

Slutrapport Charmander

DAT256 - 190607

Gustav Häger

Caroline Jönsson

Jonathan Köre

Liwia Larsson

Malin Ramström

Aron Sjöberg

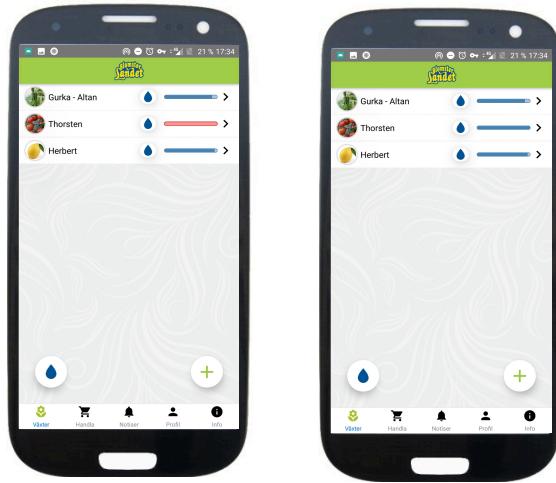
Benjamin Vinnerholt

Andrea Wester

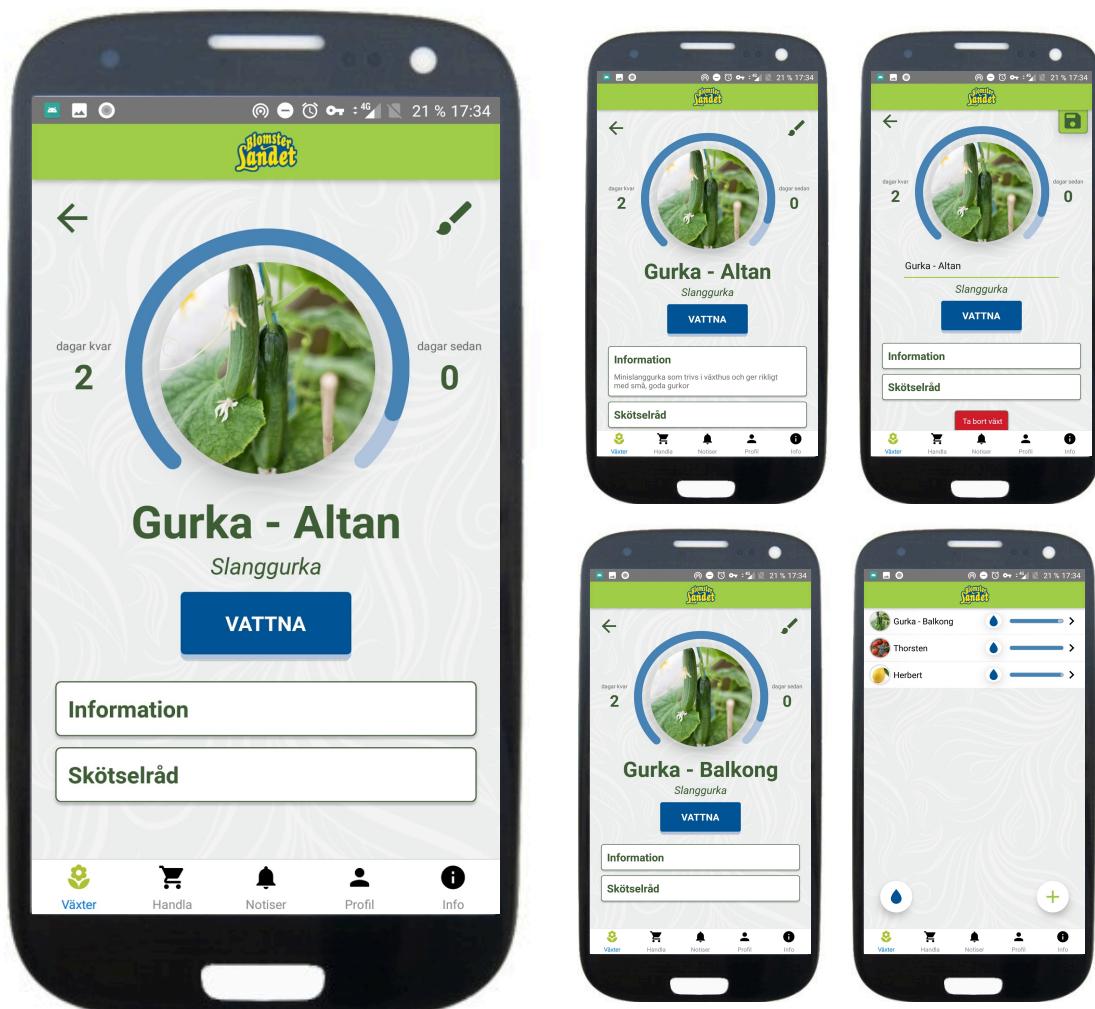
Resultat från återkopplingsmöte med blomsterlandet 05/06/2019

I slutet av projektet hade vi ett möte med Blomsterlandet för att visa upp vad vi lyckats producera. Efter att demon var klar så var VD:n Jan Larsson, och E-Handelschef Tirdad Zadeh nöjda över resultatet. Vidare följde en lång diskussion om hur den kommande utvecklingen skulle ske, och vad för potential applikationen har. Det finns många tankar kring utveckling av applikationen och dessa ska följas upp i slutet av sommaren. Blomsterlandet sa även att de eventuellt var intresserade av att anställa en av oss, för att hjälpa till med vidareutvecklingen.

Summering av resultatet



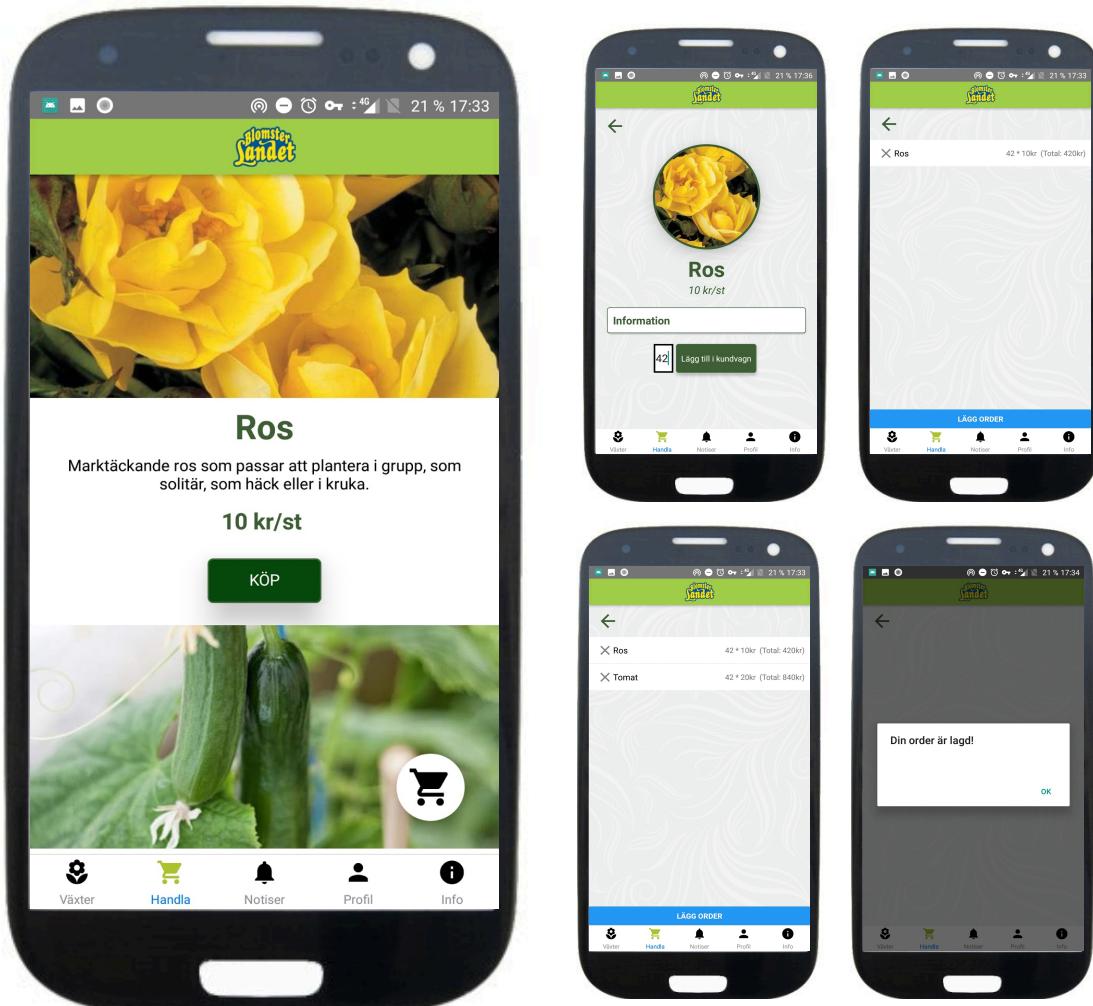
Växter: En översikt av sidan "Mina växter", både före och efter tomaten Thorsten har vattnats.



Växter: En detaljvy av Slanggurkan "Gurka - Altan" som döps om till "Gurka - Balkong".



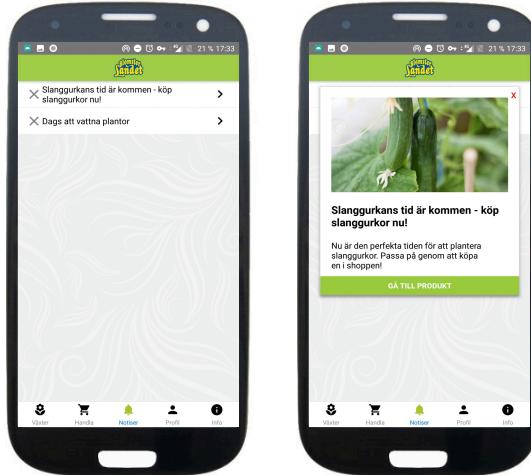
Info: Info-skärmen där man kan se en trädgårdskalender med relevant information för månaden.



Handla: I affären kan man köpa växter; i detta flödet lägger användaren till 42 rosor och 42 tomater i sin kundkorg, för att sedan lägga beställningen.

Product(s)		Buyer	Postal Code
9 st. Tomat	200 st. Slanggurka	Thorsten	22333
1 st. Ros		Charles	41261
1 st. Slanggurka	1 st. Tomat	Kalle Kulig	41326
5 st. Ros	5 st. Tomat	Kalle Kulig	41326
3 st. Slanggurka	1 st. Ros	Kalle Kulig	41326
30 st. Slanggurka		Kalle Kulig	41326
100 st. Ros	3 st. Tomat	Hugo	41326
12 st. Ros		Hugo	41326
1 st. Ros	1 st. Slanggurka	Hugo	41326
19 st. Ros		Kalle Kulig	41326
42 st. Ros	42 st. Tomat	Kalle Kulig	41326
1 st. Slanggurka	1 st. Ros	Hugo	41326
999 st. Slanggurka		Kalle Kulig	41326
1 st. Slanggurka	1 st. Tomat	Hugo	41326
1 st. Ros		Hanna	46142
3 st. Tomat	1 st. Ros	Mohammed	46142

Webbplatsen: här visas beställningar som lagts, ordnade efter postnummer.



Notiser: Användarens notifikationer samlas i en lista på skärmen "Notiser", där användaren kan klicka på notiser för att expandera dessa, och de kan i sin tur leda användaren till en relevant skärm.



