

Vad är en bra projektmetod för små IT-projekt?

Ett försök att besvara frågan görs i kursen II1302 ”Projekt och projektmetoder” vid KTH ICT

Samuel Agdelius, sameulag@kth.se

Lucas Wallén, lucaswal@kth.se

Douglas Appelberg, douapp@kth.se

Diacio Uthman, diaco@kth.se

Abstract— Kurs vid KTH ICT.

Syftet och målet med denna rapport är att redovisa för hur vår projektgrupp har besvarat frågan - “Var är en bra projektmetod för små IT-projekt?”. Genom att vid varje iteration av projektet göra reflektioner och göra eventuella ändringar, har gruppen jobbat med att ta fram trovärdiga resultat. Detta har resulterat i att de valda arbetsmetoderna kunnat utvärderas, vilket gett en grund för resultat. Vidare, kan konstateras att undersökningsfrågan kan besvaras för just vår grupp, men inte nödvändigtvis generellt för alla små IT-projekt.

Keywords - SCRUM, Extreme Programming, Projektmetoder,

I. OM DETTA DOKUMENT OCH UNDERSÖKNING

Denna rapport är främst skriven till kunden, men kan även läsas av personer med intresse för det genomförda projektet.

Rapportens disposition består av flera olika delar. En introduktion som sammanfattar de viktiga delarna, en teoridel, projektets undersökningsmetoder, resultaten, analyser och förbättringsförslag, samt en avslutande diskussion är de centrala delarna som rapporten kommer att behandla.

Eftersom rapporten är skriven av projektets deltagare, betraktas innehållet som trovärdigt. Källorna som används har granskats och betraktas som trovärdiga.

II. INTRODUKTION

Detta kapitel ger en kort introduktion om vad som undersöks, hur vi har genomfört undersökningen och varför den har genomförts.

A. Bakgrund

I denna kurs ska vi lära oss om projektmetoder för att genomföra ett småskaligt IT-projekt. I projektet använder vi oss av en raspberry pi som är kopplad till en skärm. När vi skickar meddelanden till raspberryn ska den uppdatera skärmen så att meddelandet visas. Produkten är tänkt att användas av en lärare som kan ha skärmen utanför sitt kontor för att kunna visa statusmeddelanden som till exempel ”Återkommer klockan 15:00”.

Undersökningsfrågan i rapporten är vad som är en bra projektmetod att använda i små IT-projekt.

B. Problemformulering

Problemet som rapporten kommer att ge ett svar för, är frågan: Vad är en bra projektmetod för små IT-projekt?

C. Undersökningsstrategi/lösningsstrategi

Vi kommer genomföra ett litet IT-projekt där vi använder oss av projektmetoden SCRUM, samt andra projektverktyg vi lär oss om under kursen gång. Projektet kommer användas som grund för denna rapport där vi undersöker vad som är en bra projektmetod.

D. Avgränsningar

Vi kommer i projektet att arbeta med många tekniska detaljer, till exempel PHP och HTML kod, dessa detaljer kommer inte att tas upp i rapporten då den syftar till att redogöra för vad som är en bra projektmetod.

III. TEORI OCH INGENJÖRSPRAXIS

Detta kapitel listar och i viss mån beskriver teorier och ingenjörspaxis som använts i undersökningen. Det finns två underkapitel, Litteraturstudie och Förstudie.

E. Litteraturstudie

För att få kunskap om dom metoder som ska användas under kursen gång var vi tvungna att hämta in information om ämnet. Nedan listas den litteratur vi använt oss av.

- Scrum and XP from the trenches, Henrik Kniberg[1]
- Software engineering, Ian Sommerville[2]
- Arbeta i projekt, Sven Eklund[3]

F. Förstudie

Vi har i projektet arbetat med projektmetoden SCRUM som är en iterativ projektmetod för mjukvaruutveckling. Nedan följer en beskrivning av centrala delar i metoden SCRUM.

Product backlog, se figur 1 – En tavla där kundens krav sätts upp. Dessa krav kallas för stories, varje storie prioriteras, ges en uppskattning av hur lång tid den kommer ta samt en beskrivning av hur den kan demonstreras. Utöver kraven kan annan information om projektet finnas här såsom information om projektdeltagare och en tidsplanering.

