

# Prosjekthåndbok

Vedlegg D

Bacheloroppgave 2019

Oppgave 62

Dataingeniør

NTNU, institutt for datateknologi og informatikk

Trym Vegard Gjelseth-Borgen  
Magnus Conrad Hyll

Trondheim, mai 2019

## Innholdsfortegnelse

1	Innledning.....	4
2	Fremdriftsplan .....	4
3	Møteinncallinger med referat .....	6
3.1	Veiledermøter + oppstartsmøte.....	6
3.1.1	Oppstartsmøte (10.01.2019) .....	6
3.1.2	Veiledermøte 1 (05.02.2019) .....	9
3.1.3	Veiledermøte 2 (26.02.2019) .....	11
3.1.4	Veiledermøte 3 (18.03.2019) .....	13
3.1.5	Veiledermøte 4 (11.04.2019) .....	15
3.1.6	Veiledermøte 5 (02.05.2019) .....	18
3.1.7	Veiledermøte 6 (14.05.2019) .....	22
3.2	Sprint Review/Planning-møter .....	26
3.2.1	Sprint 1 Review/Planning (08.02.2019) .....	26
3.2.2	Sprint 2 Review/Planning (14.02.2019) .....	28
3.2.3	Sprint 3 Review/Planning (21.02.2019) .....	30
3.2.4	Sprint 4 Review/Planning (01.03.2019) .....	33
3.2.5	Sprint 5 Review/Planning (25.03.2019) .....	35
3.2.6	Sprint 6 Review/Planning (01.04.2019) .....	37
3.2.7	Sprint 7 Review/Planning (08.04.2019) .....	39
3.2.8	Sprint 8 Review/Planning (12.04.2019) .....	41
3.2.9	Sprint 9 Review/Planning (24.04.2019) .....	43
3.2.10	Sprint 10 Review (14.05.2019) .....	45
4	Timelister med statusrapporter .....	48
4.1	Uke 2, 3 og 4 .....	48
4.2	Uke 5.....	50
4.3	Uke 6.....	51
4.4	Uke 7.....	53
4.5	Uke 8.....	55
4.6	Uke 9.....	57
4.7	Uke 10 og 11.....	59
4.8	Uke 12.....	61
4.9	Uke 13.....	63
4.10	Uke 14.....	65
4.11	Uke 15.....	67
4.12	Uke 16.....	69

4.13	Uke 17.....	71
4.14	Uke 18.....	73
4.15	Uke 19.....	75
4.16	Uke 20 og 21.....	77
5	Sprint Boards .....	79
5.1	Sprint 1 .....	79
5.2	Sprint 2 .....	80
5.3	Sprint 3 .....	81
5.4	Sprint 4 .....	82
5.5	Sprint 5 .....	83
5.6	Sprint 6 .....	84
5.7	Sprint 7 .....	85
5.8	Sprint 8 .....	86
5.9	Sprint 9 .....	87
5.10	Sprint 10 .....	88

## 1 Innledning

Dette dokumentet inneholder dokumentasjon i forbindelse med prosessen for utførelsen av prosjektet. For å dokumentere prosessen har vi brukt framdriftsplan, møteinnkallinger og referat, timelister, statusrapporter og Sprint Boards.

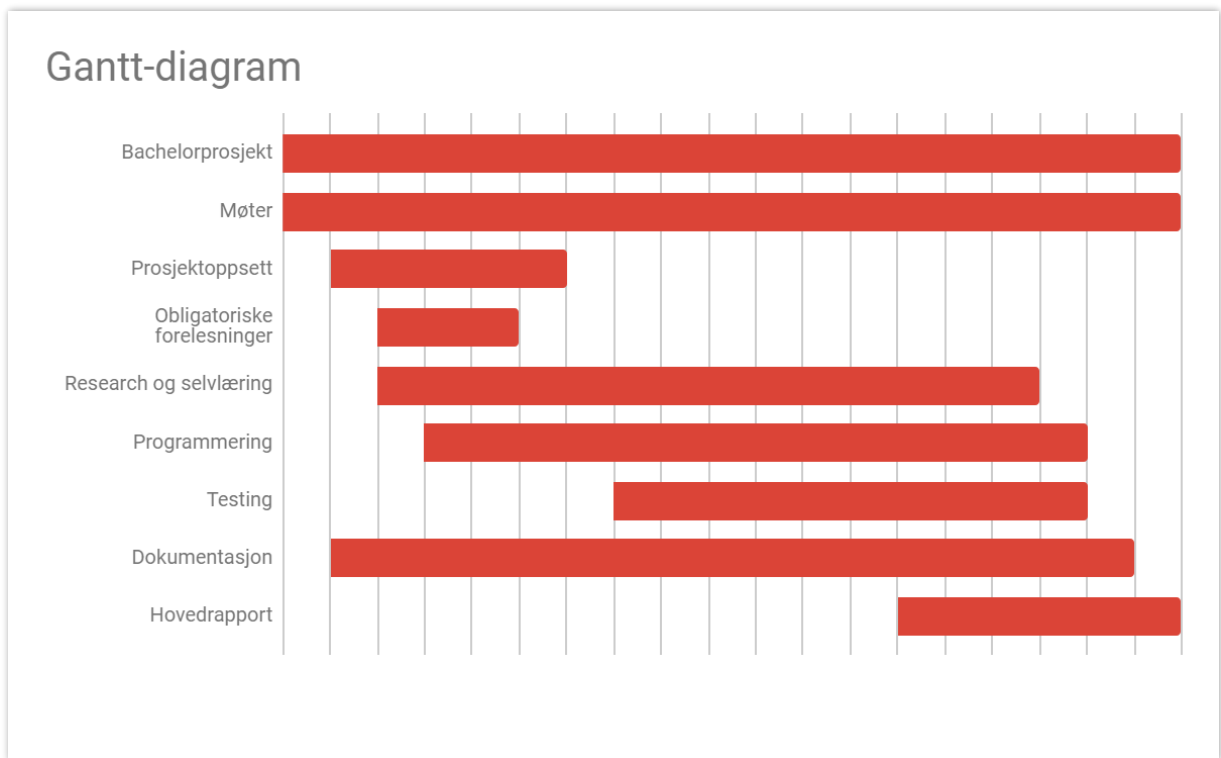
## 2 Fremdriftsplan

I starten av prosjektet lagde vi en framdriftsplan i form av en Gantt-diagram. Gantt-diagrammet viser et overordnet blick av prosjektets struktur, og er med på å definere milepæler for prosjektet.

For en mer definert fremdriftsplan se kapittel 5 - Sprint Boards. Her tar vi utgangspunkt i utviklingsprosessen Scrum, og definerer framdriftsplan for hver Sprint, samt dokumenterer resultat etter ferdig Sprint.

*Tabell 2-A Planlagt fremdriftsplan for prosjektet.*

Aktivitet	Startdato	Antall timer	Sluttdato	Dager
<b>Bachelorprosjekt</b>	<b>7. jan.</b>	<b>1000</b>	<b>20. mai</b>	<b>133</b>
Møter	7. jan.	50	20. mai	133
Prosjektoppsett	14. jan.	20	18. feb.	35
Obligatoriske forelesninger	21. jan.	20	11. feb.	21
Research og selvlæring	21. jan.	70	29. apr.	98
Programmering	28. jan.	400	6. mai	98
Testing	25. feb.	120	6. mai	70
Dokumentasjon	14. jan.	120	13. mai	119
Hovedrapport	8. apr.	200	20. mai	42
<b>Timesum</b>		<b>1000</b>		



Figur 2-1 Gantt-diagram for planlagt fremdriftsplan for prosjektet.