Profil: på skärmen "profil" kan användaren se sin profil och dess personuppgifter.

Customer Value and Scope

The chosen scope of the application under development including the priority of features and for whom you are creating value:

A:

I uppstarten av projektet diskuterade gruppen idéer för applikationer som är inom kategorin "hållbar mobility". Vi kom i kontakt med Blomsterlandet och under ett inspirationsmöte uttryckte de ett behov av en applikation som kan locka nästa generations konsumenter. Blomsterlandet hade inga krav på funktionalitet utan önskade enbart att vi skulle utveckla någonting som hade lockat oss unga att handla hos Blomsterlandet. Olika idéer för applikationen diskuterades med handledarna, vilket tillsammans med Blomsterlandets input blev grunden för vår appidé.

Den idé som uppfyllde kravet om "hållbar mobility" var vid denna tidpunkt en funktion som grundades på villaägares gräsmattor. Tanken var att hållbara transporter kunde uppnås i högre grad om villaägarna fick tips om när de borde handla gödslet till sina gräsmattor. Detta underlättade för både Blomsterlandet och dess kunder, Blomsterlandet kunde planera och konsolidera transporterna samt minimera lagerhållningen och kunderna fick tips och erbjudanden baserat på deras gräsmatta vilket möjliggör minimering av utsläpp i form av färre transporter. Detta utgjorde grunden för vår applikationsidé.

Vidare presenterades denna idé på ett möte med Blomsterlandet. De såg en begränsning i applikationens scope. De menade att den yngre generationen i stor utsträckning bor i lägenheter. Därför ansåg de att vi borde utveckla någonting som berör det önskade kundsegmentet, den kommande generationen.

En ny central funktion togs fram, som även riktar sig mot lägenhetsägare. Appen skulle ha en funktion, "Mina växter", som primärt skulle locka den yngre generationen till att engagera sig i växter. Detta är en funktion där användaren kan samla sina växter på ett ställe där den kan se när växterna vattnats senast och när det är dags att vattna dem igen. Det skulle också finnas information och skötselråd till växterna för att underlätta skötseln. Detta är något som vi som en del av det yngre kundsegment kände att vi hade haft stor användning av eftersom vi inte har så stor erfarenhet av att sköta växter och är behov av att få hjälp med skötseln.

Appens syfte med att möjliggöra konsoliderade transporter för Blomsterlandet bevarades, vilket gör att de kan minska antalet transporter samtidigt som de blir mer effektiva. Detta skulle utföras genom riktade pushnotiser till appanvändare om säsongsprodukter eller liknande för att kunna få en mer styrd och förutsägbar försäljning. En förutsägbar försäljning gör det enklare att använda resurser effektivt kring vilket gör det möjligt mer hållbart, både ekonomiskt och ekologiskt.

Fortsättningsvis diskuterade vi prioritering samt vidareutveckling av funktionerna. Gruppen beslutade att kursens krav var det viktigaste, vilket betydde att primärt fokus hamnade på transportlösningen. Kärfunktionen blev därmed notiserna, som var nödvändiga för att skicka ut anpassade erbjudanden till kunderna, vilket möjliggör konsolidering för Blomsterlandet. Vi insåg även att för att kunna möjliggöra detta så behövdes en beställning kunna läggas. Detta blev längre fram en fullskalig shop. "Mina växter" blev den näst viktigaste funktionen eftersom den skapade mycket värde för Blomsterlandet. I tabellen nedan presenteras hur de olika delarna ger värde för de olika aktörerna kring appen.

Tabell 1: Vilka krav de olika intressenterna har

Krav/Intressenter	Blomsterlandet (Kund)	Handledare (Kursen)	Utvecklingslag
Hållbarhet			
Transport			
Värde för nästa generation			

Grönt: Hög prioritering

Gult: Medel prioritering

Rött: Låg prioritering

Blomsterlandet: Hållbarhet + Värde för nästa generation → Mina växter + Notiser

Handledare: Hållbarhet + Transport → Notiser + Shop + Konsolidering

Utvecklingslag: Hållbarhet + Transport → Notiser + Shop + Konsolidering

Under arbetets gång upptäckte vi att för att underlätta för kunden vid beställningar hade även en profilsida varit nödvändig. Detta eftersom Blomsterlandets kunder då slipper fylla i sina kontaktuppgifter vid varje köp utan applikationen automatiskt hanterar det samtidigt som det blir enklare att konsolidera transporter utefter postnummer hos användare. Profilsidan började utvecklas i mitten av kursens gång men vi beslöt oss att inte lägga fokus på den eftersom att den funktionaliteten inte medför så mycket värde mer än att lagra denna data. Vi insåg att det är mer värt att ha de olika sidorna med grundläggande funktionalitet än att bara ha en av dem fast noggrann.

Slutligen valde vi att utveckla en webbsida som komplement till applikationen. Vi insåg att konsolideringsaspekten förbättrades avsevärt om vi kunde förse Blomsterlandet med en tjänst där ordrarna kunde samlas på ett effektivt sätt. Om ordrarna sorteras efter postnummer kan Blomsterlandet inte endast konsolidera transporter utan dessutom erbjuda andra i närområden fri leverans eller liknande kringtjänster. En del funktioner var tvungna att prioriteras bort. Vi använde oss av konceptet med "Värdekurvan" när vi, utvecklingslaget, tog beslut om vad som skulle utvecklas och började med det skapade mest värde snabbt.

Sammanfattningsvis har vi under projektets gång varit i kontakt med kunden samt diskuterat med handledare och produktägare för att kontinuerligt revidera scopet efter den feedback vi fått.

B:

I och med att Blomsterlandet gav oss väldigt fria händer så har vi inte haft så mycket kontakt med dem. Vi anser i efterhand att det hade varit bra att ha haft lite mer kontakt, då kunden kan ha mer åsikter än vad de från början trott. I framtida projekt hade vi tyckt det varit bra att ha gett vår kund ett större inflytande och därmed genomsyra vårt beslutfattande gällande prioritering av arbetsuppgifter. Det hade gjort att vi la ännu mindre tid på icke värdeskapande arbete.

Vi har insett att det är väldigt viktigt att definiera ett tydligt scope för att ha tydliga ramar och riktlinjer att följa i arbetet och det är något som vi vill fortsätta med i kommande projekt. Vi anser att vi hade kunnat vara tydligare än vad vi var i det här projektet. Det ska inte finnas någon tvekan om vad scopet är.

Att kontinuerligt ställa sig frågan vad som är mest värdegenererande är väldigt viktigt eftersom att det är lätt hänt att bara fortsätta koda utan att reflektera över om det är värdeskapande eller inte. Den kontinuerliga reflektion som vi haft i detta projektet om vad som är värdeskapande eller inte är något som vi vill ta med oss till kommande projekt.

Likt vi har gjort i detta projektet, vill vi kommande projekt utforma tydliga kravspecifikationer för de olika intressenterna. Att analysera vilka delar och funktioner som är värdeskapande för olika intressenter möjliggjorde för vår prioritering av funktionaliteten.

A→B:

För att ha mer kontakt med kund så hade det varit bra att i förväg bestämma tillfällen att träffas och på så sätt kunna få mer ledning av kunden. För att ha tydligare kommunikation med kund så hade det varit bra om man redan från början specificerade tydligt vilken roll, befogenheter / ansvar de skulle ha i arbetet.

För att se till att scopet blir ännu tydligare skulle vi se till att ha mer kommunikation externt och internt i början av projektet och att avsätta tid och energi till att just specificera det i tidigt. Scope är ju alltid någonting som utvecklas under ett projekts gång vilket man inte heller vill motverka men det underlättar utförandet av ett projekt ifall scopet är så tydligt och representativt som möjligt för alla tidigt.

Vi måste ta med oss våra kunskaper om ”Värdekurvan” i vidare projekt och med hjälp av detta kontinuerligt utvärdera vad som är värdeskapande. För att fortsätta utforma tydliga kravspecifikationer krävs det att vi har en kontinuerlig kommunikation genom hela projektet med de olika intressenterna.

The success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort):

A:

Vad gruppen har räknat som framgång har varierat under kursens gång. Förutsättningarna och projektets skede har spelat stor roll i de mål vi har satt upp. Det har funnits både långsiktiga och kortssiktiga mål.

Vi hade ett par långsiktiga mål i samband med projektet. Samtliga mål sattes inte från start utan en del utvecklades i samband med att projektet fortskred. Inga av de långsiktiga målen försvann eller ansågs vara ”uppnådda” utan försökte kontinuerligt förbättras.

Det första var att kunna arbeta med Scrum på ett naturligt sätt. Vi följde riktlinjerna för Scrum och såg hur vi förbättrades efter varje sprint. Våra user stories förbättrades och gick från horisontella till vertikala med hjälp av handledarna. Samtliga gruppmedlemmar är insatta i metodiken och kan beskriva syftet med varje del. Att förstå varför man gör någonting är avgörande för uppgiftens framgång. Att utföra exempelvis en Daily Scrum utan syfte skapar inget värde.

Vidare satte vi som långsiktigt mål att samtliga gruppmedlemmar skulle må bra och uppleva en rimlig arbetsbelastning. Detta valde vi att spegla i våra KPI:er. Vi hoppades att genom att mäta och diskutera allas välmående skulle situationen förbättras. Det blev tydligt att en förståelse inom laget skapades för varje individ när denne fick chansen att beskriva betydande händelser som påverkade projektet och dess arbete.

Fortsättningsvis ville vi ha ett starkt lag som kunde arbeta effektivt. Vi definierade ett starkt lag som ett lag där alla vågar fråga om hjälp och känner sig bekväma att uttrycka sina åsikter. Till detta valde vi att träffas utanför projektet för att lära känna varandra bättre. Vi anser att teambuilding är avgörande för hur väl ett lag arbetar tillsammans då vi tror att en bra grupp dynamik är nyckeln till produktivitet. Detta är även en av anledningarna till varför vi valde att parprogrammera. I samband med bra lagarbete satte vi även som mål att kommunikationen skulle vara god genom hela processen. För att uppnå detta satte vi riktlinjer kring vart man skulle skriva beroende på vad det handlade om samt träffas minst två gånger i veckan fysiskt för att uppdatera resterande gruppmedlemmar. Detta har fungerat väl och har underlättat processen.

Ett av de viktigaste långsiktiga målen vi satte var att vi ville att både gruppen samt Blomsterlandet skulle känna sig nöjda med applikationen. Vi som utvecklingslag ville inte lämna ifrån oss något vi inte var stolta över vilket satte krav på slutprodukten. Det var även viktigt att Blomsterlandet var nöjda med det som

levererades och att slutprodukten levde upp till de krav som Blomsterlandet hade ställt. Det handlade alltså inte endast om att slutprodukten var ”bra” utifrån kursens och handledarnas kriterier, utan den skulle vara anpassad efter Blomsterlandets önskemål också. För att vi skulle känna oss nöjda var det därför viktigt att avsluta värdeaddrande flöden samt eliminera de mest akuta buggarna. Detta så att inget halvfärdigt lämnades vidare, med avseende på våra flöden skulle se ut. För att Blomsterlandet skulle vara nöjda var det viktigt att de värdeaddrande flödena speglade den tidigare uttalade visionen.

