

# Final Team Reflection

Axel Hertzberg      Erik Anttila Ryderup      Erik Bengtsson  
Filip Hansson      Jonas Nordin      Oliver Österberg      Theodor Lyrheden

2 juni 2021

# **1 Kundvärde och omfattning**

Den här sektionen beskriver hur gruppen begränsat arbetets omfång och skapat värde för kunden.

## **1.1 Programmets omfattning, prioritering av funktionalitet och för vem värde har skapats**

### **Hur har det varit under projektets gång?**

Vi har skapat en bokningsapplikation för BRF Lemansgatan tillsammans med den externa intressenten Alexander Ybring som tillhör bostadsrättsföreningen. Applikationen fungerar som en hemsida där bostadsrättsmedlemmarna kan boka faciliteter. Funktionerna i webbapplikationen innefattar bokning av tvättstuga, gym och andra lokaler. De är framtagna efter de krav som Alexander ställt.

Tidigare hade bostadsrättsföreningen ett analogt bokningssystem på papper, men genom att digitalisera bokningsprocessen hänger föreningen med i tiden och värde skapas för de boende, däribland vår externa intressent. Ett konkret exempel på det är att många som i dagsläget söker bostad förväntar sig att kunna boka denna typ av faciliteter via ett webbgränssnitt. Ytterligare så skapas värde i att bostadsägarna inte behöver gå till källaren för att boka en facilitet. Gruppen har prioriterat implementation av bokningsfunktionaliteten och inloggningsfunktionalitet i applikationen.

### **Hur hade vi velat ha det?**

I framtida projekt skulle gruppen vilja ha en tydligare bild av prioriteringen av funktionalitet. Rimligtvis hade en tydligare prioritet kunnat göra hela arbetsprocessen effektivare och tillämpningen av scrum enklare. Det hade blivit lättare att fokusera på de mest relevanta delarna i rätt ordning. Det kan då bli problematiskt ifall den externa intressenten kommer med nya önskemål sent i projektet, vilket skedde i flera fall under projektets gång. Det gjorde att vissa saker i implementerades i ”fel” ordning vilket innebär att befintlig funktionalitet behövde ändras.

Vad gäller programmets omfattning går det att konstatera att det har varit ett rimligt åtagande och därför anser projektmedlemmarna att det inte hade behövts justeras ifall projektet skulle göras på nytt.

### **Vad ska vi göra för att få det som vi vill ha det?**

För att få en tydligare bild av vilken prioritet särskilda funktioner ska ha behöver gruppen ha en tydligare dialog med den externa intressenten, kanske till och med be denne att rangordna vad som är viktigast. Det är något som gruppmedlemmarna kommer fundera på

inför kommande projekt och förhoppningsvis leda till att det tydligare framgår vilka delar av projektet som ska prioriteras.

## **1.2 Mål för vad projektet vill åstadkomma**

### **Hur har det varit under projektets gång?**

Angående applikationen har huvudmålet varit att leva upp till de förväntningar som projektets stakeholder haft på slutprodukten. Efter slutlig utvärdering med kunden bedömdes detta mål vara uppnått. I termer av återkoppling till Team Reflection W5 för denna rubrik har vi fortsatt att kontinuerligt visa upp applikationen för vår kund. Ett av gruppens stora mål har varit att i så stor utsträckning som möjligt snabbt kunna visa upp lösningar kring kundens önskemål, som utvecklats under projektets gång. Två exempel på sådana önskningar under slutskedet av projektet är språköversättning till svenska och implementation av instruktionsvideo.

Distansundervisningen har inneburit både för- och nackdelar. En fördel är att vi snabbt kunnat sätta upp möten då det behövts, utan att alla ska ha möjlighet att ta sig till campus. En nackdel har varit den sociala kontakten, som trots Zoom-möten ibland känts något utebliven. Gruppmedlemmarna bedömer att exempelvis parprogrammeringen hade fungerat smidigare utan distansarbete, eftersom möjligheter att guida och ge tips åt varandra hade blivit enklare i verkligheten. Det har gått att handla via skärmdelning och att styra varandras datorer på distans, men det har upplevts som onödigt krångligt ibland. I övrigt upplever gruppen att samarbetet efter förbättringsåtgärder fungerat väl, vilket även stöds av

### **Hur hade vi velat ha det?**

Hade vi fått göra om projektet hade vi velat ha en tydligare struktur för planering och uppföljning med vår stakeholder. Vi bedömer att det hade varit fördelaktigt om samtliga gruppmedlemmar kunde närvara på alla möten med vår stakeholder, eftersom alla då hade varit mer insatta kring detaljerna i stakeholderns önskemål. Detta har dock inte alltid varit möjligt, men det är något vi hade velat uppnå till nästa projekt. Vi tror också att samarbetet hade kunnat bli ännu bättre om vi arbetat tillsammans med en ännu tydligare planering i början av projektet. Det hade gjort det ännu tydligare kring vilka user stories som hade behövts göras varje vecka, vilket framförallt hade gynnat de mest oerfarna gruppmedlemmarna med avseende på programmering.

### **Vad ska vi göra för att få det som vi vill ha det?**

Rimligtvis hade det blivit tydligare vad som behövdes göras varje vecka genom att skapa och implementera den långsiktiga planeringen tidigare än vi gjorde. Det dröjde någon vecka innan vi insåg att vi hade behov av veckovis planering, varför det hade varit fördelaktigt för samtliga gruppmedlemmar om den långsiktiga planeringen var något av det första vi gjorde innan projektet drogs igång. Att tidigt skapa en långsiktig planering är därför

något projektmedlemmarna tar med sig till nästa liknande projekt. Angående mötena med stakeholdern hade det även här blivit bättre om vi så tidigt som möjligt, vecka för vecka, planerat när mötet skulle äga rum, så att så många gruppmedlemmar som möjligt skulle få chansen att delta och därmed i detalj bättre förstå vad som efterfrågas i termer av customer value. Även denna planeringsprocess är något som gruppmedlemmarna tar med sig till nästa projekt, även om det ibland kan vara problematiskt att uppnå då kunden har begränsat med tid.

