

# TDP003 Projekt: Egna datormiljön

## Reflektionsdokument

Författare

Joakim Johansson, joajo229@student.liu.se

# Innehåll

<b>1</b>	<b>Revisionshistorik</b>	<b>2</b>
<b>2</b>	<b>Starten på projektet</b>	<b>2</b>
2.1	Första anblick . . . . .	2
2.2	Motiv med projektet? . . . . .	2
2.3	Bakgrund? . . . . .	2
<b>3</b>	<b>Partner?</b>	<b>3</b>
3.1	Vem gjorde vad? Hur har ni arbetat? Fördelar? Nackdelar? . . . . .	3
3.2	Vilka verktyg har ni använt och fungerade det bra? . . . . .	3
3.3	Hur var det att samarbeta överlag? Erfarenhet av arbetssätt? . . . . .	4
<b>4</b>	<b>Arbetsgången</b>	<b>5</b>
4.1	Projektplan, LoFi och Installationsmanual . . . . .	5
4.2	Datalagret . . . . .	5
4.3	Presentationslagret . . . . .	6
4.4	Systemdokumentationen . . . . .	6
4.5	Testdokumentationen . . . . .	7
<b>5</b>	<b>Hur påverkas arbetssättet?</b>	<b>8</b>
5.1	Möjligheten att dela upp arbetet på olika personer? . . . . .	8
5.2	Möjligheten att dela upp arbetet efter kompetens? . . . . .	8
5.3	Total tidsåtgång att lösa problemet? . . . . .	8
5.4	Möjligheten att arbeta effektivt? . . . . .	8
5.5	Kvalitén på lösningen? . . . . .	8
5.6	Möjligheten för utomstående att snabbt anamma strukturen? . . . . .	9
<b>6</b>	<b>Projektet i sitt hela</b>	<b>10</b>
6.1	Bra? Dåligt? . . . . .	10
6.2	Hur kunde detta projekt vara utformat för att vara mer givande för dig? . . . . .	10
6.3	Fyra konkreta erfarenheter som kan användas på alla typer av projekt i framtiden . . . . .	10
<b>7</b>	<b>Referens lista</b>	<b>12</b>

## 1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.?	???	161019
1.?	???	161019
1.0	Skapat dokumentet och byggt upp en struktur	161019

## 2 Starten på projektet

### 2.1 Första anblick

Jag upplevde att första anblicken av kursen och projektet var ganska rörig, det kändes liksom som att ingen visste riktigt vad som skulle ingå i kursen och vad alla skulle göra. Det var säkert inte så, det kan ha varit det att jag är ny på universitet och inte hade riktigt koll på hur allt fungerade ännu. Men jag upplevde att de andra i klassen också tyckte att det var lite rörigt.

Om jag inte minns fel, så på vår första föreläsning så fick vi höra om hur vi skulle börja projektet fast än vi inte visst riktigt vad det var för projekt vi skulle göra ännu. Detta gjorde att hela början på projektet var lite rörigt. Jag tycker att det skulle finnas en tidigt föreläsning i kursen som hade med en översikt över projektet där man gick igenom varje steg väldigt grundligt så att man senare är redo på de olika stegen när vi kom dit i projektet. Jag märkte senare att det fanns all information som behövdes på kursens hemsida så det är stor chans att det var mer klart för andra personer och att jag fortfarande är vann med gymnasiet stil att undervisa, men detta är ett reflektionsdokument så jag tyckte att det var värt att nämna det.

Senare under projektet så tyckte jag också att det var lite rörigt och ingen visste riktigt hur man skulle lämna in uppgifterna? Ibland var det till kursledaren och ibland var det till handledaren? Det hjälpte inte att de andra kurserna som går samtidigt också hade olika inlämningsätt. Det borde finnas ett inlämningsställe för alla uppgifter, för alla kurser, där man även kan se hur man ligger till i kurserna och se sina betyg i olika moment!

### 2.2 Motiv med projektet?

Motivet med projektet var att vi skulle bygga en hemsidan med tillhörande server sida som skulle lägga in dynamisk information på hemsidan. Det var ganska mycket fokus på olika arbetssätt och hur man jobbar i grupp på ett bra sätt.