### 3 Møteinnkallinger med referat

#### 3.1 Veiledermøter + oppstartsmøte

##### 3.1.1 Oppstartsmøte (10.01.2019)

#### **Innkalling**

Oppstartsmøte for Bachelorprosjekt:

E39 Fjordkryssing: Automatisering av data-pipeline for analyse av bølgedata

Tidspunkt/sted: Torsdag 10.01.2019, kl. 13:00 – 14.30, Klæbuveien 127B, Møterom 521

Følgende personer innkalles:

Atle Olsø (veileder)

Andreas Ravnestad (kontaktperson hos Norconsult)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Sak nr 01/2019	Gjensidig introduksjon.
Sak nr 02/2019	Gjennomgang av bakgrunn og hensikt.
Sak nr 03/2019	Formulere problemstilling som definerer prosjektet.
Sak nr 04/2019	Gjennomgang av prosjekthåndboka og rapportering/møter/innleveringer underveis.
Sak nr 05/2019	Innlevering, presentasjon og vurdering.
Sak nr 06/2019	Annet praktisk (tilganger, verktøy, deling av ressurser osv.)
Sak nr 07/2019	Avtalesignering.
Sak nr 08/2019	Eventuelt.

Møtet planlegges avsluttet ca kl. 14.30.

Ta kontakt med undertegnede dersom du ikke har anledning til å komme.

Mvh.

Magnus Conrad Hyll

Trondheim 09.01.2019

## **Referat**

Dato og tid: 10.01.2019, kl. 13:00-14:30

Sted: Klæbuveien 127B, Møterom 521

Til stede: Trym, Magnus, Andreas, Atle, Athul, Onno

Frafall: Ingen

### **Sak nr 01/2019 Gjensidig introduksjon**

Partene introduserer seg.

### **Sak nr 02/2019 Gjennomgang av bakgrunn og hensikt**

Bakgrunn for prosjektet presenteres.

Introduksjon av dagens løsning på problemet.

### **Sak nr 03/2019 Formulere problemstilling som definerer prosjektet**

Ta noen arbeidsmøter. Begynne å utforme en produktbacklog. Mulighet for bruk av maskinlæring. Bruk av EPPlus, for excel regneark.

Teknologivalg som må tas, og begrunnes for rapportens del.

### **Sak nr 04/2019 Gjennomgang av prosjekthåndboka og rapportering/møter/innleveringer underveis**

Lage framdriftsplan og lage timelister med framdriftsrapport.

Selvvalg av metode.

Trello høres ut som det foretrukne system for framdriftsdokumentasjon.

Dokumentere fra trello til prosjekthåndbok.

Ta skjermbilde med gjevne tidsintervall med beskrivelse fra trello til rapport.

Timeliste hver uke. Statusrapport etter hver sprint (andre, tredje uker).

Mulig ukentlige, kanskje daglige sprinter i starten.

Korte sprinter de 2, 3 første ukene, for å komme unna de første humpene.

Få fram essensen i hva som er gjort og hvordan vi tenker. Vi trenger en prosjekthåndbok som viser framdrift og timeliste. Ikke behov for Gantt-diagram.

Dokumentere hvorfor vi har valgt de prosjektmetodene vi har valgt.

Kan være greit med Gantt-diagram, men på milepæls-nivå.

**Sak nr 05/2019 Innlevering, presentasjon og vurdering**

Hovedrapport med alt det som står i prosjekthåndboken.

Ingen innlevering av kildekode, ettersom at den er lukket, men veileder må ha tilgang til koden.

Presentasjon en uke etter innlevering, hos oppdragsgiver. Vurdering med karakter. To sensorer, Atle + 1.

Innlevering av tittelside i april. Dette er problemstillingen, som vi har formulert.

**Sak nr 06/2019 Annet praktisk (tilganger, verktøy, deling av ressurser osv.)**

Quiz hver dag klokka 1400.

Gratis lunch.

Får tilganger, adgangskort, PCer og kontorplass fortløpende når det blir klart.

**Sak nr 07/2019 Avtalesignering**

Partene signerer samarbeidsavtalen.

**Sak nr 08/2019 Eventuelt****Møter**

Mulighet for flere møter underveis, inkludert alle parter.

Planning og review-møter, fram Norconsult sin del.

Møte med veileder for selve bacheloren (ikke det tekniske).

10.01.2019, Trym Vegard Gjelseth-Borgen



### 3.1.2 Veiledermøte 1 (05.02.2019)

#### **Innkalling**

Tidspunkt/sted: Tirsdag 05.02.2019, kl. 15.00-16.00, Klæbuveien 127B, Møterom 522

Følgende personer innkalles:

Atle Olsø (veileder)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

- Gjennomgang timelister og statusrapporter
- Dokumentering underveis (timelister, statusrapporter, sprint board osv.)
- Visjonsdokument
- Problemstilling
- Arbeid videre til neste møte
- Eventuelt

Mvh.

Magnus Conrad Hyll

Trondheim, 01.02.2019

---

#### **Referat**

Dato og tid: 05.02.2019, kl. 15:00-16:00

Sted: Klæbuveien 127B, Møterom 522

Til stede: Trym, Magnus, Atle

Frafall: Ingen

#### **Gjennomgang timelister og statusrapporter**

Litt mere detaljert i hva vi bruker timer på, ikke bare "programmering". Skrive opp f.eks. timer på hver feature vi jobber med. Bruk timelister og statusrapporter til å føre opp en oversikt til hva vi jobber med sånn at veilederen får noe konkret å se på.

#### **Dokumentering underveis (timelister, statusrapporter, sprint board osv.)**

Skjerm bilde fra Trello for start og slutt for hver sprint. Skriv en liten forklaring for hva vi hadde planlagt å gjøre, hva vi har gjort og hva som har vært problemer. Gå gjennom statusrapporter på møter. Dele drive med prosjekthåndbok og lignende fra Drive. Kravspesifikasjon skal kunne leses av kunde og utvikler sammen. Dette krever lettere språk. Det som er lagt ved er en mal, man kan ta vekk og legge til ting.

### **Visjonsdokument**

Visjonsdokumentet er en visjon, det gjør ikke om noe endres underveis. Burde være ferdig så tidlig som mulig.

### **Problemstilling**

Det er greit at ting endrer seg, men det betyr ikke at vi må endre problemstilling og visjon. Da har vi ikke noe grunnlag for sammenligning til slutt. "Hvordan gikk det i forhold til problemstillingen, visjonen?".

Problemstillingen bør også være ferdig så tidlig som mulig. Burde være ferdig i løpet av de neste par ukene.

Dele opp problemstillingen i flere mere håndfaste deler. For vid problemstilling kan gi trekk i karakter.

### **Arbeid videre til neste møte**

Veiledermøte hver 3. uke, helst på kalvskinnet. Nytt møte helst tirsdag 26. februar.

### **Eventuelt**

Neste agenda, hva er planen, hva har vi gjort og hvilken hindringer har vi møtt på. Det er interessant etterhvert også og se på produktet vi har laget. Produktet i seg selv er også viktig for vurderingen. Produktet vurderes for kodestruktur, god kode og tester og lignende.

05.02.2019, Trym Vegard Gjelseth-Borgen

### 3.1.3 Veiledermøte 2 (26.02.2019)

#### **Innkalling**

Tidspunkt/sted: Tirsdag 26.02.2019 kl. 15.00-16.00, Kalvskinnet, grupperom G401

Følgende personer innkalles:

Atle Olsø (veileder)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Gjennomgang timelister og statusrapporter

Gjennomgang førsteutkast visjonsdokument

Arbeid videre og tidspunkt for neste møte

Eventuelt

Meld gjerne ifra om det er andre saker vi bør gå igjennom.

Mvh.