### **1.3 User stories**

#### **Hur har det varit under projektets gång?**

Arbetet med user stories startade med att kunden, Alexander, ställde krav på systemet. Dessa krav låg till grund för de user stories sedan formulerades av gruppmedlemmarna. Varje user story utformades med samma struktur. Först skrevs en kort beskrivning om hur user storyn skapade värde för användaren. Vidare beskrevs hur det gick att bekräfta att en user story var slutförd, samt en beskrivning om hur den ska kunna testas. Varje story bedömdes också på förhand gällande hur mycket ansträngning på en skala mellan ett och tio det skulle ta att implementera den, där tio innebar att det skulle krävas mycket arbete. Gruppen gav även varje story en prioritet i form av en siffra mellan ett och fyra där en etta innebar högsta prioritet.

Ett problem som uppstod var att user stories inte riktigt fångade den grundläggande arkitekturen som behövdes för att påbörja byggandet av produkten. Att ha så stort fokus på user stories ledde till en viss teknisk skuld då det ledde till att fokus var på att bygga saker som kunden faktiskt ser. Eftersom refaktorisering inte syns i den färdiga produkten och gruppen hade en deadline så skedde bara mindre refaktorisering. För vidare utveckling av applikationen hade det varit nödvändigt med större refaktorisering för att lätta på den tekniska skulden. Ett annat problem var att våra user stories varierade väldigt i omfång. Vissa innebar en hel veckas arbete samtidigt som andra gick betydligt snabbare att utföra.

#### **Hur hade vi velat ha det?**

Det är framförallt två saker vi borde gjort för att ha åtgärdat detta problem. Första saken handlar dels om att skapa user stories med mindre omfång då vissa var för övergripande, till exempel hade man säkert kunnat dela upp vissa av våra user stories. Andra saken handlar också om att bryta ner respektive user story till olika uppgifter. Det hade gjort det tydligt redan innan man börjar arbeta vad som behöver göras och då hade det också blivit tydligare om någon user story var för stor, vilket då kunde åtgärdats genom att bryta ner den ytterligare.

I framtiden skulle det vara bättre att inkludera den externa stakeholdern ännu mer i utformandet av user stories, så de på ett bättre sätt speglar det som kunden faktiskt vill ha. Acceptance tests skulle gruppen själv göra vilket var ett stort misstag eftersom det hade varit bättre att

kunden fick ge synpunkter direkt. När en user story väl formuleras i framtiden så hade det även underlättat ifall det på ett tydligare och mer konkret sätt står hur just den funktionaliteten ska kunna testas.

#### **Vad ska vi göra för att få det som vi vill ha det?**

För att förbättra user stories skulle gruppen behöva forma acceptance tests tillsammans med kunden. Genom att ställa fler frågor och ha längre samtal om vad kunden vill ha och vad som förväntas skulle acceptance test kunna utformas mer utförligt. Det hade också som diskuterats ovan varit bra att ha brutit ner våra user stories till mindre delar vilket hade underlättat arbetsfördelningen. För att få mer kontroll på testningen av varje user story är en idé att föra in en mer tydlig beskrivning av vad som förväntas under rubriken **”testas genom...”**.

### **1.4 Acceptanstest och dess värde**

#### **Hur har det varit under projektets gång?**

Varje vecka visades förgående veckas user stories upp för den externa stakeholdern. Detta gjorde att kunden kunde ge synpunkter löpande. Dessutom fick kunden överblick över vilka user stories som avklarats respektive inte avklarats varje sprint.

#### **Hur hade vi velat ha det?**

Gruppen var mycket missnöjda med det här sättet att jobba eftersom det var oklart när funktioner blev klara. Vidare så var det svårt att få tillbaka respons varje vecka som sedan behövde ändras. Det hade varit mycket bättre att ha klara mål på hur det ska vara från början.

#### **Vad ska vi göra för att få det som vi vill ha det?**

Utforma acceptance tests tillsammans med kunden. Detta så att inte utvecklingsteamet tolkar kraven som kunden ställer.

### **1.5 KPI som används och hur de förbättrar processen**

#### **Hur har det varit under projektets gång?**

I början av projektet kom gruppen överens om att använda 3 olika kpi:er, välmående, burn-up chart och velocity.

Ett problem som uppstod var att reflektioner och annat kring projektet också behöver uppskattas eftersom tid måste användas på det. Speciellt tydligt har det varit med rapporten till projektet. Eftersom den inte varit med i planeringen så syns inte den tiden som användes för att skriva rapporten. Detta är alltså ett problem med de KPI:er som vi har använt oss av, att de inte tagit hänsyn till arbete som inte haft något värde för kunden. Ett annat problem som

funnits är att uppskattningen av den effort som krävs för respektive user story inte alltid stämte med hur lång tid det faktiskt tog att slutföra dem. Detta har också bidragit till att mycket arbete har utförts inte syns i våra KPI:er.

De KPI:er som har använts har kompletterat varandra bra och fyllt olika syften. Vårt burn-up chart har gjort det enkelt att se hur långt det är kvar tills att projektet är färdigställt vilket är viktigt för att kunna veta att projektet ligger i fas. Detta är även något som vårt velocity-diagram har hjälpt oss med. Genom att varje vecka kunna se hur mycket som gruppen åtagit sig och hur mycket av detta som färdigställts, något som varit behjälpligt vid sprintplaneringen. Med kännedom om eventuell backlog har nästkommande sprint kunnat justeras därefter. Slutligen har vårt välmående-KPI också varit viktigt. Även om det inte har tagit så stor plats i vårt projekt så har vi med den kunnat säkerställa att gruppens arbetsbelastning fungerat och det inte varit någon som mått dåligt på grund av projektet.

Resultatet av Burn up diagrammet syns i figur 1. På figuren så syns det att den totala mängden arbete ökar på grund av att kunde lägger till ny funktionalitet. Dessutom syns det att gruppen kommer närmare och närmare att färdigställa produkten.

Resultatet av Velocity diagrammet syns istället i figur 2, figuren visar att gruppen gick upp i arbetstakt i början av projektet tills vi nådde en bra arbetsnivå och sedan trappade av i slutet av kursen.