### 2.3 Bakgrund?

Jag har jobbat mycket med webb utveckling och webb design förut men aldrig gjort det med Python som server sida språk, detta var mycket intressant! Jag visste direkt att jag inte skulle ha något problem med HTML/CSS sidan av hemsidan men jag kände att det inte var mitt mål med denna kurs ändå. Jag kände att mitt mål var att lära sig mer om hur man arbetar i grupp och hur systemet fungerar när man använder Python på server, jämfört med PHP som jag använt tidigare. Det var även skönt att det var en tidigt kurs i mitt första år som handlade om webb då jag inte behövde lägga jätte mycket ansträngning på kursen och kunde istället lära mig alla system och saker som har med att gå på universitet att göra.

## 3 Partner?

### 3.1 Vem gjorde vad? Hur har ni arbetat? Fördelar? Nackdelar?

Vi jobbade i par och jag var med Arty. Jag tyckte att samarbetet gick väldigt bra. Vi försökte att dela upp arbetet så bra det gick men det var svårt när det blir oklara krav för vad man skulle göra. Då gick i stället ihop och försökte klara av motståndet tillsammans.

Vi delade upp varje del av projektet på en viss person där vi tyckte att denna person skulle ha ansvar för att den delen vart klar. Det verkade som en bra idé från början eftersom det skulle bli mer effektivt. Men det visade sig senare att många av de delarna som vi skulle göra i projektet skedde en efter den andra och inte flera under samma gång, vilket gjorde att det vart svårare att dela upp arbetsuppgifterna.

Det slutade ofta med att vi satt bredvid varandra och jobbade på samma dator. Vid dessa tillfällen vart det så klart bättre skriven kod men det var ofta väldigt ineffektivt då det tog upp bådas tid. Om man istället hade hittat två uppgifter som kunde göras parallellt så hade man kunnat bli klar mycket snabbare. Men vi försökte att dela upp jobben så ofta vi kunde, till exempel när det var en övergång mellan två delar i projekten, när datalagret börjar bli klart och man skulle starta med presentations lagret. Då arbetade vi med det vi hade ansvar över enligt vår tabell.

Precis som det står i Code Complete (Steven C. McConnell, 2004:s826) så är det väldigt viktigt att man är ärlig mot sina kollegor och säger till om man sitter fast och behöver hjälp. Annars kommer det alltid tillbaka i ett senare tillfälle och det blir en massa problem då istället. Genom att vara ärligt när man jobbar i grupp så gör man att alla i gruppen har alltid full koll hur det går i projektet och det bli lättare för alla att planera sitt jobb.

Här kan du se tabellen:

Områden	Ansvarig	Vem jobbade på det
API	Arturas	Arturas, Joakim
Data_test	Arturas	Arturas
LoFi	-	Arturas, Joakim
Flask	Joakim	Joakim, Arturas
HTML5	Joakim	Joakim, Arturas
CSS3	Joakim	Joakim

Jag tror att det skulle kunna gått lite bättre om vi hade arbetat med de olika delarna med parallellt med i detta projekt så var de olika delarna ganska beroende på att den tidigare delen var klar. Men jag antar att man kan böja på reglerna lite och börja med de andra delarna parallellt så lång man kommer utan de andra delarnas data. Ett annat alternativ hade också varit att dela upp projektets delar i mindre delar så att man kunde båda två jobba på t.ex. datalagret. Men detta hade skapat problem i sig. Våra delar var ganska små och all kod rymdes oftast i en fil. Detta gjorde att om vi skulle jobba på samma fil så skulle vi behöva vara väldigt noga med att inte redigera på samma ställe i filen då det skulle bli en konflikt och den ena personens kod skulle sluta fungera efter sammanslagningen. I t.ex. presentationslagrets Python fil så var detta en viktigt då vissa delar av koden var beroende av andra för att fungera. Datalagret var mycket annorlunda då alla funktioner var oberoende av varande och vi kunna utan problem jobba på olika funktioner samtidigt.