Magnus C. Hyll

Trondheim, 23.02.2019

---

#### **Referat**

Dato og tid: 26.02.2019, kl. 15:00-16:00

Sted: Klæbuveien 127B, Møterom 521 over Skype

Til stede: Trym, Magnus, Atle

Frafall: Ingen

#### **Gjennomgang timelister og statusrapporter**

Tenke gjennom hva vi legger inn i sprint backlog. "Hvordan vet vi hvor mye vi skal ta tak i hver sprint?". Tenker at når vi beskriver metode at vi tar utgangspunkt i scrum, men har bare brukt det og det fra scrum. Vi må ha et bevisst forhold hva vi henter fra scrum i metodedelen av rapporten. Knytte sammen timelister, trello-board og ukesrapporter.

Timeføringssystem knyttet opp mot user stories. Før timer knyttet opp mot kortene som brukes i Trello for å dokumentere sprinten. Skriv timer brukt på hvert kort/task.

### **Gjennomgang førsteutkast visjonsdokument**

Visjonsdokumentet må være så konkret at vi kan måle det vi har gjort opp mot det vi trodde vi skulle oppnå.

Ta med flere kanskje interne interessenter (alle som blir påvirket og bidrar til prosjektet og produktet). "Hvem skal godkjenne oppgaven?". Ta med interessenter som er interessert i selve oppgaven.

Ta med fysisk lokasjon for kystingeniørene. Hvor sitt dem f.eks?

Tekniske spesifikasjoner står i systemdokumentasjon, ikke visjonsdokument eller kravspek.

Overordnede funksjoner skal være med i visjonsdokumentet (Sammendrag av brukernes behov).

Alle krav som ikke kan defineres som en use-case blir et ikke-funksjonelt krav. Eksplisitt dokumentasjon av krav og "negative"-krav. Alle krav burde være målbar. FURBS kapittel 6?

### **Arbeid videre og tidspunkt for neste møte**

Krav-spek må gjøres klart til neste møte. Få på plass de nye timelistene som er knyttet til funksjoner. Neste veiledermøte mandag 18. klokka 13:00. Beskriv de funksjonelle kravene på en måte som er målbar.

### **Eventuelt**

Ingen ting å referere.

26.02.2019, Trym Vegard Gjelseth-Borgen

### 3.1.4 Veiledermøte 3 (18.03.2019)

#### **Innkalling**

Tidspunkt/sted: mandag 18.03.2019, kl. 13.00-14.00, Kalvskinnet, Møterom T04.541

Følgende personer innkalles:

Atle Olsø (veileder)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Gjennomgang timelister, sprint boards og statusrapporter

Arbeid videre og tidspunkt for neste møte

Eventuelt

Meld gjerne ifra om det er andre saker vi bør gå igjennom.

Mvh.

Magnus C. Hyll

Trondheim, 23.02.2019

---

#### **Referat**

Dato og tid: 18.03.2019, kl. 13:00-14:00

Sted: Kalvskinnet, Møterom T04.541

Til stede: Trym, Atle

Frafall: Magnus

#### **Gjennomgang av timelister, sprint boards og statusrapporter**

Kan skrive at vi gikk over til nytt format, om de blir for mye stress å endre de timelistene vi allerede har. Det viktigste er at vi dokumenterer det.

Burde heller skrive at vi hadde en pause i 2 uker, enn å skrive at vi ikke gjorde noe i de 2 første ukene av sprint 5, ettersom at dette allerede var planen. Endre det sånn at sprint 5 går fra 18. mars til 22. mars. Få med en kommentar på hvorfor det var en pause (eksamen).

#### **Arbeid videre og tidspunkt for neste møte**

Fokuser på kravspek, den må komme på plass snarest. Få skrevet den så godt som mulig ferdig, og gå gjennom den med produkteierne. Skriv deretter om det som må skrives om, sånn at vi kan gå gjennom den med Atle neste gang.

### **Eventuelt**

Ingen ting å referere.

18.03.2019, Trym Vegard Gjelseth-Borgen

### 3.1.5 Veiledermøte 4 (11.04.2019)

#### **Innkalling**

Tidspunkt/sted: torsdag 11.04.2019, kl. 15.00-16.00, Kalvskinnet, Grupperom LY1.083

Følgende personer innkalles:

Atle Olsø (veileder)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Gjennomgang timelister, Sprint Boards og statusrapporter

Gjennomgang/tilbakemeldinger på kravdokument

Arbeid videre og tidspunkt for neste møte

Eventuelt

Meld gjerne ifra om det er andre saker vi bør gå igjennom.

Mvh.

Magnus C. Hyll

Trondheim 09.04.2019

---

#### **Referat**

Dato og tid: 11.04.2019, kl. 15:00-16:00

Sted: Kalvskinnet, Grupperom LY1.083

Til stede: Trym, Atle, Magnus

Frafall: Ingen

#### **Gjennomgang timelister, Sprint Boards og statusrapporter**

Ingen ting å referere.

## **Gjennomgang av kravspek**

Kravspek er med som et krav fordi vi skal vise at vi mestrer å få på plass krav fra kunden ned på papiret.

Det ønskes at man kan sjekke kravdokumentet mot det som faktisk blir gjort. Kravene må være målbare.

### **Ting som kunne vært med**

Ta med akseptansekrav for users tories, sånn at kunden lett kan verifisere hva som er ferdig og hva som ikke er det. Ta med mer detaljerte krav i aksepantsekravene.

### **Domenemodell**

Domenemodellen skal ha med problemene for systemet.

Skriv om bruker-boksen sånn at den blir entall.

Database er en del av arkitekturen, ikke nødvendigvis domenemodellen.

Domenemodellen er mer som et klassediagram for dataklassene. Domenemodellen må være litt mer konkret.

Ta med aktivitetsdiagram, det kan være lurt.

### **Databasemodell**

Databasemodellen er kanskje litt for avansert for kravspek, ettersom at kunden ikke får så mye ut av det, sier Atle. Databasemodell hører bedre hjemme i systemdokument.

### **Wireframes**

Trenger ikke å ha med bilder som ikke lenger er med i produktet.

### **Flere krav**

Ta med flere krav om robusthet og tilgjengelighet.

### **Systemdok i sammenligning med kravspek**

Er et dokument mellom oss og andre utviklere, i motsetning til kravspek som er laget for kommunikasjon mellom oss og kunden.



Viktig å beskrive hva vi har tenkt. Viktig å dokumentere at vi har to forskjellige deler av systemet (web-grensesnitt og klassebibliotek).

Det blir som en readme på et github-prosjekt.

### **Arbeid videre og tidspunkt for neste møte**

Uke 18, 2. Mai kl. 1500

Sende innkalling

Send med referat fra forrige og denne gang.

Se på rapporten nå!

Diskutere dokumentasjon neste gang.

Til neste gang, finn en dato for presentasjon.

### **Eventuelt**

Ingenting å referere.

11.04.2019, Trym Vegard Gjelseth-Borgen

### 3.1.6 Veiledermøte 5 (02.05.2019)

#### **Innkalling**

Tidspunkt/sted: torsdag 02.05.2019 kl. 15.00-16.00, Kalvskinnet, Møterom T04.541

Følgende personer innkalles:

Atle Olsø (veileder)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Gå gjennom rapport.

Gå gjennom annen dokumentasjon (systemdokumentasjon, eventuelt endringer i krav/visjonsdokument).

Demo av fungerende produkt.

Eventuelt.

Hadde vært strålende om du (Atle) kunne booket et møterom til oss på Kalvskinnet.

Gi beskjed om du ikke kan komme, så prøver vi å få ordnet et annet tidspunkt.

Mvh.

Trym Vegard Gjelseth-Borgen

Trondheim, 12.04.2019

---

#### **Referat**

Dato og tid: 02.05.2019, kl. 15:00-16:00

Sted: Kalvskinnet, Møterom T04.541

Til stede: Trym, Atle, Magnus

Frafall: Ingen

#### **Gå gjennom hovedrapport**

##### **For problemstilling**

Vi kan peke på at vi har fjernet en del manuelle steg. Ikke behov for å ta tid med stoppeklokke.