### **Hur hade vi velat ha det?**

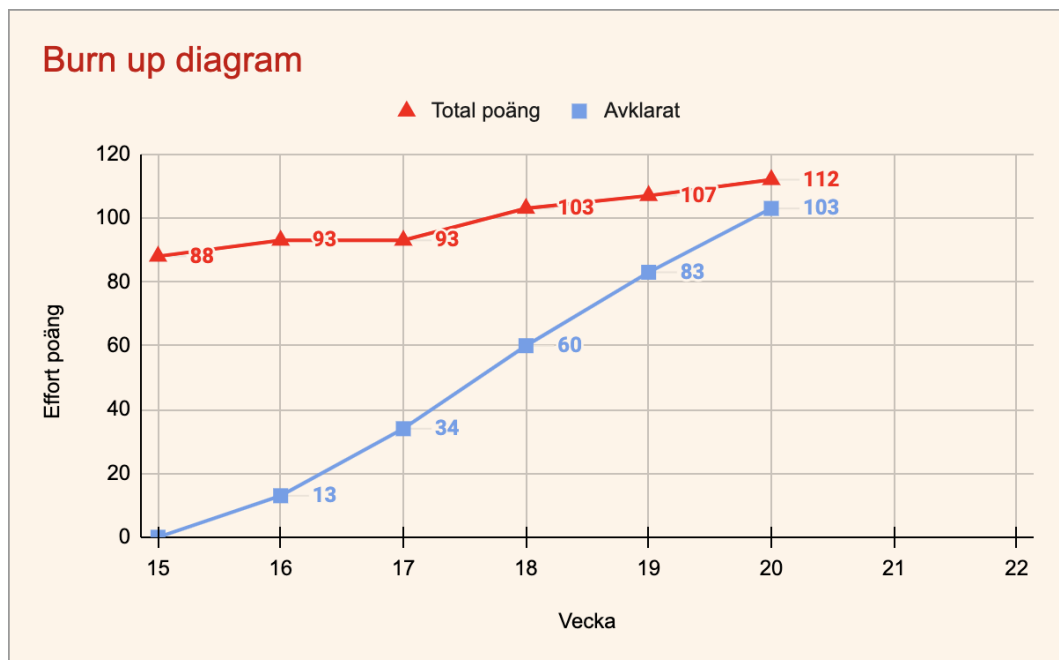
Gruppen är splittrad i frågan om KPI:erna var bra eller inte. Vissa upplever att insamlande och sammanställning av informationen i KPI:erna är slöseri med tid. Andra tycker att det varit ett viktigt verktyg. Det mest optimala hade varit att ha KPI:er som var tydligt kopplade till vår applikation. På så sätt hade det varit enklare och mindre tidskrävande att uppdatera dessa respektive vecka. Absolut bäst hade varit med KPI:er som uppdaterades automatiskt när framsteg i projektet gjordes.

### **Vad ska vi göra för att få det som vi vill ha det?**

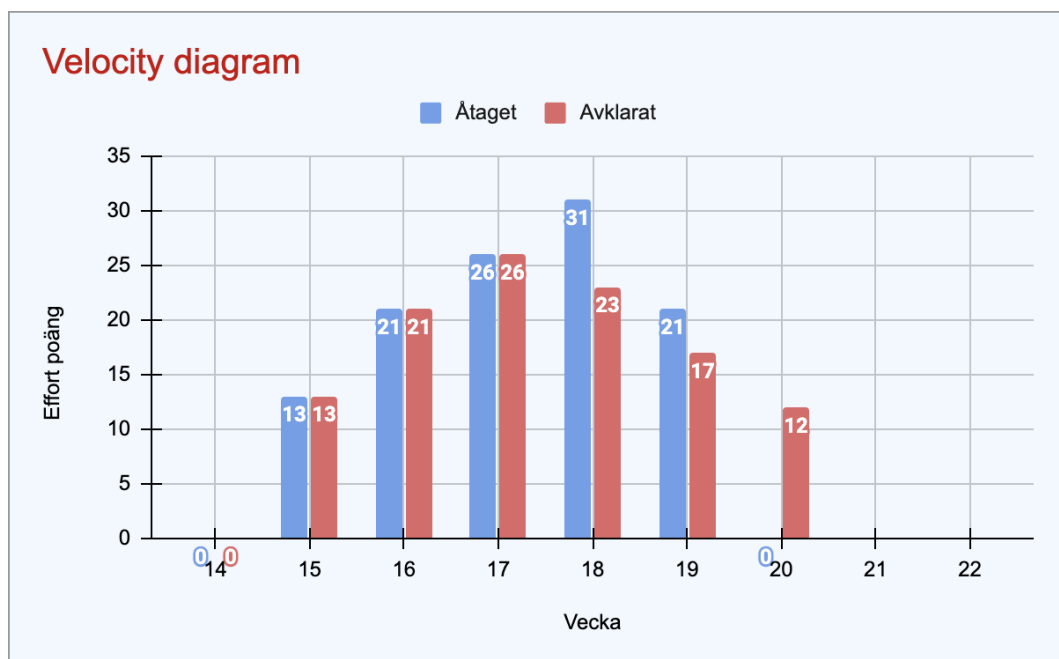
För att minska tiden som används på KPI:er så skulle det vara bra om de kunde vara bättre utformade för att dataanalysen ska ta mindre tid. Att ha en user story om välmående som bara används om någon upplever problem i gruppen är ett sådant verktyg. Det går att ställa in på Google forms att den skickar ett mejl när någon svarar. Då skulle det gå att ställa in en form som används om någon i gruppen känner av ett problem istället för varje vecka.

Uppskattning av svårighetsgrad på user stories skulle behöva uppskattas bättre för att kunna ha större värde i velocity och burn-up charts. Gruppens bild är att om informationen var mer relevant skulle de vara viktigare att ha med. För att få en bättre uppskattning av tiden behövs det läggas mer tid på det. Att verkligen fundera över vad som behöver ske kodmässigt för att user storyn ska fungera.

För att lösa problemen med att reflektion och administrativt utöver själva kodningen skulle developer stories kunna införas och sedan räknas med i KPI:erna.



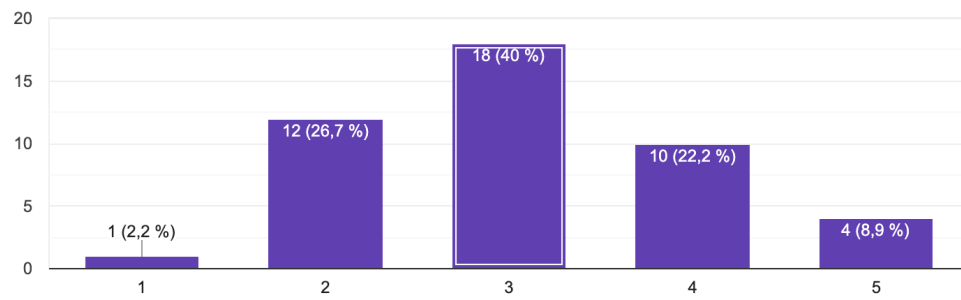
Figur 1: Burn up diagram



Figur 2: Velocity diagram

Hur upplever du din egen stressnivå under veckan?