Vidare sattes även en del kortsiktiga mål kontinuerligt under projektet. Det kortsiktiga målet i början av projektet ansåg utvecklingslaget vara att alla i gruppen skulle lära sig React Native och Git, de programvaror vi skulle använda oss utav. Detta var kritiskt för projektets fortsatta utveckling. Målet innefattade också en grundläggande förståelse för Scrum-metodiken. Dessa mål baserades på en gemensam önskan om att samtliga skulle bli trygga i gruppen och landa i gemensamma arbetsätt. Målet uppnåddes delvis, så vi fick förlänga tiden för inlärningen. Gruppen blev dock bekväm i rollerna direkt, troligtvis tack vare team building utanför projektet.

Därefter satte vi som mål att vi skulle lära oss att arbeta effektivt med en kund. För att uppnå detta så förde vi en dialog med Blomsterlandet. Vi fick möjligheten att diskutera fram olika funktioner och vi fick tillgång till den data i form av en databas med bilder och info. Vi fick även insikt i branschen och verksamheten i sin helhet. Arbetet blev mycket effektivt och fortsatte vidare med mailkontakt.

Ett kortsiktigt mål var att de olika paren skulle specialisera sig och därefter arbeta cross-functional. Några gruppmedlemmar fokuserade därför mer på de tekniska delarna och andra mer på styling. Vid våra daily scrums informerades resterande gruppmedlemmar vad som hade gjorts och hur det gick till så att alla lärde sig det. Vi ville dessutom lära oss mer om hur vi kunde använda olika paket, APIer och databaser till applikationen. Även Redux, vilket vi märkte att vi behövde när applikationen började bli mer och mer avancerad och data behövde lagras globalt. Här fokuserade en del av gruppen på denna uppgift och delade med sig av kunskapen på våra daily scrums. Anledningen till att vi delade upp och specialiserade oss var för att vi ansåg att det räckte att gruppmedlemmarna hade grundläggande kunskaper på allt men spetskompetens på en specifik. På så sätt täckte hela utvecklingslaget det som krävdes och vi kunde skapa mer värde. En grund för att detta skulle fungera var ett bra lagarbete.

De sista kortsiktiga målen i projektet vi satte var att färdigställa applikationens grundläggande funktionalitet och förbereda oss inför presentationen. Detta då det fanns en risk att vi skulle stöta på problem i slutskedet av utvecklingen och därför valde vi avsätta tid för att hantera detta innan deadlinen. För att göra detta fick vi återigen prioritera vilka funktioner som genererade mest värde och fokusera på

dessa. Det viktigaste för oss var att i både presentationen och i denna rapport visa vad, hur och för vem de olika funktionerna och flödena i applikationen bidrar med värde. Vi påbörjade planeringen av presentationen och arbetet med rapporten i god tid för att ge alla i gruppen möjliga att vara delaktiga och revidera de delar som önskas. Vi blev nöjda med vår insats på presentationen och hoppas nu göra ett lika bra jobb med rapporten.

Sammanfattningsvis är vi nöjda med slutprodukten samt processen dit. Vi anser att vi har uppnått de mål vi satt, både långsiktiga och kortsiktiga.

B:

Till nästa projekt hoppas vi få lära oss nya tekniker och färdigheter inom mjukvaruutveckling. På samma sätt som vi fick chansen att utföra uppgifter som var utanför våra tidigare kunskaper hoppas vi få chansen att utvecklas vidare i nästa projekt.

I detta projekt har vi prioriterat gruppens välmående och sammanhållning. Ett välmående lag och en god sammanhållning har gjort att det har varit roligt att arbeta med projektet och vi tror att det också bidragit till ett bättre resultat. Detta är något som vi vill fortsätta med i framtida projekt.

Vidare önskar alla i gruppen att bli ännu bättre på Scrum och i nästa projekt hade vi velat testa på metodiken igen. Vi anser att vi arbetade bra med Scrum i slutet och vill därför behålla våra kunskaper. Det krävs därför att vi fortsätter arbeta med metodiken i vidare projekt.

Fortsättningsvis hade vi velat lära oss mer om hur man arbetar med en kund. Blomsterlandet var en tacksam kund som ville ha vår input i nästan alla delar av applikationen. Detta var otroligt skönt för oss under projektet men är någonting vi antagligen inte kommer stöta på frekvent i verkligheten. Vi anser att det hade varit lärorikt att arbeta efter en tydligare kravspecifikation och behöva argumentera för våra åsikter.

A→B:

För att lära oss lika mycket i nästa projekt så ska vi likt i detta projektet avsätta tid för lärande. Dels för lärandets skull men också för att det accelererar arbetet längre fram att ha en god förståelse för de tekniker man arbetar med.

För att skapa samma fina grupp dynamik i framtida projekt så ska vi se till att likt i detta projektet göra lite roliga aktiviteter utanför projektets ramar. Vi har även tyckt det varit bra att ha KPI:er som mäter hur gruppen mår och vi vill fortsätta med att mäta detta i kommande projekt. Detta leder till att gruppmedlemmar känner sig sedda och bidrar till en god stämning.

För att arbeta bättre med Scrum krävs det först och främst att vi bibehåller de kunskaper vi har. Vi måste därför dokumentera våra lärdomar så att inlärningstiden vid nästa projekt blir kortare. Det krävs också att vi är villiga att anpassa oss för andras åsikter. Att vi i gruppen har lärt oss arbeta tillsammans med Scrum betyder inte att det kommer att fungera lika bra i nästa projekt.

För att bli bättre på att arbeta med en kund krävs det att vi utsätter oss för olika typer av situationer. Alla kunder kommer att vara olika och det finns inget bättre sätt att hantera detta än att följa mottot "learning by doing".

Your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value:

A:

Att utveckla user stories är en av de saker som initialt var svårast för oss. Vi tenderade i början att uttrycka oss horisontellt och utan att förmedla vilket värde varje user story skapade. För att förbättra detta diskuterade vi med handledarna samt varandra och omformulerade många befintliga user stories. Detta var ändå inte tillräckligt då de ofta var för omfattande och värdet låg på fel intressent. Vi insåg att vi behövde extern hjälp och Malin försåg oss med en kontakt som såg över formuleringarna och gav oss tips på förbättringar. Därefter gjorde vi en helt ny Scrum-board baserat på dessa tips. För att se till att Scrum-boarden användes korrekt och fick kontinuerligt underhåll satte vi Malin som ansvarig för Scrum-boarden. Hon jobbade dessutom mot att våra user stories följde ett standardmönster.

Tabell 2. Visandes ett exempel på utvecklingen av user stories

Vid start	<i>As a user, I want to get a notification with offers from Blomsterlandet</i>
Efter diskussion med handledare	<i>As a user, I want to get a notification in my app so that I know when I can buy plants</i>
Efter extern hjälp	<i>As Blomsterlandet, we want to send out a notification in the app so that the customers knows when it's time to buy and plant tomatoes</i>

Vidare utvecklade vi Acceptance Criterias för våra user stories. Detta gjordes i ett tidigt skede av projektet, i samband med våra första user stories. Vi upplevde samma problem med våra Acceptance Criterias, de var för vaga och otydliga. Detta ledde till förvirring då alla gruppmedlemmar inte hade samma uppfattning om när ett kriterium har uppfyllt. Vi använde även Acceptance Criterias som tasks, vilket fungerade i vissa fall men inte alla. För att lösa problemet började vi omformulera våra Acceptance Criterias så att de blev tydligare, vilket var nödvändigt för att minimera förvirringen. Detta gjorde vi i samband med att vi omformade vår Scrum board. Vi började även utforma tasks för våra user stories, vilket också minskade förvirringen och gjorde det tydligt vilka tekniska saker som krävdes för att skapa funktionaliteten till en user story. Detta gjorde det lättare att veta vad man skulle göra samt enklare att göra estimates för våra user stories.

Fortsättningsvis satte vi en initial velocity på 160 baserat på att alla skulle lägga 20 timmar var i veckan. Sedan satte vi effort estimates på våra user stories. Vi insåg redan efter första sprinten att vi inte var så exakta som vi trodde att vi skulle vara i våra estimates och det var något vi kämpade med under hela projektet. Den främsta anledningen var tekniska problem som tog mycket tid. Vidare påverkade sjukdomar,

lov och andra kurser den tid gruppmedlemmarna kunde lägga på projektet. Detta ledde till att vi hade svårt att färdigställa de user stories som låg i sprint backloggen. Vi bestämde oss för att anpassa vår velocity baserat på hur veckan såg ut vilket gjorde att vi hinner med uppgifterna. Exempelvis satte vi velocity 80 under lovet. Utöver det upptäckte vi att vår precision i våra estimates varierade, vissa user stories krävde mer än vi estimerade och andra mindre. Detta var något vi behövde förbättra. I samband med att våra user stories, acceptance criterias och tasks utvecklades blev det enklare för oss att sätta effort estimates. Vi insåg att vi hade haft en skev bild av våra user stories eftersom vi hade lagt för lite arbete på att undersöka vad de egentligen innehöll. Våra estimates förbättrades också när vi hade genomgått ett antal sprints, då hade vi mer erfarenhet att basera våra estimates på än tidigare. För att förbättra våra estimates ytterligare valde vi att pröva metoden Planning Poker. Detta ansåg vi fungera väldigt väl, det var ett effektivt sätt för oss att sätta estimates som grupp.

Något vi arbetat aktivt med under projektet var att prioritera våra user stories i projekt backloggen baserade på vilket värde de genererade, detta gjorde vi bland annat genom att göra prioriteringar enligt "Värdekurvans" principer. Det var svårt för oss att avgöra vad som skapade mest värde för kund och vi bestämde oss, efter samtal med handledare, att utforma ett acceptansflöde. Med hjälp av detta kunde vi urskilja vad som var gjort och inte vad som var centralt för vidare utveckling. Detta användes även som ett acceptanstest (som ni kan läsa om i under nästa rubrik).

För att effektivisera återkommande uppgifter valde vi att använda oss av Timeboxing. Vi insåg att vi ofta drog ut på uppgifter som tog lång tid och var tvungna att sätta någon form av deadline för dessa. Detta hjälpte oss att hålla tempot uppe, vilket då gjorde vårt arbete mer effektivt. Exempelvis lade vi cirka fem timmar på vår första Team Reflection, medan de senare tog cirka 2 timmar. I slutet av projektet använde vi oss av Timeboxing på flera av uppgifterna. Vår handledare tipsade oss om det gällande uppgifter som inte innehåller kod. Vi insåg att vi hade en tendens att överarbeta dessa uppgifter och detta hjälpte oss i projektets slutskede.

Vi har haft en definition of done för våra stories, men vi har följt den lite för lätsamt. Vi har exempelvis varit duktiga med att skapa pull-requests, men mindre duktiga med att kommentera vår kod.

Vi har avsatt mycket tid för att förbättra och uppdatera våra user stories, estimates och vår Scrum board samt prioriteringen av user stories. Vi såg detta som centralt för att förbättras i projektet och prioriterade därför detta högt.

B:

I nästa projekt hade vi velat arbeta med våra user stories på ett liknande sätt som vi gjort mot slutet av detta projekt. Våra user stories har fungerat bra, men de hade kunnat fungera ännu bättre om vi haft bättre struktur på vårt arbete med dem. Vi

hoppas i nästa projekt ha en högre standard på våra user stories redan i början.

Vidare hade vi velat förbättra utformningen av Scrum boarden i nästa projekt. Vi hade velat ge mer tid åt en framtida Scrum master att hantera Scrum Boarden i nästa projekt. Att ha en ansvarig person för detta redan från början hade underlättat för resten av gruppen och gjort arbetet med user stories mer strukturerat. Utöver detta så hade det varit intressant att få insikt i hur företag arbetar med Scrum i praktiken.