Hva betyr det at rapporten er mer komplett? Hva betyr det at rapporten er mer nøyaktig? Ikke bare anta at leseren skal forstå hva vi mener med komplett og nøyaktig.

Vi må stille opp hva som ble gjort før, med hva som blir gjort med vårt system. Hva kan skje når mennesker prøver å klippe og lime i store datamengder, blir det bedre nå som en maskin gjør denne jobben.

Problemstillingen og underpunktene ser veldig greit ut.

### **Gjennomgang av teori**

Poenget med teorien er å hjelpe leseren i resten av rapporten. Den må legge grunnlaget for det som kommer i resten av rapporten. Teknologi som drøftes under valg må beskrives i teori, selv om vi ikke har brukt den.

Vi trenger teori for hva SignalR gjør. Gå inn å se på dokumentasjon for SignalR, se hva de har skrevet at SignalR skal hjelpe til med.

Kan skrive om hvordan vi har delt opp applikasjonen, eller oppdelingen av programmer i generelt.

### **Gjennomgang av valg**

Når det kommer til valg av enten web- eller desktop-applikasjon må det forklares her, men trenger ikke å skrive teori om dette

Ta med forklaring av SignalR, men ikke forklar hvordan det fungerer kodemessig. Vi har valgt SignalR av en grunn, hvorfor ikke REST?

Skriv om hvorfor man velger ORM? Ulemper; ekstra lag, hastighet, kompleksitet.

Fordeler; uavhengig av SQL-dialekt, beskytter om utnyttelse av SQL-svakheter.

For Dependency Injection er det verdt å nevne det om vi har tatt helt spesielle valg rundt dette.

Bare de viktigste teknologiene og de viktigste valgene trenger dokumentasjon.

Forklare oppdelingen av applikasjonen. Hvorfor har vi gjort det sånn, hvilke fordeler gir det?

### **Gjennomgang av Resultater**

Vi trenger ikke å skrive om timeregnskap i “Administrative resultater”, dette kan vi henvise til. Her skal vi bare oppsummere kort f.eks. “Vi har brukt så og så mange timer totalt”. Hvis det er noe helt spesielt kan vi ta noen utdrag fra timelister eller sprint board.

### **Gjennomgang av Konklusjon**

Der skal vi vise til hva som faktisk ble resultatet av problemstillingen.

### **Referanser**

Det viktigste er at vi finner en enhetlig måte å referere til kilder.

Gå gjennom annen dokumentasjon (systemdokumentasjon, eventuelt endringer i krav/visjonsdokument)

**Gå gjennom annen dokumentasjon (systemdokumentasjon, eventuelt endringer i krav/visjonsdokument).**

### **Arkitektur**

Interessant å nevne her hvor det er applikasjonen er deployet. Beskriv miljøet vi har deployet til. Hva er adressen, tilgang, hvordan settes det opp?

### **Klassediagram**

Kan henvise til domenemodell. Prosjektstrukturen gir mye som klassediagram, men mangler koblingen mellom klassene og de viktigste metodene. Et autogenerated klassediagram gir veldig mye uinteressant. Kanskje lage et klassediagram med et subset av de viktigste klassene og koblingene.

CRC-kort, som beskriver koblinger mellom klassene kan man se på.

### **Aktivitetsdiagram**

Kan ta det med i systemdokumentasjonen. Knytt den opp mot klasser.

Aktivitetsdiagram blir mye mer lesbart enn sekvensdiagram.

### **Databasemodell**

Nevn at databasemodellen er basert på klassene i applikasjonen. At de er autogenerated av ORM.

Forklar hva Azure Blob Storage er.

### **Installasjon**

Det viktigste er å vise hvordan man får applikasjon til å kjøre. Avhengighetene blir lastet ned automatisk, forklar dette. Nevn det hvis det er noen konkrete versjoner som brukes.

### **Kontinuerlig integrasjon og testing**

Dette er viktig. Kanskje ta med noen systemtester. Konkrete tester for kjøring av programmet. F.eks. Kan laste ned en rapport. Sjekker om dette går eller ikke går. Systemtesten er kanskje det siste som kommer på plass, hvis det er tid. Atle sender en mal.

### **Demo av fungerende produkt.**

Ingen ting å referere.

### **Neste gang**

Sett opp et møte om 2 uker. Tirsdag 14. mai, 15:00. Atle booker rom.

Kom med konkrete spørsmål i forhold til rapporten, og de andre dokumentene.

Hvordan ligger vi an? Må vi prioritere noe, og legge vekk noe annet?

### **Eventuelt**

For presentasjonen, ta med oppgaven og problembeskrivelsen og forklar hvilke svar vi har funnet.

02.05.2019, Trym Vegard Gjelseth-Borgen

### 3.1.7 Veiledermøte 6 (14.05.2019)

#### **Innkalling**

Tidspunkt/sted: torsdag 14.05.2019 kl. 15.00-16.00, Klavskinnnet, Møterom T04.541

Følgende personer innkalles:

Atle Olsø (veileder)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Gjennomgang status & timelister

Finpuss og oppklaringer til hovedrapport

Finpuss og oppklaringer til systemdokumentasjon

Finpuss og oppklaringer til øvrig dokumentasjon og prosjekthåndbok

Innhold i presentasjonen

Annen praktisk info for levering

Eventuelt

Mvh.

Magnus C. Hyll

Trondheim, 14.05.2019

---

#### **Referat**

Dato og tid: 14.05.2019 kl 15:00-16:00

Sted: Kalvskinnnet, Møterom T04.541

Til stede: Trym, Atle, Magnus

Frafall: Ingen

#### **Gjennomgang status & timelister**

Ingen ting å referere.

#### **Finpuss og oppklaringer til hovedrapport**

##### **Tittel**

Tittelen er opp til oss. Den burde oppsummere hva oppgaven dreier seg om. F.eks. bare "E39 Fjordkryssing" er for vagt.

### **Sammendrag**

Tidligere har man levert en tittelside. Den har inneholdt engelsk og norsk tittel og sammendrag. Den ble levert i April.

Når vi skal levere nå vil vi bli møtt i inspera med en side for å skrive inn tittel og sammendrag på norsk og engelsk, samt andre nødvendig informasjon. I hovedrapport kan vi velge om vi vil ha med engelsk tittel og sammendrag.

Tittelside (som den andre siden i bacheloroppgaven til gard og co.), er valgfri i hovedrapporten.

Vi må uansett finne en engelsk tittel, og oversette sammendrag til engelsk sånn at vi kan skrive det inn i inspera.

### **Rettigheter i forhold til kildekode og lignende**

Det er opp til oss om oppgaven skal publiseres til slutt.

### **Forside**

Avsender, tema og mottaker bør være med. Ikke krav til at vi bruker noe malsystem for forside.

### **Refleksjonsnotat**

Vi skal bare si litt om hvordan vi opplevde prosjektet, hva vi har lært, eventuelle utfordringer. Ta opp konflikter om de har oppstått. Hovedmålet er refleksjon rundt egen læring og innsats.

### **Valg**

Vi må selv finne en god balanse for hva som skal skrives i valg, og hvordan det skal skrives (om det ble valgt fra starten, eller om det er et valg som kom senere).

Fornuftig å omtale både ORM- og python-delen i "Valg". Det viktigste er at det er en god flyt. Burde ikke inneholde masse gjentakelser. Atle er med opptatt av at det er god flyt og god lesbarhet. Det er ikke nødvendigvis en direkte kronologi i måten vi jobbet på.

## **Finpuss og oppklaringer til systemdokumentasjon**

Vi kan skrive at kildekoden har api-dokumentasjon. Kan f.eks. skrive at det ligger vedlagt kildekoden.

Istedenfor å forklare hvordan et klassediagram fungerer, men heller spesifikt hvordan flyten i prosjektet vårt er. Hvis vi ikke klarer å skrive dette kan vi skrive en kort beskrivelse, ikke en tekst som forklarer hva et klassediagram er.

