

Självreflektioner

Innehåll

1 Robert Rosborg	2
1.1 Självreflektion	2
1.2 Parprogrammering	2
2 Simon Pellgård	3
2.1 Självreflektion	3
2.2 Parprogrammering	3
3 Daniel Ågstrand	4
3.1 Självreflektion	4
3.2 Parprogrammering	4
4 Henrik Bergendal	5
4.1 Självreflektion	5
4.2 Parprogrammering	5
5 Maria Lindqvist	6
5.1 Självreflektion	6
5.2 Parprogrammering	6
6 Adam Inersjö	7
6.1 Självreflektion	7
6.2 Parprogrammering	7

1 Robert Rosborg

1.1 Självreflektion

En styrka jag har är att jag kan ta till mig en uppgift och vara helt koncentrerad på att arbeta med den tills att den är löst. Det har passat rätt bra under projektet eftersom arbetet har delats upp i mindre uppgifter och man har koncentrerat sig på sin deluppgift. Det har även varit bra när buggar skulle lösas då jag suttit länge (kanske för länge) och systematiskt letat efter fel.

En svaghet jag har är att jag har svårt att ta initiativet och ofta låter andra tilldela mig en uppgift att arbeta på, och detta kan ibland leda till att jag får uppgifter som jag inte tycker om eller inte är lämpad för. Det har dock inte varit ett problem i projektet. I framtiden borde jag kanske tänka mer på att ta mer initiativ.

Att jobba i grupp eller ensam har både fördelar och nackdelar. Om man jobbar ensam kan man själv välja när och var man vill arbeta, men i grupp måste man komma överens med övriga medlemmar om tid och plats. Men om man jobbar i grupp så kan man alltid ta hjälp av andra om man fastnar. Det är just det som är bra med grupper; ingen är bra på allt men alla är bra på något.

Jag känner att jag lärt mig väldigt mycket av projektet. Det känns som om vi har gjort någonting "riktigt" och inte bara någon godtycklig uppgift och det har varit intressant att arbeta i en lite större grupp och det är kul att det har fungerat så bra.

1.2 Parprogrammering

Parprogrammering är väldigt bra när man arbetar med en uppgift som man inte riktigt vet hur man ska lösa eftersom man hela tiden kan bolla idéer med varandra och man kompletterar varandras kunskap, två hjärnor är bättre än en, typ. Men om man har väldigt klart för sig vad som ska göras och det egentligen bara handlar om att få ner koden så är det ju mer effektivt om man programmera var för sig.

Vissa par fungerar naturligtvis bättre än andra, helt enkelt på grund av personkemi. Om båda har en jämn kunskapsnivå så kan det vara lättare att lösa uppgiften men man kanske inte lär sig så mycket, men om man har en ojämn kunskapsnivå kan båda lära sig någonting.

Ett råd till någon som aldrig har parprogrammerat tidigare är att kommunicera så mycket som möjligt. Att hela tiden prata med varandra om koden ökar förståelsen för båda och försäkrar också att båda är "på samma sida".

2 Simon Pellgård

2.1 Självreflektion

Mina största svagheter under projektet har varit stress och testdriven utveckling. Jag blir lätt stressad, vilket under ett mycket stressigt projekt som det här förmodligen innebär att jag inte presterat så bra som jag skulle kunna ha gjort annars. Den testdrivna utvecklingen har varit problematiskt dels för att jag inte har tålamodet för det när jag är stressad men framförallt för att jag har svårt att förstå problemet innan jag börjat koda. Jag tycker inte att jag haft några särskilda styrkor under projektet.

När jag arbetar i grupp så tenderar jag att bli något passiv och ser gärna att någon annan tar mer aktiva/bestämmande roller. Huruvida jag föredrar att jobba i grupp eller ensam beror på arbetsuppgiften. I lite mer konceptuellt svåra projekt som det här så föredrar jag att arbeta då det hjälper min förståelse mycket att diskutera problemet. I enklare projekt föredrar jag att arbeta själv då det ger mig större frihet.

Jag har lärt mig betydligt mindre under projektet än under övriga delar av kursen. Jag lär mig bäst genom föreläsningar/att se hur någon annan gör, blandat med individuellt arbete med färre sociala krav och stress.