Slutligen hade vi velat använda oss utav våra produktägare på ett mer effektivt sätt baserat på Scrum. De skall skriva och prioritera User Stories till backloggen, något som vi saknade. Detta diskuteras mer i en senare punkt.

A→B:

Vi måste dokumentera hur vi har arbetat med våra user stories för att inte förlora de kunskaper vi erhållit under kurserna. Vi måste även arbeta mer strukturerat från start och avsätta mer tid till att förbättra dem. Nu när vi har lärt oss att utforma acceptance criterias och bryta ner user stories till tasks är detta någonting som vi borde göra redan vid starten av nästa projekt. Utöver detta måste vi även vara bättre på att använda metoder som exempelvis planning poker för att sätta våra estimates.

Vidare måste vi utse en Scrum master redan vid starten av nästa projekt. Dessutom ska produktägarna definieras. Vi måste även se till att dessa individer får tillräckligt med tid för att utföra sina uppgifter. För att få insikt i hur företag arbetar med Scrum i praktiken måste vi utnyttja våra kontakter i arbetslivet och ta kontakt med dem.

Your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders:

A:

De första testerna vi utförde i början av projektet, kopplat till våra acceptance criterias, var att kunna få programmet att starta och köra innehållet som tas upp i Udemy-kursen. Att vara säkra på att vi kan köra React Native på allas datorer skapade ett stort värde för oss som grupp. Det gjorde att vi kunde utveckla projektet från flera enheter och var inte beroende av vissa personer.

Vidare började vi testa applikationen visuellt och kollade att detta stämde överens med vår mockup. Eftersom vi vid denna tidpunkt mestadels hade producerat vyer, vilket är svårt att testa på annat sätt än visuellt, testade vi koden "för hand". Under projektets gång insåg vi att mockupen inte innehöllade alla skärmar och komponenter som vi tänkt oss. Den illustrerade heller inte några flöden eller integrationen mellan de olika vyerna. Vi utformade därför ett acceptansflöde. Detta utgjorde ett test för att se så att de olika komponenterna tog oss dit som var tänkt men även de visuella delarna som saknades i mockupen. Detta testade dock endast de statiska delarna av applikationen och inte de dynamiska. Exempelvis olika animeringar syns inte på acceptansflödet. Detta gjorde att vi fick koppla acceptanskriterierna mer till flödet och därefter testa det visuellt. Testerna på de värdeaddrande flödena eller en funktionalitet gjordes innan det mergades in i master. Vårt acceptansflöde har därmed fungerat som utgångspunkt för acceptanstest av den nya koden.

Vi diskuterade och provade olika metoder för att testa vår kod. Exempelvis studerade vi möjligheten att använda oss utav programvaran Travis CI, som möjliggör automatiserad körning av tester och deployment av koden. Vi insåg snabbt att detta var onödigt då vi inte hade så mycket enhetstester. Enstaka enhetstester skapades med hjälp av Jest. För att kompatibilitetstesta koden valde vi att köra den på flera olika enheter. Tack vare detta upptäckte vi problem som vi behövde lösa innan merge till master. Genom att genomföra dessa test har vi kunnat undvika att introducera buggar i master-branchen och att萌ga ofärdiga funktionaliteter.

B:

Enhetstester hade underlättat utvecklingen och integrationen mellan olika delar av applikationen. Att lämna ifrån sig kod som inte är helt och hållt testad kan leda till buggar i senare skeden som kan vara extra svåra att felsöka då orsaken kan vara oklar då ursprunget kan vara svårt att lokalisera. Att säkerställa att allt i master-branchen fungerar som det ska är därför viktigt och är något som vi kommer ta med oss till nästa projekt. Tillsammans med enhetstester hade Continuous Integration varit ett mycket användbart verktyg, då det blir enklare att se vilken commit som har orsakat problemen. Detta är något som vi hade velat använda oss av i framtida projekt.

Något som hade underlättat acceptanstestningen hade varit en mer konkret produktägare som hela tiden har den rollen. Under projektet har våra produktägare även haft andra förpliktelser vilket har lett till en ScrumBut då det har varit svårt att fylla rollen. En produktägare utan övriga arbetsuppgifter hade underlättat dennes uppgifter och därmed gjort det enklare att arbeta på ett bättre sätt med acceptanstest.

I nästa projekt hade vi velat använda verktyg ämnade för framtagande av mockups då det hade gjort hela processen enklare, speciellt acceptanstesterna. Detta då interaktiva mockups hade kunnat visas upp för olika stakeholders samt underlättat uppvisande av flöden i applikationen. Vid ett längre projekt genererar mockup-verktyg mer värde och därmed kan de vara värdas den initiala tidsåtgången, men vi gjorde avvägningen att det inte var värt det för oss.

A→B:

För att uppnå detta behöver vi sätta oss in i testmiljön i ett tidigt skede av projektet så att man kontinuerligt under projektets framfart skriver testad kod och inte bara testar allt i slutet. Det är även viktigt att få igång CI-programvaran tidigt så att testerna körs automatiskt.

För att ha en mer konkret produktägare behöver gruppen tidigt utse en medlem som antar rollen och att gruppen kontinuerligt under projektet ställer frågor som berör produktägarens ansvar.

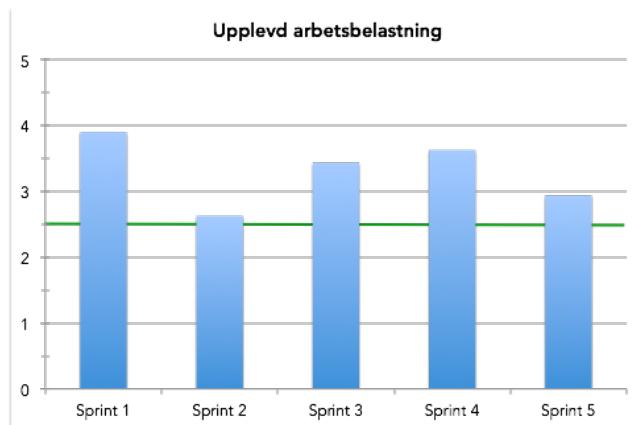
För att använda oss av mockup-verktyg till nästa projekt behöver vi lära oss om det efter att avvägande gjorts sett till tid kontra värde.

The KPIs you use for monitoring your progress and how you use them to improve your process:

A:

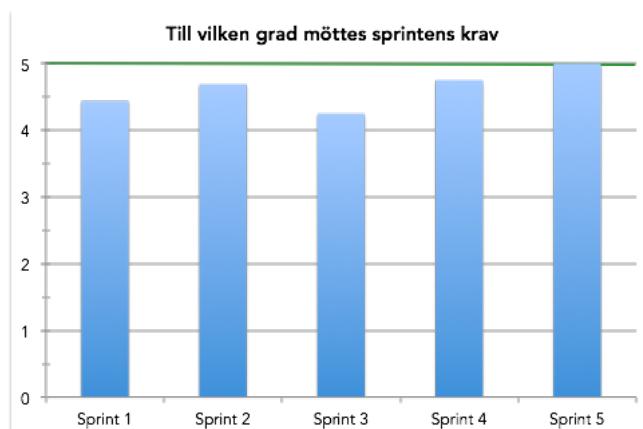
Gruppen har tagit fram fyra mätvärden som vi ansåg var centrala för att mäta hur väl processen framskred. Det första mätvärdet "Upplevd arbetsbelastning" innebär det den belastning som kursen innehåller för varje individ i gruppen. Den upplevda arbetsbelastningen mäts på en skala från 1-5 där 2,5 är en arbetsbelastning som kan anses vara rimlig. Om den upplevda arbetsbelastningen är hög har vi försökt lösa det genom exempelvis anpassa velocity så att det är hållbart för alla.

Figur 1. Visande medelvärdet för varje sprint vid KPI-mätning för upplevd arbetsbelastning.



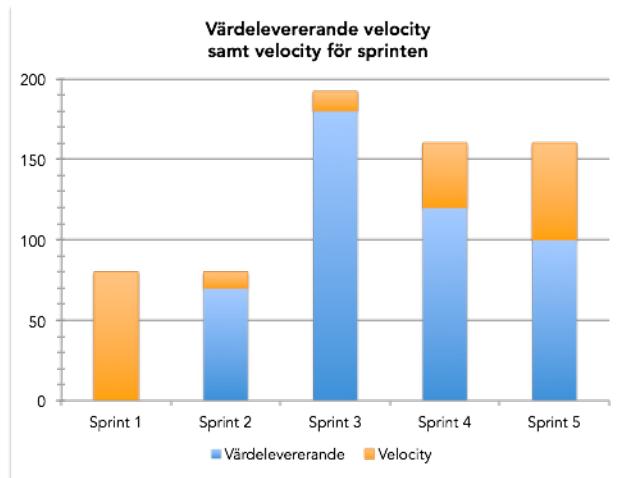
Vidare har gruppen mätt "Till vilken grad varje individ mötte sprintens krav". Detta är, liksom "Upplevd arbetsbelastning", en subjektiv bedömning där varje individ får sätta ett eget värde på sina prestationer. Värdet visade hur stort gapet var mellan vad vi producerade och vad vi hade satt upp som mål. Om medelvärdet var under 5 så visade detta att vissa user stories var ofärdiga och skulle behållas till nästa sprint. Ett värde på 5 visade att man helt uppfyllt sprintens krav.

Figur 2. Visande medelvärdet för varje sprint vid KPI-mätning för till vilken grad varje individ mötte sprintens krav.



Fortsättningsvis mätte vi värdelevererande velocity. Till detta angav vi också sprintens övergripande velocity så att en jämförelse kunde göras för att se hur mycket som genererat värde. Skattningen av velocityn var någonting som vi initialt hade problem med och de första mätningarna är missvisande då vi hade dålig koll på hur vi skulle använda värdet. Vi satte sedan velocityn till antal timmar som vi ansåg att gruppen gemensamt skulle lägga per sprint. Detta insåg vi sedan var olämpligt, bland annat då team reflection inte räknades in i vår uppsatta velocity men även efter rekommendationer från handledare. Fram tills den sista sprinten hade vi tagit hänsyn till detta och anpassade den totala estimaten varje gruppmedlem tog på sig vilket bidrog till att gruppaktiviteterna såsom team reflection även inkluderades i velocityn. Detta gjorde att den värdelevererande velocityn skiljer sig från den övergripande velocityn mer i den sista mätningen.

Figur 3. Visande medelvärdet för varje sprint vid KPI-mätning för värdelevererande velocity samt velocity för varje sprint.



Vi ville ha en arbetsbelastning på 2,5 och att sprintens krav skulle mötas till fullo, det vill säga grad 5. Dessa värden är kan motverka varandra och vi var tvungna att säkerställa att en ökning av arbetsbelastningen inte skedde på bekostnad av gruppens välmående. Vi önskade också en hög andel värdelevererande velocity. Detta kunde vi förbättra genom att arbeta med Scrum.

Fokus låg den första sprinten på att lära oss verktygen som vi skulle använda oss av i resten av projektet. Vid denna tidpunkt hade vi dålig koll på vad som räknades som värdelevererande och därfor satte vi att vi inte hade producerat något av värde under sprinten. Vi trodde alltså att det enda som utgjorde värde för kund var producerad kod och något fysiskt att visa upp. Sanningen var att vi producerade en hel del värde för kund genom diskussion gällande applikationens funktion.