## **Finpuss og oppklaringer til øvrig dokumentasjon og prosjekthåndbok**

### **Sprint Boards (Og litt om prosjekthåndbok)**

Det vi har på Sprint Boards er veldig greit. Det må komme med det store bildet også. Vi må reflektere rundt hele prosjektet (Scrummen), i hovedrapporten. Beskriv i prosjekthåndboka hva sprint board-rapportene er, at vi har brukt Trello og hvordan vi har brukt det.

### **Gannt-diagram**

Man vet at man har noen milepæler som man jobber etter. Forklar at Gannt-diagrammet er et overordnet blikk av prosessen som ble planlagt i starten. Beskriv at fremdriftsplanen (Gannt) ble for generell, og at etter diskusjon med veileder var det sprint boards som gjaldt.

### **Timelister**

Det passer bedre med de enkle timelistene. Ta med den andre spesifikke i tillegg, men som en vedlegg, zippet. Vi må sy sammen statusrapporter, timelister og alt dette sånn at det gir mening, og fin flyt.

### **Referat fra sprint review**

Ta med som et eget kapittel.

## **Innhold i presentasjonen**

Ta med det som står i eksempelet.

## **Annen praktisk info for levering**



Hovedrapport med vedlegg, hvor vedlegger prosjekthåndbok, visjonsdokument, kravspesifikasjon, systemdokumentasjon og evt. systemtester. Vedleggene har egne sidetall og lignende. I innholdsfortegnelsen til hovedrapporten står vedleggene. Vedleggene har egne sidetall, innholdsfortegnelse og referanselister og sånn.

Det er opp til oss om prosjekthåndboka skal være med som vedlegg i hovedrapporten (oppgaverapporten), eller om den skal legges utenfor. Det er fortsatt mulig å referere til prosjekthåndboka selv om den ikke er som et vedlegg i hovedrapporten. Alt vi ikke leverer i "Oppgave 1", skal være med i "Oppgave 2".

### **Eventuelt**

Ingen ting å referere.

14.05.2019, Trym Vegard Gjelseth-Borgen

## 3.2 Sprint Review/Planning-møter

### 3.2.1 Sprint 1 Review/Planning (08.02.2019)

#### **Innkalling**

Tidspunkt/sted: torsdag 14.05.2019 kl. 15.00-16.00, Klæbuveien 127B, Møterom 521

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnstad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Review av sprint 1

Planlegging av sprint 2

Eventuelt

Mvh.

Trym Vegard Gjelseth-Borgen

Trondheim 06.02.2019

---

#### **Referat**

Dato og tid: 08.02.2019 kl 15:00-16:00

Sted: Klæbuveien 127B, Møterom 521

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 1**

##### **Filnedlasting**

Det må struktureres hvor filene lagres og lignende.

##### **NetCDF-konvertering**

Bra løsning med python, hvertfall for nå. Lage kode for å sjekke at alle avhengighetene er på plass, både for pip og python.

Bedre med mindre avhengigheter, burde vurdere å implementere konverteringen direkte i C#.

Tar for lang tid som det gjøres nå. Få delt opp prosessen sånn at den kjører i flere tråder.

## **Planlegging av sprint 2**

Se på implementering av et grafisk brukergrensesnitt. Finne ut om vi skal lage en Windows-applikasjon eller en webside.

Begynn å lage kort for tegning av wireframes til GUI.

Få på plass parallellisering av NetCDF-konverteringen i python.

Se på parallellisering i C# med parallell «foreach».

Maskinlæringsbit for imputering av verdier. Det må gjøres research. Booke møte med (trenger navn?) fra Norconsult og høre med de hva de tenker rundt dette.

Helst en implementasjon av maskinlæring i C#.

Athul må være med å avgjøre om den imputerte dataen har god nok kvalitet.

Dele opp planene for parallellisering, bugfiksing, GUI og maskinlæring, og legge inn i produkt backlog. Bruke dagen ut i dag (fredag 8. feb.) til å gjøre dette. Andreas sorterer kortene vi lager på Trello etter prioritet hvor kort lengre opp har høyest prioritet. Uke 7 blir litt amputert, med tanke på mye obligatorisk oppmøte på skolen.

Log-bibliotek

Log4Net til logging av kjøring. Log alt til fil sånn at man for eksempel kan se hva som har gått galt, hvis man merker at det er noe galt i dataen.

## **Eventuelt**

Sette opp et eget visjonsdokument-møte med Andreas Ravnestad. Se på kalenderen i uke 8.

Mulig for sprint review og planning klokka 08.00 førstkommende fredag, vi sender innkalling.

Torsdag 21. Februar, planning og review-møte, vi sender innkalling.

## **Kontinuerlig Integrasjon**

Sette opp kontinuierlig integrasjon i Azure vil være veldig bra for prosessen sin del.

## **Hvis det oppstår bugs i koden**

Skriv en test som reproducerer bugen, istedenfor å fikse koden. Sånn sørger man for at den samme feilen ikke skjer igjen. Dette kalles regresjonstesting.

08.02.2019, Trym Vegard Gjelseth-Borgen

### 3.2.2 Sprint 2 Review/Planning (14.02.2019)

#### **Innkalling**

Tidspunkt/sted: torsdag 14.02.2019 kl. 08.00-09.00, Klæbuveien 127B, Møterom 521

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnestad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Review av sprint 2

Planlegging av sprint 3

Eventuelt

Mvh.

Trym Vegard Gjelseth-Borgen

Trondheim 08.02.2019

---

#### **Referat**

Dato og tid: 14.02.2019, kl. 08:00-09:00

Sted: Klæbuveien 127B, Møterom 521

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 2**

Ingen ting å referere.

#### **Planlegging av sprint 3**

Andreas Ravnestad legger inn på trello, alternativer for forslag til refactoring-pakker for .net.

Starte med python for maskinlæringsdelen er like greit.

#### **Eventuelt**

Ta kontakt med for maskinlærings-eliten:

Vigleik Lund, Einar Arbo og Viggo Wivestad.

Kanal på Slack som heter ML-AI.

Sette opp editor-config sånn at begge bruker enten tabs eller space for code indents.

14.02.2019, Trym Vegard Gjelseth-Borgen

### 3.2.3 Sprint 3 Review/Planning (21.02.2019)

#### **Innkalling**

Tidspunkt/sted: torsdag 21.05.2019 kl. 13.00-14.00, Klæbuveien 127B, Møterom 521

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnestad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Review av sprint 3

Planlegging av sprint 4

Eventuelt

Mvh.

Trym Vegard Gjelseth-Borgen

Trondheim 08.02.2019

---

#### **Referat**

Dato og tid: 21.02.2019, kl. 13:00-14:00

Sted: Klæbuveien 127B, Møterom 521

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 3**

Fikk ikke sett på maskinlæringsmetoder denne sprinten. Høre med de som kan noe på Norconsult om dette (navn, se forrige referat sprint 2).

#### **Planlegging av sprint 4**

Andreas foreslår å se maskinlæring neste sprint.

#### **Eventuelt**

##### **Hvor lage GUI**

Andreas foreslår Webapp.

Fordelene er at det er enkelt å kjøre ut nye oppdateringer uten at hver enkel må laste ned.

Continuous integration og deployment med Azure. Andreas kan hjelpe til å få deployert.

ASP .Net Core for baksiden av webappen.

Kanskje litt ekstra mikkmakk med tanke på å få kjørt Python backend.

### **Hvordan blir det med ressurser for dette?**

Viktig å tenke på performance, vil nok gå tregere på webapp.

Tenkte på med tanke på rapporten fordeler og ulemper med native- og webapp.

Andreas skal sjekke om det er enkelt å kjøre Python backend på server.

