



A-Holdet

3. Semester projekt

RO DA 18E 3C

5. Nov - 17. Dec

Joachim Frank

Anders Skov Nielsen

Frederik Sthen Hansen

Antal tegn: 64400

Trello - Kanban Board: <https://trello.com/b/NsBX1Bzi/lms-project>

GitHub Repository: <https://github.com/AndersSkovNielsen/LightManagement>

1: Introduktion	5
2: Problemformulering	5
3: Projektetablering	6
3.1: Deltagere	6
3.2: Kontakt information	7
3.3: Tidsplan	7
3.4: Gruppe kontrakt	8
3.4.1: Ambitionsniveau	8
3.4.2: Sygdom og fravær	8
3.4.3: Konflikt	9
3.4.4: Mødetid	9
3.4.5: Hjemmearbejde	9
3.5: Krav til hardware og software	9
3.5.1: Hardware	9
3.5.2: Software	10
3.6: Risiko	11
4: Projekt Manifest	12
4.1: SCRUM	12
4.1.1: Roller	12
4.1.2: Planning Game	13
4.1.3: Daily SCRUM	13
4.1.4: Sprint review:	13
4.1.5: Sprint Retrospective:	14
4.1.6: User Stories	14
4.1.7: Kanban Board, ved brug af Trello	14
4.1.8: Planning Poker	14
4.2: XP	15
4.2.1: Pair Programming	15
4.2.2: Test Driven Development	15
4.2.3: Sustainable Pace	15
4.2.4: Collective Code Ownership	16
4.3: UP	16
4.3.1: UML	16
5: Sprint 0 / Inception Deck	17
5.1: Idégenerering	17
5.2: User Stories:	17

5.2.1: Acceptance Criteria	19
5.3: Elevator Pitch	21
5.4: Product Box:	21
5.5: Product Backlog:	23
5.6: Visual WorkSpace	24
5.7: Arkitektur overblik via domænemodel	24
5.8: Sprint retrospective:	25
6: Sprint 1	26
6.1: Sprint Backlog	26
6.2: Daily SCRUM	27
6.2.1: Nov. 26	27
5.2.2: Nov. 28	28
6.2.3: Nov. 29	28
6.2.4: Nov. 30	29
6.3: Sprint Review	29
6.4: Sprint retrospective	31
6.4.1: Test Driven Development	31
6.4.2: Pair Programming	32
6.4.3: Review og Retrospective	32
6.4.4: Daily SCRUM	33
6.4.5: Kanban Board	34
6.4.6: Sustainable Pace	35
6.4.6: Velocity	36
7: Sprint 2	37
7.1: Sprint Backlog	37
7.2: Daily SCRUM	38
7.2.1: Dec. 3	38
7.2.2: Dec. 5	38
7.2.3: Dec. 6	39
7.2.4: Dec. 7	39
7.3: Sprint Review	40
7.4: Sprint Retrospective	42
7.4.1: Pair Programming	42
7.4.2: Daily SCRUM	42
7.4.3: Sustainable Pace	42
7.4.4: Review & Retrospective	43
7.4.5: SCRUM Master	43
7.4.6: Velocity	44

8: Sprint 3	45
8.1: Sprint Backlog	45
8.2: Daily SCRUM	46
8.2.1: Dec. 10	46
8.2.2: Dec. 11	47
8.2.3: Dec. 12	47
8.2.4: Dec 13	47
8.3: Sprint Review	48
8.3.1: Sidedesign	50
8.4: Sprint Retrospective	51
8.4.1: Pair Programming	51
8.4.2 SCRUM Master	51
8.4.3: Velocity	52
9: Diskussion	52
9.1: Estimering af user stories	53
9.2: SCRUM som arbejdsproces	54
9.2.1: Joachim	54
9.2.2: Anders	54
9.2.3: Frederik	55
10: Konklusion	55

1: Introduktion

Ansvarlig: Anders

I denne rapport vil vi beskrive vores arbejde med 3. Semester eksamensprojekt på Datamatiker uddannelsen, hvor vi skal afprøve en arbejdsprocess med SCRUM og XP i modsætning til det tidligere semester hvor vi arbejdede med UP.

I løbet af dette projekt kommer vi også til at arbejde med at opstille et system, der passer til vores givne opgave, men vores rapport vil stærkt fokusere på arbejdsprocessen og vores erfaringer med de metoder og artefakter, som vi har udvalgt fra SCRUM og XP.

Vores Raspberry Pi kommer vi til at referere til som "gyrosensor", en sensor man har på hånden. Sensoren, der tænder lyset bliver betegnet som "bevægelsessensor", den som sidder i væggen og tænder lyset.

2: Problemformulering

Ansvarlig: Anders

I lærerværelset på Erhvervsakademi Sjælland (EASJ) i Roskilde har de en bevægelsessensor, der slukker lyset når der er mangel på input. Lærernes arbejde er stillesiddende og giver derfor ikke altid input til sensoren, hvorved lyset slukker og lærerne skal afbryde deres arbejde for at bevæge sig for at sensoren igen tænder lyset.

Kunden ønsker at kunne arbejde uafbrudt uden at lyset slukkes. Et eksempel på en løsning fra kunden selv består i at programmere en robot arm til at lave den nødvendige bevægelse for at sensoren tænder lyset igen.

Ud fra dette problem har vi derfor stillet spørgsmålene:

“Hvordan kan vi med SCRUM som arbejdsprocess udvikle et system til lærerværelset på EASJ, så lærerne ikke behøver at rejse sig når lyset slukker?”

“Hvordan kan vi inkorporere en 3. Parts API i vores system?”

3: Projektetablering

3.1: Deltagere

Ansvarlig: Alle

Navn	Rolle	Stærke kompetencer	Svage kompetencer
Joachim Frank	SCRUM Team	GUI. Åben for ændringer.	Kan overse detaljer. Arbejde.
Anders Skov Nielsen	SCRUM Team, Kontaktperson.	Fokus på aktiv opgave. Velskrivende.	Minimalt arbejde hjemmefra. Dårlig udmelding af problemer.
Frederik Sthen Hansen	SCRUM Master SCRUM Team	Kreativitet i design.	Taber overblikket, stresser, minimalt hjemmearbejde.

3.2: Kontakt information

Ansvarlig: Alle

Navn	Telefon	Email
Joachim Frank	53 44 76 00	Jgf.easj@gmail.com
Anders Skov Nielsen	61 33 39 95	anderssnielsen@live.com
Frederik Sthen Hansen	23 66 59 02	Frederik.Sthen.Hansen@gmail.com

3.3: Tidsplan

Ansvarlig: Anders + Frederik

Fase	Estimeret Tid	Beskrivelse
Projekt Etablering	Til mandag d. 9/11.	Projektetablering og manifest
Inception Deck / Sprint 0	Til d. 22 november	Product Box, Elevator Pitch, User Stories, og Product Backlog.
Sprint 1 - 3	3 uger fra d. 25 nov - 13 dec	Kode og rapportskrivning
Rapportfinalisering	1 dag til og med 17/12	Endelig gennemgang af rapport og korrektur.

3.4: Gruppe kontrakt

3.4.1: Ambitionsniveau

Ansvarlig: Alle

Vi er alle generelt tilfredse med at vi kan bestå vores eksamen, men vil arbejde på at vi i det mindste kan få et 7-tal, mens højere karakterer selvfølgelig stadig er værd at stræbe efter, men vil ellers primært være en bonus.

3.4.2: Sygdom og fravær

Ansvarlig: Alle

I tilfælde af sygdom eller generelt fravær skal der sendes en besked til holdet via en fælles gruppe på Discord chat applikationen. Denne besked behøver ikke umiddelbart nogen grund, medmindre man forventer at være fraværende i længere tid.

Når man er fraværende forventes det at man læser rapporten igennem for ændringer dagligt for at holde sig opdateret.

Hvis man er fraværende i længere tid bedes man deltage i et par timer om dagen via Discord chatten, og at man holder sig opdateret på teorien bag arbejdet så man ikke falder bagud og glemmer hvad vi arbejder på.

Skulle det ske at kontaktpersonen selv er fraværende, skal vedkommende skrive til dagens vejleder, eller de resterende team deltagere, især hvis der ikke er en vejleder til stede på den givne dag.

3.4.3: Konflikt

Ansvarlig: Alle

I tilfælde af indbyrdes konflikt vil det givne emne først og fremmest blive taget op til diskussion i håb om at der kan findes frem til en løsning. Hvis dette viser sig ikke at være tilfældet, så vil der afholdes en afstemning hvor flertal udvælges.

3.4.4: Mødetid

Ansvarlig: Alle

Når der er timer på skemaet, så forventes det at man møder op, så mest muligt arbejde kan finde sted, samt at der er bedst mulighed for møder og fælles diskussion.

3.4.5: Hjemmearbejde

Ansvarlig: Alle

Hvis der er behov for læsning inden for et vist emne, kan dette blive allokeret til hjemmearbejde.

Derudover er hjemmearbejde ikke obligatorisk, men anmodes i tilfælde af længerevarende fravær. Hjemmearbejde kan også blive nødvendigt, hvis der ikke bliver lavet nok fremskridt i løbet af den almene mødetid ved skemaet.

3.5: Krav til hardware og software

3.5.1: Hardware

Ansvarlig: Frederik

Hver deltager i teamet skal have sin egen computer at arbejde på.

En Raspberry PI med SenseHat bruges til at simulere en gyroskopisk sensor man bærer på håndryg eller håndled, der skal registrere lærernes bevægelser, når de arbejder normalt. Denne

Pi vil fremover være benævnt "gyrosensor". Dette er for at differentiere den fra ordet "bevægelsessensor", der refererer til den indbyggede bevægelsessensor i lærerværelset.

3.5.2: Software

Ansvarlig: Alle

Rapporten vil primært blive skrevet i et fælles Google Docs dokument og Github vil bruges til at opbevare vores program og andre nødvendige filer.

For at kunne organisere vores opgaver benytter vi os af et digitalt Kanban Board på hjemmesiden Trello.com.

Visual Studio skal være opdateret til version 15.9.2 og i løbet af projektet må der ikke udføres opdateringer, for at reducere problemer med forskelligt software.

Visual Studio Code skal være opdateret til version 1.28.2, som anvendes til Typescript applikation og i løbet af projektet må der ikke udføres opdateringer, for at reducere problemer med forskelligt software.

I tilfælde af fildeling vil vi gøre brug af Github Desktop, som for alle skal være opdateret til version 1.5.0, for at undgå eventuelle konflikter mellem differentierende software.

Når der skal laves test af REST API uden for Unit Testing, vil vi gøre brug af Postman da flertallet har bedre erfaring med denne software i forhold til Fiddler, et lignende program.

