TheNext

3. Semester Eksamensprojekt i Software Design

Casper Frost, Patrick Sirich, Matthias Skou, Christian Strunge

2018

Indhold

[Introduktion 3](#_Toc532219616)

[Opsætning 3](#_Toc532219617)

[Rollefordeling og ansvarsområder\* 3](#_Toc532219618)

[Arbejdsmiljø 3](#_Toc532219619)

[Coding standards 3](#_Toc532219620)

[Design standards 3](#_Toc532219621)

[Patterns 3](#_Toc532219622)

[Udviklingsmodel 3](#_Toc532219623)

[Argumentation/Methology comparison 3](#_Toc532219624)

[Retningslinjer for udviklingsmodellen 3](#_Toc532219625)

[Analyse 4](#_Toc532219626)

[Kunden 4](#_Toc532219627)

[Risikoanalyse & risikoplan 4](#_Toc532219628)

[Identifikation 4](#_Toc532219629)

[Risikoanalyse 5](#_Toc532219630)

[Risikoplan 6](#_Toc532219631)

[Kravspecifikation 7](#_Toc532219632)

[Scope 7](#_Toc532219633)

[User stories 7](#_Toc532219634)

[Tasks 9](#_Toc532219635)

[US01 9](#_Toc532219636)

[US02 9](#_Toc532219637)

[US03 10](#_Toc532219638)

[Us04 10](#_Toc532219639)

[Us05 10](#_Toc532219640)

[US06 11](#_Toc532219641)

[US07 11](#_Toc532219642)

[US08 11](#_Toc532219643)

[Gantt-chart 11](#_Toc532219644)

[Metrics\* 11](#_Toc532219645)

[SPRINT 1 12](#_Toc532219646)

[SPRINT PLANNING 12](#_Toc532219647)

[Hvad kan vi aflevere som følge af den kommende sprint? 12](#_Toc532219648)

[Hvilke arbejdsmetoder/værktøjer skal vi anvende? 12](#_Toc532219649)

[Sprint goal 12](#_Toc532219650)

[TASK GLOSSARY 12](#_Toc532219651)

[ACCEPTANCE TEST 13](#_Toc532219652)

[Burndown chart 1 14](#_Toc532219653)

[SPRINT review 15](#_Toc532219654)

[Introduktion 16](#_Toc532219655)

[Inkrementeringen 16](#_Toc532219656)

[Demonstration & Feedback 16](#_Toc532219657)

[Diskussion og projektering 16](#_Toc532219658)

[Sprint retrospekt 17](#_Toc532219659)

[Sprint 2 17](#_Toc532219660)

[Burndown chart 2 17](#_Toc532219661)

[Tasks estimation vs. Tasks actual(\*) 17](#_Toc532219662)

[Informal reviews 18](#_Toc532219663)

[Formal reviews 18](#_Toc532219664)

[Konklusion 19](#_Toc532219665)

[Bilag 19](#_Toc532219666)

# Problemformulering

# Introduktion

# Opsætning

## Rollefordeling og ansvarsområder\*

*Af Casper*

Patrick: Product owner & stakeholder-fuldmægtig

Casper: Rapport-ansvarlig & stand-in product owner

Matthias: Kode- og task-ansvarlig

Christian: SCRUM-master & Project manager

## Arbejdsmiljø

Mangel på fast lokale, arbejdsro, forstyrrelser.  
Sideløbende undervisning og mandatory projekter(og div. fremlæggelser), svært at planlægge en sprint og overholde den.  
Problemer med intelliJ, gentagende forsinkelser, workspace xml.

### Coding standards

### Design standards

### Patterns

## Udviklingsmodel

### Argumentation/Methology comparison

### Retningslinjer for udviklingsmodellen

# Analyse

*Af Patrick & Matthias*

## Kunden

TheNext er en SaaS virksomhed som specialiserer sig inden for whitelabeling af software. TheNext tilbyder at udvikle whitelabel platformer for virksomheder – dette indebærer at de laver en markedsanalyse for at finde forskellige behov indenfor markedet. Derefter finder TheNext en kunde som skal have udviklet en platform – hvis TheNext finder at der er et større behov, for denne platform, vil TheNext udvikle en whitelabel løsning af denne platform.