### **Hvis webapp**

Blob storage for lagring av filer, veldig enkelt.

Lurt å sette opp en egen branch for deployment om vi går for denne løsningen.

Tenke på en service som laster ned ny rådata når dette finnes, ettersom at den forsvinner fra thredds.met.no sine sider fra tid til annen.

### **Hvis native**

Det går å bruke .Net Core 3 for å lage GUI.

Sjekke muligheter for andre GUI-rammeverk som er cross platform, for det er ikke .Net Core 3 sine.

### **Har fått tilgang til Azure**

Vi kan logge inn med norconsult-mailen.

Bruke resource groups for å lage en subscription.

Andreas har opprettet en som heter RG-Ferjefri-E39.

Inne der kan man opprette "storage". Kan være forskjellige ting, f.eks. table storage og blob storage. Blob storage er som et hash-system, for fillagring.

Portal.Azure.com for å oppretting av storage og lignende.

Bruke azure storage explorer tool for å se gjennom storage. Logge inn med norconsult-brukeren.

Bruke Microsoft.Azure.Store.Blob nuget pakke, den er kryssplattform.

Connection key ligger i "Access Keys"-mappe for å koble til storage. Bruk Key 1 - connection string.

21.02.2019, Trym Vegard Gjelseth-Borgen



### 3.2.4 Sprint 4 Review/Planning (01.03.2019)

#### **Innkalling**

Tidspunkt/sted: fredag 01.03.2019 kl. 15.00-16.00, Klæbuveien 127B, Møterom 521

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnstad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Review av sprint 4

Planlegging av sprint 5

Eventuelt

Mvh.

Trym Vegard Gjelseth-Borgen

Trondheim 21.02.2019

---

#### **Referat**

Dato og tid: 01.03.2019, kl. 15:00-16:00

Sted: Klæbuveien 127B, Møterom 521

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 4**

ISO/IEC 25010 for å lese seg opp på kvalitetsmodell. Dette burde tas opp i et kravdokument.

Kan også se på FURPS for det samme. Vi burde sette opp disposisjon på kravdokumentet.

Lurt å se på tidligere bacheloroppgaver for inspirasjon.

#### **Web app**

Kunden (Andreas og Athul) har bestemt seg for at de vil ha en web app.

Alle som skal bruke appen skal ha tilgang til det samme. Genererte rapporter burde være tilgjengelig for alle som har brukernavn og passord.

Burde se på en løsning som både fungerer lokalt, og som er enkel å deploye. SQL-script for oppretting av tabeller.

### **Wireframes**

Ønske om å kontrollere dato og tid for data-rapporter.

Ikke behov for grafer i GUI.

### **Planlegging av sprint 5**

Sette opp arbeidsmøte med Athul for å se på SCAT-filene.

Viktig å splitte opp GUI i små tasks.

Få på plass en Code review med Andreas. Kall inn Andreas til møte med en liste over spesifikke ting, sånn at han kan sjekke på forhånd.

### **Eventuelt**

Ingen ting å referere.

01.03.2019, Trym Vegard Gjelseth-Borgen

### 3.2.5 Sprint 5 Review/Planning (25.03.2019)

#### **Innkalling**

Tidspunkt/sted: mandag 25.03.2019 kl. 10.00-11.00, Klæbuveien 127B, Møterom 522

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnestad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjølseth-Borgen (student)

#### **Agenda:**

Review av sprint 5

Planlegging av sprint 6

Eventuelt

Mvh.

Magnus C. Hyll

Trondheim 21.03.2019

---

#### **Referat**

Dato og tid: 25.03.2019, kl. 10:00-11:00

Sted: Klæbuveien 127B, Møterom 522

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 5**

Gå gjennom sprint-board og del opp kortene slik at de blir håndterlige. Nå er det veldig store kort. Start med å lage lister innad i kortet. Hvis man ser at et sjekkpunkt på lista begynner å ta lang tid, gjør man det om til et kort.

Kall inn et arbeidsmøte til GUI på onsdag, bare Athul. Gå gjennom GUIet, hvordan det skal fungere og hva som skal være med. Ta også en gjennomgang av SCAT-filene

#### **Planlegging av sprint 6**

Endre “GUI”-kortet, det er altfor vagt. Døp det om til en kjempeenkel greie. Legg til en task for å legge til en web-solution i prosjektet.

CI får vi på plass etter at vi har fått opp webappen. Før dette finnes det ingenting å deploye.

Behandle dataen så lenge som mulig i UTC, og ikke gjør om til lokal tid før nødvendig.

Få på plass nytt sprint-møte førstkomende fredag for sprint 6.

### **Eventuelt**

Finne ut hvor konfigurertbart stasjonnavn skal være.

SignalR for å ha en åpen kommunikasjonskanal fram til frontend. Er som en websocket.

25.03.2019, Trym Vegard Gjelseth-Borgen

### 3.2.6 Sprint 6 Review/Planning (01.04.2019)

#### **Innkalling**

Tidspunkt/sted: mandag 01.04.2019 kl. 10.00-11.00, Klæbuveien 127B, Møterom 501

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnstad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjølseth-Borgen (student)

#### **Agenda:**

Review av sprint 6

Planlegging av sprint 7

Eventuelt

Mvh.

Magnus C. Hyll

Trondheim 30.03.2019

---

#### **Referat**

Dato og tid: 01.04.2019, kl. 10:00-11:00

Sted: Klæbuveien 127B, Møterom 501

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 6**

Ingen ting å referere.

#### **Planlegging av sprint 7**

##### **Bakgrunnsprosesser for generering av rapporter**

Det må genereres en id for hver task som opprettes av en bruker i backend, sånn at det er mulig å finne tilbake til denne tasken. Burde være mulig for brukeren å lukke nettleseren og komme tilbake å sjekke progresjon senere. Ordentlig avslutting av en

task. Det må være mulig for bruker å laste ned ferdig-genererte rapporter, som betyr at det må holdes kontroll på hvilke prosesser som er knyttet til hvilken bruker.

Få på plass et arbeidsmøte i forhold til oppretting av tasks i bakgrunnen, for å få på plass rett patterns. Prat også her om deployment av applikasjonen til Azure og CI.

### **Opplasting og sletting av filer**

Når det kommer til opplasting av filer og sletting av filer er det viktig at det sjekkes at disse filene ikke brukes samtidig. Her er det viktig med god funksjonalitet for å fange feil, og gi tilbakemelding til bruker.

### **KNN**

Magnus ferdigstiller KNN, og Athul sjekker funksjonaliteten. Kanskje sette opp noe kryssvalidering som ikke er manuell, men som sjekker og gir noen tall. Kanskje forhøre seg om hvordan man eventuelt får gjort en slik validering, sånn at man får noen fine metrikker.

### **Autocaching av filer**

Sette opp en singleton service i startup.cs-fila. I startmetoden må den opprette en task.

### **Eventuelt**

Ingen ting å referere.

01.04.2019, Trym Vegard Gjelseth-Borgen

### 3.2.7 Sprint 7 Review/Planning (08.04.2019)

#### **Innkalling**

Tidspunkt/sted: mandag 08.04.2019 kl. 09.00-10.00, Klæbuveien 127B, Møterom 502

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnestad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Review av sprint 7

Planlegging av sprint 8

Eventuelt

Mvh.

Trym Vegard Gjelseth-Borgen

Trondheim 03.04.2019

---

#### **Referat**

Dato og tid: 08.04.2019, kl. 09:00-10:00

Sted: Klæbuveien 127B, Møterom 502

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 7**

Burde få på plass nøye edge case-sjekking av koden for buffersystemet av rådatafiler.

Muligheter for at ting kan gå galt her, så sett try-catch og if-else sånn at feil blir logget.

Det samme gjelder også Job og JobManager.

#### **Planlegging av sprint 8**

Azure Table Storage for lagring av data.