Til opbevaring af vores databaser og web applikationer, som fx REST API og hjemmeside gennem HTML og TypeScript, vil vi gøre brug af Microsofts Azure, som er en cloud service, der gør det muligt at anvende vores programmer over nettet.

I tilfælde af fjern-møder vil der gøres brug af Discord, en gratis chat app. Discord behøves ikke opdateres til en standardiseret version.

Som 3. parts API benytter vi <http://worldclockapi.com/>. Herfra vil vi hente klokkeslæt, dato, ugedag og tidszone.

3.6: Risiko

Ansvarlig: Alle

Niveau	Beskrivelse	Betydning	Indikator	Beredskabsplan
2	Tab af link til docs.	H	Mistet adgang til Docs	Kontakt andre medlemmer for nyt link.
7	Længere sygdom.	H	Tilstedeværelse	Kontakt og reorganisation.
4	Aftagende energiniveau i mødetid	M	Lav fokus og træthed	Lad vedkommende tage en pause eller hjem for ikke at risikere fravær i følgende dage
6	Ukendt emne (Teknologi)	H	Kodning kører fast	Led efter løsning i C#-note. Konsulter andre gruppemedlemmer, og derefter læreren hvis problemet stadig ikke har løst sig.
5	Ukendt emne (Python Raspberry pi sensorer)	H	Vi har kun basis erfaring inden for emnet.	Diskuter designløsning i gruppen for at sikre at alle har samme forståelse af emnet.

4	Tekniske problemer med Visual Studio	H	Uforståelige fejl i kode	Kontakt vejleder
---	--------------------------------------	---	--------------------------	------------------

4: Projekt Manifest

Ansvarlig: Anders

Vi vil her gøre rede for hvilke praktiker og artefakter fra SCRUM og XP som vi vil tage i brug i løbet af vores projektforløb og de individuelle sprint. Dertil vil vi også tage enkelte værktøjer fra UP i brug.

4.1: SCRUM

4.1.1: Roller

Ansvarlig: Frederik + Anders

Vores Product Owner er vores vejledere, lærerne på EASJ, specifikt Vibeke. Dog har disse primært angivet business value for vores user stories. I de fleste andre af product owners ansvarsområder (User stories, endelige beslutninger vedr. Acceptance-criteria og sidste ord ved nedskæring af scope) har vi, med Vibekes velsignelse selv taget beslutningerne. De nævnte ansvarsområder omfatter: user stories, endelige beslutninger vedr. Acceptance criteria, og endelige beslutninger ved nedskæring af scope.

Efter et frivilligt valg har Frederik meldt sig som vores SCRUM Master i starten af vores Sprint 0, mens at resten af holdet er det generelle SCRUM Team.

Dog har vi op til 1. Sprint ændret SCRUM Master til at være Anders, da Frederik har meget at se til uden for projektet i forb. med flytning til Roskilde. Han vil muligvis genoptage rollen hen imod slutningen af projektet, såfremt der bliver overskud til dette. Denne ændring er også blevet

taget på delvis basis af at rotere rollen, så alle kunne prøve at være ansvarlig for opdateringer og kundekontakt med vores Product Owner.

4.1.2: Planning Game

Ansvarlig: Anders

For at få et overblik over arbejdet inden for de individuelle sprint vil vi gøre brug af et overordnet Planning Game, hvor vi ser på vores givne user stories og den udregnede velocity og derfra udpeger hvilke user stories vi vil tage fat på per sprint.

Dette vil vi primært gøre fredagen inden næste sprint, så vi ikke står med denne opgave når vi møder op om mandagen og derfor kan komme i gang hurtigst muligt.

4.1.3: Daily SCRUM

Ansvarlig: Alle

Dertil vil vi også have dagligt SCRUM møde for at skabe et hurtigt overblik over, hvor vi står i processen og hvad vi skal arbejde på i løbet af den givne dag. Vi har på nuværende tidspunkt ikke vedtaget en specifik dagsorden from SCRUM møder

4.1.4: Sprint Review:

Ansvarlig: Alle

I hvert sprint review vil vi tage et kig op hvor stort fremskridt vi har gjort med udviklingen af produktet. Fokus vil især lægges på i, hvilket omfang vores velocity ligger på og om vi følger den ideelle velocity vi fastsatte i starten af det tidligere sprint. Til dette vil vi anvende vores Kanban Board til at måle projektet og give os den info, der skal til for at tegne projektets kurs ind på et burndown chart.

4.1.5: Sprint Retrospective:

Ansvarlig: Alle

Efter sprint review afholder vi sprint retrospective hvor vi finder forklaringen på afvigelser i velocity fra det forventede, ved at diskutere hvad der fungerer godt i vores arbejdsmetoder, og hvad der skal strammes op på.

4.1.6: User Stories

Ansvarlig: Alle

Vi vil i Sprint 0 opstille diverse user stories som vi kan arbejde ud fra vigtigheden via Business Value (forkortet BV), som er afsat af vores kunde, også kendt som vores Product Owner.

4.1.7: Kanban Board, ved brug af Trello

Ansvarlig: Frederik

Vi har valgt at organisere vores arbejdsopgaver ved brug af et Kanban board, frem for et SCRUMboard, da Kanban boards er mere gennemtestede og formaliserede end SCRUM boardet, der stadig har flere experimentelle træk.

4.1.8: Planning Poker

Ansvarlig: Alle

Når vi skal til at estimere vores User Stories i Sprint 0 vil vi gøre brug af metoden Planning Poker, da vi føler at denne fungerer bedst for os da alle får mulighed for at komme med deres mening og argumentere for deres valg af score points.

4.2: XP

4.2.1: Pair Programming

Ansvarlig: Alle

For at overkomme de mere komplekse dele af vores user stories, og for at undgå synkroniseringsfejl i vores kode, vil vi gøre brug af Pair Programming til at samarbejde om at få løst et problem. Vi tænker at gøre brug af denne metode i 2. Sprint, men vi har på fornemmelsen at metoden kan være gavnlig løbende og vi vil derfor angive de steder i rapporten hvor vi vælger at gøre brug af Pair Programming.

4.2.2: Test Driven Development

Ansvarlig: Alle

Når vi skal i gang med at programmere i 1. Sprint vil vi afprøve Test Driven Development, da vi ikke har gjort os megen erfaring inden for denne arbejdsmetode og mener at det bedste tidspunkt at foretage et sådant arbejds eksperiment vil være i starten af programmeringen, hvor vi ikke har opbygget megen kode endnu.

4.2.3: Sustainable Pace

Ansvarlig: Frederik

Sustainable pace bruges i projektet for at kunne fastholde det nødvendige arbejdstid i løbet projektet. Derved kan vi sikre stabile sprint, hvor vi ikke springer noget arbejde over.

Derudover vil vi også kort nævne at tirsdagene d. 27/11 og d. 4/12 har vi ikke skemalagte timer og derfor, på nuværende tidspunkt, vælger at holde fri, medmindre vi senere tænker at vi har brug for den ekstra tid.

Da vores projekt adskiller sig fra et virkeligt udviklingsprojekt ved at vi primært skal skrive en rapport og at det er denne, der er karaktergivende, har vi besluttet at afvige fra enkelte

sustainable pace ideer, men på måder vi mener beholder kernen i metoden. Vi vil til kodning have faste arbejdstider som ikke må overskrides, for at simulere metodens krav om aldrig at overarbejde, men i stedet udelukkende bruge evt. overarbejdstid på rapportskrivning.

Vores kodningstid er i starten sat til 4 timer om dagen, i modsætning til 6-8 man burde se i erhvervslivet af samme årsag. For os er det jo rapporten, og kun i mindre omfang vores produkt, der er udslagsgivende for dette projekts succes. For at vurdere om dette er nok arbejdstid til kodning, vil efter behov diskutere om vi mener at vi skal øge kodningstiden. Ved projektets start vil vi kode fra 09:00 til 13:00, og skrive stikord til rapporten ud fra vores kode. Efter kl 13 vil vi udelukkende skrive på rapporten og derefter er der ingen faste tider for fyraften fra rapportskrivning.

4.2.4: Collective Code Ownership

Ansvarlig: Alle

Da dette er et fælles projekt, som skal gavne os alle, vil vi gå ud fra at ingen ejer specifikke dele af koden og alle har rettighed til at komme med ændringer og rettelser, selv hvis de ikke er den originale forfatter af koden.

4.3: UP

4.3.1: UML

Ansvarlig: Frederik

I dette projekt ønsker vi at benytte UP's UML-diagrammer til at visualisere vores design ideer og give os en arbejdstegning til første udkast når vi begynder at kode i 1. Sprint. Dette ville fx være at lave en domænemodel for at få en ide om hvad vi skal have stillet op i vores arkitektur.

Vi bryder med vilje med UML syntaksregler i første omgang da vores UML diagrammer i dette projekt ikke er ment som dokumentation, men som værktøj til idegenerering til at skyde kodningen i gang med.

5: Sprint 0 / Inception Deck

5.1: Idégenerering

Ansvarlig: Frederik + Anders

Som allerede beskrevet i starten af rapporten blev vi præsenteret med en potentiel løsning af kundens problem i form af en robotarm. Vi har dog efterfølgende prøvet at komme frem til andre potentielle løsningsforslag for at se om en bedre løsning kunne forekomme.

1. Bevægelsessensor på hånden via gyrosensor, som sender besked til sensoren om at lyset skal holdes tændt
2. Tryksensor i stolene, der ville virke som en override på sensoren ved at lyset altid holdes tændt hvis nogen sidder ned i rummet.
3. Varmesensor, igen på hånden eller i lommer, som kunne fornemme ændring af varme fra brugere frem for hvis den bare lå på bordet.

Efter en diskussion i gruppen kom vi frem til at gyrosensoren var den mest realistiske plan.

5.2: User Stories:

Ansvarlig: Alle

Vi har valgt user stories fordi det er en overskuelig og letforståelig måde at holde overblik over programmets funktionalitet på og hvem der skulle ville have brug for hvert enkelte funktion.