2.2 Parprogrammering

Parprogrammering genererar definitivt bättre kod än enskild programmering. Eftersom koden måste förstås av två blir den mer lättförståelig. Dessutom är det väldigt användbart att ha någon att diskutera med när man inte förstår någonting eller kör fast.

Vissa par fungerar bättre än andra. Detta beror främst på skillnader i parmedlemmarnas kunskapsnivå och personkemi. Personligen så tycker jag att parprogrammering fungerar bäst när parmedlemmarna har ungefär samma kunskapsnivå. Det är då lätt att förstå och diskutera problem/lösningar tillsammans. När parmedlemmarna har stora kunskapsskillnader händer det lätt att den kunnigare personen tar över arbetet helt så att det i praktiken är enskild programmering men med en person som sitter brevid. Man skulle kunna argumentera att den mindre kunniga personen ser andra fel och simplare lösningar än den kunnigare. Vidare måste den kunnigare personen förklara tydligare för att den andra personen ska förstå, vilket kan leda dels till att den kunnigare personen tvingas tänka mer på problemet/koden och därigenom se potentiella brister/förbättringar, dels till att koden blir mer lättförståelig. I min mening väger dock dessa potentiella fördelar inte upp för nackdelarna.

3 Daniel Ågstrand

3.1 Självreflexion

Jag anser mig själv vara hyfsat snabblärd och snabbtänkt i allmänhet. Dessa egenskaper kommer väl till pass när man ska programmera och under projektet så förstod jag hyfsat fort vad uppgiften skulle gå ut på och fick mig snabbt en bra överblick av hur vi skulle kunna åstadkomma vårt mål. Jag kunde sedan hjälpa mina gruppmedlemmar om det var något som var oklart. Självklart kunde jag inte svara på alla frågor, men jag kunde hjälpa till och brainstorma fram ett svar till nästan alla frågor. Jag anser också att jag är hyfsat bra på att hålla uppe en god stämning inom grupper, med hjälp av skämt och tack vare att jag går hyfsat bra med de flesta människor. Jag brukar bland annat alltid teama med den personen som blir lite utanför och försöker därefter få in han/hon i gruppen igen. Dock har detta inte behövs i vår projektgrupp eftersom det inte var någon som blev lämnad utanför.

Min absolut största svaghet är att jag är väldigt otålig och gärna kastar mig in i uppgifter, bland trots att jag inte förstått uppgiften till 100%. Men tack vare min grupp så hände inte detta under detta projekt eftersom de andra gruppmedlemmarna i Humrarna inte tillät mig kasta mig in i uppgiften. Vilket var väldigt tur. För jag hade med väldigt stor sannolikhet missuppfattat något, om vi inte hade haft lika långa planeringstid (ca. en vecka) som vi nu hade.

I vanliga fall föredrar jag att jobba ensam och inte i grupp. Men på grund av komplexitet på detta projekt så var grupparbete att föredra. Jag har aldrig några problem att jobba i grupp, problemet jag har med grupparbete är att man inte alltid får göra som man vill och saker tenderar att ta längre tid i grupparbeten kontra jobba själv. Iallafall i mindre komplexa uppgifter. Men detta beror också mycket på vilka man hamnar i samma grupp som. Under detta projekt har jag haft turen att hamna i en väldigt bra grupp.

Jag har lärt mig en hel del under projektets gång, eftersom det varit så intensivt. Dock känns det lite som om dessa kunskaper är mer så utvecklingar av kunskap vi redan lärt oss. Eller rent av bara repetition av redan lärd kunskap. Men jag antar att det lite är poängen med ett "examens-projektarbete". Att man ska använda sig av kunskap man fått under kursens gång för att uppnå ett mål.

3.2 Parprogrammering

Jag föredrar verkligen att jobba i par när man kodar så pass svåra uppgifter. Problemet jag har om jag skulle koda ensam på ett så pass svårt program som under projektet är att jag skulle fasta lite här och där under längre perioder. Detta tror jag är sant för de flesta. Men jobbar man i par så kan ens partner hjälpa när man fastnar. Detta gör att man fastnar betydligt mer sällan.