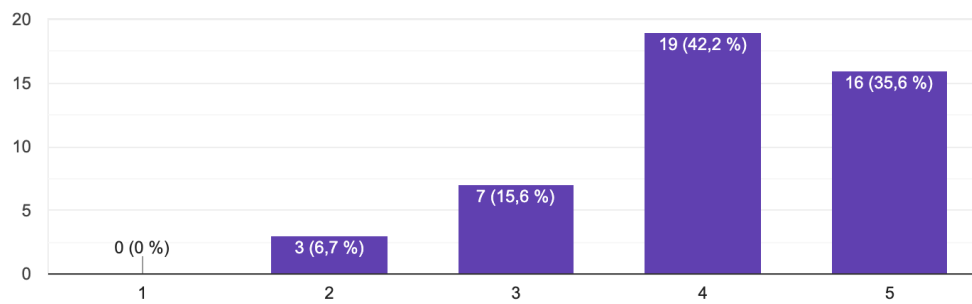
45 svar



Figur 3: KPI stressnivå

Hur upplever du att samarbetet i gruppen har fungerat?

45 svar



Figur 4: KPI Samarbete



## **2 Socialt kontrakt och nedlagd arbetstid**

Här beskrivs det sociala kontraktet och den tid som är nedlagd på kursen.

### **2.1 Gruppens sociala kontrakt**

#### **Hur har det varit under projektets gång?**

Gruppen skrev tillsammans under den första veckan av kursen ett socialt kontrakt som sedan inte uppdaterades mer. Syftet med det sociala kontraktet var att gruppen skulle komma överens och bestämma en del riktlinjer för arbetet. Exempel på detta var gruppens förväntningar på arbetet, hur kommunikationen i gruppen ska fungera och hur eventuella konflikter ska lösas. Att skriva det sociala kontraktet så pass tidigt i projektet gav en bra start och möjligheten för gruppmedlemmarna att lära känna varandra. Övergripande så har de stora delarna i det sociala kontraktet följts, även om alla medlemmarna i gruppen är överens om att gruppen vid vissa tillfällen hade kunnat vara bättre på att hålla tider. Då gruppen fick allt arbete gjort i god tid och haft ett samarbete så blev inte detta ett särskilt stort problem.

Andra saker som gruppen varit sämre på att efterfölja i det sociala kontraktet utöver förse-ningar har varit schemaläggning, beslutstagande och stresshantering.

Problemet med schemaläggningen var främst det inte gjordes en strukturerad övergripande schemaläggning från början, vilket stundtals gjorde att det blev svårt att planera in gemensamma arbetstillfällen.

Gruppen hade satt upp ett antal punkter kring hur beslut skulle tas, vilket bara efterföljdes delvis. Enligt det sociala kontraktet skulle de flesta beslut tas gemensamt via röstning, men det var något som visade sig vara ineffektivt i praktiken.

Att gruppmedlemmarna själva skulle ansvara för att få en jämn fördelning av arbetsbördan var inte heller en så bra idé i efterhand. Till viss del berodde detta på att user stories, se 1.3, ibland var väldigt stora och därför skapade en stor arbetsbörda för den individ som tagit på sig en så stor uppgift. Arbetsbördan diskuteras vidare i 2.2.

#### **Hur hade vi velat ha det?**

Även om det sociala kontraktet inte användes särskilt mycket så har gruppen tyckt att det har funnits en trygghet i att ha en överenskommelse i botten av projektet som alla har ställt sig bakom. Om projektet skulle göras om hade det sociala kontraktet definitivt behållits, samtidigt som det hade varit bra ifall gruppen hade lagt ned mer tid på att fundera kring vilka situationer utöver de som stod i färdiga mallen som skulle kunna uppkomma. Då hade kontraktet kunna bli ännu mer relevant. En annan del som gruppen missade var att hålla kontraktet uppdaterat. Det hade exempelvis kunnat vara en stående mötespunkt där en fråga kring kontraktet togs upp och diskuterades kring. Det är något som gruppen tar med sig till nästa projekt.

### **Vad ska vi göra för att få det som vi vill ha det?**

Genom lägga till sociala kontraktet som en återkommande punkt på veckomöten tror gruppen att det hade kunnat hållas mer relevant och uppdaterat med vad som hände i projektet.

Att viss svårighet med scheman förekom var inte helt oväntat då deltagarna var från två olika klasser. Det hade gruppen kunnat vara mer förberedda på, och haft en diskussion om schemaläggning och planering i samband med att kontraktet skrevs. Även det är något som gruppen känner att de kommer vara mer tydliga med till nästa projekt.

## **2.2 Tid spenderad på kursen och hur det relaterar till den levererade produkten**

### **Hur har det varit under projektets gång?**

Med tanke på projektets omfång och de 6 - 7 veckor vi haft för att genomföra projektet har förväntningarna fått justeras därefter. Detta gjordes i ett väldigt tidigt skede för att kunna hålla en jämn och passande arbetstakt genom hela projektet. Detta har inneburit att gruppen i huvudsak endast behövt arbeta på vardagar, vilket också var en punkt som togs upp i det sociala kontraktet. Ser man även till vår stress-KPI så har arbetsbelastningen legat på en lämplig nivå. Att gruppen huvudsakligen arbetat på vardagar har förmodligen bidragit till att hålla stressnivåerna på en bra nivå, då gruppmedlemmar kunnat hämta andan på helger. Dessutom så har förmodligen sprintplaneringen och gruppens goda samarbete hållit nere stressnivåerna. Att samtliga bidragit med en god arbetsinsats har vidare påverkat detta positivt.

Tiden som användes till projektet varierade vecka för vecka om man kollar på gruppen.

### **Hur hade vi velat ha det?**

### **Vad ska vi göra för att få det som vi vill ha det?**

### **3 Designval och produktstruktur**

Den här sektionen beskriver de designval vi fått göra och varför vi gjort dem.

#### **3.1 Hur projektets designval stödjer kundvärde**

##### **Hur har det varit under projektets gång?**

För att bygga programmet användes flera olika bibliotek med ett fokus på att snabbt visa upp saker. React för att snabbt få upp något på skärmen som faktiskt syns. För att bygga programmets server i java användes Spring-Boot därför att alla gruppmedlemmar redan kunde java och det därför gick snabbt att skriva i det språket.

