produktägare, men vi hade en scrum-master. Sprintar sattes upp, en backlog skapades och delades ut till olika kodpar.

3.2 Implementation

Vi implementerad sprintar och en backlog till en början men vi tappade processen rätt snabbt. Vi började kommunicera mer via discord och tappade då trello tavlan och den tappade sin mening. Discord och IRL möten användes främst för beslutfattande.

Något som saknades i vår process var en tydlig struktur för ansvar, vem som ha ansvar för att processen följs och vem som har ansvar för att vi går i rätt riktning. Det funkade okej till en början, men när det blev jul och när deadline närmade sig så tappade vi tråden. Både gällande projektet och processen. Besluts togs fortfarande och vi hade en kanal i discord ämnad för det, men det skedde inte på ett strukturerat sätt.

4 Användning av verktyg

Under projektets gång har vi använt oss av discord, github samt trello. Discord användes för vanlig chatt, samt för röstchatt.

4.1 Discord

Vi hade separata chatt-kanaler i discord för "decision-making" och "status-updates" som vi i princip ersatte trello med allt eftersom. Där dokumenterades de flesta saker som behövdes göras. Exempelvis när en person hade gjort en pull request på github så kunde någon annan i gruppen snabbt se det och ta tag i granskningen. När vi använde trello så kollades det inte allt för ofta, och det kändes överflödigt att på discord skriva att man uppdaterat trello.

4.2 Github

Github använde vi till det sedvanliga för att dela kod med varandra, merge:a kod/moduler, se till all kod som pushas upp till master-branchen är kommenterad/testad och är vettigt skriven helt enkelt.

4.3 Trello

Trello användes i början "som det skulle", det vill säga att vi lade upp planering och 'TODOs' för varje sprint och programmeringspar. Men det tog bara någon vecka innan gruppen gav upp på det, ca efter juledigheterna.

I efterhand så kunde trello varit användbart med planeringen, men eftersom de flesta var vana vid discord samt att det fanns röstchatt så övergick vi dit. Det var lite svårt att komma ikapp på trello när vi använde discord så mycket under jul, då det var enklare samt de flesta hade det på telefonen.

5 Kommunikation, samarbete och koordination

I allmänhet har grupparbetet fungerat bra. Vi har haft regelbundna fysiska möten (med undantag under jul-dagarna då vi hade discord-möte istället) och varit aktiva både på github och discord. Just discord har använts väldigt mycket, vi har i snitt haft ca 45 meddelanden per dag under projektet.

Den största spänningen i gruppen var kanske i början när vi diskuterade tekniska detaljer. Till slut kom vi fram till att diskussionen inte ledde någonstans och efteråt bestämdes det att implementationsdetaljer bestäms av kodparen.

6 Arbetsuppdelningsstruktur

En av de första sakerna vi gjorde när projektet startade var att bestämma ett system för hur vi fördelar arbetet i gruppen. Ett enigt beslut togs att dela upp gruppen i kodpar.

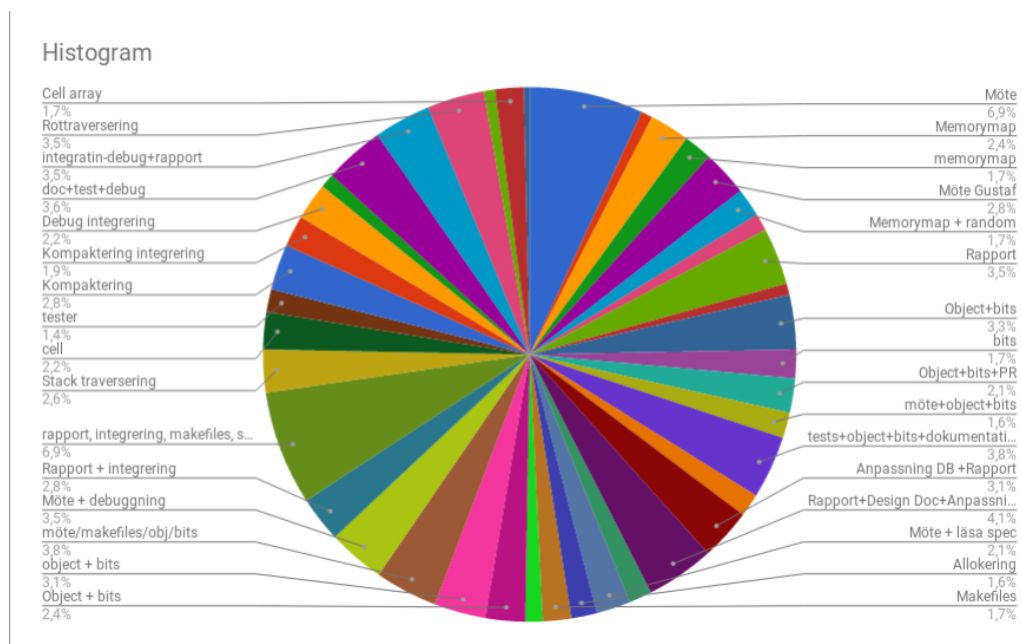
6.1 Kodpar

Varje kodpar skulle få en del uppgifter per sprint, sen så skulle paren roteras efter varje sprint och nya uppgifter delas ut. Ett problem var att vi är ett

udda antal projektmedlemmar men det löste genom att en person skulle få jobba själv per sprint. Upplägget var en bra plan på pappret.