### 3.2 Vilka verktyg har ni använt och fungerade det bra?

Vi har använt git som versionshanteringssystem. Jag har testat på det i projekt förut men jag kunde det långt ifrån flytande. Det var härligt att kunna jobba med det igen. Men jag märkt att efter en stund så

puttade vi upp data till git med dåliga namn på commiten. Om vi hade varit lite mer seriösa och skrivit bra meddelande i git pusharna så hade vi haft en väldig hjälp när vi skulle skriva dagboken. Det hade även blivit enklare om vi skulle behöva rulla tillbaka till en äldre version av systemet ifall vi hittar en bugg ifall varje förbättring i systemet hade en egen commit. Men så var det inte för oss. Vi använde det lite mer som lagring så som Dropbox, då vi inte riktigt var vana och som tur är så behövde vi inte använda oss av någon avancerad GIT funktion, så som att rulla tillbaka, så det gick ganska bra för oss ändå.

GIT verkar som en bra sak att kunna väldigt bra i framtiden så man med stor sannolikhet kommer att arbeta med det ute i arbetslivet om några år. Det är ett väldigt smidigt system och när man väl har vant sig med att använda det så kan man inte dra sig ifrån det!

Latex är ett annat verktyg som vi använde oss av. Jag måste säga att det inte är min favorit. Jag kan se poängen med det och det är ett optimalt sätt att hantera dokument. Men när jag skriver ett dokument så vill jag gärna att den texten jag skriver ska se fin ut, då jag får mer motivation att fortsätta skriva. Det arbetssättet vi har kört med i våra dokument är att man ska skriva all sin text i en .tex fil och sedan kompilera den varje gång man vill se resultatet. Texten som man skriver i Latex blir så klart väldigt mycket funktionalitet före design men jag tycker att det går lite över gränsen när man ska skriva kod liknande kodblock för att skapa en stycke, rubrik eller andra saker som man behöver i sitt dokument. Detta tycker jag göra att hela dokumentet ser ut som kod och det är svårt att hitta när det blir långa texter. Men jag kan se poängen med att kunna skicka Latex filer till varandra då de är universella och alla kan kolla på dem så länge de ha en kompilator installerad på sitt system. Fördelen är också så klart att man kan ändra designen på sitt dokument på ett enkelt sätt utan svårigheter, men detta tror jag är möjligt även i en vanlig texteditor så som Word eller OpenOffice.

### 3.3 Hur var det att samarbeta överlag? Erfarenhet av arbetssätt?

Jag tycker att samarbetet har fungerat bra överlag. Jag är mycket nöjd med Artys insats och även min egna. Som jag skrev i tidigare stycke så var det svårt att dela upp projektet i delar som gick att arbeta med parallellt men om man lyckas med detta bättre så är det en riktig fröjd att arbeta grupp för att man går så många saker gjorda på så kort tid!

Steven McConnell skriver i sin bok Code Complete (Steven C. McConnell,2004:s483) att man inte ska trycka på med parprogrammering på de enkla sakerna. Jag håller fullständigt med. Det är inte värt att ta upp flera personers arbetstid för att kunna göra en enkel uppgift som vem som helst skulle kunna göra. Det är mer när det kommer till svåra programmerings uppgifter som det är bra att man jobbar i par eftersom den ena kan skriva och den andra letar buggar och jobbar med problemlösning.

Jag kommer ta med mig erfarenheten om att jobba i par till framtida projekt. Då ska jag komma ihåg att försöka dela upp projekten lite mer i delar som går att jobba i parallellt.

