

Matteus Laurent, Johan Levinsson, Oscar Petersson, Erik Peyronson

MatLabb - Erfarenhetsdokument

Högskoleingenjörsutbildning i datateknik, 180 hp

Erfarenhetsdokument - 27 november 2015
Programmeringsprojekt, HT15
TDDI02, Linköpings universitet

Handledare:
Johan Frimodig
Institutionen för datavetenskap

Innehåll

1	Oscar Petersson: Att introducera Git	3
2	Matteus Laurent: Ledning och koordinering i ett projekts tidiga skede	3
3	Johan Levinsson: Designplanering	4
4	Erik Peyronson: Att sätta sig in i ett externt bibliotek	4
5	Regelbundna möten	5

1 Oscar Petersson: Att introducera Git

Valet att använda Git och Github för central lagring och versionshantering togs på ett tidigt stadie 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 sparat 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.

2 Matteus Laurent: 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 förhoppningvis 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.

3 Johan Levinsson: 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.

4 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 ända man skulle börja i. Varje funktion eller klass man stötte på var kopplad till en annan vilket ledde till långa sidospår och många saker man behövde förstå för att förstå.

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

Initialt kändes 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 hjälpt mig att spara tid i slutändan.

5 Regelbundna möten

Även om vi under hela första perioden haft frekventa gemensamma arbetspass så kände vi att det, när projektet nu gått in i det stadie där enskilt arbete blivit en större faktor, är hög tid att ha någon form av regelbundet koordinationsmöte för att underlätta distribuering av arbetsuppgifter, samt för att säkerställa att alla är med på banan och att alla vet vad som väntar och förväntas den närmaste tiden.

Vi bestämde oss därför för att i samband med handledarmötet varje vecka ha ett separat möte inom gruppen. Någon spikad dagordning har i skrivande stund inte upprättats, men de punkter som cementerades vid första mötet var:

- Vad har varje medlem gjort under den gångna veckan?
- Vad ämnar varje medlem göra under den kommande veckan?
- Arbetsmetodik: Vad fungerar bra, vad kan förbättras?
- Stundande deadlines, delmål, planerad frånvaro etc.
- Avstämning mot kravspec.

Att ha hållt dessa möten redan från start hade möjligen kunnat öka effektiviteten en aning, men då mycket av arbetet på det stadiet skedde gemensamt och gruppen även var en delmängd av den parallella projektkursens grupp så skedde en hel del av denna kommunikation fortlöpande. Datatyper och returtyper i designspecifikationen

Vid utformandet och skrivandet av en designspecifikation krävs beslutstagande i ett skede där detaljer ännu ej har stakats ut.

Man har en idé i stort om hur programmet ska se ut, eller snarare, hur det *skulle* kunna se ut och hur det *skulle* kunna implementeras. Det existerar många bra tankar och förslag, men just därför är det viktigt att som grupp bestämma sig för vilka lösningar man vill använda och på vilket sätt man vill implementera dem.

Väntar man, som vi gjorde, kan det senare under kodningsfasen komma upp oklarheter och egna tolkningar. Drjösål kan uppstå för att man är osäker på vilket format man får information som ska användas för den modul man ansvarar för.

Ser man till att reflektera senare designändringar i specifikationen, behövs antagligen mer ineffektivt spenderad tid i det stadiet med en undermålig specifikation.

I vårt fall har vi missat på denna punkt. Många returtyper och parametertyper blev lämnade åt senare”. Tanken var att tillåta bättre idéer växa fram och undvika att låsa in sig, men resultatet av denna metod har istället haft mer negativa än positiva effekter på framfarten av vårt arbete.