Sprint – Projektet delas upp i sprintar. Inför varje sprint

hålls en sprintplanering, där tydliga mål för vad som ska göras denna sprint bestäms utifrån dom stories som finns på backloggen.

Sprint backlog, se figur 2 – En tavla som visar hur gruppen ligger till i den aktuella sprinten. Alla stories, som kan ha brutits ner i mindre stories gruppen arbetar med under den aktuella sprinten sitter på denna tavla. En storie befinner sig i en av tre faser. Fas ett är att den inte har påbörjats, fas två att den har börjat arbetas med och fas tre att den är färdig.

Sprint demo – I slutet av varje sprint hålls en demo som visar vilka resultat gruppen har åstadkommit under den gångna sprinten.

Sprint retrospective – Efter varje sprint håller gruppen ett möte där de går igenom vad som gått bra och vad som gått mindre bra under den gångna sprinten.



Figure 1, Product Backlog



Figure 2, Sprint Backlog

IV. UNDERSÖKNINGSMETODER

Detta kapitel beskriver vilka metoder som använts i undersökningen. Metoderna är valda och specificerade så att de skall kunna ge svar på ett antal följdfrågor som identifierats i denna undersökning. Först anges frågorna och sedan följer metodbeskrivning.

G. Frågor att besvara i undersökningen

Frågorna kategoriseras i följande kategorier

1. Hur skall man bedöma/redovisa om en delprojektmetod eller praktik är bra?
2. Vilka ansvarsroller skall användas som ansats i projektet?
3. Vad består ett projekt av och vilka metoder/praxis skall användas, undersökas och bedömas? Vilken ansats skall göras?
3. En viktig roll är Projektledaren, detta för att kunna hålla gruppen sammanfogad och fokuserade på samma mål. Man behöver någon som kan ta de slutgiltiga besluten och bestämma lite om när saker ska göras. Kundansvarige är viktig för att man som projektgrupp ska veta vad som ska produceras, och ha en direktlänk till kunden ifall man behöver mer information eller stöter på problem. Utvecklingsansvarig och Testansvarig är viktiga för produkten. Utan dessa två ansvarsroller skulle produkten förmodligen ha en sämre kvalitet. Arkitekturansvarig är viktig för att man ska ha ett bra överseende över vilka moduler som ska produceras.
4. Det är viktigt att man sätter upp ett mål som man vill nå inom projektgruppen. Man har en projektstart där man planerar hur projektet ska gå till, denna fas är väldigt viktig eftersom det är under denna tid som man kan göra som störst ändringar i projektet. En bra start ökar sannolikheten att projektet blir lyckat. Därför kan man spendera lite extra tid att se till att man gjort alla rätta val, så sparar man eventuell tid i slutändan. Man ser till att alla i projektgruppen vet sina roller och att man går igenom kraven för projektet så att de följs. Att göra en tidsplan för hela projektet, samt att ta reda på vilka resurser som krävs, är viktigt för att skaffa sig en bra förståelse. Gå igenom riskerna som finns i projektet, och besluta hur man kan hantera/undvika dem. När projektstarten är avklarad kan man sätta igång med själva projektet.

H. Metodbeskrivning

Den centrala metoden i undersökningen är att avgöra om olika valda projekt-praktiker och arbetssätt är ”bra” och om de bidrar till att göra hela projektprocessen bra. Åstadkommer/skapar projektet rätt saker och konstrueras

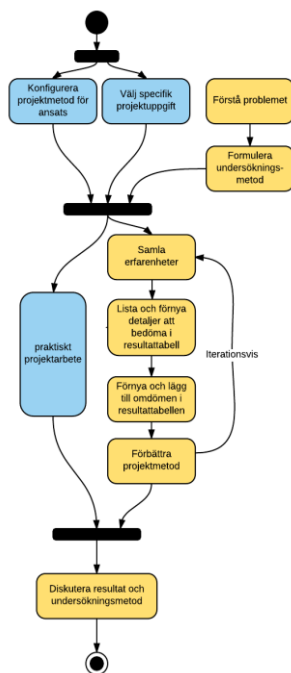
lösningar på bästa sätt?