IHostedService for både SignalR-hub og for rapportdata som skal skrives til database.

For logging. Send hele exception med i loggen sånn at stack trace kommer med, ved feil. Ha gode navn på loggene sånn at det går an å spore feil tilbake.

Lag en klasse for infoen om rapporten som blir lagd (med antall manglende rader, filer som ikke ble lastes ned ol.).

### **Eventuelt**

Sprint review 8 på førstkommande fredag.

08.04.2019, Trym Vegard Gjelseth-Borgen



### 3.2.8 Sprint 8 Review/Planning (12.04.2019)

#### **Innkalling**

Tidspunkt/sted: mandag 12.04.2019 kl. 10.00-11.00, Klæbuveien 127B, Møterom 521

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnestad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjølseth-Borgen (student)

#### **Agenda:**

Review av sprint 8

Planlegging av sprint 9

Eventuelt

Mvh.

Magnus C. Hyll

Trondheim 08.04.2019

---

#### **Referat**

Dato og tid: 12.04.2019, kl. 10:00-11:00

Sted: Klæbuveien 127B, Møterom 521

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 8**

Ingen ting å referere.

#### **Planlegging av sprint 9**

Sjekke ut Entity Framework core for database-tilkobling og kall.

Athul vil ha med informasjon om manglende data i excel-rapportene.

Skjemaet for sql-databasen burde lagres, sånn at systemet enkelt kan settes opp.

#### **Eventuelt**

Kalle inn til to møter. Et vanlig planning- og review-møte og et arbeidsmøte for CI/CD. Møte onsdag 24. april, etter lunsj.

12.04.2019, Trym Vegard Gjelseth-Borgen

### 3.2.9 Sprint 9 Review/Planning (24.04.2019)

#### **Innkalling**

Tidspunkt/sted: mandag 24.04.2019, kl. 12.00-13.00, Klæbuveien 127B, Møterom 502

Følgende personer innkalles:

Athul Sasikumar (Produkteier)

Andreas Ravnestad (Scrum Master)

Magnus Conrad Hyll (student)

Trym Vegard Gjølseth-Borgen (student)

#### **Agenda:**

Review av sprint 9

Planlegging av sprint 10

Eventuelt

Mvh.

Magnus C. Hyll

Trondheim 17.04.2019

---

#### **Referat**

Dato og tid: 24.04.2019, kl. 12:00-13:00

Sted: Klæbuveien 127B, Møterom 502

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Review av sprint 9**

Ingen umiddelbare spørsmål eller kommentarer fra Andreas. Athul skal starte å jobbe med å lage rapporter i neste uke.

Vi sender ut en epost med link og eventuelle instruksjoner for testing av appen. Andreas og Athul vil deretter teste og legge ut eventuelle bugs på Trello.

I forhold til applikasjonen som krasjer i Azure, så skal Andreas ta en liten titt på det. Vi skal også se på det, sånn at vi sørger for at applikasjonen er stabil.

#### **Planlegging av sprint 10**

Vi kjører i gang en liten sprint med noen kort, for å få finpusset systemet. Dette fordi at vi trenger å få satt i gang å skrive rapport. Rapporten har høyest prioritet. Andreas og Athul stiller til disposisjon om de trengs i forhold til rapporten.

### **Eventuelt**

Vi har veldig mange kule forskjellige teknologier i applikasjonen. Vi må prøve å få med så mye som mulig i rapporten.

24.04.2019, Trym Vegard Gjelseth-Borgen

### 3.2.10 Sprint 10 Review (14.05.2019)

#### **Innkalling**

Tidspunkt/sted: mandag 14.05.2019, kl. 10.00-11.00, Klæbuveien 127B, Møterom 521

Følgende personer innkalles:

Athul Sasikumar (Produkteier)  
Andreas Ravnstad (Scrum Master)  
Magnus Conrad Hyll (student)  
Trym Vegard Gjelseth-Borgen (student)

#### **Agenda:**

Gå gjennom sprint 10  
Gjennomgang av produktet  
Tilbakemeldinger på produktet  
Hovedrapportrelaterte aktiviteter

Mvh.

Trym Vegard Gjelseth-Borgen

Trondheim 08.05.2019

---

#### **Referat**

Dato og tid: 14.04.2019, kl. 10:00-11:00

Sted: Klæbuveien 127B, Møterom 521

Til stede: Athul, Andreas, Magnus, Trym

Frafall: Ingen

#### **Gå gjennom sprint 10**

#### **Gjennomgang av produktet**

#### **Tilbakemeldinger på produktet**

- Uklarheter rundt hva “values imputed” betyr og hva “manually uploaded files” betyr.
- Feil start- og sluttdato for x-akse i tidsserier
- Feil tittel for tidsseriregraf, burde inneholde navn på stasjon
- Fikse MAX i rød kolonne på SCAR i rådata-fanen.

- Fikse average i scatterplots i Hs\_Tp
- WindDirection trenger ikke desimaler.
- Trenger bare 2 desimaler og format number
- Mangler rådatafilen for scat\_swell og scat\_vind
- Få ordnet PK for tabellene i databasen

## **Hovedrapportrelaterte aktiviteter**

### **Hvordan skal sensor få tilgang til koden?**

Vi kan zippe sammen koden og sende med innlevering. Vi må bare fjerne connection string og lignende privat info.

### **Automatisk API-dokumentasjon**

Swashbuckle for HTTP-endepunktdokumentasjo

Doxygen for klassebibliotekdokumentasjon

## **Intervju med Athul**

### **Lager dere rapporter for Vegvesenet?**

“Vi analyserer data, sendt til oss fra Fugero.” “Rapportene vi lager er for vårres del, og blir sendt til.”

### **Hvorfor sender dere rapporter?**

“Bruker rapportene til å avgjøre om hvordan broer, eventuelt hvilke broer som skal bygges”

“Vi kommer bare med data og analyse, og kommer ikke direkte med forslag til avgjørelser som skal tas”.

## **Prosess med å lage rapport**

Ocean Fugro sender data som excel format, eller henter data fra thredds.met.no.

Bruker to forskjellige MatLab-script. Et for å konvertere fra NetCDF til txt og et for å konvertere datoer. Da kommer hver feature ut som en fil, og må deretter legges inn i malene.

Når dataene har kommet inn i Excel-filene måtte det først lages en kolonne over alle forventede datoer lages. Denne ble brukt for at excel skulle ordne og fylle inn kolonner for alle manglende rader. Må sjekke om det er riktig range for vlookup.

Måtte også justere y- og x-akse for grafer.

Må sjekke range for mange verdier, sånn at formlene blir rett. Malene måtte tilpasses avhengig av størrelsene på dataene.

For å lage alle rapporter for 1 stasjon, kunne ta opp mot 2 timer.

Filene fra Fugro Ocean kan være utdatert er oppdatert. Vårt nye system er mer oppdatert en filene sendt direkte fra Fugro Ocean, hvertfall i et tilfelle.

14.05.2019, Trym Vegard Gjelseth-Borgen

## 4 Timelister med statusrapporter

I dette kapittelet finnes timelister og statusrapporter for hver uke i prosjektet. Timelistene er en forenklet versjon av timelistene som ble brukt for timeføring under prosjektet.

Timelistene som ble brukt under prosjektet, inneholder timer ført på hvert kort/task for sprinten. For kort/tasks, se kapittel 5. For utvidet timeliste, se i mappen «utvidet timeliste» i leveranse 2.