## 4 Arbetsgången

### 4.1 Projektplan, LoFi och Installationsmanual

Vi gjorde en projektplan ganska tidigt i projektet, vilket var bra, men vi kände att vi borde ha skrivit och planerat lite mer om de olika delarna av projektet. För att senare i projektet så fanns det ibland tillfällen där vi inte visste riktigt vad vi skulle göra. Det ledde till att vi flera gånger i rad vände oss emot kursens planering på kurs hemsidan för att kolla upp vad som var nästa deadline och så gjorde vi det som behövdes göras för att klara det. Det optimala vore förstås att ha alla deadlines och uppskattade tider i projektplanen så att vi alltid kunde vända oss dit för att kolla vad vi ska göra. Vi hade förstås med deadlines och mål för när vi skulle vara klara med vissa uppgifter men de var inte tillräckligt bra skrivna för att vi skulle vända oss dit istället för kurs hemsidan. Vi upplevde även att saker och till flyttades runt och ändrades i kursens planering vilket gjorde att vi inte riktigt litade på den planeringen som vi skrev vid en tidigare tidpunkt och istället förlitade oss på kurs hemsidans planering och försökte höra oss för i klassen vad som behövde göras i kursen.

Vi gjorde en LoFi först på papper lite grovt för att få en överblick över hur hemsidan och systemet skulle vara uppbyggt och se ut och sedan så gjorde vi en renare version med enkla boxar och texter i Photoshop. Detta vart väldigt bra tycker jag och det passade bra in i vårt projektplans dokument. Men eftersom denna kurs inte hade några krav på hur hemsidan skulle vara utformad så blev det en väldigt enkel hemsida och jag upplevde att den var så enkel att det inte hade behövts någon LoFi alls. Men jag förstår att det är viktigt när det gäller större hemsidor.

Vi skulle även skapa en Installationsmanual, först en i varje grupp men det vart senare ändrat så att man skulle skapa en gemensam för hela klassen. Detta vart mycket rörigt då vi hade börjat på en manual och sedan så skulle vi lägga ihop den med alla andras. Men det vart en bra installationsmanual tillslut som var lätt att läsa och följa.

### 4.2 Datalagret

Datalagret byggde vi upp genom att skapa en pythonfil som har en massa funktionen i sig för att hantera data. Den har till exempel en funktion för att ladda in data ifrån databasen, olika sök funktioner och även andra nödvändiga funktioner som kan vara bra att ha. Denna fil importeras sedan som en modul in i presentationslagrets python fil där flask körs.

All data sparades i en JSON databas-fil som låg på servern. Fördelen med att ha ett API byggt på detta sätt är att om man behöver ändra på hur datat sparas på servern så behöver man bara ändra på sitt API och resten av systemet kan vara kvar som det är. Detta fungerar så länge det finns stadiga krav på vad varje funktion i sitt API sak ta för indata och ge för utdata och att dessa aldrig ändras.

Det var Artys del att jobba med datalagret för det mesta så han vet mest av hur det fungerar. Men vi brukar väldigt ofta gå ihåg och förklara för varandra vad vi gjort så att båda är med på vad som händer i koden och får förståelse för varför det fungerar som det gör. Men eftersom datalagret är uppdelat i separata funktioner så gjorde vi så att vi jobbade tillsammans i början för att försöka förstå oss på vad vi ska göra. Sedan när vi har förstått vad som måste göras så delade vi upp arbetet lite och vi gjorde några funktioner parallellt. Men efter en stund så började jag med presentationslagret istället medans Arty gjorde klart datalagret. Vårt arbetssätt fungerade mycket bra tycker jag!

Jag tycker att slutprodukten av datalagret vart ganska bra. Det vart några funktioner som vart lite röriga att följa med även fast man hade kommentarer. Men detta var inte ett problem när man väl fått en förklaring om hur det fungerar av sin partner. Men jag ser att man skulle kunna förbättra denna del av projektet lite

gran. Kanske även lägga till några fler bra att ha funktioner. Men i vårt scenario så behövdes det så klart inte då vi hade redan bestämda funktioner enligt kravspecifikation som täckte alla saker som senare behövdes i presentationslagret.

### 4.3 Presentationslagret

Presentationslagret är uppbyggt av en python fil som körs helatiden med hjälp av modulen som heter flask. Detta är själva webserver som svarar på HTTP förfrågningar. Denna fil importerade sedan funktioner ifrån vårt API för att hantera data som den sedan lade in i HTML mallar med hjälp av modulen som heter Ninja. Väldigt intressant kombination av programspråk som jag aldrig testat förr.

