

Matteus Laurent, Johan Levinsson, Oscar Petersson, Erik Peyronson

MatLabb - Erfarenhetsdokument

Högskoleingenjörsutbildning i datateknik, 180 hp

Erfarenhetsdokument - 12 november 2015
Programmeringsprojekt, HT15
TDDI02, Linköpings universitet

Handledare:
Johan Frimodig
Institutionen för datavetenskap

Innehåll

1 README

Detta dokument ska ni skriva för att få anledning att begrunda den process ni gått igenom, samla ihop de tankar den givit upphov till, och summera erfarenheter som kan ligga till grund för framtida projekt/verksamhet. Rapporten i sin helhet bör vara ett par A4-sidor.

Relatera till den 'tidsplan' ni (väl?) satt upp, eller de förväntningar ni med all säkerhet haft. Reflektera annars om hur en bättre planering kunde undvikit några problem ni haft eller hjälpt er på annat sätt.

Gå igenom projektets olika faser (krav, design, implementation, leverans), en i taget. Skriv för var och en hur er plan sett ut, och huruvida ni lyckats hålla er till den eller ej. Analysera orsakerna till att ni eventuellt blev sena, och beskriv hur ni löste de problem ni stötte på. Eller tvärtom; av vilket skäl gick det bättre än ni förväntat er. Blev slutprodukten från denna fas bra nog för att ligga till underlag för nästa fas, eller inte? Vad bestod dess brister i? Vad kunde gjorts bättre, sett ur det perspektivet?

Blev till sist slutprodukten vad ni från början hade hoppats? Vilka faktorer bidrog till att det gick som det gick?

Slutligen ska varje gruppmedlem för sig (ange vem!) beskriva den allra viktigaste personliga erfarenheten/kunskapen som projektarbetet bidragit till, och som säkerligen kommer till nytta i kommande programmeringsprojekt (eller vidare yrkesliv).

Sammanlagt bör detta bli minst 8 erfarenheter, individuella och gemensamma sammanräknade.

OBS! Dokumentet är INTE en kursutvärdering, utan en samling arbetserfarenheter som är värda att komma ihåg till nästa projekt ni är med i. En form av självutvärdering". Har ni synpunkter på kursen lämnar ni dem separat till examinator.

Vad räknas som en dokumenterad erfarenhet? Att skriva ned sina erfarenheter kräver lite eftertanke. Det gäller att skriva ned tillräckligt för att en oinsatt läsare skall kunna göra samma positiva erfarenhet i ett eget projekt, alternativt kunna undvika en negativ erfarenhet. Det bör för varje erfarenhet framgå:

- Hur ni gick tillväga, vad ni gjorde, hur ni gjorde, och varför ni valde den metoden?
- Vad resultatet blev, om tillvägagångssättet ledde till något positivt eller negativt?
- Varför ni är nöjda eller missnöjda med resultatet och varför det blev som det blev?
- Hur kommer det här påverka hur ni närmar er en liknande situation i framtiden? Det vill säga, vad blir det långsiktiga lärdomen från detta?

Exempel på en intetsägande "erfarenhet": Revisionshantering via Git har fungerat mycket bra och hjälpt alla utvecklare i projektet."

Exempel på en bättre formulerad erfarenhet: "Vi använde revisionshantering via Git. Detta hjälpte alla utvecklare hålla reda på senaste versionen och underlättade distributionen av färdiga moduler mellan utvecklarna. Vid två tillfällen kunde vi snabbt hitta fel som uppstod tack vare historiken. Ett problem var att kod som inte fungerade checkades in vid flera tillfällen och det slarvades även med att kommentera vad som checkades in. Detta gjorde att utvecklare spenderade onödig tid på felsökning. Vi borde kommit överens om att bara checka in testad och fungerande kod, men ändå ofta, en modul i taget. I framtida projekt kommer vi försöka

använda revisionshantering, med ett gruppkontrakt där vi kommer överens om att bara checka in fungerande kod för att underlätta samarbetet.”

Döm själv vad som saknas i de två exemplen enligt punkterna ovan. Döm vilket exempel ni själv helst skulle vilja läsa i förberedelsefasen för ett projekt.

2 Att introducera Git

Valet att använda Git och Github för central lagring och versionshantering togs på ett tidigt stadié då det är ett populära och kraftfulla verktyg i sammanhanget. Givetvis innebar detta även en instegströskel extra då jag var den enda som i någon utsträckning använt mig utav det tidigare, och då endast i enkelanvändarmiljö.

Inledningsvis såg jag till att alla fixade användarkonton och därefter bifogades några länkar till vidare läsning samt en kortare hemskrivna introduktion. Jag är dock tveksam till huruvida endera har lästs av någon involverad i projektet, och tack och lov gavs en översiktlig introduktionsföreläsning till Git relativt tidigt vilket gett gruppen ett hum om hur det fungerar.