Vi var relativt nöjda med resultatet för upplevd arbetsbelastning och hur väl sprintens krav möttes då vi stötte på mycket tekniska problem vid nedladdning av

de olika programvaror vi använde oss utav. De flesta problem uppstod för ägarna av Mac-datorer.

Sprinten därefter började vi producera kod. Det var denna sprint vi började basera vår velocity på antal timmar gruppen gemensamt skulle lägga. Anledningen till att den var så pass låg var att det var lov så vi ansåg för gruppens välmående att vi skulle få lite ledighet.

Vi ökade velocityn för nästkommande sprint tillbaka till snittet på 20 timmar per vecka. Detta medförde att vi ville att samtliga medlemmar skulle lägga 24 timmar var under sprinten. Här producerades mycket kod vilket resulterade i en hög siffra för värdelevererande velocity. Tyvärr insjuknade två gruppmedlemmar, Andrea och Benjamin under denna sprint. De olika delarna skulle dessutom integreras denna sprint vilket medförde att några par var beroende av andra. Dessa orsaker medförde en lägre siffra på till vilken grad sprintens krav möttes. Vidare berodde den höga arbetsbelastningen på att det var slutspurt i kandidatarbetet för de som läser Industriell ekonomi.

Den upplevda arbetsbelastningen i sprinten därför var relativt oförändrad från föregående sprint då slutsprutens i kandidatarbetet fortfarande pågick. Dessutom då detta projekt var nära sitt slut lades extra mycket tid, för att säkerställa att vi skulle leverera det till kunden som vi satt som mål. Vårt snitt för hur väl vi upplevde att vi mött sprintens krav var denna gång 4.75. Detta kan jämföras med den föregående sprintens snitt på 4.25. Vi menade att anledningen till denna ökning berodde på två saker. Dels att vi blev mer och mer bekväma i utvecklingsmiljön vilket gjorde att det gick fortare att få saker gjort. Dels att vi insåg att vi inte skulle ta på oss uppgifter som krävde 20 timmars arbete så att de gemensamma uppgifterna inkluderades i velocityn. Därför sjönk den värdelevererande velocityn i förhållande till den avklarade.

Det var först vid den sista sprinten som vi blev helt nöjda med våra värden. Detta berodde troligtvis på att vi hade lärt oss att anpassa estimates och inkludera kringliggande uppgifter i velocityn. Vi hade även lärt oss från tidigare veckor och enkla misstag slutade förekomma. Troligtvis berodde det även på att mycket av det producerade var nästintill färdigt och vi finslipade detta under denna sprint.

B:

Våra KPI:er har inget sätt att visa på om en individ har gjort mer på en vecka än den tagit på sig. Har en person 5 i arbetsbelastning, och 5 i uppfylld grad av sprintens krav, så säger det inget om hen har gjort "mer" än de ursprungliga user stories som hen tog på sig, därför hade vi velat ha ett mer konkret sätt att visa vad en person har gjort till nästa projekt. Till exempel att ha att man mötte sprintens krav fullt ut som värdet 3 istället för 5. Detta hade tillåtit att registrera både om man gjort för lite eller mer än vad som förväntats.

Vidare vill vi kolla närmare på vad företag använder för KPI:er och hur de använder resultatet för att förändra sitt arbete

A→B:

För att tydligare visa vad en person har gjort krävs det att vi utformar nya KPI:er. Ett exempel på ett mer konkret KPI för framtida projekt är avklarad velocity per person och sprint. Detta hade visat tydligare hur mycket varje person producerade och det hade också möjliggjort att man skulle kunna skapa individuella velocities som i sin tur skulle göra det enklare att planera inför sprinter. Detta är också någonting som vi hade velat förbättra till nästa projekt.

Vidare måste vi utnyttja våra kontakter för att få insikt i hur företag arbetar med KPI:er. Vi känner en handfull personer som jobbar inom mjukvaruindustrin. Vi kan fråga dem hur de gör med KPI:er och vilka de använder. Det finns även gott om litteratur och artiklar om ämnet.

Social Contract and Effort

Your social contract, 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:

A:

Gruppens första social contract sammanställdes under projektets första vecka. Vi utgick från en mall som vi ansåg var lämplig för uppgiften. Det var svårt att förutse vilka problem som kunde uppstå och vilka regler som hade kunnat förebygga detta. Ett exempel på ett sådant problem som uppstod under kursens gång var att vi hade problem med kommunikationen, var all kommunikation skulle ske vid olika syften. Vi löste detta genom att skapa trådar i Slack och reviderade vårt Social Contract baserat på detta. Vi märkte dessutom att vi behövde klara riktlinjer gällande var vi sparar och delar filer. Dessa riktlinjer blev direkt en del av vårt social contract och gruppen arbetade mot att följa detta därför.

Vidare insåg vi efter att vi började producera kod att vår ansvarsuppdelning inte fungerade som vi ville. Punkten i social contract om att varje user story ska ha en huvudansvarig och en co-creator hade inte följts och skapade bara mer förvirring. Vissa user stories löstes på egen hand, vilket vi insåg inte behöver vara negativt, samt att vi arbetade med parprogrammering och då blev två stycken huvudansvariga. Vi plockade därför bort denna punkt ur kontraktet.

Vårt mål med vårt social contract var att skapa ett heltäckande kontrakt som kunde fungera som en grund för allt arbete. Vi anser att vi har lyckats med detta efter några modifieringar av kontraktet. Anledningen till att vi lyckades bra är att vi tog upp problem som uppstod i gruppen när saker inte fungerade. Vi diskuterade hur vi ville att arbetet skulle fungera och tog tillsammans fram en lösning på det här. Därefter reviderade vi och uppdaterade vi kontraktet.

Sammanfattningsvis har kontraktet följts väl av gruppens medlemmar och har reviderats när detta har ansetts vara nödvändigt. Alla har arbetat aktivt kontinuerligt mot att följa och förbättra kontraktet.

B:

Vi är väldigt nöjda hur vårt arbete kring vårt sociala kontrakt har fungerat i detta projektet och vi är eniga om att vi hade velat arbeta likadant i kommande projekt. Vi har förståelse för att ett social contract ska vara anpassat till situation och gruppmedlemmar och skulle därför ha ett öppet sinne för att behöva göra förändringar baserat på det.

A → B:

Detta skulle vi uppnå med genom att fortsätta att uppdatera kontraktet under projektets gång för att hantera de utmaningar som man inte kan förutse i början. Vi kommer att ta med oss hur vi arbetat med detta till kommande projekt.

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):

A:

I början av projektet satte vi att vi skulle ses varje måndag 8-17 och torsdag 8-12, vilket kvarstod under hela projektet för att slippa lägga onödig tid på planering. Utöver detta skulle gruppmedlemmarna få styra sin tid själva. Då hade vi inget system för att mäta den nedlagda tiden på kursen. Detta berodde på att en majoritet av arbetet gjordes i grupp i form av planering och gemensamt arbete. Vi insåg dock att det kunde underlätta för en själv att logga sina timmar, både gemensamma men även individuella. Därför skapade vi ett excelark som fungerade som tidslogg där varje gruppmedlem enkelt kunde logga sina timmar. Detta gjordes för att alla själva skulle kunna kontrollera att man lade den tid som förväntades och ha det som riktlinje.

Vi nöjda med hur vår tidslogg fungerade. Vi var alla olika snabba på att lära oss saker och arbetet såg väldigt annorlunda ut för de olika paren på grund av specialiseringar. Paren tog på sig uppgifter baserat på specialiseringarna vilket gjorde arbetet tidseffektivt. Upplägget genererade mycket värde på kortare tid än vad det hade tagit om alla jobbat med alla delar. Med hjälp av tidsloggen kunde vi kontrollera att vi klarade av de åtaganden som man satt upp för sig själv och den man parprogrammerade med på den tid som vi beräknat vår velocity och estimates med. Gjorde vi inte detta eller blev klara med våra antaganden tidigare, kunde vi ta det i beaktning när vi bestämde estimates under nästa sprint. Tidsloggen var också hjälpsam när någon körde fast och inte kunde leverera så mycket som planerat. Då kunde man ändå se att man hade lagt ned tid och gruppen kom fram till att samtliga i gruppen har lagt den tid som krävts. Någonting vi alla har lärt oss är att det i början av ett projekt är väldigt svårt att uppskatta hur lång tid hur någonting ska ta, men att allt eftersom man får mer erfarenhet i området så blir det lättare att estimera. Den har också bekräftat att vi under projektet hållit en sustainable pace.

B:

I ett nytt projekt hade vi velat arbeta med en tidslogg på samma sätt som vi gjort i det här projektet. Vi hade velat se till så att alla fyller i vår tidslogg redan från början så att vi kunnat följa vår tidsåtgång genom hela projektet och på så sätt ha det som hjälpmittel när vi förbättrar våra estimates och velocity.

A→B:

För att uppnå det här skulle vi satt upp en tidslogg redan från start och låtit en punkt i social contract klargöra hur den ska fyllas i. Vi hade även tagit upp tidsloggen under våra reviews.

Design decisions and product structure

How your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value:

A:

Det första designbeslutet vi tog i gruppen var gällande vilket programmeringsspråk och ramverk vi skulle använda. Efter att ha undersökt vilka alternativ som skulle kunna vara relevanta efter våra önskemål föll valet på Javascript och React Native. Dessa valdes för att det möjliggjorde utveckling för både Android och iOS, vilket ansågs vara värdefullt för Blomsterlandet.

Det har varit relativt tydligt för gruppen hur applikationen skulle utformas visuellt. Vår kund Blomsterlandet har riktlinjer som de själva följer som vi fick ta del av vid uppstarten av projektet och kvarstod under hela. Detta innebar att samtliga färger och bilder vi inkluderade i applikationen är identiska med Blomsterlandets och det ser därmed ut som att det är en applikation gjord utav Blomsterlandet.

För att utforma applikationens flöden och funktioner skapades en mockup. Vi insåg snabbt att den inte var tillräcklig för att visualisera de olika flödena i applikationen. Därmed valde vi att skapa en wireframe för hand. Denna visade de grundläggande flödena i applikationen, hur komponenterna skulle se ut på varje sida men även i vilken ordning det skulle produceras. Efter diskussion med handledaren utformades även ett acceptansflöde baserat på denna och mockupen. Det var som en förbättrad variant av de tidigare nämnda med tydligare design då den var gjord på datorn.

Vi valde att använda Redux för att hantera appens tillstånd (där t ex växter och notiser m.m. lever under körtid). Detta då vi stötte på stora problem när vi inte hade ett förutsägbart, flexibelt och centraliserat sätt att hantera appens tillstånd. Användningen av Redux gjorde koden mer lättläst och det blev lättare att hantera data i applikationen.

Under kursens gång har vi använt oss utav ett antal API:er och bibliotek / paket. Många av dem implementerades av estetiska skäl. Detta innehöllade iconer, animationer och grafiska representationer av data. Till exempel implementerade vi ett CSS-ramverk till webbplatsen där Blomsterlandet kan se ordrar för att underlätta utvecklingen av den. Dessa paket gav applikationen och webbplatsen ett professionellt intryck. Vi gjorde bedömningen att det skulle vara för tidskrävande att utveckla dessa komponenter från grunden. Halvvägs in i projektet upptäckte vi att vi hade gjort en del dubbearbete och letat efter liknande paket / bibliotek som andra par redan använt sig utav. För att undvika detta bestämde vi att man alltid skulle informera resterande gruppmedlemmar när nya paket / bibliotek importeras. Vi vände oss även vid att fråga runt i gruppen när nya paket skulle importeras om det fanns något liknande som redan används.