Jag byggde början på presentationslagret medan Arty färdigställde datalagret och dess funktioner. Eftersom presentationslagret inte var så stort och det var väldigt mycket kopplingar mellan den och datalagret så kunde jag inte göra så mycket för än datalagret var klart. Men när datalagret vart färdigt så var även början på presentationslagret klart vilket blev perfekt tid för oss att sätta ihop dem. Detta blev ett ganska stort och komplicerat jobb så vi jobbade mycket med presentationslagret tillsammans. Under en senare stund så började vi även jobba parallellt igenom då jag jobbade med HTML mallarna och CSS filerna medan Arty jobbade flitigt med datalagret. Dessa två blev sen också ihop satta och de fungerade bra tillsammans.

Jag tycker att resultatet blev mycket bra och det är lätt att lägga till fler undersidor i framtiden ifall det skulle behövas. Vi kunde ha förbättrat vår felhantering. I nuläget så renderar den error-mallen bara om det blir error 404 eller om projektets id inte hittas. Man skulle kunna ha med fler felhanteringar för att göra det snyggare för kunden, men vår sida är helt okej för vårt ändamål tycker jag.

### 4.4 Systemdokumentationen

När systemet började närma sig sin slutliga form så började vi skriva på en systemdokumentation där vi beskriver hela systemet i detalj för framtida utvecklare på systemet ska för en snabbare start att vänja sig med allt.

Jag tycker att vår systemdokumentation var bra, vi kanske skulle kunna ha skrivit något mer på vissa delar men globalt så tycker jag att det vart ett bra dokument då vi beskrev mycket bra om det mest nödvändiga delarna av systemet vilket gjorde att en utomstående skulle kunna sätta sig in i systemet relativt enkelt utan för mycket möda.

När vi skrev vår systemdokumentation så upptäckte vi att det fanns problem som kunde uppstå då vi inte hade gjort vårt system så generellt som vi skulle ha velat. Men vi byggde enligt kravspecifikation men om det fanns en riktig kund så hade jag tagit kontakt och frågat om förändring. Vi upptäckte att när man laddar in databasen med vårt API så skickas all data till presentationslagret och sedan skickas tillbaka för att sedan filtreras och minskas till en storlek som är okej att arbeta med och visa för användaren. Detta är inget problem för det system som är upp just nu då både vårt API och datalagret körs på samma dator. Men om vårt API skulle ligga på en annan dator relativt till datorn med presentationslagret så skulle det bli stora fördröjningar och väntetider för användaren då hela databasen skickat mellan datorer för att göra filtreringar. Detta kanske inte är ett så stort problem när det gäller ynka fyra projekt i databasen men om det kommer upp emot tusental så blir det lite att skicka mellan datorer. När man kollar på andra publika API:er så brukar de ofta ha så att man göra endast en förfrågning med en massa parametrar för vilken data man vill få tillbaka och sedan så jobbar deras API och skickar tillbaka den förväntade datan. På detta vis skulle jag vilja att vårt API också fungerade med det är ett projekt för en annan dag.

## 4.5 Testdokumentationen

Testdokumentationen tycker jag var väldigt rolig att göra. Jag har aldrig gjort en testdokumentation förut och är en erfarenhet som jag tror är bra att ha till framtida projekt.

Våran testdokumentation är ett enkelt dokument som beskriver alla krav som systemet ska klara och sedan en logg på alla tester som har körts men en viss version av systemet och vilket krav som har testats. Jag tycker att det blev ett bra resultat men den data vi lade in var för det mesta exempeldata för att då ett dokument alls. Detta eftersom vi inte hade någon bra version hantering av vårt färdiga projekt. Jag kan förstå nu efteråt att vi borde ha gjort en ZIP med namnet och versionen när vi tycker att projektet är klart. Detta göra att vi kan göra tester på just den versionen och sedan om det inte går igenom så kan vi fortsätta utvecklar och sedan göra mer ZIP filer och tester. Detta skulle vara det optimala arbetssättet att arbeta med systemets versioner. När en version lyckas igenom testerna så skulle det även kunna vara dags att pusha den versionen till det skarpa projektet på Internet där personer t.ex. skulle kunna ladda ner det eller hur man nu publicerar sitt projekt.