Vi udvikler en platform for TheNext som tager udgangspunkt i virksomheden YourLocal, men som senere skal udvides til at flere virksomheder kan benytte sig af platformen.

## Risikoanalyse & risikoplan

*Af Patrick*

I dette afsnit vil vi identificere diverse risikomomenter hvor der er en risiko for at slutproduktet ikke lever op til forventningerne og krav. Derefter vil vi lave en risikoplan hvor vi tager de risiko som har et moment på over 10 og planlægge præventive tiltag samt løsningsforslag samt de ansvarlige for disse.

### Identifikation

I nedenstående tabel vil vi identificere de risikomomenter som har en indvirkning på projektet slutprodukt.

|  |  |
| --- | --- |
| Risikomomenter: |  |
| Projektrisiko | * Misforståelse af krav til kunden * Drastisk ændring af krav mod slutningen af projektet |
| Personalerisiko | * Længerevarende sygdom * Gruppemedlem frafalder projektet * Tidsoverskridelse i sprints |
| Kunderisiko | * Nedskæringer i budget * Nedlukning af virksomhed |
| Ressourcerisiko | * Hardwarenedbrud * Tab af data (overskrivning, fejl i version control) * Servernedbrud |
| Estimeringsrisiko | * Misforståelser i planlægning * Større udfordringer i koden * Tests viser store fejl * Ikke god nok tidsestimering |
| Produktrisiko | * Et produkt der ikke er brugervenligt nok * Fejlkommunikation der resulterer i manglende/forkerte features |

### Risikoanalyse

Efter identifikationen, vil vi analysere sandsynligheden for at risiciene hænder, hvor stor en konsekvens disse vil have på projektet, og ud fra disse udregne et produkt af, hvor stor en indvirkning de vil have på vores projekt:

|  |  |  |  |
| --- | --- | --- | --- |
| Risikoanalyse |  |  |  |
| Risikomoment | Sandsynlighed (skala: 1-5) | Konsekvens (1, 3, 7, 10) | Produkt (Sandsynlighed x Konsekvens) |
| Misforståelse af krav til kunden | 3 | 7 | 21 |
| Drastisk ændring af krav mod slutningen af projektet | 1 | 7 | 7 |
| Længerevarende sygdom hos gruppemedlem | 1 | 10 | 10 |
| Gruppemedlem frafalder projektet | 1 | 10 | 10 |
| Tidsoverskridelse i sprints | 3 | 7 | 21 |
| Nedskæringer i budget | 1 | 7 | 7 |
| Nedlukning af virksomhed | 1 | 10 | 10 |
| Hardwarenedbrud | 2 | 3 | 6 |
| Tab af data (overskrivning, fejl i version control) | 2 | 10 | 20 |
| Servernedbrud | 1 | 7 | 7 |
| Misforståelser i planlægning | 2 | 7 | 14 |
| Større udfordringer i kode | 3 | 7 | 21 |
| Tests viser store fejl | 2 | 3 | 6 |

### Risikoplan

*Af Patrick*

Efter vi har fået identificeret vores risikomomenter og udregnet hvor stor en indvirkning de muligvis kan have på vores projekt, kommer vi her med en løsningsmodel for, hvordan vi kan undgå eller løse problemstillingerne. Vi har taget udgangspunkt i de risikomomenter der har et risikoprodukt på over 10, og derfor har en markant indvirkning i slutproduktet som gør det værd at analysere:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risikomoment | Præventive tiltag | Ansvarlig | Løsningsforslag | Ansvarlig |
| Misforståelse af krav til kunden | Hav løbende dialog med kunden, og sørg for at stille uddybende spørgsmål | Project owner | Hvis der er opstået en misforståelse mellem kunden og os, vurderes det om funktionen skal implementeres eller forkastes. | Project owner og kunde |
| Gruppemedlem frafalder projektet | Klare retningslinjer og kommunikation i gruppen. Støt og hjælp hinanden. | Hele gruppen | Den resterende gruppe aftaler indbyrdes hvordan projektet fortløbende gribes an, og laver de nødvendige tiltag for at projektet kan færdiggøres. | Den resterende gruppe |
| Tidsoverskridelse i sprints | Vær realistisk omkring estimering, kend vores begrænsninger og arbejd systematisk mod målet. | Hele gruppen | Gruppen skal analysere fejlestimeringen, og om tiden kan fordeles bedre. | Hele gruppen |
| Tab af data (overskrivning, fejl i version control) | Sørg for at have backups på flere forskellige platforme: lokalt, GitHub, server | Hele gruppen | Alt afhængig af tabets omfang og tidspunkt i processen, vurderes det om arbejdet skal overføres til næste sprint. | Hele gruppen |
| Misforståelser i planlægning | Konstant kommunikation samt klare regler omkring mødetid og dage. Der skal gives klar besked forinden, så resten af gruppen kan lave en plan for dagen. | Hele gruppen | Medlemmet der ikke kan arbejde den dag skal selv sørge for at medlemmets tasks for dagen er blevet overtaget af en anden eller at en anden plan er lagt forinden. | Individuelle gruppemedlemmer |
| Større udfordringer i kode | Læs op på stoffet hjemmefra, inden koden påbegyndes. Sørg for at have lavet acceptance test forinden programmeringens påbegyndelse | Hele gruppen | Søg hjælp hos hinanden eller hos andre grupper. Hvis løsningen ikke er opnåelig, diskuteres andre løsninger. | Hele gruppen |