Metoden för att samla data i denna fråga blir induktiv då erfarenheten i gruppen är ytterst liten. Arbetssättet blir att efterhand som projektet fortskrider så förs bedömningsområden som anses vitala in i tabeller, se resultatdelen.

För att kunna få ett uppfattning om huruvida en delprojektmetod är bra krävs att man samlar information om redan etablerade metoder. Detta görs lämpligen genom att gå en kurs i projektmetodik, alternativt att läsa dokumentation om respektive metod. När man har en grundläggande uppfattning om de olika metoderna som finns tillgängliga gäller det att kunna avgöra vilken som passar bäst för det projekt man försöker genomföra. Detta kan vara svårt, (därför frågeställningen till denna rapport), vilket gör att man får testa flera olika metoder var för sig, för att sedan utvärdera dessa.

Detta går till som så att man väljer en metod att använda sig av under en tid, till exempel under en iteration, och går igenom vad som var bra och vad som var dåligt med metoden. Om projektgruppen kände att metoden har fungerat bra har man behållit den till senare iterationer. Men om den har fungerat dåligt får man diskutera vad som var dåligt, och huruvida man kan förändra någon aspekt av den som skulle leda till en förbättring. Om det inte går att göra en mindre förändring för att metoden ska bli bra får man se sig om efter någon annan metod att byta ut den mot och utvärdera den efter nästkommande iteration. Genom att följa det här mönstret får man testa flera olika metoder, och kommer sannolikt fram till en metod som fungerar bra för just den specifika projektgruppen. Här samlas även viktiga erfarenheter inom olika metoder, som kommer vara nyttiga senare vid arbete i andra projekt.

Metod 1: Undersökningsmetod, se figur. De gula fälten är aktiviteter som kopplar till själva undersökningen. Metoden följer principer för vetenskaplighet enligt Andersson och Ekholm (Andersson & Ekholm, 2002 pp 17).



Figur 1: Undersökningsmetod för "Vad är bra projektmotod för små IT-projekt".

V. GENOMFÖRANDE

I följande kapitel redovisas viktiga beslut, förändringar och anpassningar som gjorts i projektmotod, projektpraktiker, värderingar, beslut mm som gjorts under studiens genomförande.

(Här kan det vara lämpligt med en indelning i olika ansvarsområden inom projektet?)

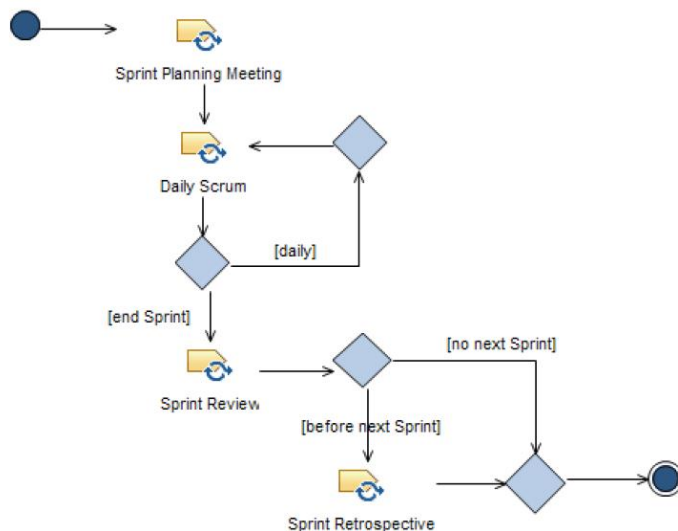
I. Projektleddning (Samuel)

I början av projektet under iteration 1 så hade det varit svårt ifall projekt-ledandet gjordes utav en person. Eftersom rollerna för alla projektdeltagande var något nytt för varje person så kunde man inte gå in i sin roll till hundra procent direkt. Istället fick alla vara delaktiga inom projektledandet.

Desto längre projektet led, ju mer satte sig rollerna för alla. Under iteration 2 så kunde man applicera sin roll mycket bättre på projektet. Som ledare var det lättare att se vad som skulle göras, och med hjälp av productbackloggen, avgöra när saker skulle göras. Arbetstavlan var ett bra redskap för att hålla reda på alla olika moment och områden. Det underlättade som ledare att ha en bra överblick över de olika beslut projektgruppen tagit.

