

Customer Value and Scope

Situation A (“här är vi”):

1. I nuläget har vi kommit till en punkt där vi anser att vi har en MVP sett till hela applikationen och det slutresultat som den ska producera. (slutprodukt: docs med hela order + kupong för varje individuellt anställd + ordersystem)
2. Gällande framgångskriterier för projektets mjukvaruresultat är gruppen överens om att en MVP nu har skapats ungefär i proportion till vad den initiala planen var i början av projektet. Det som fallit bort är en ytterligare integration med externa system utanför vår mjukvara och anpassade format för att få kompatibla format som underlättar manuella hanteringen. Sett till lärdomar så har kursen resulterat i en djupare förståelse för agil-arbetsmetodik och för gruppen väl fungerande implementation för att utföra delstegen som skapande av user stories och estimeringen av dem etc.
3. Gruppen har landat i en återkommande rutin för hur vi tar fram våra user stories, förstår väl och kan skapa värde utifrån.
4. Vi har varit dåliga på att använda acceptance tests i form av exempelvis JUnit tests för varje metod eller liknande. Istället har vi använt oss av att inför varje merge granskat varandras kod och på så sätt sett den från ett annat perspektiv för att se eventuella uppenbara missar. Då vi har en relativt simpel applikation som inte använder sig av någon form av multithreading eller liknande har detta fungerat relativt bra för oss.
5. I dagsläget mäter vi stress, samarbete och värdeskapande varje vecka med mätsiffror och motiveringar till svaret + förbättringsförslag.
Ev. KPI:er lite onödigt trubbigt verktyg för en så liten grupp. Kan diskutera istället för att mäta kvantitativt.

Situation B (“hit vill vi komma”):

1. Givet en vecka till hade vi velat gå vidare med lite “polering” av applikationen, exempelvis gjort den mer visuellt tilltalande och implementerat mer säkerhet på inmatningsfälten så de inte fylls i fel. Även utskriften programmet producerar hade fått sig en uppdatering.
2. Sett till produkten och en ytterligare vecka så hade förväntningarna på produkten sett relativt lika ut. kunskapsmässigt, skulle gruppen nog gynnas av att få extern input för att större värde i teoretiska förståelse skulle uppfattas. Vi har nått den punkt där vi i gruppen hittat ett fungerande arbetssätt som följer ett minimalt teoretiskt ramverk som vägleder projektet. Ytterligare behov för nya definitioner, klassificeringar och arbetssätt skulle sporra gruppens intresse av djupare förståelse och högre läromål.
3. Utan någon större radikal förändring av projektets ramar eller kraven på oss skulle nog den standard vi arbetat fram över veckorna inte förändra sig nämnvärt.
4. En logisk utveckling av denna punkt hade varit att bättre använda sig av något i stil med JUnit tester för att lättare kunna debugga och testa alla funktioner som implementeras i programmet.
5. Det skulle kunna vara önskvärt att få mer data på den kod vi producerar. Sett till hur mycket arbete som mätningarna av KPI:er redan kostar hade en denna sortens förändring behövt andra förutsättningar dock. Skapandet av gemensamma

referenser av hur skalan av KPI:er fungerar

Funktionen A -> B (“så här tar vi oss från A till B”):

1. Exempel på hur ett mer visuellt tilltalande format hade kunnat skapas är med olika typstorlek, fetstiltä nyckelord och en tydligare tabell i dokumentet som skrivs ut för ordern. Applikationen i sin helhet skulle kunna vara mer personifierad och anpassad till den grafiska profilen av PO:ns företag. Säkerhetsmässigt så kunde inmatningen även detektera undantagsfall till de mer generella reglerna som vi har kodat efter. Tex. skottårsdagar och att varje månad har “på dagen” rätt begränsningar.
 2. Som nämnts ovan hade gruppen behövt lite ytterligare teoretisk input för att veta vad näst man bör fokusera på och bli bättre på. I nuläget har vi funnit ett fungerande mönster för hur en sprint planeras och avslutas etc.
 3. Utifrån rätt förändringar som skulle skapa behovet av att vi anpassar vårt arbetssätt så skulle nog mer resurser gå till att skriva koden framtidskompatibel och med mer hänsyn till säkerhet, underhåll och noggrannhet.
 4. Användningen av mer avancerade tester och mer grundlig granskning av kod skulle tillämpas för att komma till den i B önskade situationen.
 5. Införandet av hårda KPI:er, dvs. testa koden och mäta antalet buggar eller dylikt. Flera än 3 KPI:er. En ny skala hade kunnat testas för att undersöka hur användbarheten och tolkningen hade påverkats. Vi hade även kunnat utöka eller byta ut områdena vi mäter inom.
För att göra verktygen mindre trubbiga, har vi utökat dem med förslag.
-