## Kravspecifikation

## Scope

## User stories

*Af alle*

Vi vil i projektet tage udgangspunkt i nogle user storys, som danner ramme for de funktionaliteter applikationen skal kunne. User storys er skrevet i en ”som x skal jeg kunne y” form, for at gøre forståelsen mellem udvikler og kunde lettere. Dette er første udkast, som vi efterfølgende vil bryde op i tasks, lave en tidsestimering på disse, og til sidst præsentere for kunden, som vil prioritere hvilke user storys der vægter højest:

|  |
| --- |
| User stories |
| Som kunde og firma skal jeg kunne logge ind, og have forskellige rettigheder. |
| Som kunde skal jeg kunne oprette en bruger, hvor mine interesser bliver gemt. |
| Som firma skal jeg kunne oprette en bruger |
| Som firma skal jeg kunne oprette nye tilbud som vises til relevante kunder. |
| Som kunde skal jeg kunne se en oversigt over tilbud i mit nærområde og sortere i disse ud fra kategorier. |
| Som kunde skal jeg kunne købe et tilbud. |
| Som kunde skal jeg modtage push-meddelelser om tilbud indenfor en geolokation på 5 km. |
| Som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt. |
| Som ansat skal jeg have administrative rettigheder over firmaer og kunder. |

## Tasks

*Af Casper*

Ud fra vores user storys vil vi nedbryde hvert enkelt i flere tasks. Vores tasks er de opgaver, som vi vil arbejde ud fra i et sprint. Disse tasks vil blive estimeret ud fra to planlægningsteknikker*: three point value estimating* og *planning poker*.

Herunder en opstilling af tabeller, som hver indeholder en user story, dennes tasks, og en three point value og planning poker estimering ud fra:

### US01

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US01 | User Story: Som kunde skal jeg kunne oprette en bruger. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US01.1 | Firma-, bruger- og tilbudstabel oprettes i databasen. | 1 | 2 | 1,5 |
| US01.2 | Underside til firmadata inklusiv kontooplysninger. | 2 | 2 | 2 |
| US01.4 | Opret gem-funktion for firma til databasen. | 3 | 1 | 2 |
| US01.5 | Tilknytning af kategorier til en bruger. | 2 | 2 | 2 |
| US01.6 | Funktion til redigering af stamdata. | 4 | 1 | 2,5 |
| US01.7 | Login-oplysninger gemmes i database. | 2 | 1 | 1,5 |
| US01.8 | Login-funktion der adskiller de forskellige brugere via login. | 3 | 4 | 3,5 |

### US02

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US02 | User Story: Som kunde skal jeg kunne oprette en bruger, hvor mine interesser bliver gemt, og logge ind, med relevante rettigheder. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US02.1 | Kundetabel oprettes i database. | 1 | 1 | 1 |
| US02.3 | Underside til kundeinformation med kategorier. | 1 | 4 | 2,5 |
| US02.4 | Underside til betalingsinformationer. | 2 | 1 | 1,5 |
| US02.5 | Gem-funktion til kunde. | 3 | 1 | 2 |
| US02.6 | Funktion til redigering af stamdata. | 4 | 2 | 3 |