Under resterande iterationer när arbetet flöt på var det viktigaste att man såg till att alla följde tidsplanen och att vi hela tiden kom närmare målet för iterationerna.

Genomförandet av iterationer följer helt och hållet mallen för s k "sprint" i Scrum [ref] och beskrivs enklast med följande aktivitetsdiagram, se bild.



Figur 4: Iteration/Sprint(Elvesæter et al., 2013)

J. Systemutveckling (Douglas)

Utvecklingen av produkten har skett i iterationer där vi för varje iteration implementerat nya funktioner på produkten. I slutet av varje iteration har en liten release av produkten gjorts i form av en kort demonstration.

När vi arbetat med koden för produkten har vi använt oss av GitHub[6] som är en plats där folk kan lägga upp och dela kod med varandra. Vi har lagt upp källkoden som sedan används av raspberryn i vårt gemensamma projekt på GitHub som alla medlemmar har tillgång till.

Innan vi började skriva koden togs en arkitektur fram, som byggde på kraven vi fått från vår uppdragsgivare. Den första versionen av arkitekturen motsvarade dom grundkrav som gavs och inget mer. Utifrån den första arkitekturen skrevs uppgifter upp, så kallade stories, som delades ut till medlemmar i projektet. När alla grundkraven var uppfyllda kom vi i gruppen överens om vilka nya funktionaliteter vi ville implementera och gjorde sedan en ny utökad arkitektur som låg till grunden för vilka nya uppgifter som skulle göras.

En liten justering vi gjorde i den fjärde iterationen var att vi slutade ge våra stories en tidsuppskattning. Detta beror på att dom uppskattningar vi gjort i flera fall var väldigt felaktiga. Detta kan lätt hända om man inte har någon tidigare erfarenhet av att lösa en teknisk uppgift, ibland kan man köra fast och uppgiften tar längre tid än man förväntade sig.

När mjukvaran har tagits fram har flera element från extreme programming [4][5] använts, nedan är en lista av metoder som använts från extreme programming.

Små releaser – Efter varje iteration har en ny prototyp släpps med ny funktionalitet.

Inkrementell planering – Inför varje iteration väljs uppgifter ut att göras, så kallade stories. Dessa stories bryts ner i mindre uppgifter som fördelas mellan deltagarna i projektet, i slutet

av iterationen ska dessa stories demonstreras i form av en ny release av produkten.

Enkel design – Tillräckligt med design i form av sekvensdiagram och arkitektur av produkten har tagits fram, men inte mer än nödvändigt.

Återanvändning – Identisk kod som används på fler ställen skrivs en gång och återvinns sedan.

Parprogrammering - Ibland har vi suttit i par och programmerat då den ena skriver och den andra ser på och ger feedback på vad den andra skriver.

Arbeta kollektivt – Alla medlemmar i projektet har varit involverade och haft insyn i alla delar av koden.

Kontinuerlig infasning – Så fort en del har blivit färdigskriven har den lagts in i projektet och testats enligt skrivna test.

K. Kundrepresentant (Lucas)

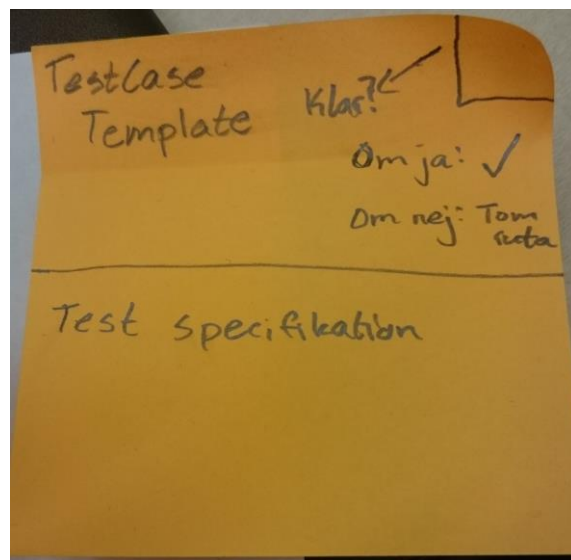
Först efter första iterationen när vi hade kraven från kunden på plats kunde alla usecases skapas. Eftersom arkitekturen har skapats utifrån dessa var således en av de viktigaste bitarna på plats. Våra usecases ändrades inte under projektets gång till och med i början av den fjärde iterationen, då det ansågs att även säkerhet bör ses som ett grundkrav, så ett usecase för detta lades till.