Biblioteket React Bootstrap användes för att snabbt skapa ett fint gränssnitt. Detta gav värde för kunden för att utvecklingstid inte behövde läggas på att skapa grundläggande gränssnittskomponenter.

Istället för att lägga en hel del tid på att skriva en egen kalender hittade gruppen en som redan var färdig, React-Calender [1].

##### **Hur hade vi velat ha det?**

Det problemet som uppstod var att React inte var något som gruppen använt förut. I framtiden skulle det gått snabbare och skapat värde för kunden att använda något som gruppmedlemmarna redan kunde. Att använda en färdig kalender hade stora brister i att den inte var utvecklad för projektets ändamål. Detta ledde till vissa uppoftningar i användargränssnittet. Ett exempel där det fungerade bra att använda ett färdigt bibliotek är React Bootstrap.

Dessutom ville kunden ha en webbsida så det gick inte att använda ett programmeringsspråk som gruppmedlemmarna redan använt.

##### **Vad ska vi göra för att få det som vi vill ha det?**

I det stora hela så är gruppen nöjd med tanken att skapa värde snabbt för kunden. Att se något på skärmen är trots allt viktigt för kunden.

För att undvika att bli begränsade av de val av färdiga lösningar som används ska gruppen i framtiden mer noga kontrollera så att lösningen faktiskt stödjer användningsområdet.

#### **3.2 Teknisk dokumentation som används och varför**

##### **Hur har det varit under projektets gång?**

Den tekniska dokumentationen av applikationen har samlats i ett övergripande System design document. Där presenteras en generell överblick och beskrivning av applikationen, kodstrukturen i form av ett enklare UML-diagram där även domänmodellen kan utläsas,

applikationens grafiska gränssnitt, hur datalagringen hanteras, en genomgång av säkerheten samt hur vi utför tester under utvecklingens gång.

Dokumentet har under projektets gång uppdaterats då nya funktioner eller större refaktoreringar gjorts i koden. Anledningen till användningen av denna typ av dokumentation är att den ger ett värde, inte bara för oss, utan även för gruppens stakeholder. För oss internt har dokumentet främst hjälpt oss att enkelt få en övergripande blick av strukturen i programmet, både kod- och designmässigt och på så sätt tidigt kunna upptäcka och korrigera brister med applikationen under utvecklingen. Dokumentet fungerar utöver det som ett hjälpmedel att kommunicera med och få feedback från stakeholder, där stakeholder enkelt kan se vad som implementeras eller ändras i applikationen under projektets gång.

Dokumentation har även gjorts till viss del direkt i koden i form av Jsdocs. Dokumentation i Jsdocs format var något som började göras efter ett par veckor och skrevs hela vägen till projektets slut.

### **Hur hade vi velat ha det?**

Hade vi fått chansen att göra om projektet eller att göra ett liknande projekt i framtiden hade större hänsyn tagits till den tid det faktiskt tar att göra teknisk dokumentation. Större fokus hade framförallt lagts redan i projektets start på att i samband med att vi lärde oss det nya språket och ramverket även lära oss i detta fall hur man på rätt sätt dokumenterar i Jsdocs format. Dels för att vi inte efter ett par veckor in i projektet ska få en mindre chock över hur mycket dokumentation vi måste klämma in i planeringen utan även för att dokumentationen, om den görs från start, ger oss ett betydligt effektivare samarbete i gruppen då skriven kod blir mycket mer lättförståelig och lättanvänd för näste som ska vidareutveckla eller använda den redan skrivna koden på annat sätt.

Utöver hur vi hanterat Jsdocs är vi i gruppen nöjda med kvalitén på projektets System design document. Vi har under tiden även insett vilket värde som tillkommer både internt och externt genom att skriva ett sådant dokument. I ett framtida arbete skulle vi absolut använda ett liknande dokument för teknisk dokumentation.

### **Vad ska vi göra för att få det som vi vill ha det?**

För att i ett liknande projekt i framtiden uppnå en högre kvalitet på den tekniska dokumentationen och tidigare i projektet kunna använda de fördelar som kommer med den skulle vi behöva redan från start lägga upp en plan där vi beräknat hur mycket tid det tar och hur man kan lära sig om i detta fall Jsdocs. Den planen skulle sedan räknas med i den mer övergripande interna projektplanen vilket skulle ge alla gruppmedlemmar en tydligare bild av vad som väntas av var och en och när under utvecklingen av applikationen, främst under kodning. Genom att göra detta skulle vi slippa att chockas av hur mycket som behöver dokumenteras ett par veckor in i projektet och skulle utöver det få ett effektivare samarbete och kodande.

### **3.3 Hur dokumentation används och uppdateras under sprinter**

#### **Hur har det varit under projektets gång?**

Teknisk dokumentation under projektets gång har gjorts som nämnt tidigare i form av Jsdocs och i ett System design document. Jsdocs började vi skriva efter ett par veckor efter vi insåg hur kodbasen växte till den punkten där det blev svårt att hela tiden hålla sig uppdaterad på all kod i projektet under utvecklingen. System design dokumentet har uppdaterats då större ändringar eller refaktoreringar i koden implementerats.

Utöver den tekniska dokumentationen har även dokumentation förts och uppdaterats kontinuerligt varje vecka med en scrumboard online som kallas Trello. Scrumboarden har vi gått igenom varje måndag för att följa upp förgående vecka och för att planera inför kommande vecka. Det är genom denna vi delegerar userstories och andra uppgifter till gruppens medlemmar för att hålla koll på vem som gör vad och när man kan vänta sig att det blir klart.

I slutet av varje vecka har även en så kallad Team reflection och en Individual reflection skrivits för att både i grupp och individuellt reflektera över veckan som gått.

För att stödja planeringen av applikationen skrevs också en långsiktig planering med planer för vad som skulle bli gjort varje vecka. Fördelen med en planering som alla gruppmedlemmar kommit överens om är att det då finns tydliga mål för när saker ska vara gjorda. Vidare är det bra att förhålla sig till vid planering av kommande sprint.

#### **Hur hade vi velat ha det?**

Hade vi fått chansen att göra om projektet eller att göra ett liknande projekt i framtiden hade större fokus lagts på att oftare uppdatera både System design dokumentet och Jsdocs, så att uppdateringar sker redan efter mindre ändringar i programmet, främst i Jsdocs. Hade vi gjort det hade vi förmodligen kunnat förbättra och effektivisera samarbetet i gruppen eftersom den dokumentationen internt då kan användas för att lätt se och förstå de senaste ändringarna i programmet. Det skulle därmed bli lättare att kritisera och utvärdera ny kod och nya designval vilket skulle leda till en mer robust kodbas med mer genomtänkta designval.