### US03

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US03 | User Story: Som firma skal jeg kunne oprette nye tilbud, som skal vises til kunder. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US03.1 | Underside til oprettelse af tilbud, med tilknytning af billede. | 2 | 4 | 3 |
| US03.2 | Redigerings- og slettefunktion af eksisterende tilbud | 4 | 4 | 4 |

### Us04

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US04 | User Story: som ansat hos Your Local skal jeg have administrative rettigheder over firmaer og kunder. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US04.1 | Godkendelsesfunktion til firmaoprettelse. | 3 | 8 | 5,5 |
| US04.2 | Redigering af information for både kunder og firmaer. | 3 | 4 | 3,5 |
| US04.3 | Slettefunktion der fjerner en kunde, en firma eller en butik. | 3 | 1 | 2 |
| US04.4 | Underside til godkendelse af nyoprettede firmaer. |  |  |  |

### Us05

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US05 | User Story: som kunde skal jeg kunne købe et tilbud. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US05.1 | Funktion der ændrer lagerantallet for det pågældende tilbud. | 1 | 1 | 1 |
| US05.2 | Funktion der kan bekræfte køb, og få tilsendt en kvittering. | 2 | 2 | 2 |
| US05.3 | Underside for salg af tilbud. | 2 | 2 | 2 |

### US06

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US06 | User Story: som firma skal jeg kunne se en oversigt over hvor mange tilbud der er blevet købt. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US06.1 | Funktion der henter salgsinformation fra databasen. | 1 | 1 | 1 |
| US06.2 | Underside til oversigt af salg. | 2 | 1 | 1,5 |

### US07

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US07 | User Story: som kunde skal jeg kunne se en oversigt over tilbud i mit nærområde, og sortere i disse ud fra kategorier. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US07.1 | Som kunde skal jeg kunne modtage en meddelelse om tilbud indenfor en geolokation på 5 km. | 7 | 8 | 7,5 |
| US07.2 | Underside til visning af tilbud i nærheden. | 6 | 1 | 3,5 |

### US08

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: US08 | User Story: som bruger skal jeg have en tilpasset navigeringsbar afhængig af min brugertype, med en kontoredigeringsfunktion. | Three point value (timer) | Planning poker (timer) | Estimeret produkt ((TPV+PP)/2) |
| US08.1 |  |  |  |  |
| US08 |  |  |  |  |

## Gantt-chart

## Metrics\*

Planning poker

Three-point-estimation

# SPRINT 1

## SPRINT PLANNING

*Af Christian & Matthias*

### Hvad kan vi aflevere som følge af den kommende sprint?

Vi kan aflevere et produkt, som gør det muligt for firmaer at oprette en bruger på applikationen, hvor informationerne bliver korrekt gemt i databasen. Vi vil lave en grafisk brugergrænseflade, hvor der hhv. er en underside til firmadata og en til konto-oplysninger. Når et firma opretter sig kan de derefter tilknytte forskellige kategorier til deres bruger. Derudover vil der være mulighed for at redigere disse brugeres stamdata. Når et firma har oprettet en bruger, vil de få et login som er tilknyttet, med de korrekte administrative rettigheder.

Vi vil derudover aflevere et produkt, som gør det muligt for kunder at oprette en bruger, hvor kundens interesser bliver gemt til database. Kunden får et login når de opretter en bruger og ud fra dette login får de specifikke rettigheder, som er forskellige fra et firma-login. Der vil også være grafiske sider tilknyttet til oprettelse af en kunde – herunder side til betalings-information, kunde-information(med GDPR samtykke i mente) og en underside til deres foretrukne kategorier. Derudover vil det også være muligt for en kunde at redigere sine informationer og præferencer.

### Hvilke arbejdsmetoder/værktøjer skal vi anvende?

Vi vil i løbet af denne sprint, arbejde i pair-programming og løbende sørge for at holde vores SCRUM-board samt burndown-chart opdateret. Vi vil benytte os af JPA til at oprette samt administrere vores database. Vi vil følge vores aftalte coding-standarder og arbejde ud fra aftale design-patterns, med især MVC i fokus. Vi vil benytte os af Spring til at lave web-applikationen og bootstrap(eller evt. egen css) til designet af applikationen. Vi vil skrive acceptance-tests til alle vores tasks og løbende sørge for at de er blevet opfyldt, før tasken kan godkendes.