## 5 Hur påverkas arbetssättet?

### 5.1 Möjligheten att dela upp arbetet på olika personer?

När man jobbar ensam i ett projekt så tar det oftast väldigt lång tid och det är lätt att man råkar skapa buggar och man inte kan hitta på en lång tid. Genom att vara två eller flera i ett projekt så kan man dela upp arbetet vilket gör att det går i nästan dubbla farten när man jobbar. Fördelen är också att om den ena personen sitter fast i sin del så kan de andra utvecklarna hjälpa till och med stor sannolikhet hitta felet eftersom varje person att ett eget perspektiv på hur man kollar på koden.

Ett annat sätt att jobba med flera personer på ett projekt är att man jobbar i par. Då sitter en person vid datorn och den andra sitter bredvid och hjälper till med att hitta fel och lösningar på problemen. Detta gör att buggar kan hittas tidigare, vilket medföljer att det blir mycket mindre framtida buggar. Det kommer även med stor chans skrivas finare och mer effektiv kod. Jag har även märkt att när man sitter i par och programmerar så blir det en mycket mer defensiv programmering (Steven C. McConnell, 2004:s186). Detta gjorde att vi var mycket mer vakna och mer defensiva när vi programmerade. Eftersom vi inte vet vad t.ex. användaren kommer skriva i sökbaren, vilket skapade en risk att de förstör vår kod och data. Detta gjorde att det var enklare att hitta dessa senarion när vi jobbade i ett par.

### 5.2 Möjligheten att dela upp arbetet efter kompetens?

Genom att dela upp arbetet efter kompetens så kan varje person jobba med den delen av projektet som dem är bra på. Detta medför givetvis att allt går mycket snabbare och man kan leverera produkten till kund mycket snabbare. Nackdelen är att utvecklarna lär sig och utvecklas inte så mycket i andra gemensamhang så mycket, men detta kanske inte är en dålig sak ifall man har utvecklare med spetskompetens och man vill ha det så.

### 5.3 Total tidsåtgång att lösa problemet?

Om man är ensam och arbetar så tar det självklart längre tid att genomföra arbetet gentemot om man var ett team på flera utvecklare.

Om man skulle dra ner på hur lång tid utvecklarna har på sig att göra arbetet så är det står risk för att deadline inte kan nås och att det blir en ineffektiv kod med många buggar. Genom att ge utvecklarna mer tid på sig så kan man minimera risker för buggar och problem i framtiden. Nackdelen är pris och tid.

### 5.4 Möjligheten att arbeta effektivt?

Det blir mycket mer effektivt att arbeta i grupp. Eftersom då kan man jobba med delar av projektet parallellt och på det sättet jobba mer effektivt

### 5.5 Kvalitén på lösningen?

Kvaliteten på projektet skulle nog bli bäst ifall man körde parprogrammering då den ena personen satt vid datorn och den andra bredvid. Men eftersom detta tar mycket tid och pengar så är detta ofta inte möjligt är min gissning.

## 5.6 Möjligheten för utomstående att snabbt anamma strukturen?

När man jobbar i grupp så har jag märkt att jag skriver omedvetet finare kod då jag vet att det är någon annan som kommer kolla på den och försöka förstå vad som händer. När jag jobbar på ett projekt ensam så är det lätt att man skriver kod som ser jätte konstig ut men att man kan själv förstå den, och då bryr man sig inte om att försöka rätta till det.

Så om det är någon utomstående som ska snabbt försöka anamma strukturen och börja jobba på projektet så kan det vara bra om det tidigare har varit flera utvecklare sen innan och att de har haft gott om tid på sig med deras del så att de har hunnit göra deras del av projekt på ett bra sätt med fin skriven kod och bra kommentarer och dokumentation

## 6 Projektet i sitt hela

### 6.1 Bra? Dåligt?

Jag tycker att projektet har varit en nyttig erfarenhet för mig, kanske inte för webbsidans kod skull, utan mer för systemet i översikt skull. Då jag aldrig gjort ett python baserad backend förut så var det ganska intressant. Jag hade visserligen velat använda någon typ av SQL databas för att lagra datan men det är förstås lätt att ändra i framtiden då vi har byggt en generell presentationsdel och behöver bara ändra på API:et och anpassa den för att kontakta en SQL server.