Tyvärr så höll den inte så bra. Den första sprinten gick smärtfritt, då gjorde vi mest saker som handlade om att sätta upp teststrukturen. Den andra sprinten så började vi med projektet på riktigt.

Ett par jobbade med hur vi skulle allokerat objekt på vår heap, en grupp jobbade på stacktraversering och den som jobbade solo började med en memorymap. Det visade sig att objekt paret hade fått en väldigt stor del och de var tvungna att fortsätta med det sprinten därefter. Arbetet fortsatte med att jobba på olika moduler, men sprintarna glömdes bort. Mot slutet av projektet så har vi suttit tillsammans i skolan och integrerat modulerna.



6.2 Vad kunde varit bättre

För att hålla våra tidsbudgetar så borde vi brutit ner projektets delar i mindre delar. Hur vi allokerar objekt var en rätt så stor modul med många delar. Det hade vi insett tidigare om vi brutit ner stegen tillsammans och då hade vi kunnat få ett bättre tidsestimat.

Vi pratade om hur vi kunde ha löst detta bättre nu med facit i hand. Då kom

vi överens om att vi borde tillsammans gjort gränssnitten för att få bättre översikt vad det faktiskt är vi vill kunna använda vardera modul till. Då skulle vi även ha kunnat dela upp arbetet i funktioner, haft mindre sprintar men mer koll på arbetsprocessen.

7 Kvalitetssäkring

Vid projektets start så satte vi oss ner och utarbetade en policy för hur vi skulle jobba för att hålla en bra kvalitet på arbetet. Vi skapade en kodstandard och formulerade hur vi skall jobba med pull requests och branches muntligt.

7.1 Kodstandard

Resultatet blev en fil "Kodstandard.txt" i repot som omfattade dels hur vi ska strukturera koden, men även hur pullrequest ska gå till. En pullrequest inte får merge:as med master repot om den inte granskats av någon som inte skrivit koden. Det här har ändrat hur de flesta i gruppen tänker på sin kodning. En del har försökt att göra så tydliga commits som möjligt så att det finns en röd tråd i historiken, andra har tänkt mer på läsbarheten i sin kod än vanligt medans en del inte kodat annorlunda. Det har varit en lärorik erfarenhet för alla.

7.2 Pull requests

Under projektets gång har också kvalitén på pullrequests ökat. Vi har gått från namn som inte säger så mycket till väldigt beskrivande namn. Ett exempel på ett tidigt namn är "Hoppas jag gör rätt". Mot slutet så hade vi bättre namn som "Div. ändringar i gc-modul, compact-modul lades till, memmap-modul är ändrad". När det kommer till feedback till pull requests så har det också kännas okej för gruppen att komma med kommentarer. De flesta kommentarer och ändringar på pullrequests har varit på läsbarhet av koden.

7.3 Testning

Vi har försökt att testa så mycket som möjligt, men med tanke på naturen av vårt projekt så har det varit svårt att göra integrationstester tidigt. Istället så har vi förlitat oss på enhetstester som skrivits av implementatörerna. Man

skulle kunna anse att det finns en stor risk i det då de antingen kan förstå specen fel eller har ett bias när de skriver sina tester.

7.4 Kvalitén på arbetet

Kvalitén är okej givet förutsättningarna, det fanns en del oklarheter i början av kursen om vad vi skulle implementera. Det löstes dock efter diskussioner inom gruppen. Så här i efterhand har många känt att vi kanske haft för hårda regler när det kommer till pull requests och att det saktat ner arbetet. Speciellt har det saktat ner integrationen. Allt som allt blev det 23 pull requests gjorda med ca 1 kommentar per pr. Mycket feedback som kanske skulle hamnat under kommentarerna hamnade istället på discord.

8 Reflektion

På en skala 1-7 så bedömer vi vår tillfredsställelse med processen 4/7. Vi är nöjda med processen och scrum fungerade bra. Det som drar ner det främst är att vi kunde ha delat upp uppgifterna i mer detaljerade delar. Vi hade också en inbalans i arbetsbelastning sent i projektet relativt tidigt i projektet. På en skala 1-7 så bedömer vi vår tillfredsställelse med produkten 4/7. Den är ej komplett men vissa delar fungerar helt ok. Kanske kunde vi ha fokuserat mer på funktionalitet än optimering.

På en skala 1-7 så bedömer vi vår tillfredsställelse med kvalitetsäkringen 4/7. Vi är nöjda med vår kodstandard och arbetet med pull requests har fungerat helt ok, men vi kan ha varit lite för hårda med vad de ska innehålla. Vissa tester är helt ok.