L. Arkitekt (Lucas)

När kraven var på plats diskuterade gruppen tillsammans olika idéer på tillvägagångssätt att uppfylla dessa. Efter att vi hade enats om en lösning tillverkades en första arkitektur som innehöll de delar vi behövde för att uppfylla målet för en första prototyp. I början av varje iteration efter detta uppdaterades arkitekturen för att kunna täcka alla iterationsmål, samt att vissa komponenter utökades för att tydligare beskriva hur allt hänger samman.

M. Testare (Diacio)

I rollen som testare är det viktigt att anpassa sig till arkitekturs strukturer. I början av projektet, när arkitekturen inte var utvecklad och uppsatt på tavlan, blev således rollen som testare, rätt så passiv. Under senare iterationer, när utkast av den tilltänkta arkitekturen var på plats, blev arbetet kring testning mer central. Beslutet att jobba enligt TDD (Test Driven Development) medförde att gruppen blev mer beroende av att samlas och göra beslut kring hur testerna skall genomföras. Genom diskussion beslutades det att följa mallen angiven i Figur 5.M.1.



Figur 5.M.1: Visar den använda mallen för ett TestCase.

Beslutet togs av hela gruppen tillsammans. Eftersom gruppen tidigare hade jobbat enligt modellen Utveckla/Testa/Slutföra, bidrog den nya metoden till ett byte av arbetssätt.

VI. RESULTAT

I tabellerna nedan tas resultaten av de undersökta metoderna upp. Vi listar resultaten i tre kategorier. I den första kategorin keep listas saker vi anser vara nyttiga och användbara i projektet, i problems-try listas saker vi tycker har haft problem men även varit användbara. I kategorin problems-skip listas saker vi tycker inte har varit bidragande till projektet.

| Keep | | |
|-----------------------------------|---|--|
| Keep | Motivering | Förbättring |
| Product backlog | Ger en bra bild över vad som behöver uppfyllas. | |
| Stories | Ett bra sätt att bryta ner en stor uppgift i mindre och mer överskådliga uppgifter | Ta bort tidsestimering en då det kan vara mycket svårt att göra en korrekt sådan i vissa fall. |
| Kontinuerligt skriven arkitektur. | Med en överskådlig arkitektur går kod-skrivandet snabbare. Bidrar till förståelse inom gruppen. | |
| Kontinuerlig infasning | Underlättar för | |

| | | |
|--------------------------|--|--|
| av ny kod. | testning. | |
| Små releaser/Sprint demo | Bra sätt att visa uppdragsgivare att man ligger i fas i projektet. | |

Figur 6.1: Visar “keep” förslag.

| Problems - try | | |
|------------------|---------------------------|--|
| Problem | Testa vad? | Motivering |
| Parprogrammering | Använd när man kört fast. | Kan vara tidskrävande om två sitter med samma kod om det gäller enklare uppgifter. |
| Sen ankomst | Daily Scrum | Se till att alla kommer i utsagd tid, samt skapa en kommunikation i gruppen. |

Figur 6.2: Visar “try” förslag.

| ”Problems – Skip” | | |
|---------------------------------------|---|------------|
| Problem | Ta bort eller ersätt | Motivering |
| Burndown | Uppskattningen av hur lång tid en storie tar kan vara svår att göra | |
| För många stories bearbetas samtidigt | Ersattes med en gräns på 2 stories per person. | |

Figur 6.3: Visar förslag som tagits bort eller ersatts.

VII. ANALYS / FÖRBÄTTRINGSFÖRSLAG

Med tanke på projektets relativt korta tid, fick många metoder användas i väldigt korta perioder. Därför gjordes ändringar nästan så fort problem upptäcktes. Vilka ändringar som skulle göras och när, bestämdes vid slutet av varje iteration. Genom så kallade retrospectives. Dessa dokument har sparats och de viktiga besluten har inkluderats i figurerna 6.1, 6.2 och 6.3.