### Sprint goal

Målet med denne sprint, er at vi ud fra vores kundes prioriterer, kan aflevere et færdigt produkt som kan benyttes selvstændigt. Kunden har prioriteret disse user-stories, da det er vigtigt at firmaer tidligst muligt kan oprette sig og være forberedt inden applikationen officielt bliver opført.

## TASK DESCRIPTION

*Af Matthias & Christian*

### USER story 1

|  |  |
| --- | --- |
| ID | BESKRIVELSE |
| US01.1 | Alle tabeller skal være oprettet korrekt, og der skal være forbindelse imellem tilbudstabel og companytabel, der skal være kolonner til: ID, e-mail, fornavn, efternavn, password, telefonnummer, konto samt adresseoplysninger og konto-oplysninger. |
| US01.2 | Man skal kunne indtaste firmadata og dette skal gemmes til DB via en knap, herunder: ID, e-mail, fornavn, efternavn, telefonnummer, firmanavn, CVR og adresseoplysninger. |
| US01.3 | Man skal kunne indtaste kontooplysninger på undersiden for oprettelse af et firma, som skal gemmes til databasen via gem-knap, med følgende informationer: account number og registration number. |
| US01.5 | Et firma skal kunne tilknytte en til flere kategorier til deres firma, efter de har udfyldt firmainformationer. Disse kategorier skal gemmes korrekt [i en](https://www.facebook.com/) ManyToMany tabel indeholdende id på category og id på firma. Et firma skal også kunne vælge ikke at tilknytte kategorier. |
| US01.6 | Man skal kunne redigere et firmas informationer, hvor de redigerede inputs bliver gemt korrekt i databasen. Email skal ikke være muligt at ændre(da det er bruger-login).  Det skal ikke være muligt at indtaste ugyldige inputs, de skal følge anvendte patterns og titles magen til dem der gør sig gældende, når man opretter et firma. |
| US01.7 | Når et firma opretter sig, skal der på samme underside som de andre informationer, være mulighed for at man indtaster et ønsket password. Dette password skal sammen med e-mail gemmes korrekt i databasen, og vil efterfølgende være det login som de benytter på applikationen. Når et firma gemmes, er det vigtigt at de ikke kan benytte en e-mail som allerede eksisterer for et andet firma. |
| US01.8 | Der skal laves en login-side, hvor en bruger af system kan indtaste e-mail og password og logge ind. Disse login-informationer skal stemme overens med informationer i databasen.  Hvis login-oplysningerne er forkerte, skal brugeren ikke logges ind og der skal komme en fejlmeddelelse. Når en bruger er logget ind, skal system kunne differenciere hvorvidt om det er et firma eller ej. |

### USER story 2

|  |  |
| --- | --- |
| ID | BESKRIVELSE |
| US02.1 | Der skal oprettes tabel til kunde i database med de korrekte informationer: city, country, email, firstname, lastname, password, phonenumber, streetname, zipcode og GDPR consent, cardtype, creditcardnumber, cvv, expirationDate, nameOnCard, samt en junction-tabel til category indholdende kategori id og kunde id. |
| US02.3 | En underside skal laves med input fields til kundeinformation. Følgende felter skal inkluderes: city, country, email, firstname, lastname, password, phonenumber, streetname, zipcode og GDPR consent checkbox.  Disse felter skal garderes mod fejl 40(brugerfejl), således at der kun kan indtastes korrekte oplysninger der følger vores patterns og titles der viser til brugeren, hvilket pattern der skal følges.  Derudover skal der laves checkboxes til alle eksisterende kategorier, så kunden kan vælge hvilke kategorier personen har interesse for.  Alle disse felter skal gemmes korrekt i databasen, såfremt at kundens input er gyldigt. |
| US02.4 | Der skal oprettes en ny underside som kunden bliver redirected til efter de har udfyldt alle brugerinformationer og kategorier. Denne underside skal indeholde cardtype, creditCardNumber, cvv, expirationDate og nameOnCard input fields med korrekte patterns og titles så en bruger ikke kan indtaste ugyldige informationer. Disse informationer skal gemmes til databasen i samme kunde-række som er blevet gemt i US02.3. |
| US02.5 | Der skal implementeres en gem-funktion, der sørger for at informationerne i US02.3 og US02.4 bliver gemt korrekt og at kunden bliver redirected til korrekte undersider efterfølgende. |
| US02.6 | Man skal kunne redigere en kundes informationer, hvor de redigerede inputs bliver gemt korrekt i databasen. Email skal ikke være muligt at ændre(da det er bruger-login). Der skal både laves en underside for redigering af kundens stamdata og en for deres betalingsoplysninger. Begge disse undersider skal være udfyldt med de eksisterende data, når undersiden vises til at starte med.  Det skal ikke være muligt at indtaste ugyldige inputs, felterne skal følge anvendte patterns og titles fra opret firma. |