Vidare importerade vi även Firebase till applikationen. Firebase innehåller en stor mängd olika tjänster som var användbara i vår applikation. De tjänster vi använde oss av var cloud messaging för notiser, samt firestore som databas. Vidare anledning till att vi valde Firebase över andra system var att vi redan hade redan använt oss av Firebase i kursen på Udemy, vilket gjorde att vi redan visste att vi enkelt kunde få det att fungera i vår applikation.

Sammanfattningsvis har vi behövt ta många designbeslut under projektets gång. En del beslut har varit självtaliga medan andra har krävt en djupare diskussion av oss gruppmedlemmar och en del beslut borde tagits tidigare, t.ex. användandet av Redux. Vi är nöjda med de beslut vi tagit och anser att samtliga API:er och paket har skapat värde för en potentiell användare och/eller Blomsterlandet.

B:

I ett nytt projekt hade vi velat använda en resurs som Redux tidigare i projektet för att få ett förutsägbart, flexibelt och centraliserat sätt att hantera appens tillstånd redan från början. När vi implementerade detta i mitten av projektet blev det ett avbrott i vår framfart när vi behövde lära oss något nytt och skriva om en del kod.

Under nästa projekt hade vi velat testa att använda nya API:er. En del av de vi valde att använda var inte optimala, de var för krävande för en del av våra datorer. Det hade varit fördelaktigt att hitta API:er som var lika snygga estetiskt men som inte förhindrade arbetet. Vi hade även velat testa på att använda fler av Googles API:er.

Till nästa projekt där vi skriver kod i javascript vill vi använda oss utav typescript då det resulterar i att man enklare kan implementera olika designmönster, samtidigt som det också ökar typsäkerheten och modulariteten i applikationen.

A→B:

För att uppnå detta måste vi planera bättre inför eventuella problem som ska uppstå. Vi borde ha kunnat förutspå att vi var tvungna att använda Redux. Om vi i nästa projekt försöker att kontinuerligt diskutera projektets framtid kommer det underlättा otroligt mycket.

För att finna nya API:er, paket och bibliotek krävs det att vi söker på nätet. Vi måste vara öppna för att använda API:er, paket och bibliotek från sidor som vi inte använder i det här projektet. Vi måste även undersöka vilka API:er, paket och bibliotek som är relevanta för nästa projekt.

Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents):

A:

Från den sprint då vi började producera kod så skapade vi det första interaktionsdiagrammet tillsammans på tavlan. Detta utvecklades senare till ett acceptansflöde vilket utgjorde grunden för våra user stories och prioriteringen av dessa. Under projektets gång försökte vi kontinuerligt förbättra kommenteringen av koden. Detta gjorde vi så att det skulle vara enkelt att se tillbaka på koden och förstå de andra gruppmedlemmarnas kod. I efterhand anser vi att vi hade kunnat vara bättre på att kommentera koden. Detta var ett område som vi slarvade mycket med.

I slutet av projektet skapades UML-diagram som berörde de viktigaste delarna i applikationen, så som exempelvis shoppen, mina växter och backenden som helhet. Detta gör det enklare för oss, och för eventuella framtida utvecklare att förstå hur koden hänger ihop, och för att finna eventuella förbättringar.

Ursprungligen var planen att ha hög code-reuse, och en mapp med namnet "common" för diverse återanvändbara komponenter skapades. Det här tankesättet följdes inte alls så bra, och lathet ledde oss till att i många fall inte abstrahera komponenter för bättre code-reuse, utan istället kopiera och modifiera befintliga lösningar. Detta på grund av att vi inte redan från början strukturerade upp med hjälp av UML-diagram. Detta ledde till att vår kod blev svår att navigera i, trots ett antal försök att strukturera upp den.

Allteftersom projektet fortskred insåg vi att det hade varit användbart att dokumentera förändringen av vår Scrum-board. Vi har gjort flera förändringar och för att vi skulle kunna utvärdera dessa behövde vi kunna se över tidigare versioner av Scrum-boarden. Vi har en gammal version sparad på Trello, vilken kan jämföras med den befintliga.

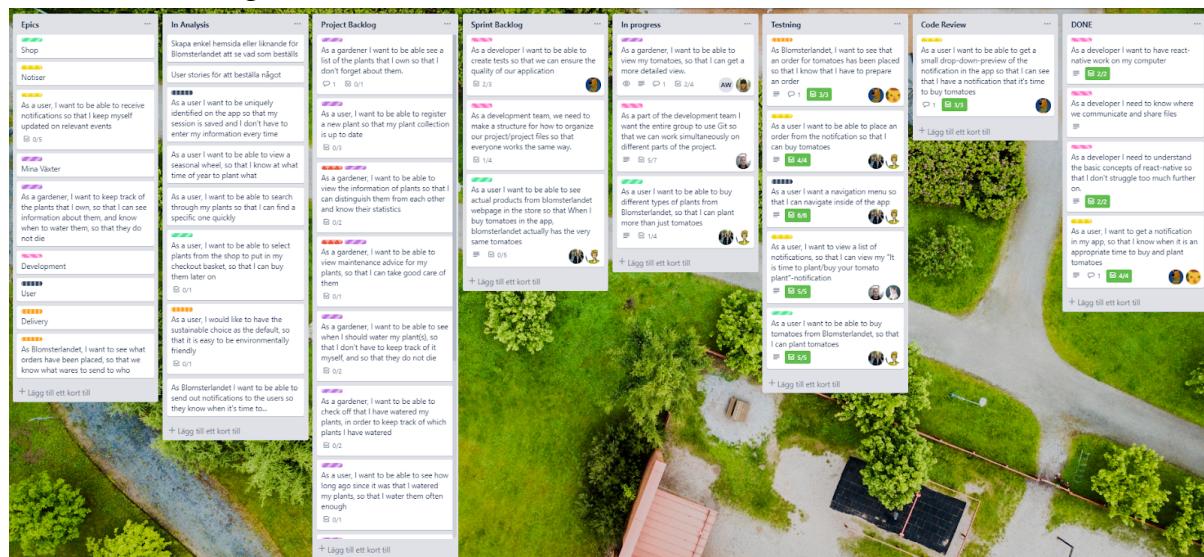


Bild 1: Äldre version av Scrum-board

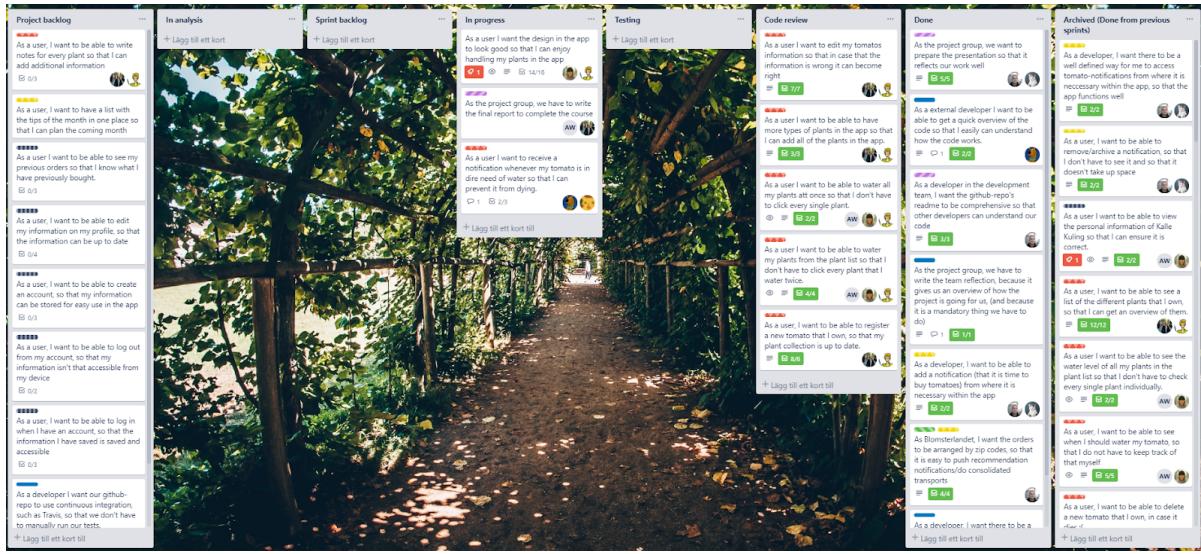


Bild 2: Uppdaterad version av Scrum-board

B:

Som nämnt ovan har vi under projektet haft problem med att få en strukturerad och lättförståelig kod med mycket code-reuse och återanvändbara komponenter. Detta är något vi vill göra bättre i ett nytt projekt.

I ett nytt projekt hade vi velat vara bättre på att kommentera vår kod medan vi skriver den för att göra det lättare för alla gruppmedlemmar att navigera i och förstå koden. Genom att kunna läsa vad koden gör och vem som skrivit den blir det lättare att koppla ihop olika funktionaliteter och återanvända komponenter.

Vi hade velat dokumentera vår scrum board under projektets gång för att kunna följa utvecklingen och i slutet av projektet kunna granska detta när vi utvärderar vårt arbete och utveckling.

A→B:

För att uppnå detta måste vi ha möten där vi diskuterar vilka komponenter som skulle kunna användas på flera ställen i vår kod och sedan sätta några personer på att utveckla dessa komponenter. Detta kommer leda till mindre levererat värde i början av projektet, men underlättar för mer producerat värde ju längre in i projektet man kommer. Detta kan vi åstadkomma genom att strukturera upp koden med hjälp av UML-diagram redan från början som sedan uppdateras när mer komponenter adderas.

Vi måste bestämma från början att all kod som skrivs ska kommenteras och att detta arbete ska räknas in när man bestämmer ett estimate för våra user stories. Vidare måste vi ta bilder inför varje sprint på vår scrum board.

För att se till att vi dokumenterar vår Scrum board under nästa projekt måste vi integrera denna arbetsuppgift på ett naturligt sätt. Detta kan vi exempelvis göra genom att lägga in det på dagordningen inför varje ny sprint.

How you use and update your documentation throughout the sprints:

A:

Allt eftersom projektet fortskred ökade betydelsen av dokumentation. Vi blev bättre på att använda oss av och uppdatera vår dokumentation ju längre tiden gick. Detta beror antagligen på att när applikationen blev mer och mer komplex så blev det svårare att hålla i huvudet vad som hade gjorts eller inte. Det gjorde vi genom att använda oss mycket av acceptansflödet som skapades i mitten av projektet.

Acceptansflödet utgjorde grunden för hur applikationens flöden skulle designas och vad som skulle göras. Med hjälp av detta var det lätt för varje gruppmedlem att på egen hand se hur långt vi hade kommit i projektet.

Under projektet så har vi också försökt dokumentera vad som gjorts genom att kommentera i koden. Detta är något som vi inte fullföljt till den grad vi önskat. Vi har kommenterat ibland och ibland inte, vi har även gått tillbaka i koden och försökt kommentera kod som vi glömt kommentera innan. Detta beror bland annat på att vi varit otåliga och velat gå vidare med att skriva mer kod istället för att kommentera den. Detta har bidragit till att vår kod ibland varit otydlig och svår att navigera i.

Sammanfattningsvis är vi delvis nöjda med hur vi använder och uppdaterade vår dokumentation under kursens gång och det finns definitivt rum för förbättring. Vi har alla förstått värdet i att lägga tid på att dokumentera vårt arbete och vi kommer antagligen bli bättre och bättre på det.