Enda nackdelen som jag kan hitta med detta projekt är att det var lite rörigt under tidens gång, ibland visste vi inte vem man skulle skicka in uppgifterna till, ibland visste vi inte när uppgifterna skulle vara klara då de vart uppskjutna och det hjälper inte att hela kursen startade en vecka senare än planerat. Men allt rede ut sig till slut ändå men det kan kanske bli en förbättring till nästa gång kursen körs?

### 6.2 Hur kunde detta projekt vara utformat för att vara mer givande för dig?

För mig så hade det varit lite mer givande ifall det var mer saker som händer på serversidan i vårt system. Nu är det bara enkel sortering av list och data. Det vore coolt om man kunde använda serverns kraft för att göra någon sorts tung bild- eller videokomprimering, kryptering, fillagring eller något annat tungt.

### 6.3 Fyra konkreta erfarenheter som kan användas på alla typer av projekt i framtiden

En bra erfarenhet som jag fick är att det är bra att kunna använda Git. Vi använde det men det kändes som att vi inte helt gick med på det i början. Detta gjorde att vi inte pushade så ofta och vi var hela tiden rädda för att pusha olika versioner då det skulle bli ihopslagnings problem. Men senare i projektet så lärde vi oss hur man gjorde för att pusha på ett mer effektivt sätt så att det inte blev ihopslagnings problem. Men ibland hände det ändå vilket gjorde att vi lärde oss att klara av det också. Nu tycker jag att det är väldigt smidigt att pusha till git och slå ihop olika versioner. Jag till nästa projekt så skulle jag vilja testa att köra så att varje utvecklare i gruppen har en egen gren som de kan pusha till och sedan vid olika tillfällen så sätter man ihop de olika grenarna till master grenen.

Jag märkte när vi gjorde testdokumentationen att den kunde vara riktigt nödvändig. I nästa projekt så kommer jag även försöka exportera projektet då och då och sätta ett namn och en version på den så att det blir som en släppt version. Då skulle man även göra några tester och dokumentera allt i ett testdokument. Det är underbart att ha ett sådan dokument när man får en bugg och man vet inte i vilken version som den dök upp. Då kan man kolla i testdokumentationen och i de olika versionernas kod för att lista ut vad som orsakade buggen. Underbart.

Vi märkte att det är viktigt att göra en bra projektplan. I projektets gång så upptäckte vi att vår projektplan inte riktigt hade med så många saker som var vad vi skulle jobba med när vi var klara med det vi höll på med just nu. Det vara bara mest uppenbara saker som stod i vår projektplan. Men vill förstås också ha med de uppenbara sakerna men man vill också ha med saker som beskriver hur man ska genomföra projektet. Vi hade inte så mycket sådan information vilket gjorde att vi inte hittade det vi ville när vi öppnade projektplanen mitt i projektets gång. Detta gjorde att vi började mindre och mindre använda projektplanen, vilket var ett misstag! Vi skulle haft en bättre skriven projektplan, vilket hade underlättat för oss under flera tillfällen under projektets gång.

När vi skulle börja på projektet så var vår plan att dela upp olika delar av projektet till olika personer så att det skulle bli mer effektivt och gå snabbare. Detta är en bra plan i teorin men i vårt fall så var det så att vissa delar var tvungen att bli klara före andra vilket gjorde att vi inte kunde jobba parallellt lika ofta som vi ville. Detta säger det att vi skulle ha planerat lite mer på vilka delar vi delade upp och vilka vi gör tillsammans.

## 7 Referens lista

Steven C. McConnell (2004).Code Complete -2nd ed.Microsoft Press. Redmond, Washington 98052-6399