Social Contract and Effort

Situation A (“här är vi”):

1. Vi har inte ändrat något i kontraktet denna veckan.
2. Vi har lagt ungefär lika mycket tid som förra veckan. Men det är ju mindre än de 10 timmar som man förväntas lägga på kursen kodmässigt. I mötestid har vi ungefär lagt 10 timmar. När det gäller tid har vi dock varit överens om att det är bättre med färre effektiva timmar än att försöka klämma in något som känns orimligt. I de flesta andra kurser lägger man inte den tiden som man “förväntas” lägga. Inte för att kursen är för enkel, utan för att det finns många andra delar i den intellektuella delen av att lära sig saker. Vila, repetition och träning är alla saker som kanske inte anses ta en framåt i en projekt, men som alla är dolda delar i en framgångsrikt koncept.

Situation B (“hit vill vi komma”):

1. Det finns inget vi just nu känner att vi behöver utveckla.
2. Vi är nöjda där vi är, vi har hittat en lösning som funkar för oss och ger oss mycket värde.

Funktionen A -> B (“så här tar vi oss från A till B”):

1. ...
2. Inga åtgärder vidtas.

- your [social contract \(Links to an external site.\)](#), i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives)
There is a [survey \(Links to an external site.\)](#) you can use for evaluating how the team is perceiving the process and if it is used by several teams it will also help you to assess if your team is following a general pattern or not.
- the time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)

Design decisions and product structure

Situation A(“här är vi”):

1. Denna vecka har vi implementerat “avbryt”-knappar, så nu är applikationen inte längre beroende av operativsystemets egna bakåt-knappar. Vi har även gjort fält och knappar större för att öka läsbarhet och gjorde det enklare att klicka rätt med fingret.
2. Vi har inte anpassat vår applikation för mobiltelefoner, utan den är enbart designad så den funkar på en Pixel C-platta.
3. Vi hade som mål att designen på plattan skulle fungera både lodrät och horisontellt
4. Sen några veckor tillbaka kan applikationen skriva ut en word-dokument med orderinformation.
5. I nuläget har designen inte varit prioriterad därav har det gjorts mycket uppoffringar i den fronten så att programmet fungerar men kanske inte ser väldigt snyggt ut eller är helt självklart hur det används. I nuläget används mycket av de basala funktionerna som Android själv ger oss, så vi har t.ex. inget färgschema för hur vi vill att applikationen ska se ut.
6. När det gäller design så har det inte varit ett särskilt högt prioriterat krav från vår PO. Även om vi ser ett värde i att göra appen fungerande för flera plattformar och göra den snyggare, så hade det i ett verkligt projekt kanske varit något som kostat mycket pengar och som vår PO inte hade velat satsa på. Vi har därför fokuserat mer på funktion än design.

Situation B (“hit vill vi komma”):

1. Läsbarheten är bra, men vi hade definitivt kunnat jobba mer med färgtema och mönster för att göra appen ännu mer läsbar och lätt att navigera i.
2. Vid mer tid skulle det vara bra om programmet gick och köra på ospecifik plattform. Så att vi vet att applikationen fungerar för kunden oavsett vad de använder.
3. Det var svårt att få till detta i Android Studio, och vi har ännu inte knäckt hur man ska använda constraints för att designen ska fungera både lodrät och horisontellt.
4. Word-dokumentet hade kunnat vara mer läsbart och väldesignat än var det är i nuläget. Ett annat förbättringsområde hade varit om word-dokumentets fil hade varit tillgänglig i själva applikationen och inte bara i plattans filbibliotek. Man hade definitivt kunnat utveckla en funktion där användaren lättare hittade sin word-fil och kunde skicka vidare den - antingen direkt till sin mejl, eller till en mapp på Google Drive.