B:

I ett nytt projekt hade vi som sagt skapat våra acceptansflöden i ett tidigare skede. Vi hade, som nämnt under tidigare punkt, velat ha bättre och mer uppdaterad kommentering av koden och ha ett UML-dokument som vi kunnat uppdatera för att ha överblick över vår kod.

A→B:

För att uppnå detta hade vi redan från början gjort en plan över hur vi ska kommentera vår kod, dokumentera i ett UML-diagram och för varje sprint uppdatera våra acceptansflöden.

How you ensure code quality and enforce coding standards:

A:

För att säkerställa att samtliga gruppmedlemmars kod skulle hålla ett liknande format valde vi att använda oss av ESLint med rallycodings regler som är baserade på AirBnBs regler. ESLint underhöll vår kod och underlättade för oss i vårt arbete genom att varna vid avvikelser av hur koden skulle formateras så att vi kunde korrigera och hålla en enhetlig stil genom hela kodbasen.

Vidare ville vi se till att koden var av högsta kvalitet vid merge till master. När någons kod var redo att mergas till master lades en pull request på github. En annan gruppmedlem testade då programmet så att allt fungerade och såg över koden för att se till att allt hade gjorts korrekt. Om allt höll standard så mergades sedan koden in till master.

Fortsättningsvis insåg vi att vi hade en stor andel duplicerad kod. Detta löste vi genom att skapa återanvändbara komponenter. Det underlättade även för gruppen i det dagliga arbetet då en hel del filer kunde tas bort. Detta, som tidigare nämnt, fullföljdes dock inte till fullo, vilket ledde till att vi fortfarande hade relativt rörig kodstruktur.

Mot slutet av projektet spenderades mycket tid på att lösa ett flertal buggar i syfte att öka vår code quality. De flesta buggarna var sedan tidigare kända men nya hittades också genom att gå igenom acceptansflöden och aktivt försöka ”breaka” applikationen.

B:

I ett nytt projekt hade vi velat lära oss mer om samt använda oss av Continuous Integration då det hade underlättat utvecklingen på många sätt. Dels hade det underlättat testningen av mjukvaran och därmed försäkrat körtiden kvalitetsmässigt. Dels hade det underlättat genom att förtydliga när något går fel.

För att säkerställa att kodstandarder säkerställs vill vi även i framtiden använda oss av linting med gemensamma regler så att allas kod lintas på samma sätt. På så vis blir koden enhetlig men givetvis kommer inte alla programmera på samma sätt.

A→B:

För att lära oss mer om Continuous Integration i framtiden vill vi arbeta med projekt där det är passande att implementera samt sätta oss in mer i testramverk som är relevant för det projektet.

För att säkerställa att alla använder sig av samma linting och regler tänker vi inom nästa arbete prata om det inom gruppen och säkerställa att alla använder samma. Vi vill göra detta tidigt då vi anser att det hjälpt oss mycket att vi implementerade det tidigt i detta projektet.

Application of Scrum

The roles you have used within the team and their impact on your work:

A:

I början av projektet utsedde vi inte någon Scrum master eller produktägare då de user stories vi som utvecklingslag behandlade syftade till inlärning. Initialt ansåg vi att det var onödigt att tillsätta en Scrum Master. Vi använde oss istället av en mötesledare. Mötesledaren tog på sig en del av en Scrum Masters uppgifter och vi ansåg att det var tillräckligt bra.

Vidare under projektet uppkom ett behov av tydligare roller. Vi prövade på att tillsätta en Scrum Master samt utsåg en mötesledare och sekreterare inför varje möte. Det var fortfarande oklart vad en Scrum Master hade för arbetsuppgifter och vi skapade alternativa uppgifter som passade våra behov. Det var inte förrän i slutet av projektet som vi lyckades använda oss av Daily Scrum och Scrum Masters på ett effektivt sätt. Vi tillsatte ett av våra programmeringspar, Gustav och Malin till rollen. Det underlättade att de kunnat både övervaka och ha inblick i allas arbete samtidigt som de haft projektets värde i fokus. Detta bidrog till att vi kunde fokusera vårt arbete på endast värdeaddrande aktiviteter.

Tidigt in i projektet utsedde vi två produktägare som hade ansvar för att definiera de värden som efterfrågades av Blomsterlandet. Vi tillsatte Andrea och Caroline till denna roll då det var de som hanterade kontakten med Blomsterlandet, vilket blev naturligt. Denna rollen har vi inte utnyttjat när vi skapat våra user stories till den grad vi önskat.

B:

I ett nytt projekt hade vi utnämnt Scrum Masters varje sprint redan från början för att få mer struktur på vårt arbete och undvika dubbeltarbete exempelvis. Vi hade behövt låta produktägarna ta en större roll och tillfråga dem mer aktivt när vi planerat våra sprinter för att lättare kunna prioritera rätt saker i rätt tid.

A→B:

För att uppnå detta skulle vi ha uppdaterat vårt social contract med rollfördelning och inför varje sprint ha låtit produktägarna same Scrum Master göra sina uppgifter.

The agile practices you have used and their impact on your work:

A:

Under projektets gång har vi arbetat med flera agila metoder, exempelvis wireframing. Vi gjorde i början av projektet flera skisser över hur vi ville att vår app skulle se ut. Vi började med att rita på griffeltavla för att bestämma vilka vyer som vi ville ha med i applikationen och som en grov skiss över hur de skulle se ut.

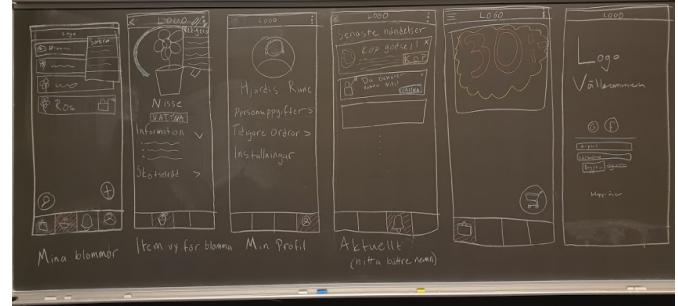


Bild 3: Skisser

Vi gjorde sedan ett acceptansflöde på papper över det flöde i applikationen som vi i som vi ansåg krävdes för att vi skulle uppfylla kursens mål samt att skapa det värde som Blomsterlandet ville ha (Bilaga 2). Detta gav oss en bra överblick över vad som var den viktigaste funktionaliteten och hjälpte oss att prioritera våra user stories.

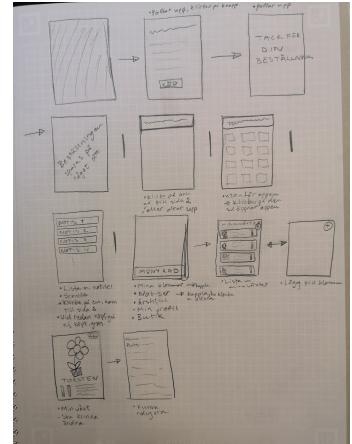


Bild 4: Acceptansflöde
på papper

Slutligen skapade vi en förbättrad version av acceptansflödet (Bilaga 3). Vi fortsatte att använda det som grund för hur vi prioriterade user stories och började också använda den för att bocka av vilka vyer/funktioner som var färdiggjorda. Detta var väldigt användbart under de sista sprintarna då det blev väldigt tydligt hur mycket och vad som fanns kvar att göra.



Bild 5: Acceptansflöde uppdaterat med alla flöden

Värt att notera är att det är ett par skärmar i acceptansflödet som vi inte hunnit implementera eftersom vi inte visste hur lång tid det skulle ta. Detta gör dock att vi skulle ha något att gå efter om vi skulle göra en till sprint och fortsätta utvecklingen.

Under projektets gång har vi arbetat med versionshantering med hjälp av Git. Detta har gjort att vi kan arbeta effektivt med parallella påbyggnader av koden.

Under hela projektet har vi använt oss utav parprogrammering, bortsett från de sprints då vi inte producerade någon kod. Vi försökte para ihop personer som hade kompletterande kunskapsprofiler för att bilda cross-functional teams även inom programmeringsparet vilket möjliggjordes av att projektgruppen utgjordes av studenter från olika program, därför bestod alla grupper av en person från IT och en från Industriell Ekonomi. Parprogrammeringen har gjort att problem som uppstod lösades snabbare och att vi fick olika perspektiv under utvecklingens gång. Vi var inte heller lika beroende av vår tekniska utrustning. Vi insåg dock att vi var tvungna att koordinera paren emellan så att inget dubbelarbete utfördes. Vi var noga med att sätta parets markering på de kort de hade ansvar över på scrum-boarden. Ett ytterligare problem som uppstod var att paren utvecklade olika specialiteter. Detta gjorde att paren blev beroende av kunskap som de inte själva besatt. Vi arbetade hårt för att främja arbete över parens gränser och vi kom snabbt in i ett effektivt arbetssätt.

Vi har under hela projektets gång försökt arbeta med Timeboxing för att minimera tidsslöseri. Bland annat har vi använt oss av det då vi skrivit våra team reflections. I början märkte vi att det tog väldigt mycket tid för oss att skriva våra team reflections, men efter att vi började med Timeboxing blev vi mer effektiva. Vi har arbetat för att uppnå sustainable pace där vi har försökt planera vårt arbete för att inte behöva arbeta övertid. Tack vare att vi använde oss utav god planering kunde vi färdigställa ungefär så mycket som vi ursprungligen tänkt oss. Vi var dock noga med att inte låta planeringen gå före ett eventuellt behov av förändring. Under projektets gång insåg vi att det var omöjligt att planera allting i förtid då vårt scope ändrades flera gånger. Det var viktigare för oss att skapa en framgångsrik produkt än att följa vår planering.

Under projektets gång har vi arbetat med relativt korta sprintar och applicerat principen short iterations. Vi har haft som mål att萌gra in producerad kod i master branch så snabbt som möjligt efter att det färdigställts. Att vi萌grat in vår nya kod så snabbt som möjligt kan ha gjort att vi undvek större merge conflicts och att alla hängde med i utvecklingen men det möjliggjorde också snabb feedback både inom project team och mot handledare och produktägarna.

B:

I ett nytt projekt måste vi vara bättre på att uppdatera våra acceptansflöden. Som man kan se i våra flöden ovan så är bilderna baserade på skisser som skapades vid början samt mitten av projektet. Slutprodukten har ungefär samma sorts flöde som våra bilder, men utseendet stämmer ej överens, vilket kan bli förvirrande.

Vi hade också behövt ha tydligare gränser när vi har daily scrum, review, retrospective och sprint planning. Vi har under projektet genomfört alla delar men

de har ibland flutit samman och rörts ihop. Något som kan ha bidragit till att de inte utnyttjats till sin fulla potential

A→B:

För att uppnå detta skulle vi, när vi börjar känna att en vys utseende har uppdaterats/blivit klart, tagit en bild av detta och byta ut bilden i acceptansflödet mot den. Detta gör också att vi måste uppdatera de framtida delarna i flödet så att de stämmer någorlunda överens med den nuvarande produkten.

Vi hade också under våra möten gjort tydligt när vi går över från en aktivitet till en annan, till exempel genom att göra daily scrum stående och timeboxa våra reviews och retrospectives.

The sprint review and how it relates to your scope and customer value (in terms of your meetings with the product owner):

A:

I början av projektet hade vi dålig koll på vad som hade gjorts under sprintarna då vi inte avsatte tid för att aktivt utvärdera arbetet. Vi var alla fokuserade på våra egna arbetsuppgifter och vad vår kund Blomsterlandet önskade. Lite senare i projektet införde vi produktägare. Produktägarna hjälpte oss att driva arbetet framåt och se till att produkten hela tiden genererar värde.