Disse kan ses nedenfor, hvor at BV står for Business Value og SP for Story Points:

1. Som bevægelsessensor, vil jeg have en override switch, så jeg ikke slukker lyset når der er folk til stede.
 - BV: 3 | SP: 4

2. Som TS-applikation, vil jeg kunne sende data til væggens bevægelsessensor, så vi kan kommunikere.
 - BV: 3 | SP: 4
3. Som lærer, vil jeg kunne sende input til gyrosensor ved at arbejde normalt på computeren.
 - BV: 5 | SP: 2
4. Som gyrosensor vil jeg kunne videresende input til TypeScript applikationen.
 - BV: 2 | SP: 5
5. Som lærer vil jeg med TS-applikationen kunne justere bevægelsessensorens refresh-rate.
 - BV: 4 | SP: 7
6. Som lærer vil jeg kunne logge mig ind på TS-applikationen for at kunne begrænse, hvem der styrer lyset på kontoret.
 - BV: 4 | SP: 3
7. Som admin/udvikler vil jeg kunne tilføje og fjerne brugere fra TS-applikationen.
 - 7.1 Som admin vil jeg have en liste af brugere af denne service.
 - BV: 5 | SP: 5
8. Som TS-applikation vil jeg kunne modtage tidspunkt fra 3rd part webservice (world clock API, kalender).
 - BV: 1 | SP: 5
9. Som TS-applikation vil jeg have at de ændringer er gemt permanent selv når URL adressen ikke besøges.
 - BV: 2 | SP: 5
10. Som Lærer vil jeg med TS-applikation kunne bestemme, hvornår bevægelsessensoren skal holde op med at modtage input fra TS-applikationen.
 - BV: 5 | SP: 5
11. Som Lærer vil jeg kunne indstille Pi'ens sensitivitet i TS-applikationen, så den registrerer mine bevægelser.
 - BV: 5 | SP: 5
12. Som REST Service med CORS vil jeg kun kunne modtage requests fra IP-adresser på skolens adresse.
 - BV: 1 | SP: 5

Herfra kan vi se at vi har en score point sum på 55, som ville være ideelt at nå igennem inden projektets slutning. For at dette skulle lykkes ville det være nødvendigt for os at opnå en velocity på 18-19 score points per sprint.

5.2.1: Acceptance Criteria

Ansvarlig: Joachim + Anders

1. Override
 - a. (Success) Aktivering af override forhindrer bevægelsessensor i at slukke lyset.
 - b. (Failure) Override switch virker ikke og lyset slukker.
2. Data til væg sensor
 - a. (Success) Data bliver sendt fra applikationen, og registreres af væg sensor.
 - b. (Failure) Data sendes ikke fra applikationen.
 - c. (Failure) Bevægelsessensor modtager ikke de sendte data.
3. Input til Pi
 - a. (Success) Gyrosensor registrerer bevægelse.
4. Input fra Raspberry Pi til TS-applikationen
 - a. (Success) Input der sendes fra Raspberry Pi, sendes til, og modtages af TS-applikationen
 - b. (Failure) Input bliver ikke modtaget af Raspberry Pi.
 - c. (Failure) Input sendes ikke af Raspberry Pi.
 - d. (Failure) Inputtet modtages ikke af TS-applikationen
5. Sensor Refresh
 - a. (Success) Tid til lys slukker er ændret efter nye indstillinger.
6. Login
 - a. (Success) Bruger info står i hjørnet af hjemmeside.
 - b. (Success) Knapper på hovedmenu kan tilgås.
 - c. (Failure) Knapper kan tilgås uden login.
7. Tilføj/Fjern Bruger - Brugerliste
 - a. (Success) Tilføjet bruger kan hentes via API.
 - b. (Success) Længde af brugerliste er faldet med 1 efter at fjerne bruger.

- c. (Success) Brugerliste kan hentes via API.
 - d. (Failure) Tilføjet bruger findes ikke i brugerliste.
- 8. Clock/Kalender API
 - a. (Success) Ur kan ses på hjemmeside.
 - b. (Success) Dato kan ses på hjemmeside.
- 9. Permanent Indstilling
 - a. (Success) Ændringer i indstillinger ses stadig efter hjemmeside er lukket ned.
- 10. Sluk Pi
 - a. (Success) Pi deaktiveres ved tryk af knap på hjemmeside.
 - b. (Failure) Pi er stadig aktiv efter tryk af knap på hjemmeside.
- 11. Pi Sensitivitet
 - a. (Success) Ændring på sensitivity slideren i programmet, ændrer hvor store eller små bevægelser Pi'en registrerer efter behov.
 - b. (Failure) Pi'en registrerer uønskede bevægelser.
 - c. (Failure) Ændring på slideren gør intet.
- 12. Rest modtager request fra skolens ip-adresser
 - a. (Success) Rest-Service modtager request fra skolen IP-adresser som ønsket.
 - b. (Failure) Rest-Service modtager ikke request sent med skolen ip-adresser
 - c. (Failure) Rest-Service modtager request fra andre IP-adresser, der ikke kommer fra skolen af.

5.3: Elevator Pitch

Ansvarlig: Frederik

Som et ekstra punkt er vi blevet bedt om at fremstille en Elevator Pitch, som et eksempel på, hvordan man ville kunne sælge ideen om vores aktuelle opgave, og denne ses nedenfor.

Til: Lærerne på EASJ Roskilde

Som: Skal kunne arbejde uforstyrret

The: LMS (Lights Management System)

Er et/en: Hjemmeside der gør brug af Raspberry Pi, egen og 3rd party World Clock API

Som: Gør det muligt for lærerne at indstille og afsende input til lys-sensoren (holde lyset tændt)

I modsætning til: At skulle afbryde arbejdet for at vinke til sensoren

Vores produkt: Er digitalt, automatisk og fleksibelt.

5.4: Product Box:

Ansvarlig: Frederik + Anders

På samme måde som vores Elevator Pitch blev vi også bedt om en Product Box som eksempel på en visuel og hurtig fortælling om, hvad vores endelig system ville involvere.

(Product Box ses på næste side)

LIGHT MANAGEMENT SYSTEM



Wave goodbye to obstinate sensors!

**Automatic
Adjustable online
Secure**

5.5: Product Backlog:

Ansvarlig: Alle

Nu, hvor vi har fået styr på vores User Stories og de aspekter involveret deri, kan vi sætte vores Product Backlog op for et let overblik over, hvad vi har at arbejde med og begynde at opstille en rækkefølge som vi vil arbejde ud fra i vores opkommende Sprint.

Her har vi fx. valgt at vi hurtigst muligt skal have sat en liste af brugere op, da denne er kritisk for at få løst User Story #7, som er blandt de højeste Business Values vi har at arbejde med. Samtidig har vi også rykket User Story #8 højere op end egentlig nødvendigt i en sand arbejdsprocess, da denne er vores involvering af 3rd Part API, som vi er krævet at skulle arbejde med i dette eksamensprojekt.

Product Backlog	Sprint Backlog	In progress	Testing	Done
Story 1: Override Switch				
Story 2: Data fra TS-applikation til Vægs Bevægelsessensor				
Story 3: Gyrosensor får input fra bruger				
Story 4: Kommunikation				
Story 5: Refresh-rate				
Story 6: Styrebegrænsning v. Login				
Story 7: Tilføj/fjern Brugere				
Story 8: World clock API				
Story 9: Permanente ændringer				

Story 10: On-off switch til TS-app				
Story 11: Pi Sensitivity				
Story 12: CORS til REST				

5.6: Visual Workspace

Ansvarlig: Anders

Vi har valgt ikke at gøre stort brug af et visual workspace, da vi syntes at vi har fint overblik ved kun at gøre brug af vores Kanban Board på Trello.

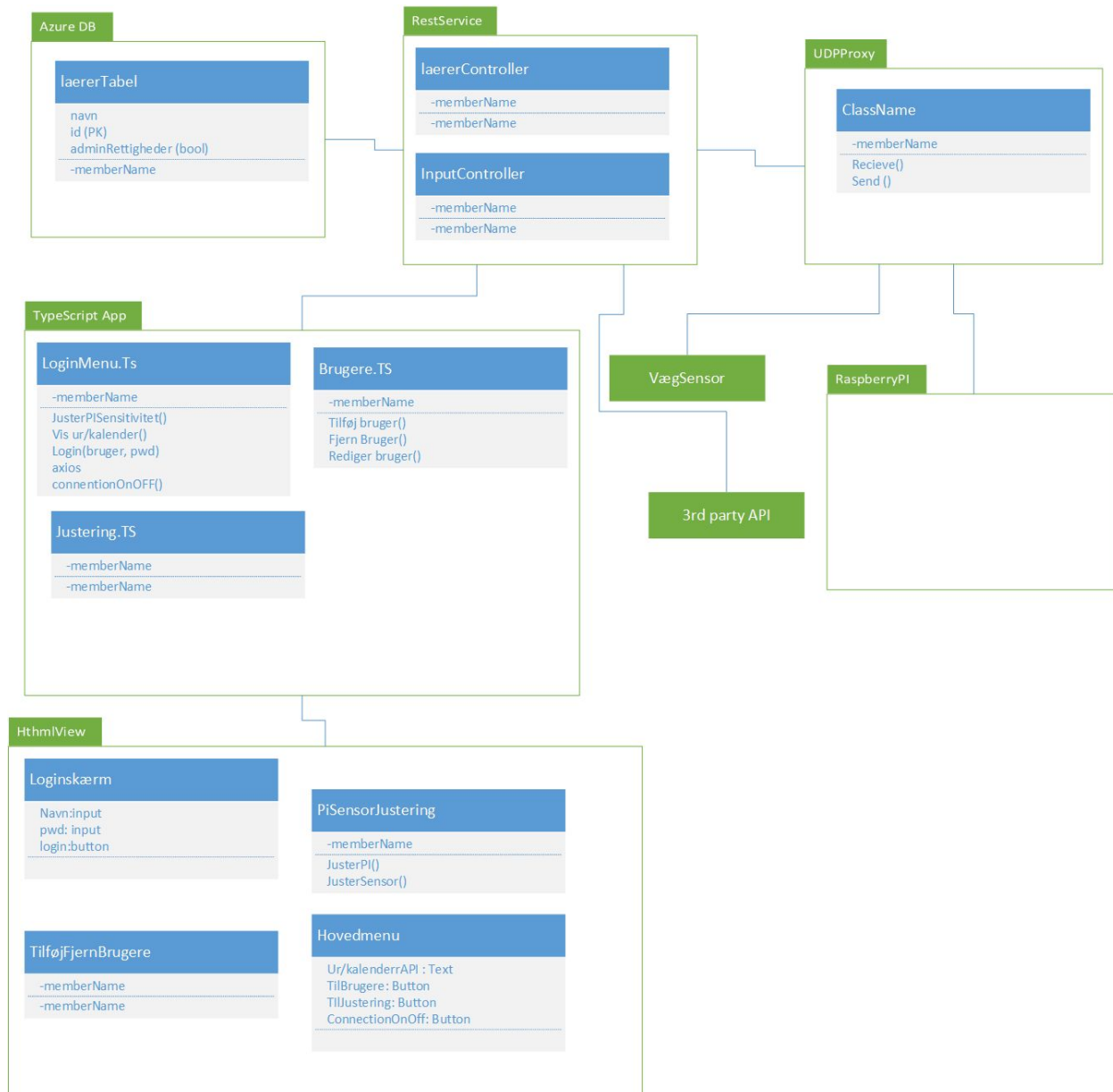
Vi vil også løbende i slutningen af hvert sprint opdatere et Burndown Chart for at vise vores ideelle velocity, samt den aktuelle velocity vi vil kunne se ud fra hvor meget arbejde vi får lavet i de individuelle sprint.