Vissa kombinationer av par fungerar självklart bättre än andra, på grund av att alla har olika personkemi. Men så länge alla personer är hyfsat anpassningsbara så bör inte skillnaderna i olika parkombinationer vara så stora. Detta upplevde jag under projektets gång, att jag jobbade bättre med vissa och sämre med andra. Men att det bara var ett anpassa sig lite till den personen man jobbade med för tillfället för att få samarbetet att fungera bra. Dock så finns det en parkombination som jag tror verkligen inte fungerar bra. Den kombon är om ena personen är väldigt blyg och tillbakadragen och den andra är väldigt utåtriktad. Då finns det en risk att den utåtriktade personen kör över den blyga och den blyga slutar att bidra. Märker man att detta sker i något par bör man peppa den blyga så att han/hon vågar bidra. Om man ska jobba i par för första gången är det viktigt att man från första starten försöker ha så bra kommunikation som möjligt. Det är också viktigt att man vågar bidra och säga sitt, men även att man kan kompromissa och lyssna på sin partner.

4 Henrik Bergendal

4.1 Självreflektion

Att arbeta i ett projekt ställer krav på att kunna lyssna på sina medarbetare och vara öppen för förslag. Det är också vad jag tycker var den största utmaningen för mig under projektet. Efter att ha skaffat mig en bild av hur något ska se ut vet jag att jag kan ha svårt att ändra uppfattning. Generellt tycker jag att det gått hyfsat för mig att inte bita mig fast vid ideer utan kunna ändra dem under arbetets gång, men det är något som jag vet att jag kommer behöva öva på även i fortsättningen.

Jag har inget emot att arbeta i grupp, och föredrar det många gånger framför att arbeta själv, däremot är inte parprogrammering något jag alltid finner bra. Jag skulle hellre göra "lätta" delar självständigt och använda parprogrammering för delar av projekt som kan vara tekniskt svåra eller kräva många designbeslut.

När det kommer till lärande så har det många gånger gått väldigt fort att utbyta kunskap och jag har lärt mig en hel del saker väldigt fort. Det har hjälpt när vi bytt par och man har fått sätta sig in i ny kod och teknik och fått en introduktion av någon som är välbekant med det.

4.2 Parprogrammering

Jag tycker att parprogrammering i stor utsträckning leder till kod med bättre kvalitet, i alla fall så länge man i paret lyckas med att kommunicera sina tankar och ideer. Jag tror att mängden kod som produceras med parprogrammering kan variera väldigt mycket i förhållande till den kod två personer som suttit enskilt hade producerat. Vid enkla problem, där tiden mest tid går åt för att skriva kod tror jag att man producerar mindre än hälften än om man suttit enskilt då viss tid går åt till kommunikation. Däremot vid svårare stycken kod kan produktiviteten gå upp då man har två personer som kan knäcka kluriga tankenötter i stället för en. Dessutom hjälper här samarbete för att lösa problem.

Det är uppenbart att vissa par fungerar bättre än andra, personkemi spelar stor roll i hur samarbetet fungerar. Kunskapsnivån hos de båda personerna spelar roll i vissa fall. Är den ojämn kan det gå åt extra tid vid förklarande av koncept, men att förklara saker kan ofta ge en bättre förståelse av saker även för den som förklarar.

Till någon som inte parprogrammerat tidigare skulle jag rekommendera att planera arbetet innan man börjar programmera, detta gör att båda deltagarna är införstådda i en plan som man kommer försöka följa, och det kan leda till att arbetet flyter på bättre.

5 Maria Lindqvist

5.1 Självreflektion

Jag tycker grupparbetet har varit väldigt givande. Dels fick man verkligen fördjupa sig i detaljer inom C, typomvandling, stack, minneshantering med mera. Dels lärde man sig väldigt mycket om att kommunicera och jobba tillsammans i en grupp. Det som var annorlunda mot resten av kursen var att man var mer motiverad då man inte bara pluggade för sin egen skull, utan för att projektet skulle bli klart. Det var även väldigt intressant att se “produkten” växa fram, då man i början jobbade på en liten del, som sedan sattes ihop med resterande delar till något som faktiskt fungerar. Det som var svårast i projektet var att förstå specifikationen i början, och veta var man skulle börja.