I ett framtida projekt skulle vi även se till att oftare uppdatera och framförallt hålla oss själva uppdaterade om Trello för att veta vem som gör vad och vad som behöver göras innan man sätter sig och börjar arbeta på projektet. Gör alla i gruppen det minskar man risken för att något görs dubbelt på två olika håll eller motsatsen att något glöms bort att göras och upptäcks i efterhand. Med andra ord skulle man lyckas följa sprinternas planering bättre under projektets gång och då få en jämnare arbetsbörda från start till slut.

En långsiktig planering av hur applikationen successivt ska byggas tycker gruppen ger stort värde och har fungerat bra då den hjälper mycket i planeringen och struktureringen av hur

kommande sprinter behöver läggas upp för bästa möjliga slutresultat. Något som absolut skulle användas i ett framtida projekt.

#### **Vad ska vi göra för att få det som vi vill ha det?**

För att hålla dokumentationen uppdaterad inför ett tydligt krav i Definition of done att koden ska vara uppdaterad och sen neka ändringar som inte håller sig till detta krav. Vidare så behöver tid också avsättas och schemaläggas till dokumentation.

För att hålla sig uppdaterad i Trello bör ett större krav ställas på individer i det sociala kontraktet att det förväntas att kontrollera om någon annan redan jobbar på uppgiften.

I ett framtida projekt ska det vara hög prioritet på att skapa en långsiktig planering tidigt så att den sedan kan användas för att få fördelarna från förra sektionen.

### **3.4 Hur kodkvalité säkerställs och kodstandard hålls**

#### **Hur har det varit under projektets gång?**

Under projektets gång såg vi att det behövdes en kodstil så att all kod ser enhetlig ut. Detta löstes genom att installera StandardJS som automatiskt formaterade koden. För att tvinga gruppen att använda den här kodstilen användes den kontinuerliga integreringen för att automatiskt köra en kontroll att standarden följdes. En fördel med den här lösningen var att stilkontroll inte glömdes. En fördel med att använda samma kodstil var att merge problem på grund av att någon formaterat koden i sin IDE inte skedde. Dessutom så behövdes inte någon annans kodstil tolkas.

Vi har mest testat visuellt och ett några kontinuerliga tester på viktig gemensam funktionalitet.

#### **Hur hade vi velat ha det?**

Vi hade velat arbeta test-drivet. En anledning till att vi inte har gjort det är att vi på förhand hade för dålig kunskap kring tester i JS/React.

Att använda kontinuerlig integration för att automatiskt kontrollera koden var en bra lösning som gruppmedlemmarna även kommer att använda i framtiden.

I ett framtida projekt skulle det varit bättre att undvika visuella tester.

#### **Vad ska vi göra för att få det som vi vill ha det?**

För att arbeta så som vi hade önskat hade vi dels behövt inkludera tester DoD, varit bättre på att faktorisera ut gemensam kod ifrån komponenterna till gemensamma funktioner för att testa "centralt". Detta gör även så att visuella tester kan undvikas

## 4 Tillämpning av Scrum

Den här sektionen beskriver gruppens användning av Scrum.

### 4.1 Gruppens roller och dess påverkan på arbetet

#### Hur har det varit under projektets gång?

Under projektets gång har rollerna varit desamma som bestämdes under den första veckan. Erik Ryderup har haft den mest centrala rollen i egenskap av scrum master, vilket inneburit att han har haft ansvar för varje veckas sprintplanering och sprint review.

Övrigt nämnbara roller har varit att Axel Hertzberg har fungerat som produktägare då han har varit mest insatt i domänen och skött det mesta av kontakten med projektets stakeholder. Att ha en produktägare på utvecklarens sida är möjligtvis lite okonventionellt men det upplägget passade både gruppen och stakeholder bra.

Projektet började således med att Axel genom möte med stakeholder satte sig in i domänen. Så fort Axel blev insatt i domänen kunde han hjälpa projektets stakeholder att formulera domänkrav, som gruppen senare formulerade user stories utefter.

Filip Hansson har varit ansvarig för att mäta och utvärdera KPI:erna. Det har varit en kontinuerlig process som varje vecka har presenterats för gruppen i samband med sprintreflektioner. Även övriga gruppmedlemmar hade roller men i slutändan flöt många utav rollerna ihop, vilket kanske inte är så konstigt. Till en viss del har den otydliga rollfördelningen i gruppen gjort att vissa har fokuserat mycket på kod och andra mer på reflektion.

#### Hur hade vi velat ha det?

Då medlemmarna i gruppen kommer från olika program är det naturligt att intressen, styrkor och svagheter skiljer sig åt. Det gjorde att rollerna, utöver de som delades ut från början, naturligt fördelades mellan medlemmarna. Överlag är samarbetet något som medlemmarna enligt välmåendeformuläret, upplevt vara bra. Det hade varit lättare att fördela tydligare och mer relevanta roller på förhand. En diskussion kring vilka roller varje enskild medlem skulle ha fördes i början av projektet. Gruppen kom då fram till att roller helst skulle roteras, framförallt när det kommer till rollen som Scrum master. Så blev det dock inte, mycket tack vara att Erik R fungerade väldigt bra i denna roll. I efterhand så är gruppen överens om att ur ett lärandeperspektiv så hade det varit bra att rotera just denna roll så att alla kan testa på det.

Att rollerna blev suddiga desto längre projektet led borde varit något som gruppen hade kunnat förutse på förhand.

Idealt hade vi fördelat rollerna i gruppen så att arbetet av projektet blev så effektivt som möjligt, samtidigt som gruppmedlemmarna gavs möjlighet att lära sig nya saker inom både mjukvaruutveckling och agila arbetsmetoder.

Att vissa arbetsuppgifter inte var helt jämnt uppdelade mellan alla i gruppen behöver inte nödvändigtvis vara något dåligt. Dock hade det för både lärandets och arbetsfördelningens skull varit önskvärt att få gruppmedlemmarna i högre grad delaktiga i alla delar av projektet. Det hade också gett en tydligare bild på vad som förväntas av varje individ.