5.7: Arkitektur overblik via domænemodel

Ansvarlig: Alle

For at få et hurtigt, umiddelbart overblik over arkitekturen af det arbejde vi skal foretage os i de næste par uger, har vi gjort brug af programmet Visio, hvori vi har opsat de elementer, der typisk bruges i opsætningen af en domænemodel fra UP.

Dette er dog ikke ment som en formelt korrekt domænemodel og er kun taget i brug som en visualisering af de mange elementer vi skal arbejde med, og hvad disse kan komme til at indeholde i fremtiden, både inden for den givne arbejdstid vi har at gøre godt med og i tilfælde af at vi havde mere tid til at tilføje ekstra funktionalitet.

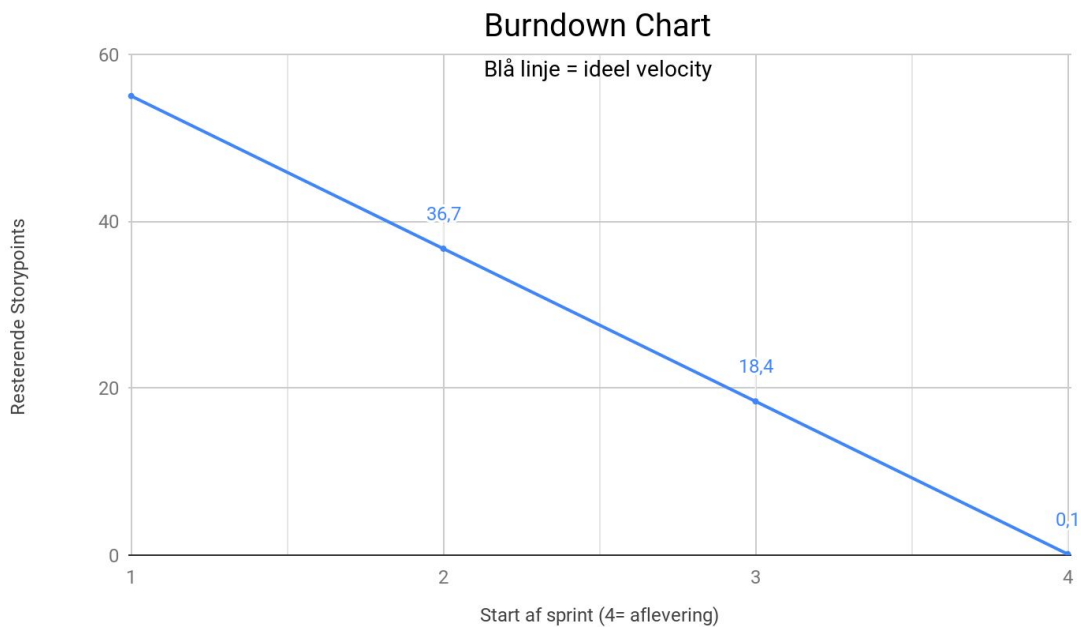


5.8: Sprint retrospective:

Ansvarlig: Alle

I starten af november blev Joachim ramt af influenza, og Anders i midt på måneden. Dette gjorde i begyndelsen SCRUM som metode er svær at opretholde når gruppen er ramt af sygdom. Vores diagnoser gør nogle gange vores vilje til kommunikation mangelfuld, hvilket gør det svært at opretholde de specificerede retningslinjer i vores gruppekontrakt.

Det lykkedes os at indhente det tabte arbejde og vi fik samlet op og re-grupperet tæt på slutningen af sprint 0. Nu hvor vi er klar til vores 1. Sprint, har vi fået opsat et Burndown Chart, som kan ses nedenfor, hvori vores ideelle velocity er vist ud fra den mængde story points vi har i øjeblikket.



6: Sprint 1

6.1: Sprint Backlog

Ansvarlig: Alle

Blandt de vigtigste User Stories fra vores kunde står vi med at skulle begynde på tilføjelse og fjernelse af brugere, som vil gøre brug af systemet, samt at få vores gyro sensor til at registrere bevægelse fra brugerens hånd, og at man kan indstille sensitiviteten for denne sensor.

Vi står dog i en situation ,hvor vi ikke har et nuværende system at bygge op på, så vi bliver derfor nødt til først at sætte vores arkitektur op og skabe forbindelse mellem de individuelle

elementer før at vi kan i sandhed komme i gang med vores arbejde. Dertil kræver vi også en database til opbevaring af listen af vores brugere, så disse kan gemmes i det at de bliver tilføjet.

Product Backlog	Sprint Backlog	In progress	Testing	Done
Story 1: Override Switch	Story 3: Gyrosensor får input fra bruger			
Story 2: Data fra TS-applikation til Vægs Bevægelsessensor	Story 4: kommunikation			
Story 5: refresh-rate	Story 7: tilføj/fjern Brugere			
Story 6: Styrebegrænsning v. Login	Story 11: Pi Sensitivity			
Story 8: world clock API				
Story 9: Permanente ændringer				
Story 10: On-off switch til TS-applikation				
Story 12: CORS til REST				

6.2: Daily SCRUM

6.2.1: Nov. 26

Ansvarlig: Anders

Efter en langsom opstart pga. forsinkelser og fravær blev det besluttet at det umiddelbare startpunkt ville være at få start på vores arkitektur i form af at sætte en REST service op, så vi kunne gå i gang med at få funktionaliteten klar til tilføjelse og fjernelse af bruger.

Vi ville også gerne være kommet i gang med vores gyro sensor, men pga. Netproblemer var dette ikke muligt på nuværende tidspunkt.

5.2.2: Nov. 28

Ansvarlig: Anders

Eftersom at vi ikke kom i gang med UDP relateret arbejde var dette en af punkterne vi gerne ville tackle i løbet af dagen men pga usikkerhed om dele af funktionaliteten dertil, så har vi valgt at vente med dette til senere på dagen, hvor vores vejleder inden for teknik er til stede.

Dernæst skal vi også helt klart i gang med TypeScript/HTML delen, så vi kan fuldføre vores arbejde med tilføjelse og fjernelse af brugere via vores komplette REST API.

6.2.3: Nov. 29

Ansvarlig: Anders

Vores arbejde med brugere er overstået og mangler kun potentiel finpudsning i fremtiden.

Vi vil nu komme i gang med at arbejde med vores Raspberry Pi, men dette er dog ikke muligt fra vores standard arbejdslokale, da forbindelsen til det krævede netværk ikke er kraftig nok eller ikke kan tilgås overhoved.

Derfor vil vi til at starte med arbejde via en emulator hvor vi kan skrive nok kode til at få styr på vores registrering af bevægelses input og justering af indstillinger for sensitiviteten af input. Dog kan vi ikke arbejde med at sende beskeder fra vores Pi til vores UDP-Proxy, medmindre vi rykker op i klasselokalet.

6.2.4: Nov. 30

Ansvarlig: Anders

Inden at holdet dukkede op til dagens arbejde og SCRUM møde, blev der lavet test på vores forbindelse mellem Gyrosensoren og vores UDP-Proxy, da denne forbindelse ikke blev lavet i løbet af gårsdagen.

Dette viste sig så at være pga vi havde gjort brug af det forkerte netværk og blev derfor hurtigt bragt i orden. Derfor kunne vi nu vælge at gå i gang med kommunikationen mellem vores UDP-Proxy og REST, så vi kunne få forbindelse mellem alle elementerne af vores system.

6.3: Sprint Review

Ansvarlig: Frederik + Anders

Nu hvor vi er kommet til vejs ende for vores første sprint ser vores Kanban Board således ud.

Product Backlog	Sprint Backlog	In progress	Testing	Done
Story 10: On-off switch til TS-applikation		Story 4: kommunikation	Story 7: tilføj/fjern Brugere	Story 11: Pi Sensitivity
Story 8: world clock API		Story 6: Styre-begrænsning v. Login		Story 3: Gyrosensor får input fra bruger

Story 5: refresh-rate		Story 9: Permanente ændringer		
Story 1: Override Switch				
Story 2: Data fra TS-applikation til Vægs Bevægelsessensor				
Story 12: CORS til REST				

Den største succeshistorie, på et rent fremskridts niveau, inden for dette sprint var vores arbejde med at få vores gyrosensor til at registrere bevægelse, som er kommet fint på plads efter en opfriskning af kode (user story #3).

Derudover har vi fået fuldendt #11 til et funktionelt stadie, dog efter at vi har ændret i kravene til denne user story uden at have konsulteret vores kunde. Dette vil vi komme mere ind på i vores retrospective.

Derefter har vi dog diverse user stories, som stadig ikke er kommet fuldt på plads.

#7, som består af vores administration af brugere er komplet men vi har dog glemt at få lavet en unit test for at vise at der sker en ændring i længden af en brugerliste, når vi fjerner en bruger. Derfor står denne stadig under Testing men forventes at være simpel at løse senere.

Vores kommunikation i #4 har fået samtlige elementer sat op, så der kan kommunikeres den ene vej mellem alle delsystemer. Dog går denne kommunikation ikke hele vejen igennem men i stedet foregår halvvejs for to forskellige elementer: Bruger og Sensor. Da denne user story kræver at kommunikation foregår hele vejen fra vores gyrosensor til vores HTML side, er denne

stadig In Progress og er muligvis blevet fejlestimeret da denne user story har vist sig at være den mest besværlige for os i løbet af 1. Sprint.

Undervejs i vores arbejde indså vi at vi allerede har taget delvis hånd om #9, da vi har sat database tabeller op, hvori vores indstillinger kan gemmes. Derfor ser vi denne som værende In Progress, selvom denne ikke skulle være igangsat på nuværende tidspunkt.

Til sidst har vi også taget enkelte skridt ind i #6 ved at oprette en HTML side for login. Denne er dog stadig ikke udført betydeligt, men er stadig blevet sat igang og derfor In Progress. Ud fra disse refleksioner er vi kommet frem til at vi har fået taget hånd på 13 af vores 17 story points, der stod i vores sprint Backlog for dette print.

Siden at vi ikke nåede vores planlagte velocity, planlægger vi at få det indhentet i sprint 2, eftersom det virker som om sprint 2 vil tillade det, rent arbejdsmæssigt.

6.4: Sprint retrospective

Ansvarlig: Frederik + Anders

Men vigtigst af alt er et indblik i vores arbejdsprocess i løbet af vores første sprint, som kunne være gået betydeligt bedre.