Jag är inte den som tar någon ledarroll i en grupp. Däremot är jag bra på att försöka lösa eventuella konflikter, genom att sammanställa en överblick över läget och komma fram till var problemet egentligen ligger och vilka alternativ som finns. Min styrka ligger i att jag är noggrann och systematisk, vilket är bra vid felsökning och även för att minska antalet fel som uppstår från första början. Något jag borde bli bättre på är att hänga med i diskussioner, då jag är mer inåtriktad och tenderar att bli ofokuserad av att lyssna länge.

5.2 Parprogrammering

Personligen tycker jag att parprogrammering under projektet har fungerat bra och varit väldigt givande. Dels så är det mindre risk att man kör fast på rent logiska problem, eftersom man är flera som kan “tänka” och den som inte skriver kan se saker som den andra kanske missar, då den som skriver är fokuserad på ett visst ställe i koden precis då. Jag tror även att koden blir bättre skriven, då man blir noggrannare och genom att man inte kan skriva något konstigt utan att förklara det för sin kodpartner. Om ett kodstycke är helt obegripligt för någon annan än den som skrivit koden, kan man tillsammans skriva om det tills det går att läsa även för andra. Detta bidrar till bättre och underhållsvänlig kod.

Att personerna inom paret ligger på olika nivå tror jag inte är något problem. Den som kan mer får en chans att förklara och dela med sig, vilket är ett tillfälle för repition och befästa sina kunskaper, och den som kan mindre lär sig av den som kan mer. Viktigt är dock att båda är ödmjuka inför den andras kunskaper, och att det inte blir att endast en person skriver, eller att ena inte hänger med utan att säga till om det.

6 Adam Inersjö

6.1 Självreflektion

Jag känner verkligen att jag har lärt mig mycket av projektet. I början tog jag för mycket ansvar över hur projektet skulle fortgå då jag blev Team-leader men snabbt märkte jag att jag mådde bättre av att leda men samtidigt inte behöva planera saker i förväg. Jag har också lärt mig mycket om Test-driven utveckling (TDD) och om hur man får samman något i en större grupp.

Det jag behöver bli bättre på är att släppa kontroll över hur andra delar av projektet utförs. Jag hade svårt att känna tillit till att saker skulle göras på ett bra sätt, och denna känsla var kontraproduktiv då det fick mig att abstrahera mindre och grotta ner mig mer i alla detaljer.

Att jobba i team har varit väldigt givande. Det var intressant att se hur jag agerade när det inte fanns någon annan i gruppen som tog ledarrollen. Jag brukar trivas bäst när någon annan leder och jag bara gör min grej, men det var nyttigt för mig att testa på att leda. Jag känner att när jag leder igen någon gång så kommer jag ta med mig av det jag lärt mig under det här projektet för att göra det bättre, både för mig själv och min grupp.

Projektet var som sagt väldigt lärorikt, men det är i kombination med hur tidigare delar av kursen varit som jag verkligen kunde ta till mig allt. Parprogrammeringen tidigare i kursen byggde upp för att jobba i grupp och jag behövde förstå grunderna för att kunna komma vidare. Jag känner att projektet fått många av de tidigare delarna av kursen att verkligen fastna och att många teoretiska delar blev applicerade konkret.

6.2 Parprogrammering

Jag känner att parprogrammeringen har varit givande, men i vissa tillfällen frustrerande. Det har varit givande för att det har hjälpt till att hitta fel man gjort med koden eftersom man behöver förklara hur man tänker. Det har också varit givande med tanke på hittandet av lösningar på problem man stöter på, två personer ser olika på saker och detta leder till att t.ex. det varit lättare att komma på tester som verkligen täcker allt. Det har varit frustrerande när man gjort saker som när man väl ska skriva ner testerna, dvs. när man gör saker som inte kräver lika mycket tänkande eller diskuterande. I dessa fall har det känts som att parprogrammering varit en last istället för en tillgång.

Sammantaget har det dock varit till stor nytta under projektet. Jag tycker att en jämnare kunskapsnivå har gett bäst samarbete för ett så stressigt projekt som detta. Det känns som om att tiden för att verkligen gå igenom saker inte riktigt funnits.

Mitt råd skulle vara att tidigt acceptera att vissa saker kommer ta längre tid när man parprogrammerar, men att se att det kommer bli bättre kvalitet på programmet i slutändan.