## ACCEPTANCE TEST

*Af alle*

Vi har valgt at benytte denne form for acceptance test, hvor vi i ID kolonnen har opsat det på følgende made: Testen er sat op således:

* **ID:** er skrevet i formatet “USERSTORY-ID.TASK-ID.TEST-ID”.
* **HVIS:** hvad skal der ske for, at testen kan igangsættes?
* **NÅR:** kriteriet for testens udfald.
* **SÅ:** testens output.

Vi har taget udgangspunkt I en række tasks fra vores US01, for at demonstrere hvordan testsne skal udføres:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | HVIS | NÅR | SÅ |
| US01.5.1 | Et firma opretter sig og ønsker at tilføje en kategori | Et firma har valgt en ønsket kategori | Bliver den enkelte kategori gemt korrekt i databasen, i en junction med kategori id og firma id |
| US01.5.2 | Et firma opretter sig og ønsker at tilføje flere kategorier | Et firma har valgt flere kategorier | Bliver kategorierne gemt korrekt i databasen i en junction, med en række for hver valgt kategori |
| US01.5.3 | Et firma opretter sig og ønsker ikke at tilføje en kategori | Et firma har ikke valgt nogle kategori(er) | Alt resterende data gemmes korrekt i databasen, men intet data gemmes i junction med kategori id og firma id |
| US01.6.1 | Et firma ønsker at redigere deres brugeroplysninger | En bruger bliver redirected til undersiden for at redigere | Vil alle eksisterende informationer være udfyldt i redigerings-templaten |
| US01.6.2 | Et firma har ændret deres ønskede informationer og trykker gem | De udfyldte informationer er gyldige | Overskrives eksisterende data i databasen korrekt |
| US01.6.3 | Et firma har ændret deres ønskede informationer og trykker gem | De udfyldte informationer er ugyldige | Vises der en fejlmeddelelse med det ugyldige data og ændringerne gemmes ikke. |
|  |  |  |  |
| US01.4.1 | Et firma opretter en konto | Informationerne er gyldige | Opret bruger I database |
| US01.4.2 | Et firma opretter en konto | Informationerne er ugyldige | Gemmes data ikke, og fejlmeddelelse vises |
| US01.7.1 | Et firma opretter en konto | E-mail allerede findes I database | Gemmes data ikke, og fejlmeddelelse om duplikering vises |
| US01.8.1 | En bruger logger ind på applikationen med et gyldigt login | Login-typen er firma | Programmet genkender login-typen og gemmer det korrekt i en session |
| US01.8.2 | En bruger logger ind på applikationen | Login-typen er ikke et firma | Programmet genkender ikke login-typen og viser en fejlmeddelelse |
| US01.8.3 | En bruger forsøger at logge ind på applikationen | Login-informationerne er ugyldige | Programmet viser meddelelse ’Email or password is incorrect’ og brugeren bliver ikke logget ind |

## Burndown chart 1

Eftersom denne sprint indeholdt en del opsætning og nogle opstartsproblemer, samt et sideløbende projekt, ser vores burndown-chart derfor lidt atypisk ud.

## SPRINT review

*Af Alle*

Eftersom vi følger en SCRUM-inspireret udviklingsmodel, er sprint review og sprint retrospekt essentielle dele af vores proces. I reviewet vil udviklerne og stakeholders holde et uformelt møde, der fokuserer på den inkrementering, som er blevet leveret som produkt af sprintet, hvor især feedback, kommentarer og ændringer er i højsædet.