6.4.1: Test Driven Development

Ansvarlig: Frederik + Anders

Tilbage i sprint 0 havde vi tænkt os at prøve at arbejde med test driven development i løbet af sprint 1, da vi tænkte at det bedste tidspunkt for dette ville være i starten, hvor vi ikke havde noget kode at arbejde med. Men pga sygefravær og tekniske problemer med at forbinde til vores Raspberry Pi, og derfor kunne arbejde på vores gyrosensor, måtte vi revurdere vores prioritet af test driven development. Denne beslutning var også for at vi kunne gøre os nogen nævneværdige fremskridt i fremstillingen af vores software produkt, så test driven development blev ikke taget i brug på noget tidspunkt.

På nuværende tidspunkt ser vi ikke denne arbejdsmetode komme på banen igen, da det ideelle tidspunkt at gøre brug af den er passeret i vores øjne. Dette skyldes at test driven development involverer at man først opstiller en test, før at man begynde at bygge koden op. Samtidig foretrækker vi stærkt at kunne arbejde med test efter at vi har fået opsat kode på forhånd og derfor ser vi ikke denne mangel som et stort tab.

6.4.2: Pair Programming

Ansvarlig: Frederik + Anders

På den anden side fik vi allerede nu taget Pair Programming i brug. Dette skyldtes at vi begyndte at arbejde på systemer og kode, som vi ikke har megen erfaring med i forhold til hvad vi lærte i vores første studieår. Folk risikerede derfor ofte at køre fast og løsningen for os blev at sætte en eller flere andre medlemmer på som pair-programmere til at komme over disse forhindringer.

Disse tilfælde var arbejde med TypeScript og Python.

Vores første oplevelse med at tage Pair Programming i brug fremkom efter nogle timers arbejde om onsdagen på vores TypeScript applikation, som viste sig at være besværlig at arbejde med i forhold til C# kode, da vi var nødt til at arbejde på den samme fil, og versionsstyring blev et mindre mareridt, når alle tre i holdet prøvede at lave ændringer de samme steder.

Processen blev dog forbedret i løbet af dagen og vi kom ind i en mere optimal rytme, dog med enighed om at når vi arbejdede med denne type kodning, så burde vi holde os til at en enkelt person stod for at skrive mens resten hjalp til fra sidelinjen.

Dette blev så tilfældet om torsdagen, hvor vi endelig gik i gang med vores Gyrosensor, og nu holdt vi os til primært til en enkelt mand ved tastene, som gjorde processen betydeligt lettere at håndtere.

6.4.3: Review & Retrospective

Ansvarlig: Frederik + Anders

Da vi endelig nåede til fredag og skulle i gang med at evaluere vores arbejde i løbet af sprint 1 kom vi dog frem til at vi havde en meget kaotisk struktur og at vi nu, efter en længere dag, havde rigtig mange punkter at skulle komme igennem og skrive om for at officielt kunne blive færdige med sprint 1.

I forhold til vores review og retrospective, som begge var planlagt kun at bruge tid på i slutningen om fredagen, så kom vi frem til at vi burde prøve at sprede disse diskussioner ud over ugens strabadser. Dette tænker vi at gøre ved at diskutere individuelle user stories, når vi mener at disse er i deres endelige stadier. På denne måde bliver arbejdet bedre fordelt i løbet af ugen, samt at vi både får snakket om arbejdet, når det er mest friskt i hukommelse, og at vi har mere rapportskrivning på dagen udover fredag, hvor vi førhen kun havde vores Daily SCRUM.

6.4.4: Daily SCRUM

Ansvarlig: Frederik + Anders

Angående Daily SCRUM så har disse været et betydeligt element i vores kaotiske struktur, som ikke kan ses som en umiddelbar struktur overhoved da vores mangel på erfaring gjorde at disse møder primært blev til en enkelt person, som kom med de fleste ideer på varierende tidspunkter om morgenen, og pga dårlige perioder for andre medlemmer af holdet var vi aldrig fuldtallige til SCRUM møder.

Derfor har vi nu, efter en opfølgning på den typiske procedure at Daily SCRUM, besluttet af stramme op, så vi kan prøve at gøre korrekt brug af dette værktøj. Dette har vi tænkt os at gøre ved først og fremmest at fastsætte tidspunktet på vores møder til kl. 10:00. Herfra vil vi enkeltvis, i en fastsat rækkefølge, nævne:

- Hvad vi sidst har arbejdet på.
- Hvad vi vil gå i gang med på den givende dag.
- Hvad kan forekomme som problematisk i løbet af dagens arbejde.

Dette skal ikke tage længere end 10 minutter og med vores gruppe størrelse vil det give 3 minutter hver, hvor hvert emne får et enkelt minut.

6.4.5: Kanban Board

Ansvarlig: Frederik + Anders

Som tidligere nævnt, så valgte vi i 1. Sprint at arbejde på review om fredagen. Dette betød også at vi blev disorganiseret i forhold til vores Kanban Board, som blev opdateret løbende men ikke under de korrekte omstændigheder.

I den kaotiske process glemte vi at tjekke vores acceptance criteria inden at vi rykkede rundt på vores user stories. Dette gjorde at da vi kom til fredagen, fandt vi frem til at en del som vi mente var færdige, ikke var så langt pga anden formulering da vi satte vores user stories op i forhold til den fremgangsmåde vi senere kom frem til da vi begyndte at kode.

Derfor har vi gået imod den akkurate metode ved at fx sige at vores user story #11 er Done ud fra det resultat vi har opnået, men dette ville ikke være acceptabelt ellers når man arbejder med SCRUM. Den korrekte fremgangsmåde ville have været at vi diskuterede den givne ændring med kunden og argumenterede for, hvorfor dette ville være en bedre løsning på nuværende tidspunkt. Med denne erfaring gjort vil vi samt også stræbe på at være bedre til at opholde vores user stories som de er ud over #11 og i tilfælde af ændringer, tage dem op med kunden.

Med disse erfaringer blev vi derfor enige om at inden man rykker en user story, skal man huske at tjekke deres acceptance criteria og teste derudfra. Hvis alle disse er på plads kan man derefter flytte den givne user story til Done efter fælles vurdering, og når en user story er nået til Done skal der skrives i vores Sprint Review, så dette bliver gjort i forhold til vores nye fremgangsmåde per 2. Sprint.

6.4.6: Sustainable Pace

Ansvarlig: Frederik + Anders

For at vi bedre kan gøre brug af Sustainable Pace, så i 1. Sprint ikke udmundede sig til meget, bl.a fordi vi ikke havde meget at skrive om i løbet af ugen, pga vores ide om at review og retrospective skulle holdes til fredagen, har vi besluttet at fastlægge rapportskrivning til kl. 13 og derefter.

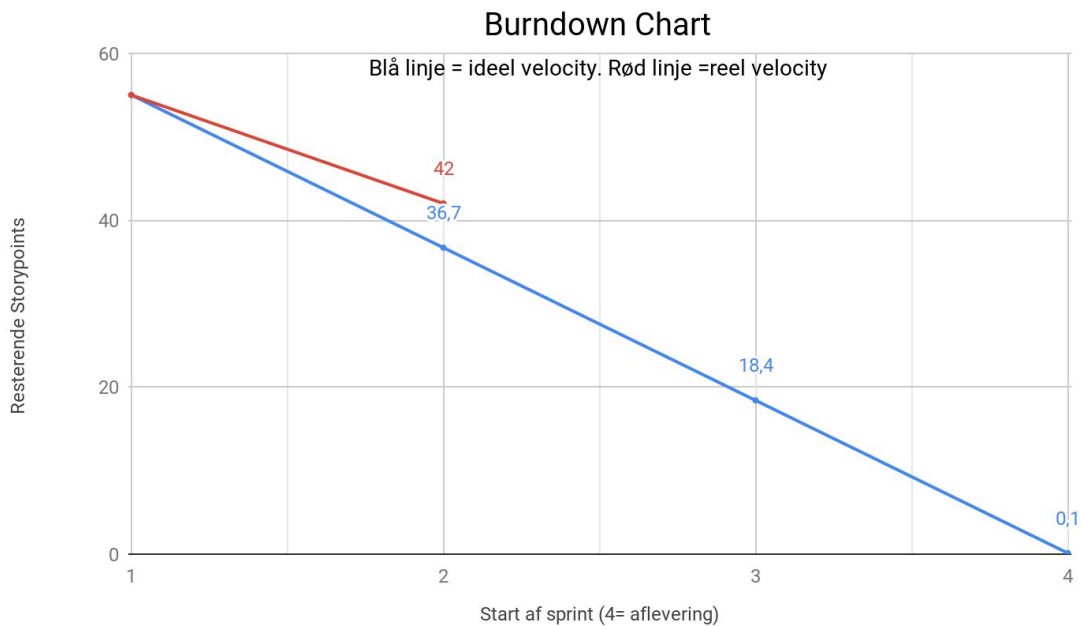
På denne måde kan vi garantere os at der altid bliver skrevet på rapporten og at der er tid sat af til gennemløbende review, når dette er tilfældet. Samtidig, i tilfælde af at der ikke er noget nyt at tilføje til rapporten, vil denne tid blive brugt på gennemlæsning og omskrivning, således at vi alle holder os opdateret på rapportens stadie.

Denne tid med fokus på rapportskrivning vil altid være på en lektions varighed af 45 minutter og dette, samt tidsplanen på Daily SCRUM og opdatering af Kanban Board, vil blive bedre opholdt af vores SCRUM Master, så vi kan få en betydelig bedre oplevelse af SCRUM i vores 2. Sprint.

Som det sidste punkt på 1. Sprint har vi opdateret vores Burndown Chart med vores aktuelle velocity som vi har udregnet til at være omkring 13 ved at kigge nærmere på vores estimering af user stories og hvor meget fremskridt vi har opnået for hver af dem. På nuværende tidspunkt finder vi denne velocity er i orden da vi mener at vi kan fange op på de manglende user stories i næste sprint.

6.4.6: Velocity

Ansvarlig: Frederik



Velocity er altså 5-6 mindre end vores ideelle velocity på mellem 18 og 19 sp/sprint. På dette tidspunkt var vi stadig optimistiske om vores evne til at indhente dette, da vi endnu ikke havde stødt på vores tungeste stories.