#### **Vad ska vi göra för att få det som vi vill ha det?**

För att undvika denna typ av frågor hade det varit mer fördelaktigt att ta upp denna diskussion på förhand. Både när det kommer till vad som förväntas av varje gruppmedlem samt redan på förhand kunna uppskatta vilka roller som kommer att vara relevanta i projektet. Vi borde också ha roterat rollen som scrum master trots att Erik fungerade såpass väl i denna roll i syfte att samtliga medlemmar skulle få prova på rollen och få en bättre förståelse för samtliga delar av att arbeta med scrum.

Ett sätt att uppnå mer rulljängs bland rollerna skulle vara att man redan i det sociala kontraktet definierar regler för när roller ska roteras. Det hade lett till att medlemmarna "tvingats" till att testa på olika delar.

## **4.2 Agila processer som använts och deras påverkan på arbetet**

### **Hur har det varit under projektets gång?**

I början varje vecka hade gruppen ett planeringsmöte där den kommande sprinten planerades. Planeringsmötet hjälpte till att hålla antalet uppgifter som kodoas samtidigt lågt, vilket hjälper till att hålla fokus på de uppgifter som ska göras den veckan.

Gruppen använde också Trello för att sätta upp ett scrum board. På gruppens scrum board märktes vem som skulle jobba på vad. I samband med veckans planeringsmöte visade Axel upp produkten.

I slutet på varje vecka hade gruppen ett sprint review möte. På det här möten kunde feedback på gruppens process ges direkt. Gruppen hade dock svårt med feedback delen för att det var svårt att se vad som skulle ändras.

I slutet på varje vecka leverades också en fungerande version till skillnad från vattenfallsmodellen vilket gjorde att kunden direkt kunde se arbetet och ge feedback. Under projektets gång hände det att kunden haft åsikter om vilka ändringar som skulle göras. Att direkt få synpunkter på arbetet var bra, då det blev lättare att undvika att lägga tid på saker som inte skapade värde till kund.

Ett mål som fanns var att minska tidsåtgången, med detta menas att inte sitta på möten i onödan. Detta gjorde att gruppen för det mesta endast planerade in mötena Sprint planning och Sprint review. Med det sagt så var det dock inte så att gruppen inte sågs mer än två gånger i veckan, utan gruppen hade ett starkt samarbete via mer spontana Zoom-möten. Där



jobbade teamet tillsammans och hjälpte löste problem. Även parprogrammering skedde via Zoom.

KPI:erna ”burn up chart” och ”velocity chart” användes för att hålla koll på att vi varje vecka når de mål som gruppen tillsammans med kunden kommit överens om. Mer om dessa kan läsas under avsnitt 1.5.

### **Hur hade vi velat ha det?**

De agila metoderna vi använt har hjälpt gruppen mycket och är något som vi tänker fortsätta med. Något som fattats som hade varit bra är testdriven utveckling. När koden var färdigskrivnen fick vi inte riktigt till tester. Något mer som hade varit bra är ett dagligt stand-up möte så att feedback kan komma mycket tidigare. Som det varit nu har man fått vänta tills på fredag för att komma med synpunkter om processen. Detta kanske har gjort att viktiga bidrag till processen har glömts bort.

### **Vad ska vi göra för att få det som vi vill ha det?**

För att få en löpande feedback så skulle dagliga stand-up möten kunna hållas. För att möjliggöra detta hade gruppen på ett tydligare sätt behövt göra en gemensam planering, där samtligas scheman slås samman och gruppen kan bestämma en gemensam tid för något sådant. Även om det hade varit det bästa så är frågan om det hade varit möjligt att få det att fungera med tanke på att gruppmedlemmarna är så pass många, läser olika kurser och inte har en avsatt arbetstid varje dag. Det gör att det hade varit svårt att få till möten varje dag, även om mötena är väldigt korta. Frågan är också hur mycket det verkligen hade gett, speciellt med tanke på att vi ofta satt i zoom och arbetade tillsammans och tog upp problem och funderingar löpande under arbetets gång.

## **4.3 Sprint review och hur den relaterar till projektets omfattning och värde för kunden**

### **Hur har det varit under projektets gång?**

Varje fredag hade gruppen en sprint review där den föregående veckans sprint diskuterades. På det mötet fick alla medlemmar i gruppen ge synpunkter på den föregående veckans sprint. En kontinuerlig utveckling av sättet som gruppen arbetar är värdefullt eftersom gruppen då kan få ett så effektivt arbetssätt som möjligt. Att utvärdera vad som fungerat bra och vad som fungerat mindre bra och ta med sig de bra delarna samtidigt som de mindre bra förbättras. Detta översätts till värde för kunden genom att slutprodukten kan bli klar snabbare och hålla högre kvalitet.

Det var scrum master som ansvarade för både utvärderings- och planeringsmötet. Vissa saker behövde gruppen förhandla med kunden efter det framkommit på sprint review att det varit för omfattande att få med.

Sprint reviewen hade en koppling till Definition of Done genom att de user stories som blivit avklarade förra veckan togs upp. Då kunde varje gruppmedlem se att de blivit avklarade.

Ett exempel på vilken feedback som framkom på möten var att gruppmedlemmarna tyckte det var svårt att veta vem som jobbade på vad. Detta var något som gruppen agerade på och införde att alla skulle notera vad de jobbar på så vi inte får situationen där flera jobbar på samma uppgifter.

Ett problem som upplevdes med mötet var att gruppen inte kom på så mycket feedback. Det är svårt att avgöra om detta är för att gruppen fungerade bra eller okunskap.

### **Hur hade vi velat ha det?**

Även i framtiden ser gruppen att sprint review är ett bra verktyg att använda. Det är viktigt för gruppen och dess välmående därför att det ger gruppmedlemmar en kontroll över processen som använts. Det är viktigt för kunden eftersom gruppen då kan få den mest effektiva arbetsmetoden och leverera produkten snabbare.

Att ha ett möte för sprint review är dessutom bra eftersom problem inom gruppen kan lösas snabbare. Sådana problem kan tas upp redan när sprinten avslutas.

Att ha en tydlig struktur på sprintreviewen med vilka punkter som tas upp på mötet skulle varit bra. Detta så att gruppmedlemmar skulle kunna förbereda sig till mötet och vad som ska tas upp.

Något som också upplevts är att det hade varit bra med en tydligare Definition of done så att det är enklare att markera saker som färdiga.