Patrick vil udover at være product owner i vores projekt desuden også repræsentere stakeholders, og dermed have deres fulde opbakning ift. produktets design, funktionalitet og andre relevante nøglepunkter.

Vi følger en disposition over vores sprint review der indeholder følgende:

1. En introduktion til det første Sprint review (punkt 1 vil ikke blive fulgt i fremtidige reviews med mindre andre stakeholders vil være til stede). Introduktionen vil understrege reviewets formål og de deltagenes roller.
2. Et oplæg fra product owneren der fortæller, hvilke Product Backlog produkter der er blevet færdiggjort (og eventuelle user storys der ikke blev).
3. Udviklerne forklarer kortfattet hvilke problemer der blev mødt, og hvordan de blev løst. Fokus vil være på problemer, som stakeholders kunne have interesse i, og ikke tekniske komplikationer.
4. En demonstration af inkrementeringen, hvor stakeholder stiller spørgsmål og giver feedback sideløbende.
5. En diskussion af Product Backloggen og et sandsynligt færdiggørelsestidspunkt for projektet.
6. Alle deltagere planlægger næste skridt, så udviklerne har noget at gå ud fra i deres sprint planning.
7. Mødet hæves.

### Introduktion

Formålet med sprint reviews er et uformelt møde, og ikke et statusmøde; alle involverede parter skal have en indgangsvinkel til reviewet som værende en udveksling af idéer og en konstruktiv dialog. Det er derfor vigtigt at understrege, at udviklerne ikke står og skal sælge det produkt de har udviklet til stakeholderen – selvom de naturligvis skal stå på mål for deres tanker og handlinger under sprinten, og produktet deraf. Uenigheder under et sprint review udviser tegn på miskommunikation mellem parterne.

Ligeledes skal de/den repræsentative stakeholder(s) ikke føle at det der bliver demonstreret, er et færdigt produkt, som der skal bedømmes som helhed. Stakeholders job er at kommunikere deres behov så klart som muligt til udviklerne, og komme med indskud, ændringer og idéer til, hvordan de forestiller sig deres kundebase tænker.

Product owneren vil under et review stå for det bindende led mellem stakeholders og udviklingsteamet. Han vil blandt andet introducere hvilket arbejde der er blevet udført, præsentere oversigten over product backloggen efter teamets demonstration, og sætte en realistisk slutdato for projektet.

### Inkrementeringen

Under dette sprint har vi fået færdiggjort to User Storys fra Product Backloggen: US1 og US2. Kundens vigtigste prioriteter for produktet var, at give både firmaer og kunder mulighed for at oprette sig i systemet. Vi har lavet det mest basale skelet for at kunne oprette sig som virksomhed, med tilknyttede kontooplysninger, og et småt udvalg af kategorier som firmaet kan tilknytte sig. Ligeledes kan en kunde nu oprette sig, vælge hvilke kategorier vedkomne er interesseret i, og samtidig give samtykke til at få sine private data behandlet.

### Demonstration & Feedback

Efter demonstrationen af iterationen havde stakeholder følgende kommentarer:

1. Layout på siderne var som første udkast okay, men i fremtiden vil han gerne have en mere strømlinet og overskuelig side, som også inkluderer Your Locals farveskema.
2. GDPR-samtykket var en god tilføjelse, men han vil have ændret det til ’vilkår og betingelser’, som linker til en pop-up med en Lorem Ipsum.
3. I forlængelse af punkt 1 vil kunden gerne have en navigationsbar i toppen, så man kan navigere rundt når man er logget ind. Kunden vil også gerne have en funktion i navigationsbaren, hvor man kan gå ind og redigére sine oplysninger, oppe i højre hjørne ved siden af en ’log ud’ knap.
   1. Ud fra dette krav oprettede vi en ny User Story, da vi blev enige om, at kravet ikke hørte til under nogen eksisterende User Story, og i øvrigt også var for stor til at være en enkeltstående task.
4. Efter vi var færdige med at diskutere de allerede implementerede funktioner, diskuterede vi hvordan flowet i programmet skulle være. Kunden kunne godt tænke sig at man, som kunde, kunne gå ind på forsiden og se en liste over tilbud, uden at være logget ind. Hvis kunden så trykkede på et tilbud, vil vedkomne blive omdirigeret til en enkel side der giver besked om, at man skal oprette en bruger for at benytte sig af servicen, og i bunden skal der være en mulighed for at kunne logge ind hvis man har en eksisterende bruger.