5. Det skulle vara väldigt bra och underlätta för användare av programmet om det fanns tydliga design mönster inorporerad i vår applikation. Så att det blir enkelt att använda samt så att användaren inte hakar upp sig på inkonsekvent utseende och beteende för programmet.
6. Vi hade kunnat fråga vår PO tydligare om vilka krav denne hade för designen. Just nu har vi bara gått på att han gillade mock-upen vi gjorde efter att ha hört hans pitch.

Funktionen A -> B (“så här tar vi oss från A till B”):

1. Det enklaste hade varit om vi redan med vår mock-up hade gjort vissa designval med exempelvis färgval och läsbarhet. För att gå vidare bör man nog göra någon slags designplan och gå igenom hela applikationen och revidera den efter en satt standard.
2. Det hade helt enkelt behövts läggas tid på att få det att fungera, man hade behövt någon form av User Story som behandlade applikationen för flera plattformar och göra utseendet för appen relativt plattformen.
3. Nästa vecka skulle vi kunna kolla upp om det finns någon tutorial online som visar hur man kan uppnå detta.
4. För att gå vidare så bör vi kolla in vilka befintliga lösningar som finns i andra applikationer med liknande process som vår. Eftersom vi inte just nu vet hur vi skulle lösa problemet så hade vi kanske kunnat ha ett möte där vi bara spånade idéer.
5. Designen hade behövt en genomgång där man tog hänsyn till flera av de etablerade designmönster som finns och sedan försöker se till att applikationen följer dem. Detta kan ta otroligt lång tid beroende på hur genomgående man är. Men endast vid lite mer tid hade det kunnat göra stor nytta. Så man hade t.ex. kunnat rätta på knapparna så de hamnar mer i linje. Man hade också kunnat röra sig från det basala android utseendet som erbjuds och få ett mer unikt utseende för applikationen.
6. Vi bör boka in ett designmöte med PO:n.

- how your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value
- which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)
- how you use and update your documentation throughout the sprints
- how you ensure code quality and enforce coding standards

Application of Scrum

Situation A(“här är vi”):

1. Josefine har varit gruppens ständiga Scrummaster under projektet projektet. Hon har även blivit tilldelad rollen som ständig sekreterare. Vilhelm tilldelades rollen som intern PO. Ingen av rollerna har dock påverkat arbetet nämnvärt. Vi har tagit gemensamt ansvar för skrivande och koordinering av arbetet. Rollen som intern PO har mestadels utmärkt sig som kontaktperson gentemot extern PO, vilket fungerat bra med tanke på dessa personers goda relationer.

Informellt ser vi att rollerna som attribueras scrummaster och sekreterare har delats ut över resten av gruppen / vi tar gemensamt ansvar.