Tack vare att mycket av projektets initiala del bestod av dokumentation (huvudsakligen via \LaTeX) så blev inkörningsperioden (där användarna bekantade sig med aktuella kommandon) tämligen överkomlig. Därmed dock inte sagt att det har gått smärtfritt att jobba med det. Git är ett kraftfullt och användarvänligt verktyg, men i likhet med det mesta som Torvalds haft fingrarna på så är det selektivt vilka användare det är vänligt emot. Missförstånd kring exempelvis hur commits fungerar har lett till ett par klassiska tillfällen där man valt att spara undan aktuellt arbete och radera lokala mappar för att därefter “pulla” hem repot på nytt. Det är givetvis inte “rätt” sätt att lösa problemen på, men det är åtminstone en lösning som sparar tid när det kommer till lösandet och fokus har då kunnat läggas på hur problemet undviks och på att jobba vidare på projektet.

Git är dock ett verktyg väl värt att bekanta sig med, och jag tror att vi alla kommer att vara glada över att ha gjort det förr eller senare.

3 Ledning och koordinering i ett projekts tidiga skede

Innan utformandet av en designspecifikation kommer flera viktiga steg under ett projekts gång. Utgående från en väldigt grundläggande idé om en produkt, så ska idéer diskuteras och funktionalitet sammanställas för en kravspecifikation. Detta dokument lägger grunden för diskussionen kring designspecen. Det är i dessa tidiga skeden som projektet i stort tar form, och det är viktigt att alla ansvariga hänger med och kan bidra med konstruktiva åsikter för att fullborda en väl genomtänkt design. Detta leder till en mer realistisk implementering med färre frågetecken och effektivare gruppkommunikation - som förhoppningsvis leder till att många timmar kan sparas in.

Som ledare för vår projektgrupp kan man anse att min roll är kritisk för att säkerställa det ovannämnda. Sålunda, en erfarenhet jag kommer att ta med mig från detta arbete är mitt agerande i denna roll, vad för påverkan det hade på dagordningen, och några förslag på korrigeringar för framtida projekt.

Några insatser jag inte är nöjda med:

- Jag försökte egga brainstorming utan att få först få alla ordentligt inblandade.
- Jag var rädd för att styra och ställa för mycket, vilket ledde till:
- Jag pressade inte individer tillräckligt för input.
- Jag var ibland kritisk utan att kunna sätta fingret på varför.

Detta ledde till en skillnad i engagemang under designdiskussionen. De uttalande fick mest utrymme och det fanns en tvekan till att framföra åsikter innan folk hade nått samma nivå. Diskussionen hann inte få tid att mogna. Detta rörde till och ödslade tid när designdokumentet skulle skrivas.

Utöver att ha ovannämnda i åtanke, så är min egna åtgärd att försöka dela upp diskussion i ytterligare pass, så det finns tid att smälta första passet och fundera kring frågeställningarna. Kanske även ge i hemläxa att förbereda programpunkter, för att få alla mer involverade från första början.

4 Designplanering

En upplevelse jag har haft i Programmeringsprojektet så har jag haft en upplevelse kring att planera designen. Jag har upplevt stor förvirring kring att planera designspecen. Jag fann modulerna i diagrammen att vara högst vaga till en början, man var nästan tvungen till att dyka rakt in i det utan en klar vision och förståelse för att faktiskt få gjort någonting av substans, och först efteråt försöka förstå sig på det på en djupare nivå. När vi satt ner och ritade modulerna, så blev det till en början väldigt många. Vi har i TSIU03 gjort en liknande grej. Där går varje modul att konstruera i verkligheten, plocka ut, ta och titta på. I ett programmeringsprojekt på såpass hög nivå så är det mer vagt. Det fanns inte riktigt en tydlig definition om vad som skall vara en modul och vad som inte får vara en modul. Jag antar att det bär med sig att väldigt få saker inte får vara moduler. Denna frihet skulle, som mycket annat med få saker som är fel”, lämna många möjligheter för hur slutprodukten skall se ut. Då blir det förmodligen även svårt att visualisera hur det skall se ut när man är klar med det. Jag gillar inte hur vingligt den här approachen har känts för mig. Jag har som förhoppning att detta förvirrande moment kommer att gardera mig inför framtida liknande situationer, så att även om jag kanske inte fann det så bekvämt nu så kan det iallafall ge mig någon form av färdighet att ta med mig. Om det nu blir så eller inte återstår helt enkelt att se när projektet börjar närma sig klart.

5 Erik Peyronson: Att sätta sig in i ett externt bibliotek

I vårt projekt använder vi två externa bibliotek. QT samt MySql och jag som databas-ansvarig har varit tvungen att sätta mig in i båda, vilket inte var helt lätt. När vi skulle designa vårt system började jag försöka läsa på både om MySQL som jag visserligen hade lite erfarenhet av sen tidigare men framförallt QT och det var oerhört svårt att veta vilken ordning man skulle börja i. Varje funktion eller klass man stötte på var kopplad till en annan och sidospåren blev många.

Den stora miss jag gjorde var att inte direkt börja skriva ett litet testprogram och testa de olika funktionerna som jag stötte på, dels för att lättare förstå hur dom fungerar och interagerar med varandra men också för att undvika missar i designen.

Initialt känndes det som att det skulle vara ett slöseri med tid att börja koda så tidigt i projektet då ingen användbar kod skulle produceras men proceduren att få allting att fungera med kompilering, inkluderingar etc var något som jag ändå var tvungen att göra när kodningen väl satte igång och att testa mer hade definitivt hjälp mig att spara tid i slutändan.