### **Vad ska vi göra för att få det som vi vill ha det?**

Om gruppen tillsammans tar fram en agenda för sprintreview och vilka punkter som ska vara med löses problemet med struktur eftersom alla kan ta med sina punkter. Om alla varit med och skapat agendan så vet alla gruppmedlemmar vad som tas upp på mötet och ges möjligheten att förbereda sig bättre.

För att få en tydligare Definition of done kan en checklista med saker som behöver vara klart för att en user story ska vara färdigt [2].

## **4.4 Metod för att lära sig och använda nya verktyg och teknologier**

### **Hur har det varit under projektets gång?**

Gruppen har använt Git och plattformen Github för att versionshantera produkten och arbeta gemensamt. En funktion som Github erbjuder som varit värdefull är pull requests. Eftersom det kan verka svårt första gången presenterade gruppens scrum master funktionen och visade hur den har använts. Detta så att alla kunde använda funktionen direkt. Med Githubs pull

requests går det att på ett smidigt sätt ge kritik på kod som andra gruppmedlemmar vill lägga in i applikationen och på så sätt höja kodkvalitén. För att utveckla expertis om hur Git används genomfördes Zoom-genomgångar där de som redan kunde arbetssättet berättade och förklarade hur det skulle användas till de som inte visste. Ett verktyg som varit värdefullt för gruppen är Learn git branching [3]. Verktöget användes för att på ett enkelt och interaktivt sätt lära sig mer avancerade koncept inom Git.

För att bygga applikationens gränssnitt har plattformen ReactJS använts, för att lära sig använda denna har Youtube-videor, tutorials och stackoverflow använts. När vissa personer i gruppen fått mer erfarenhet med denna så var det även frågor till andra teammedlemmar. Något som även fungerat bra är att experimentera själv.

Design av gränssnittet gjordes i programmet Figma. Vissa i gruppen hade tidigare använt detta program men även för de som inte bekantat sig med det tidigare var det inga svårigheter.

Trello var nytt för flera i gruppen. Det fanns inte så mycket att lära sig dock, så det har fungerat smidigt. En sak gruppen gjorde var att en i gruppen pekades ut som ansvarig för trello och sedan hade ansvaret för att allt såg snyggt ut.

Rapporten är skriven i latex på webbsidan Overleaf. Detta var nytt för vissa i gruppen vilket vi löste genom att de som kunde bättre hjälpte till med formateringen.

Pakethanterade som maven och npm var nytt för flera i gruppen. Detta löstes genom att de med mer erfarenhet visade hur man gjorde.

### **Hur hade vi velat ha det?**

Ambitionen när man startar ett nytt projekt är att ingen tid behöver användas till att lära sig nya program. Verkligheten är dock att nya program kommer att förekomma. För att hantera detta på bästa sätt tycker gruppen att färdiga resurser borde användas. Att ha genomgångar av program från andra gruppmedlemmar är också givande.

### **Vad ska vi göra för att få det som vi vill ha det?**

Det första som behöver göras är att kommunicera med gruppen för att kontrollera vilka som redan kan verktyget i gruppen och vilka som behöver lära sig det. Sen får de som redan kan verktyget visa de som inte kan. Gruppmedlemmarna har sedan eget ansvar att själva leta efter resurser och se till att dela resursen med varandra.

## **4.5 Koppling till föreläsningar och litteratur**

### **Hur arbetade vi under projektets gång?**

I samband med kursen fanns flera föreläsningar som berättade om ett agilt arbetssätt. Den delen från föreläsningarna som gruppen har applicerat mest handlar om KPI:er och framförallt

hur de ska mätas och följas upp. Som tidigare nämnts så användes KPI:erna främst i samband med utvärdering och planering av sprint. En stor del av kursens föreläsningss innehåll handlade också om user stories och task, framförallt hur user stories ska formuleras och hur det påverkar projektet. Det var något som gruppmedlemmarna hade färskt i minnet när arbetet med att formulera user stories påbörjades. Trots det så är gruppen inte helt nöjda med hur arbetet med user stories flöt på, då alla varit överens om att de varit för stora och inte nedbrutna i tasks. Detta är något som diskuteras mer ingående i punkt 1.3. Föreläsningen kring användandet av scrum har också varit en viktig bidragande faktor till att gruppen kunde komma igång bra i början. Även om det inom gruppen fanns oklarheter kring vad det faktiskt innebär att arbeta med scrum så var de första föreläsningarna och framförallt övningen viktig för att få en förståelse för hur veckorna skulle komma att se ut.

### **Hur hade vi velat ha det?**

Gruppen är överens om att det hade underlättat att från början bestämma mer konkret hur KPI:erna hade tänkts att användas löpande genom projektet. Det hade förmodligen också varit en fördel ifall gruppen i enighet med rekommendationerna från föreläsningarna varit bättre på att bryta ner user stories redan från start då det hade varit något som kunnat underlätta flytet i projektet. Trots att gruppen fick introduktion och övning på scrum i början av kursen hade det varit önskvärt om alla medlemmarna själva hade haft en klarare uppfattning om vad det faktiskt innebar att jobba agilt med scrum.

### **Hur ska vi komma dit?**

För att KPI:erna skulle komma till större användning hade det varit bättre ifall gruppen förde en diskussion om hur de skulle kunna bidra till att förbättra och utvärdera projektet redan innan det påbörjades. Det hade förmodligen lett till att gruppen valde de mest relevanta KPI:erna och kanske på ett mer naturligt sätt inkluderade dem mer i de löpande processerna. För att scrumprocesserna skulle fungerat bättre redan från start hade det varit bra ifall alla gruppmedlemmarna tagit individuella ansvar och läst in sig på materialet som fanns tillgängligt redan inför första sprinten. Det hade exempelvis varit något som gruppen hade kunnat formulera i det sociala kontraktet. Då user stories haft en så central och betydande roll i projektet hade det varit bra ifall de hade gjorts noggrannare redan från första början. Gruppen hade exempelvis kunnat planera in en workshop där det gick igenom hur user stories skulle skrivas för att slippa lägga ner mer arbete senare under projektet.

## Referenser

- [1] jacobsidford, *React Time Calendar*, Tillgänglig på <https://www.npmjs.com/package/react-timecalendar> (2021/05/21).
- [2] H. Kniberg, *SCRUM AND XP FROM THE TRENCHES*. C4Media, 2015, ISBN: 9781329224278.  
URL: <https://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2/>.
- [3] pcottle, *Learn Git Branching*, Tillgänglig på <https://learngitbranching.js.org/> (2021/05/24).