### Diskussion og projektering

Efter demonstrationen vil vi kigge på vores Product Backlog, og lave et sandsynligt tidsestimat af hvornår projektet forventes færdigt.

## Sprint retrospekt

*Af Alle*

Hvad har vi lavet internt og hvordan?

* Redegørende afsnit om hvad vi har lavet. Evt. nogle smarte features.

Ændringer i arbejdsfremgang?

* Sidste sprint krævede flere timer og relokering hjem til Matthias. Næste sprint fokuserer på at være til stede på skolen og i nærheden af vores Scrum-board.
* Tirsdag og fredag er vi på skolen næste sprint, og bruger rådgivning til rapport. Andre dage aftales dagen inden.
* Pair programming: fortsætter med at skifte internt, alt afhængig af tasks. Skifter som udgangspunkt mellem User Stories.

Hvad var godt?

* At vi som team var gode til at tilpasse os omstændighederne, både ift. lokation, tidspres og omstilling ift. fravær.
* God kommunikation mellem udviklerne
* God struktur: dagsplan: en klar rollefordeling og et godt arbejdsmiljø.
* God morale: folk lægger en indsats indtil de er færdige.

Hvad var skidt?

* Stand-up meeting: gik skidt, aldrig fuldtallige lavet dagsplan i stedet for. Afsnit om stand-up meeting først, og dagsorden baseret på det møde.
* Baseline: prøvet at følge den 100%, men det er ikke gået så godt i første sprint. Generelt mere en hæmsko end et godt værktøj, da det henvender sig til større projektgrupper.
* Reviews har ikke været veludført – manglede en ordentlig måde at teste på.
* Fravær – folk skal være bedre til at tage ansvar og spørge ind til hvad der skal laves. (Hvordan har vi håndteret det?)

# Sprint 2

## Burndown chart 2

# Tasks estimation vs. Tasks actual(\*)

*Af Casper*

Her vises en tabel indeholdende alle tasks, både færdige og ufærdige. Tabellen indeholder tre kolonner: task ID, det estimerede produkt, og det faktiske produkt.

|  |  |  |
| --- | --- | --- |
| TASK ID | ESTIMATED TIME (TIMER) | ACUAL TIME (MIN) |
| US1.1 | 1,5 | 110 |
| US1.2 | 2 | 80 |
| US1.4 | 2 | 30 |
| US1.5 | 2 | 90 |
| US1.6 | 2,5 | 55 |
| US1.7 | 1,5 | 120 |
| US1.8 | 3,5 | 70 |
| US2.1 | 1 | 25 |
| US2.3 | 1,5 | 70 |
| US2.4 | 2,5 |  |
| US2.5 | 2 |  |
| US2.6 | 3 |  |
| US3.1 | 3 |  |
| US3.2 | 4 |  |
| US4.1 | 5,5 |  |
| US4.2 | 3,5 |  |
| US4.3 | 2 |  |
| US5.1 | 1 |  |
| US5.2 | 2 |  |
| US5.3 | 2 |  |
| US6.1 | 1 |  |
| US6.2 | 1,5 |  |
| US7.1 | 7,5 |  |
| US7.2 | 3,5 |  |

# Informal reviews

# Formal reviews

|  |  |  |  |
| --- | --- | --- | --- |
| ID | HVIS | NÅR | SÅ |
| US01.7.1 | Et firma opretter en konto | Informationerne er gyldige | Opret bruger I database |
| US01.7.2 | Et firma opretter en konto | Informationerne er ugyldige | Gemmes data ikke, og fejlmeddelelse vises |
| US01.7.3 | Et firma opretter en konto | E-mail allerede findes I database | Gemmes data ikke, og fejlmeddelelse om duplikering vises |

# Konklusion

Cays opfordring: Skriv om hvordan det har været at have en kunde, som på forhånd har stillet krav til databaseopsætningen. Fordel/ulempe?

# Bilag