7: Sprint 2

7.1: Sprint Backlog

Ansvarlig: Frederik

Da vi dog har taget mindre hånd ved andre user stories i løbet af 1. Sprint ser vores Kanban Board således ud:

Product Backlog	Sprint Backlog	In progress	Testing	Done
Story 1: Override Switch	Story 10: On-off switch til TS-app	Story 4: kommunikation	Story 7: tilføj/fjern Brugere	Story 11: Pi Sensitivity
Story 2: Data fra TS-APP til Vægs Bevægelsessensor	Story 8: world clock API	Story 6: Styregrænsning v. Login		Story 3: Gyrosensor får input fra bruger
Story 12: CORS til REST	Story 5: refresh-rate	Story 9: Permanente ændringer		

Tanken her var at vi ville gå i gang med de næst vigtigste opgaver udvalgt af vores product owner, dog med den ekstra tilføjelse af story 8, som ikke var af stor vigtighed ud fra business value, men er vigtig for vores projekt da denne er et arkitekturkrav til programmeringsopgaven. Derfor har vi valgt at hæve denne i vigtighed og begynde arbejde på denne i løbet af dette sprint, så vi ikke risikerer at haste med brug af 3. Parts API i vores sidste sprint.

Vi har derfor valgt at få bedre styr på det vi har tilbage fra 1. Sprint, samt at fortsætte med dele af de user stories vi havde opstillet til vores 2. Sprint, primært #9, der allerede er sat i gang og #8 da denne som tidligere nævnt er krævet til vores projekt.

7.2: Daily SCRUM

7.2.1: Dec. 3

Ansvarlig: Anders

Desværre starter vores nye struktur af Daily SCRUM dårligt ud, da vi denne dag kun er en enkelt til stede mens resten af holdet er utilpas.

Derfor kan vi kun dele at Anders har arbejdet på rapport, som ikke blev færdiggjort for 1. Sprint og vil fortsætte med dette, så gruppen kan komme godt i gang med 2. Sprint, når folk er tilbage på benene. Det eneste umiddelbare problem med dette er enkelte dele af rapporten, der kræver nogle spørgsmål til en vejleder for at vi kan fastlægge hvor et par af vores user stories ligger i forhold til vores arbejdsprocess.

7.2.2: Dec. 5

Ansvarlig: Frederik + Anders

Vi har nu haft vores første, strukturerede Daily SCRUM og er kommet frem til følgende for hver enkelt medlem af gruppen:

- Frederik har sidste fredag arbejdet med Python og sikret at denne kode blev gemt andetsteds for garantiens skyld, men derudover ikke gjort andet pga flytning af lejlighed. Vil i dag kigge rapporten igennem og derefter assistere Joachim med TypeScript via Pair Programming og forudser ingen umiddelbare problemer udover dem stillet af Joachim selv.
- Anders har i mandags skrevet på rapport og i dag haft kommunikation med kunden, hvoraf mere af rapporten er blevet skrevet på. Vil arbejde på en unit test af vores user story #7 og rydde op i dele af koden fra 1. Sprint.

- Joachim har i fredags arbejdet på konstruktion af en login side i vores HTML og vil fortsætte med dette i dag. Dog forudses nogle problemer med hvordan værdier bliver henter fra vores database via TypeScript, som Frederik vil prøve at hjælpe med.

7.2.3: Dec. 6

Ansvarlig: Frederik + Anders

- Frederik har arbejdet med 3. Parts API, været Pair Programmer for Joachim i forb. med login ved at sætte et array op. Vil arbejde videre med 3. Parts API. Grundet at vores API leverer JsonP kode, forudser jeg besvær med at få det til at køre, da dette involverer installering af nye npm packages vi ikke er blevet undervist i. Hvis JsonP bliver for stort problem, bruger jeg UCT tidszone istedet for CET, da UCT fås i regulær JSON.
- Anders har bygget videre på user story #7 pga ændringer i database tabellen og ryddet op i dele af koden, samt forberedt basis elementer til brug i fremtiden. Dertil er der også arbejdet på noget error prevention men dette virker ikke fuldt ud endnu og er en af dagens punkter, som har givet et problem i at få en if-statement til at fungere med statuskoder.
- Joachim har arbejdet på login, men har mistet det meste af koden fra igår, grundet uheld med Github. Vil i dag sætte sig ind i login arrayet og genopbygge det tabte kode. Forudser vanskeligheder med kodningen af login funktionen. Frederik stiller sig til rådighed som pair-programmer.

7.2.4: Dec. 7

Ansvarlig: Frederik + Joachim

- Frederik har arbejdet med world clock API og er gået væk fra JSONP for i stedet at benytte alm. JSON. API'en skulle være consumeret, mangler kun en hurtig unit test og så er den klar til at flyttes til Done. Han vil foretage test i dag og ellers skrive rapport. Forudser visse vanskeligheder med at lave unit test.
- Anders er sygefraværende og derfor har Frederik i dag har taget rollen som SCRUM-Master. Har i går arbejdet med at lave en slider til at justere gyro-sensorens sensitivitet. Har, grundet sygefravær, ikke planlagt arbejde i dag.

- Joachim har arbejdet på login som der har været en masse hø at arbejde med. Brugte pair-programming i går som hjalp da hele gruppen deltog. Har brug for hjælp fra vejledere for at kunne få lavet login funktionen færdig. Vil kigge på rapporten og de andre user stories få at se om der er andre ting der kan gennemgås og måske gå i gang med en ny user story for at indhente tabt tid. Det eneste problem i dag er med at gjort login færdig, som har været et enormt problem.

7.3: Sprint Review

Ansvarlig: Alle

Ved afslutning af 2. Sprint så vores Kanban Board sådan ud:

Product Backlog	Sprint Backlog	In progress	Testing	Done
Story 1: Override Switch	Story 10: On-off switch til TS-applikation	Story 4: kommunikation		Story 11: Pi Sensitivity
Story 2: Data fra TS-APP til Vægs Bevægelsessensor	Story 5: refresh-rate	Story 6: Login		Story 3: Gyrosensor får input fra bruger
Story 12: CORS til REST		Story 9: Permanente ændringer		Story 8: world clock API
				Story 7: tilføj/fjern Brugere

Vi havde altså kun kunne gennemført 8 nye points, i forhold til de 18 vores ideelle velocity dikterede. Dette kan forklares med skæv estimering af sværhedsgraden ved vores user stories, samt at alle gruppe-medlemmer havde en dag med sygefravær under dette sprint. Vi har således hver kun arbejdet tre dage ud af de fire planlagte i denne uge, hvorved det er rimeligt at antage at vi mistede 25% af vores velocity samlet.

Grundet arbejde og komplikationer med andre user stories har vi ikke fået gjort meget på #4 omkring kommunikation mellem vores systemer ud over en enkelt dags opsætning af nye HTML sider og forsøg på at opsætte en slider, der ideelt skulle kunne justere sensitivitet.

Login har været en besværlig kode, som var udfordrende både teoretisk og praktisk. Koden der skulle skrives, minder om noget som vi har programmeret før, men at det foregår i TypeScript har givet flere uventede problemer, der ikke kunne løses på egen hånd. Derfor har der været brug for pair-programming for at bedre kunne få skrevet koden. Denne user story har derfor været en for stor forhindring, der har holdt os tilbage fra at fortsætte med andet kode og vi har overskrevet vores tidsbegrænsning på denne user story. Login user story må fortsættes i 3. Sprint eller springes over hvis ingen løsning findes.

#7 er nu færdiggjort efter at der blev lavet en Unit Test, som var den sidste del af vores acceptance criteria. Dog pga ændringer i koden for at denne Unit Test kunne udføres var det nødvendigt at ændre andre dele af kode i forskellige delsystemer, men som også er blevet bragt i orden.

Brugen af 3. Parts API i user story #8 var den eneste story fra 2. Sprint som vi fik færdiggjort. Selvom funktionaliteten er på plads, har vi dog problemer med at kalde den effektivt via Multi-Threading, som ville optimere brugen i forhold til vores umiddelbare løsning.

Vi har derfor endnu engang ikke haft den ønskede velocity i vores sprint, hvilket af denne omgang virker til at skyldes uerfarenhed inden for bestemte typer af kode. Samtidig har vi også tydeligt fejlestimeret user stories, som fx. #6 angående login, en opgave vi ville have mindre problemer med inden for XAML og C# kode, men grundet vores nuværende erfaring med TypeScript og HTML var en betydeligt større opgave end vi troede.

7.4: Sprint Retrospective

7.4.1: Pair Programming

Ansvarlig: Joachim

Pair programming har vist sig at være en mere brugbar metode end som så, da det har hjulpet stærkt med et kæmpe problem som der blev stødt på i form af user story #6.

Som tidligere nævnt estimerede vi denne til en mindre mængde af story points, men når det skal foregå i kode man ikke er stor erfaring i, bør der afsættes mere tid til den.

I næste sprint vil der helt klart bruges mere pair programming, hvis man sidder fast med en opgave. Specielt da en del af vores arbejde består af TypeScript, som vi fælles mener er vores svageste punkt pga mindre struktureret undervisning og træning inden for denne type kode.

7.4.2: Daily SCRUM

Ansvarlig: Frederik

Daily SCRUM fungerer betydeligt bedre i forhold til vores 1. Sprint. Det virker til at være en simpel, men effektiv måde at holde sikker kommunikation mellem medlemmerne på holdet. På den nuværende måde giver det også et let overview over, hvad der er eller kan blive et problem for projektet. Møderne giver mere retning med arbejdet og man reflekterer mere over sit arbejde og sine mål, når man skal formulere dette overfor de andre teammedlemmer. Den punktformede struktur forbedrer også effektiviteten af mødet som koordineringsværktøj betydeligt.

7.4.3: Sustainable Pace

Ansvarlig: Frederik

Sustainable pace fungerer bedre under denne sprint, men kan være lidt begrænsende for tidligere sektioner af rapporten. Den har sikret at vi får lavet, hvad vi skal når tiden kommer, men reglen har været sværere at overholde, hvad enten på grund af over- eller under arbejde.

Der kan komme, og har været uventede problemer, hvor vi følte at vi ikke burde stoppe med at arbejde, selvom at sustainable pace sagde at det var tid til at fokusere på rapporten alene. Vi har flere gange i dette sprint udskudt ophør af kodning for lige at få den sidste funktion til at give det rigtige resultat og lignende situationer. Dertil skal siges, at hjælp fra programmerings- og teknik vejledere til tider er kommet udenfor timerne vi måtte kode, som har gjort vores sustainable pace mere forhindrende end vi ønskede.

7.4.4: Review & Retrospective

Ansvarlig: Anders

Vores ændring af fokus på hvornår vi kigger tilbage på vores arbejde har forbedret sig en del ved at vi løbende laver noter og diskutere enkelte user stories, når de når et nyt punkt af enten succes eller frustration.

Vi vil derfor fortsætte med denne fremgangsmåde i forhold til at placere alt dette arbejde på den sidste dag af vores sprint, så dagene flyder bedre sammen.