2. Det känns ändå att estimeringen i kombination med sprint-upplägget gör att man alltid vet vad man behöver jobba på, och när det behöver bli färdigt. Det gör sannolikt också att man inte snöar iväg och lägger lång tid på att utveckla avancerad funktionalitet som ligger utanför MVP-formen.
Dock gör de tydliga funktionella kraven och tidspressen ibland att vi upplever att det sätts en onödigt stor betoning på funktionalitet (dvs att "det funkar" prioriteras betydligt högre än "det är en bra lösning"). Det har lett till mycket refraktering och omskrivning av kod sista veckorna.
3. Sprint reviewsen och PO-mötena har nog inte påverkat vårt arbetssätt jättemycket. Vi har å ena sidan använt mötena för att prioritera vilka nya features och user stories vi ska arbeta med nästkommande sprint. Men å andra sidan har vi haft en konceptuellt rätt enkel produkt, där vi tidigt haft en klar bild av vad som behövs för att den ska kunna användas i praktiken (och det kanske har varit ett mer diskret steg än för vissa andra applikationer). I de tidigare stadierna hade vi emellertid tät kontakt med PO, och fick många dokument och instruktioner för hur olika typer av dokument (som nu skulle få en digital motsv.) skulle se ut.
4. Vi har fortsatt att arbeta på med de verktyg vi satte upp i början av kursen, som GitHub, Android Studio, osv. Bästa sättet att lära sig om hur de fungerar är att använda Google när saker krånglar. Problemet är att lösningarna som hittas är ofta väldigt specifika och svåra att översätta till ens egna problem. I veckan hade vi ett stort problem med Git som vid en merge tog bort massa kod och en del filer, däribland individuella reflektioner. Där löstes problemet tillslut genom googling, men det tog mycket tid att hitta någonting som hjälpte mot just vårt problem.
- En nackdel med scrum som vi har uppmärksammat är när man ska jobba vertikalt. Om samma team är ansvariga för både design och programmering så blir oftast den ena delen bristande, särskilt i små team som vårt. Den som är bra på att programmera är inte nödvändigtvis bra på design och vice versa.
- En annan nackdel för oss, i vårt projekt är att sprintsen har varit väldigt korta, samtidigt som vi alla har haft en annan kurs som tar vår tid. I de bästa av världar hade man bara fokuserat på denna kurs och har längre tid på sig i varje sprint. Vi löste detta genom att estimerade många stories, men sedan välja under veckan hur många av de vi skulle göra. Detta går inte helt hand i hand med Scrum-teorin, men blev en lösning som funkade för oss. Bättre att välja mellan flera små stories än att få en alltför stor som känns övermäktig, som man inte lyckas med.

Situation B ("hit vill vi komma"):

1. Vi hade gärna sett att rollerna hade större påverkan på arbetet, och att alla får lärochansen att testa på de olika rollerna.
2. Behålla fördelarna, men inte nackdelarna. Stories som vägleder arbetet, men inte kan vilseleda det, genom att inte bara specificera funktionalitet men även lite implementationskrav typ. Kravet på att kunna leverera värde ska inte överspegla bra utvecklingssed.

3. Av utbildningstekniska skäl så kanske lite tuffare kritik från PO-möten, där de utöver att erbjuda hjälp med information i utvecklingen och få en indikation om vad som är mest värdefullt, även få testa på lite mer "förhandling" och dyl.
4. Att vi har en bredare kunskap om de verktyg vi använder, så att googling inte blir det enda alternativet. Sedermera att också kunna översätta lösningar mellan olika problem som kan uppstå, vilket också kan uppnås genom en bättre kunskapsbas.
5. nästa vecka bör vi förbereda en process för att rekrytera nya teammedlemmar med bättre spetskunskap inom programmering och design.

Funktionen A -> B ("så här tar vi oss från A till B"):

1. Även om rollerna inte påverkat så mycket, och således fallit lite i glömska, så tycker vi det hade varit bra att börja rotera framförallt scrum master rollen till nästkommande projekt. Då kan man undersöka om rollen får större påverkan på arbetet, på ett positivt eller negativt sätt, beroende på vem som förvaltar ansvaret.
2. Genom att utveckla definition of done, och införa mer stringenta krav på hur lösningar ska designas och implementeras, hoppas vi på att skapa en motvikt mot incitamentet att snabbt och enkelt kunna leverera värde. Mer rigid lösning i slutet. Hållbar!
3. Tydligare kommunikation gentemot PO gällande vad PO-mötena ska ge.
4. Att mer läsa på om dokumentation och liknande och inte endast leta lösningar på specifika problem. Det är skönt i stunden att snabbare kunna gå vidare, men det är ännu skönare att nästa gång direkt veta lösningen på problemet.
5. Känns som vi behöver få in lite mer kapital innan vi kan utöka vårt team.

- the roles you have used within the team and their impact on your work
- the agile practices you have used and their impact on your work
- the sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?)
- best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)
- relation to literature and guest lectures (how do your reflections relate to what others have to say?)