Exempelvis när det gäller problemet med sen ankomst i figur 6.2, så visade det sig att medlemmar i gruppen ofta kom sent till samlingen. För att lösa detta, infördes en sorts mer formell samling vid starten. Ett så kallat “Daily Scrum” möte. Det medförde att gruppmedlemmarna visste att något viktigt skedde vid början av varje arbetspass. Det resulterade i bättre tidspassning.

Ändringarna som gjorts genom så kallade retrospectives har alltså lett till att gruppen kunnat bilda en bättre uppfattning av

vad som är en bra projektmetod. Besluten har alltså varit direkt kopplade till faktumet att gruppen velat bilda sig en egen uppfattning om vad en bra projektmetod är.

Ett förbättringsförslag är att ge grupperna en bättre kunskap om projektmetoder innan kursen startar. Det skulle exempelvis kunna vara genom andra kurser. Att på 5 iterationer skapa sig en bild, applicera och anpassa metoder, gör att slutsatserna kan vara baserade på vad som råkade stämma just i någon iteration. Det innebär inte nödvändigtvis att metoden är bra under en lång period. Detta är dock i detta sammanhanget svårt att ordna.

VIII. DISKUSSION

Det vi kan säga om resultatet är att vi kan endast avgöra ifall de metoder vi använt oss av är bra eller inte. De metoder som vi inte använde oss av men diskuterade kring, kan vi endast spekulera om de hade varit bra eller inte. Vi kan alltså inte till hundra procent avgöra ifall en annan projektmetod hade varit bättre utifrån resultatet.

Eftersom vår tidsperiod var så pass kort (10 veckor ungefär) så hanns det inte med mycket experimenterande mellan olika projektmetoder. Det vi fick göra istället var att testa olika delprojektmetoder och jämföra dessa. Anledningen till detta är att testtiden för en projektmetod ifall vi testat fler än en hade givit för lite underlag för att bedöma/avgöra om en projektmetod för små IT-projekt är bra.

En delprojektmetod från SCRUM som vi ansåg var onödig för oss under varje sprint, var burndownen. I och med att projektgruppen var så pass liten (4 personer) så hade vi bra koll på alla stories, och hur många vi hann med på en sprint. Burndownen kändes alltså överflödigt eftersom vi hela tiden kände hur vi låg till. En positiv sak med burndownen var att vid reflektionen efter en sprint så kunde man se tillbaka på hur sprinten hade gått på ett övergripande sätt. Men den hjälpte som sagt inte så mycket under själva sprinten.

N. Metoddiskussion

Ett första steg för att få en uppfattning om vad som är en bra projektmetod är lämpligtvis att utföra ett projekt, använda sig av olika metoder och sedan analysera vilken effekt metoderna har fått på projektet vilket vi har gjort.

För att få en bättre uppfattning om vad som är en bra projektmetod hade det antagligen varit bra att testa fler metoder, det fanns dock inte tid för det i vårt projekt.

O. Resultatdiskussion

Resultaten bör tolkas som trovärdiga eftersom de berör våra egna upplevelser under projektet. Huruvida man kan dra slutsatser kring vad som är en bra projektmetod utifrån resultaten, är dock inte lika säkert. Varje projekt stöter på olika problem, och därför är det inte säkert att just dessa resultat säger vad som generellt är en bra projektmetod. Däremot är dessa knep och ändringar som resultatet beskriver, ett sätt för just denna projektgrupp att försöka bilda sig en uppfattning om vad en bra projektmetod är.

REFERENSER

- [1]Kniberg, Henrik. *Scrum and XP from the trenches* 2.uppl. C4Media
- [2]Sommerville, Ian. *Software engineering*. 9.uppl. Pearson
- [3]Eklund, Sven. *Arbeta i projekt* 4.uppl Studentlitteratur
- [4]Sommerville, Ian. Agile software development *Software engineering* 9.uppl. Pearson
- [5]Kniberg, Henrik. How we combine XP with SCRUM *Scrum and XP from the trenches* 2.uppl. C4Media
- [6]GitHub, Inc. [Online] Available:www.github.com