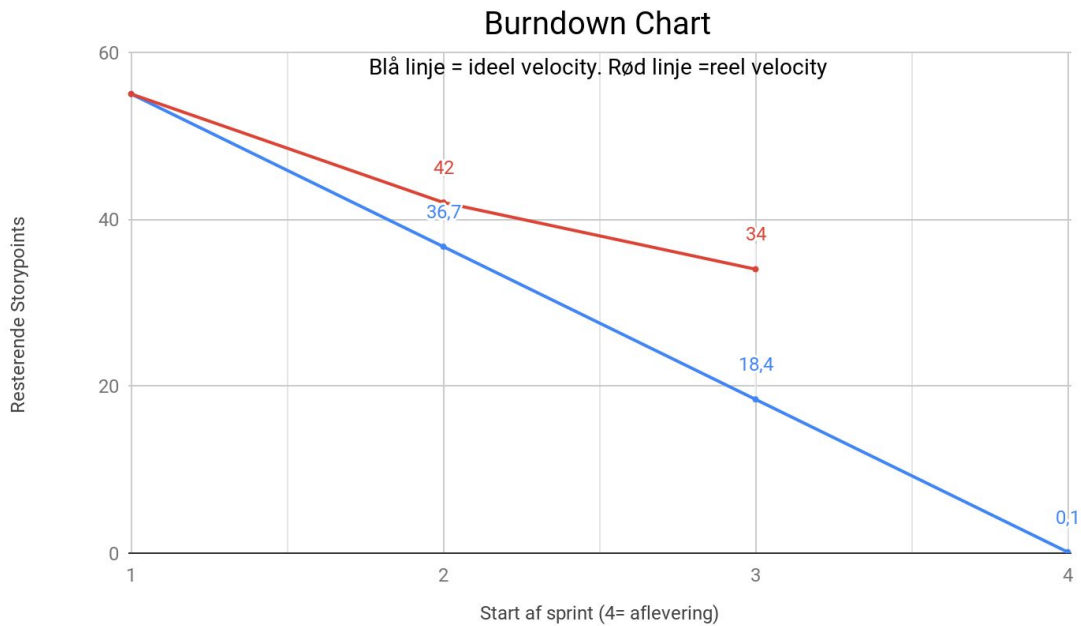
7.4.5: SCRUM Master

Ansvarlig: Frederik

Vi havde torsdag d. 6 en uformel forberedende snak til retrospektive om hvem, der skulle varetage rollen som SCRUM Master. Det blev her besluttet at Anders umiddelbart godt kunne vedblive i rollen for at sikre stabilitet i de grupper rutiner vi har fået oprettet igennem 2. Sprint. Pga. Anders' sygefravær d. 7, har Frederik og Joachim diskuteret om Anders skal aflastes og er kommet frem til at i tilfælde af Anders fravær vil Frederik tage rollen på sig. Når Anders vender tilbage vil vi spørge ham om han ønsker at fastholde rollen eller om Frederik skal være SCRUM Master i 3. Sprint.

7.4.6: Velocity

Ansvarlig: Frederik



Efter de forskellige problemer vi er kommet ud for i løbet af dette sprint, kan vi se at vores velocity er faldet til 8 story points. Vi har derfor valgt at vi i planning game for 3. Sprint vil fjerne mange user stories, for at reducere vores scope markant. De stories vi arbejder med på nuværende tidspunkt er tungere end vi forventede, og dette sænker vores velocity. Forklaringen ligger i vores begrænsede erfaring med andre kodesprog end C#.

8: Sprint 3

8.1: Sprint Backlog

Ansvarlig: Anders

Da vi mødte op til starten af 3. Sprint var status på vores Kanban Board det samme som i slutningen af 2. Sprint. Dog har vi pga. vores refleksioner valgt at fokusere på de user stories vi på nuværende tidspunkt har In Progress, da vi tydeligvis har overvurderet vores arbejdshastighed, når det kommer til de nye kodesprog vi arbejder med i dette projekt.

Derfor har vi valgt at lægge det meste af vores Backlog til siden, da vi i øjeblikket ikke mener at det vil lykkedes os at nå betydeligt fremskridt med disse og derfor mener at vi bør gøre hvad vi kan for at vores eksisterende kode bliver bragt i orden. Dog har vi beholdt user story #12 i vores sprint Backlog, da denne virker som noget vi kan få sat orden på i tilfælde af at vi får løst problemerne med de user stories, som er i gang.

Product Backlog	Sprint Backlog	In progress	Testing	Done
	Story 12: CORS til REST	Story 4: Kommunikation		Story 3: Gyro sensor får input fra bruger
		Story 6: Login		Story 7: Tilføj/fjern Brugere
		Story 9: Permanente ændringer		Story 8: World clock API

				Story 11: Pi Sensitivity
--	--	--	--	-----------------------------

Som det kan ses på boardet, har vi på dette tidspunkt valgt at skære ind til benet og reducere vores scope betydeligt. Denne beslutning tages, fordi det er tydelig for os, at vores velocity ligger lige under halvdelen af det krævede for at gennemføre alle 55 story points vi havde estimeret projektet til. Dette er hovedsageligt fordi vi har estimeret flere stories omkring TS-applikationen for lavt, i forhold til hvor meget arbejde de i praksis krævede af os. Denne markante reduktion i scope burde teknisk set være sket mere gradvist i løbet af vores tidligere planning games, med start i sprint 2. For at sikre et nært samarbejde med vores product owner, ville vedkommende skulle have sidste ord omkring hvilke stories der skulle skrottes. Grundet de praktiske begrænsninger i adgang til vores product owner, Vibeke, har vi selv ageret product owner i denne forbindelse.

8.2: Daily SCRUM

8.2.1: Dec. 10

Ansvarlig: Anders

- Frederik har været med til at hjælpe med login og været SCRUM-Master mens Anders var sygemeldt. Han er desværre ikke til stede under dagens møde pga ukendte årsager.
- Anders har kigget rapporten igennem efter hans fravær og har redigeret efter de andres arbejde. Dette vil han fortsætte med, samt finde frem til hvilke user stories, der bør lægges til side indtil gruppen er på rette vej igen. Kan dog være svært at fuldt beslutte uden hele gruppen til stede.
- Joachim har fortsat kampen med vores login og har fået løst en del fejl indtil videre. Arbejdet vil fortsætte i dag, men hvis denne user story ikke bliver bragt i orden i løbet af dagen vil der muligvis tages fat i en ny opgave. Tidligere problemer med denne user story fortsætter ind i dagens arbejde.

8.2.2: Dec. 11

Ansvarlig: Joachim + Anders

- Frederik er stadig fraværende og har derfor ingen aktive opgaver for i dag.
- Anders har arbejdet på et import problem af ts filer som han vil fortsætte med i dag, eller prøve at finde en anden løsning da dette har vist sig at være et større problem i forhold til at kunne arbejde videre med at kode.
- Joachim har fået løst problemerne med login og vil bruge dagen på oprydning af TS og GUI, som ikke menes at skulle forårsage nogen problemer.

8.2.3: Dec. 12

Ansvarlig: Joachim + Anders

- Frederik er endnu engang fraværende og fremgangsmåde herfra vil bringes på bane senere på dagen.
- Anders har gjort user story #9 færdig og fået fint fremskridt med #4 og vil i dag først og fremmest kigge på hvorfor diverse knapper ikke virker mere og derefter arbejde på hans tidligere error prevention for brugere. Ingen umiddelbart store problemer forudses.
- Joachim: Vil bruge dagen på at fuldføre alle login criteria. Bruger info skal stå i hjørnet af hjemmesiden. Gøre Login siden til den første side man kommer på.

8.2.4: Dec 13

Ansvarlig: Alle

- Frederik har været pair programmer på Login torsdag d. 12/12, efter han dukkede op. Har også nedfældet ideer til rapporten fra vores samtaler med Peter Levinsky. Planlægger at hjælpe med færdiggørelsen af Login som pair programmer, samt færdiggørelse og renskrivning af rapporten.
- Anders har forbedret error prevention på user story #7 og har været med til pair programming for #4. Vil i da rydde lidt op i koden, sætte kommentarer og ellers kigge på rapport.

- Joachim har arbejdet på at færdiggøre vores sidste acceptance criteria på Login og planlægger at bruge dagen på rapporten og at færdiggøre login, hvis muligt.

8.3: Sprint Review

Ansvarlig: Alle

Ved vejs ende af vores 3. Sprint og projektet er vores endelig Kanban Board nået til dette punkt:

Product Backlog	Sprint Backlog	In progress	Testing	Done
				Story 3: Gyrosensor får input fra bruger
				Story 4: Kommunikation
				Story 6: Login
				Story 7: Tilføj/fjern Brugere
				Story 8: World clock API
				Story 9: Permanente ændringer
				Story 11: Pi Sensitivity

Vi er nu endelig fået løst user story 4 ud fra dens acceptance criteria, som krævede at vi kunne hente info fra sensoren på vores HTML side. Man kan argumentere for at dette skulle kunne ske uden nogen interaktion, som på nuværende tidspunkt er indtastning af ID og tryk på en knap, men da dette ikke har været specificeret og denne user story kun krævede at rejsen kunne fuldendes, er denne nu officielt Done.

Blandt andre længere stående user stories har vi nu også endelig fuldført vores Login i user story 6. Denne har helt klart også været et tilfælde af fejlestimering, som kan ses ud fra vores mange problemer med kode vi ikke var vant til og fremgangsmåder vi ikke kendte til. Mens at vi

fx tænkte at der skulle være forskellige sider, kunne man også have valgt at arbejde med en enkelt side, og når man fik logget ind vil Login elementerne fjernes og udskiftes med elementerne man ønskede i fx en hovedmenu. Samt fik vi også lært om at gøre brug af cookies for at løse det sidste acceptance criteria om at brugernavnet skulle stå på siden, hvilket vi brugte meget tid på at komme frem til pga vores setup med flere sider.

Da vi har fået sat vores kommunikation op kunne vi også færdiggøre user story 9. Denne krævede at man kunne gemme indstillinger til gyrosensoren fra hjemmesiden, og at disse ville blive gemt, selv når siden er lukket ned. Selve muligheden for at gemme permanent var allerede på benene via vores database. Det krævede derfor kun en smule tilføjelse af GUI med enkelte funktioner til at justere på sensorens sensitivitet. Dette var dog i første omgang besværligt pga. vores initiale opsætning med flere sider. Efter megen besvær valgte vi at samle vores essentielle funktioner på vores hovedmenu. På denne måde kom koden let på benene og vi kunne nu justere gyrosensorens sensitivitet fra både hjemmesiden og på selve sensoren.

Som man kan se i starten af vores 3. Sprint, havde vi beholdt en enkelt user story da vi reducerede vores scope. Dette var user story 12 om at tilføje CORS til vores REST service for den ekstra sikkerhed. Desværre endte vi med at skulle bruge meget tid på vores igangværende user stories og kom derfor ikke i gang med user story 12. Denne er derfor, her til slut, blevet fjernet sammen med de resterende stories, der måtte lægges til side.