När vi sedan började införa mer ordentliga sprint reviews med en tydligare struktur och syfte senare i projektet så fick vi alla ännu bättre koll på hur vi låg till i utvecklingen och blev mer samspelta. Vi ansåg att det viktigaste med en sprint review var att få veta status för de inplanerade sakerna, det vill säga vad som är klart eller inte och kunna koppla detta till projektbackloggen. Även att se över vad som ska göras näst är en viktig funktion. Att ta reda på vad som gått bra samt mindre bra är ett förvillkor för en förbättrad arbetsprocess.

Något vi aldrig använde oss av på ett ordentligt sätt genom projektet var demonstration under sprint reviews, något som vi i början tyckte var överflödigt och sedan inte tänkte på. I efterhand kan man se problem med att inte göra rejälä demonstrationer eftersom det kan leda till att parprogrammeringspar banar iväg åt ett eget håll utan faktisk input från gruppen. Det hade nog varit bra att köra demonstrationer av ens leveranser för att förtydliga den gemensamma processen och hålla sig så samstämmiga som möjligt som grupp. Detta var dock inte något som vi, som tur var, fick problem med under projektet.

B:

Vi hade velat ha en bättre struktur på våra sprint reviews till nästa projekt. När man utvärderar det som gjorts kan det vara hjälpsamt att följa någon typ av mall. På samma sätt kan det vara lämpligt att ha tydligare roller än vad vi har haft under det här projektet.

Vidare vill vi bli bättre på att göra riktiga sprint retrospectives. Vi var bra på att säga vad vi gjort och bra på att diskutera hur processen hade gått, men inte lika bra på att säga vad som kunde förbättras.

Någonting vi hade velat göra under nästa projekt är att demonstrera det som gjorts under sprinten på sprint reviewen. Det är viktigt att demonstrera det som gjorts, särskilt i början när projektet formas som mest. Vi pratade främst om vad som gjorts under sprinten, det gjorde att vissa inte förstod till fullo vad de andra hade arbetat med.

Under kommande projekt vill vi fortsätta att ha värde i fokus när vi utvärderar våra sprints. Vi är nöjda med hur diskussionen har förts även om den inte har varit särskilt strukturerad. Därför hoppas vi på många intressanta diskussioner inför vidare projekt.

A→B:

För att skapa en bättre struktur inför nästa projekt krävs det att vi avsätter tid och planerar hur vi vill att sprint reviewsen ska fungera. Vi måste ta fram en mall som kan användas för att gå igenom och utvärdera arbetet.

Vi måste ha en mall för hur sprint retrospectives ska fungera, precis som med sprint reviewsen. Vi måste dokumentera vilka områden som väljs ut under varje sprint för förbättring. Utan att konkretisera vad som önskas uppnås blir det svårt att avgöra om målet har nåtts.

För att bli bättre på att demonstrera det värde som skapats under sprinten måste vi avsätta tid för att gå igenom allas arbete.

Slutligen måste vi fortsätta diskutera värde när vi pratar om det arbete som ska utföras. Det är upp till var och en att påminna gruppmedlemmarna om vad som är det viktigaste.

Best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.):

A:

I början av projektet valde vi att använda oss av React Native för att utveckla koden. Vi lärde oss grunderna för React Native av en video tutorial på Udemy för att sedan komplettera med hjälp av varandra där det behövdes. I tutorialen utvecklade man en enklare app där vi lärde oss att skapa, hantera och styra komponenter. Genom att testa oss fram har vi fått lära oss om ramverket och kringliggande teknologier.

Fortsättningsvis insåg vi att vi behövde lära oss versionshantering. För att lära oss att använda versionshantering har vi använt oss av en interaktiv Git-tutorial som rekommenderades på kurshemsidan samt fått hjälp av andra gruppmedlemmar med mer erfarenhet. Vi har även kontinuerligt genom användandet av Git lärt oss ytterligare om teknologin och kunnat använda den inom projektet.

Under projektets framfart har vi alla stött på problem och har löst dessa delvis genom att ta hjälp av varandra men framförallt genom att använda sig av forum såsom stackoverflow där liknande problem och lösningar hittats. Genom det cross-functional team vi har haft och att vi har parprogrammerat har vi troligtvis minimerat problem eftersom vi hela tiden har haft olika perspektiv och kunnat bolla nya idéer hela tiden. Detta har troligtvis lett till att vi lärt oss mycket av varandra och fått ett effektivare gemensamt lärande.

Under senare delar av projektet var vi tvungna att lära oss implementera Redux för att på ett effektivt sätt kunna hantera data. Delar av gruppen återvände till Udemy-tutorialen för att lära sig mer om denna typ av state management.

Vidare utvecklade vi en Scrum board för projektet. För att lära oss använda vår Scrum board på ett effektivt sätt baserade vi grunden på andras scrum boards för att sedan anpassa dem efter vårt projekt. Vi har dessutom fått hjälp och guidning av handledare för att förbättra vår Scrum board. De presenterade olika sätt för hur Scrum boarden kunde delas upp samt hur tidigare elever har gjort.

Vidare insåg vi att vi behövde lära oss mer om att skapa goda estimates för våra user stories. Under projektet började vi använda oss av planning poker för att estimera våra user stories, det fungerade bra. Vi upplevde att vi blev bättre på att estimera men det är svårt att avgöra om det enbart berodde på vår ökade erfarenhet eller för att vi började använda planning poker.

B:

Vi vill fortsätta lära oss av varandra i nästa projekt. Vi har sett att inlärningen går snabbare när gruppmedlemmarna har ett stabilt gemensamt underlag. Vi har

upplevt att inlärningen har fungerat mycket bra i detta projekt,, troligtvis till stor del på grund av att vi har arbetat metodiskt med samma material (t.ex. Udemy-kursen).

Vi hade velat få inspiration och lära oss av andras genomförande av projekt för att se andra möjliga tillvägagångssätt och metoder.

A→B:

För att fortsätta lära oss av varandrar så ska vi fortsätta parprogrammera när vi kodar. Vi vill även fortsätta dela lärorika klipp / kurser med varandra. Vi ska även söka upp andra projekt för att kunna studera dem.

Relation to literature and guest lectures (how do your reflections relate to what others have to say?):

A:

För att underlätta att använda Scrum på ett bättre sätt har vi läst Atlassians Agile coach Scrum Guide för att därifrån ta inspiration, speciellt våra Scrum Masters. Detta gjorde att gruppens roller blev tydligare och det underlättade användandet av Scrum tack vare konkreta exempel. Det vi tog oss an från denna litteratur var alltså hur man arbetar med dessa roller. Ett exempel där vi använt oss av tips från guiden är att en Scrum Master ska se till att utvecklingsteamet ägnar sin tid åt rätt saker och att teamet inte störs av andra idéer eller problem ”utifrån”. När vi applicerade detta i vårt projekt så tycker vi att det gick bra och gruppens arbete flöt på väl.

Vi har samlat en del information inför utformningen av vår Scrum board och de epics, user stories och tasks vi har haft där. I ”The New User Story Backlog is a Map” skriver Jeff Patton om att det är viktigt att inte behandla sina Epics som något ouppnäligt, utan att man istället ska se på dem som kontext och som en ”ryggrad” till sitt projekt. Han skriver också att det är viktigt att gå igenom sin map för att testa det och få en överblick. Detta har vi använt oss av när vi skrivit våra user stories och speciellt våra epics, då vi skrivit våra epics så att de ger kontext snarare än att vara något ouppnäligt och detta är någonting vi är nöjda med eftersom det har gett projektet en riktning så som litteraturen påstod.

Något vi kontinuerligt diskuterat under arbetets gång är att ”slice the cake” på ett bra sätt vilket vi diskuterat under föreläsningar i kursen. Vi har försökt att skapa så vertikala user stories som möjligt och haft det i åtanke hela tiden. Det är någonting vi har tyckt varit svårt eftersom det frångår det tänket vi haft sedan tidigare, men det är någonting vi har strävat efter och jobbat med. Det som legat till grund för vårt problem är att när vi skapade stories vertikalt, exempelvis en enklare notifikationsfunktion, var att vi insåg att det tog väldigt lång tid att implementera även en enklare variant av den vilket ledde till att våra user stories tenderade att bli för stora när vi försökte hålla dem vertikala. Vi fick istället utgå från ett vertikalt tänk men splitta upp varje del till flera stories som var hanterbara över iterationerna.

I The Complete React Native and Redux Course på www.Udemy.com gjord av Stephen Grider, som är en professionell front-end utvecklare, går grunderna till React-native och Redux igenom. Kursen går igenom grundläggande koncept och visar hur man gör dem efter best practice. Alla gruppmedlemmar började första sprinten med att kolla igenom denna. Att alla utgått från samma utbildningsmaterial gör att alla lärt sig göra saker på samma sätt, vilket lett till en konsekvent kod i projektet.

Forumet Stackoverflow har varit till stor hjälp under projektets gång. På forumet finns en uppsjö frågor om diverse programmeringsrelaterade problem. Så fort man stött på något problem har google ofta hänvisat en till stackoverflow där det många

gånger funnits fungerande lösningar. Även githubs forum har använts på samma sätt. Från dessa forum har vi som sagt fått hjälp med specifika problem som uppkommit under projektets gång. Utöver dessa forum har även diverse artiklar och guider hjälpt gruppen vid behov.

Boken "Up & Going" i bokserien "You don't know JS" som några i utvecklingsteamet läst har givit en god grundläggande förståelse för javascript, vilket underlättade utvecklingen. Detta accelererade det tidiga inlärandet och underlättade gruppens arbete mycket då vi kunde fokusera på andra teknologier i Udemy-kursen, istället för att även samtidigt lära oss javascript utan tidigare förkunskaper. Så denna litteratur har gett oss en bra grund för utvecklandet i javascript vilket vi använt oss av under projektets gång.

B:

I kommande projekt vill vi applicera fler av de kunskaper och den information vi fått genom litteratur inom denna kurs. Främst när det handlar om de olika agila metoder vi utforskat så att vi kan implementera dessa för att underlätta och förbättra vår framtida process. Vi vill även lära oss mer om detta, teoretiskt såväl som praktiskt.

Vi vill i framtida projekt fortsätta att ha epics för att ge en större kontext som alla andra user stories bidrar till att uppnå.

I ett framtida projekt skulle vi vilja arbeta vidare med att utforma vertikala user stories för att nyttja de fördelar som presenterats kring det. Vi vill utveckla vårt tänk och sätt att se på värde för att kunna få till detta enligt de rekommendationer vi fått med oss av denna kurs.

A→B:

Vi måste aktivt fortsätta läsa på om dessa metoder och implementera dessa i vårt dagliga arbete för att fördjupa de kunskaper de ger oss. Då våra handledare aktivt förespråkar Kniberg, H. (2015) Scrum and XP from the Trenches, så är det nog bra att ha denna boken i åtanke som utgångspunkt till framtida projekt.

För att fortsätta ha epics som ger kontext är det viktigt att lägga tid på dessa i början av projektet för att fånga upp essensen av målen i projektet. Så att man under projektets gång kan förhålla sig till dessa och hela tiden röra sig i rätt riktning.

För att kunna utforma user stories på ett mer vertikalt sätt så skulle vi behöva lägga mer aktiv tid på utformningen av dessa i början och diskutera oss fram till bra lösningar så att de inte blir för stora och oanvändbara.