8.3.1: Sidedesign

Ansvarlig: Frederik

Som en sidste ting angående vores produkt vil vi kort nævne lidt om vores design og arkitektur som kan ses nedenfor:

Thursday 13/12/2018 11:14 CET

frsh

Log ud

Bruger Administration

Brugerliste

Hent

ID for specific item.

input

Tilføj/Fjern

Tilføj

Bruger ID (Tal)

Brugernavn (Max 4 tegn)

Kodeord (Max 20 tegn)

Fjern

Bruger ID (Tal)

Sensor Indstillinger

Vælg sensor

Hent

Sensor ID (Tal)

Sense (+)

Sense (-)

Værd at bemærke er at siden adskiller sig fra vores domænemodel,¹ da vi har samlet elementer fra tre HTML-sider på én enkelt. Dette var en måde at reducere i scope på da Login-siden viste os at flytning af data mellem flere HTML-sider var ekstremt arbejdskrævende. Denne afvigelse fra vores domæne model betragter vi dog som helt naturlig, I og med vi sagde fra dag 1 at vores brug af UML udelukkende var som værktøj til idegenerering.²

¹ Punkt 5.7, side 24

² Punkt 4.3.1, side 16

8.4: Sprint Retrospective

8.4.1: Pair Programming

Ansvarlig: Anders

Pair Programming har endnu en gang været i brug en hel del i løbet af dette sprint og er nu blevet til en bestemt favorit for at løse større problemer inden for at skrive kode. Det er blevet meget naturligt at dette er vores generelle løsning for at håndtere komplikationer og vi er yderst tilfreds med at have gjort brug af denne metode siden starten af vores arbejdsproces.

8.4.2 SCRUM Master

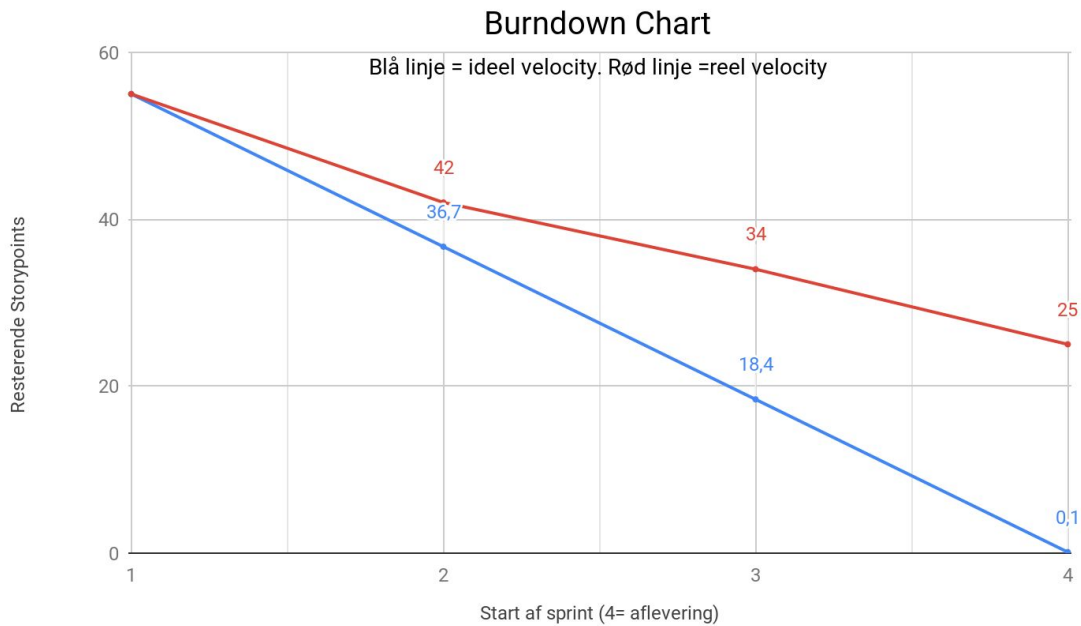
Ansvarlig: Anders

Diskussion om at bytte rollen som SCRUM Master fra at være Anders til Frederik kom desværre ikke på bane da Frederiks fravær pga. stress gjorde denne overførelse irrelevant. Anders ville alligevel have stået med rollen de først tre dage på ugen.

Dog var der ikke den store nødvendighed for dette rolleskift og arbejdet fortsatte som det skulle. Det ville dog stadig havde været rart med muligheden for officielt at bytte rolle igen, så det ikke kun var Anders, der prøvede at være ansvarlig for den daglige tidsramme og generelt overblik over vores user stories.

8.4.3: Velocity

Ansvarlig: Frederik



I vores endelige sprint nåede vi en velocity på 9 point. På dette tidspunkt var det besluttet at vi udelukkende gennemførte story 6 - Login med 1 point tilbage ud af 3, story 4 - Kommunikation, på 5, og story 9 - Permanente ændringer med 3 point. Dette var en beslutning taget ud fra vores 8 points velocity i den foregående sprint.

9: Diskussion

Ansvarlig: Anders

Vi vil nu komme dybere ind på nogle overordnede punkter omkring vores projekt, som vi har gjort os i løbet af vores arbejde med SCRUM og XP.

9.1: Estimering af user stories

Ansvarlig: Frederik + Joachim

I tilfælde af fremtidige SCRUM projekter skal vi blive bedre til at gennemskue, hvilke arkitekturkrav der er forbundne med vores user stories og deres acceptance criteria, inden at vi estimerer dem. Vi var også for optimistiske omkring estimering af andre typer kode, specielt TypeScript og derved fik vi estimeret langt fra hvor vi burde være.

Det store eksempel er vores Login i user story #6, som vi husker som værende simple, hvis vi havde arbejdet med C# og XAML. Dog skulle vi nu lave et lignende system i TypeScript og HTML, som nu var betydeligt mere kompleks pga vores manglende erfaring. Dog har vi nu lært at vi skal være betydeligt mere skeptiske over for user stories, der involvere kode vi ikke har arbejdet med før i en projekt relateret situation.

En anden del af vores fejlestimering kom også fra at vi kom frem til en arkitektur, der havde flere HTML-sider at arbejde med. Dette skyldes også vores tidligere erfaringer, specifikt med MVVM-arkitekturen, og vi overvejede derfor slet ikke en single-page arkitektur. At arbejde med flere sider i HTML var samtidig også et aspekt vi ikke havde arbejdet med overhoved før projektet, så det var en yderst ambitiøs beslutning af tage i vores optimistiske sind. Men pga dette valg fik vi samtidig gjort os nogle erfaringer inden for dette område og derved har lært mere end vi måske ville hvis vi havde holdt os til en anden fremgangsmåde. Alt i alt havde vi generelt undervurderet tyngden af flere af vores user stories.

En anden måde at sikre en bedre velocity ville være at foretage nogle programmeringsopgaver som gruppe inden projektet gik i gang for alvor, for at få en bedre fornemmelse for vores velocity og en bedre erfaring inden for den type kode, som vi primært endte med at arbejde på i projektet.

9.2: SCRUM som arbejdsproces

Ansvarlig: Alle

Som det sidste inden vores konklusion at projektets problemformulering vil vi hver især komme ind på vores umiddelbare tanker om hvordan det har været at arbejde med SCRUM.

9.2.1: Joachim

Ansvarlig: Joachim

At arbejde med SCRUM har været nyt, svært men brugbart. Den har været brugbar når det kommer til få den nødvendige mængde af kontrol, men den har ladet os tilpasse vores projekt, hver gang vi løb ind i de problemer som der har opstået, ved hjælp af struktur fra diskussioner og SCRUM masterens overblik. En brugbar men lidt stressfyldt process, der kræver ekstra papirarbejde. Den kan sagtens blive brugt i andre projekter, eller ihvertfald flere metoder fra den kan sagtens bruges igen.

9.2.2: Anders

Ansvarlig: Anders

Det har været en yderst interessant arbejdsproces, som i starten var en smule kaotisk indtil der kom en overordnet struktur, der gav mere ro over arbejdet. At få opsat denne struktur i starten er tydeligvis også en smule problematisk, når man er i en gruppe som består af folk, som ikke er specielt komfortable med at interagere på større basis og derfor kan gøre det svært at få nogen til at tage beslutningen af at få igangsat en fast struktur. Men med lidt hjælp blev det initielle skridt taget og derefter har processen været udmærket.

Dertil har det været rart ikke at have megen dokumentation at holde styr på i forhold til fx UP hvori man hyppigt kan komme ud for at man skal opdatere diverse diagrammer, når man tilføjer et nyt aspekt i sin overordnede arkitektur.

9.2.3: Frederik

Ansvarlig: Frederik

SCRUM er som projektstyringsmetode udmærket: Det flade hierarki og fokus på klar kommunikation, både ml. SCRUM master og product owner, samt ml. team og SCRUM master/internt i teamet, gør det lettere at forventningsafstemme og sikre effektivt samarbejde, både internt og eksternt. Især SCRUM møderne er guld værd til at koordinere teamet effektivt. Review sikrer at product owner har tillid til udviklerne, og SCRUM/kanban boards samt burndown chart, gør det let visualisere fremskridt. Velocity giver os muligheden for bedre at kunne lave realistiske planer, selvom den i dette projekt var noget svingende.

10: Konklusion

Ansvarlig: Alle

Ved brug af SCRUM som vores agile arbejdsproces, kan vi ikke nødvendigvis få lavet alt det planlagte arbejde. men vi er bedre istand til at kunne takle alle de uforudsigelige problemer vi støder på.

Vi skulle i dette projekt bruge både en Raspberry Pi og vi skulle skrive kode i TypeScript. Begge er felter som vi kun har minimal erfaring i og det var sikkert at vi ville løbe ind i problemer. men vi var for optimistiske. Vi blev alligevel forsinket flere gange men kunne ved hjælp af vores agile udviklingsmetode nedskære vores scope tilpasse vores projekt til den resterende tid vi havde tilbage.

Desuden har vi erfaret at vi, vha. 3. Parts API, kan få ekstra funktionalitet til vores applikation. I vores tilfælde endte det med at være noget så simpelt som en mulighed for dato og klokkeslæt på vores HTML. Men selv noget så simpelt har givet os en erfaring inden for at arbejde med

elementer fra 3. Part, som er en vigtig del af programmering. Dog skal det siges at 3. part elementer indebærer sine egne problemer, da programmets funktion og sikkerhed ender med at hvile på elementer, man ikke har magt over. På den anden side betyder dette at man som udvikler skal yde mindre support til sit program i det lange løb og derfor er det en vigtig beslutning at gøre sig i løbet af et projekt: Om man vil lave alt fra bunden eller søge hjælp udefra.