### 4.1 Uke 2, 3 og 4

#### Hva er gjort

##### Magnus og Trym

- Avholdte oppstartsmøte
- Fikk gjennomgang av rutiner og praktisk informasjon hos Norconsult
- Satte opp PC-er og kontorplass
- Avholdte første arbeidsmøte hvor vi gikk gjennom krav til første versjon av systemet
- Satte opp GitHub-repository, Trello-boards og Visual Studio-prosjekt
- Deltatt på obligatoriske forelesninger i vitenskapsteori og -metode

Tabell 4-A Timer for uke 2

Uke	2			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter	1,5	1,5	3	3
Div. admin			0	0
Hovedrapport			0	0
Dokumentasjon			0	0
Obligatoriske forelesninger			0	0
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)			0	0
Maskinlæring			0	0
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	1,5	1,5		
Sum timer totalt	1,5	1,5		3



Tabell 4-B Timer for uke 3

<b>Uke</b>	<b>3</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter			0	3
Div. admin	6	6	12	12
Hovedrapport			0	0
Dokumentasjon			0	0
Obligatoriske forelesninger			0	0
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)			0	0
Maskinlæring			0	0
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	6	6		
Sum timer totalt	7,5	7,5		15

Tabell 4-C Timer for uke 4

<b>Uke</b>	<b>4</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter	1,5	1,5	3	6
Div. admin	1,5	1,5	3	15
Hovedrapport			0	0
Dokumentasjon			0	0
Obligatoriske forelesninger	8	8	16	16
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)			0	0
Maskinlæring			0	0
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	11	11		
Sum timer totalt	18,5	18,5		37

## 4.2 Uke 5

### Hva er gjort

#### Magnus

- Deltatt på obligatorisk problemstilling-workshop
- Undersøkte og testet .NET-biblioteket EPPlus for skriving til excel-filer
- Undersøkte og testet .NET-biblioteket SDSLite for lesing av NetCDF-filer i C#-applikasjonen. Bestemte oss for å vente med dette til senere
- Gjorde research og leste meg opp på dataformatet og -strukturen til NetCDF
- Utviklet Python-script for konvertering av NetCDF-filer til csv
- Undersøkte kravene til og jobbet med prosessdokumentasjon
- Innkalt til veiledermøte

#### Trym

- Deltatt på obligatorisk problemstilling-workshop
- Lagd en klasse for å laste ned rådata fra MET gitt start og slutt dato for ønsket data.
- Undersøkt .NET-biblioteket NUnit for å laga unit-tester.
- Lagd en excel-mal for dataen.
- Startet på en klasse for å lese CSV-filer, konvertere data og skrive det inn i en excel-mal.

Tabell 4-D Timer for uke 5

Uke	5			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter			0	6
Div. admin	4		4	19
Hovedrapport			0	0
Dokumentasjon			0	0
Obligatoriske forelesninger	2	2	4	20
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)	26	29,5	55,5	55,5
Maskinlæring			0	0
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	32	31,5		
Sum timer totalt	50,5	50		100,5

## 4.3 Uke 6

### Hva er gjort

#### Magnus

- Laget førsteutkast til framdriftsplan med Gantt-diagram
- Avholdt og deltatt på veiledermøte
- Laget maler for ukentlige statusrapporter og timelister, samt mal for dokumentering av sprint boards
- Jobbet med å integrere python-skript i C#.NET-applikasjonen vha. IronPython
- Jobbet med å kjøre Python-skript som egen prosess ifra C#-appen
- Laget unit-tester for NetCDFConverter
- Laget slik at feilkoder fra Python-skriptet genererer exceptions i C#-appen
- Review & planning-møte for sprint 1

#### Trym

- Forsøkt å ferdigstille klassen for å skrive data inn i excel-malen. Har kommet så langt at vi nå har en klasse som kan brukes.
- Oppdatert FileGetter-klassen sånn at den nå kan laste ned filer over flere år.
- Skrevet tester for ExcelWriter- og FileGetter-klassen.
- Sydd sammen klassene til en demo.
- Fikset excel-malen slik at alt blir laget.

### Hvilke problemer har eventuelt oppstått

- .NET-biblioteket EPPlus virket til å ha en feil som gjør at det ikke får til å lagre noen typer grafer i Excel.
- Problem med Excel-malen som gjorde at grafene ble feil når alle cellene med formler ikke hadde tilhørende data.
- Sliter med å integrere Python-skriptet (NetCDFConvert.py) i C#.NET-applikasjonen med IronPython, bl.a. pga. vansker med å få med nødvendige Python-pakker. IronPython egner seg nok ikke til oppgaven, da dette i utgangspunktet er en Python-interpreter og ikke en Python-kompilator.

### Hva er planen neste uke

- Det vil gå bort mye tid i obligatorisk oppmøte i faget Systemtenkning med økonomi.
- Fikse bugs i nåværende kode. Kanskje se på implementering av parallellisering av NetCDF-konverteringen i Python.

Tabell 4-E Timer for uke 6

Uke	6			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter	2	2	4	10
Div. admin	12,5	1	13,5	32,5
Hovedrapport			0	0
Dokumentasjon			0	0
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)	17	27,5	44,5	100
Maskinlæring			0	0
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	31,5	30,5		
Sum timer totalt	82	80,5		162,5

#### 4.4 Uke 7

##### **Merknad**

Mandag, tirsdag og onsdag denne uka gikk bort til obligatorisk Innovasjonscamp. Torsdag var det tre timer obligatorisk forelesning i CCD. Sprinten ble avsluttet fredag kl. 8 i review & planning-møtet. Dette gjorde at denne uka ble svært amputert, og at vi fikk lite tid til å jobbe med bachelorprosjektet.

##### **Hva er gjort**

###### **Magnus**

- Måtte endre navn i prosjektfiler og namespaces i C#, da bindestreker i navnene ikke er tillatt og skaper trøbbel.
- Sett på parallellisering av pythonskript-kjøring for konvertering av NetCDF-filer
- Review & planning-møte etter sprint 2

###### **Trym**

- Review & planning-møte etter sprint 2
- Hvilke problemer har eventuelt oppstått
- Bindestreker i navnet på prosjektfiler og namespaces i C#-programmet skapte problemer, men dette ble løst ganske raskt.
- Hva er planen neste uke
- Skrive førsteutkast til visjonsdokument og kravdokument
- Begynne å undersøke hvilke maskinlæringsteknikker som egner seg til å generere manglende måldata
- Rette opp bugs i nåværende kode
- Parallellisere konvertering av NetCDF-filer

Tabell 4-F Timer for uke 7

<b>Uke</b>	<b>7</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter	1	1	2	12
Div. admin			0	32,5
Hovedrapport			0	0
Dokumentasjon			0	0
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)	4		4	104
Maskinlæring			0	0
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	5	1		
Sum timer totalt	87	81,5		168,5

## 4.5 Uke 8

### Hva er gjort

#### Magnus

- Laget felles, cross-platform metoder for å finne data- og programfiler i riktig mappestruktur
- Code refactoring i hovedloopen av programmet for mer lettlest kode
- Parallel-konvertering av NetCDF-filer
- Unit-testing av NetCDF-konvertering og Resources-klasse
- Begynt på visjonsdokument
- Review & planning-møte etter sprint 3
- Begynt såvidt på kravdokument
- Begynt å lese på og implementering av kNN-regresjon for imputering av manglende måledata

#### Trym

- Starter på å lage en metode som kan lage sine egne grafer, framfor å bruke de som allerede finnes i malen.
- Testet den nye metoden som lager grafer i excel, og skrevet unit-tester for den.
- Begynt på visjonsdokument.
- Fikset bug i CSV-innlastingskoden og i tidsforskyvning for data.
- Review & planning-møte etter sprint 3
- Tatt kontakt med med folk på huset som har god kontroll på maskinlæring, og spurt om passende teknikker for å fylle inn for manglende data.
- Delt opp ExcelWriter-klassen sånn at datainnlastingsfunksjonene kom inn i en egen klasse kalt DataLoader.
- Lagt til en funksjon i ExcelWriter-klassen som fyller inn SCAT-excel-malene.

### Hvilke problemer har eventuelt oppstått

- Dataene fra MET sine nettsider forsvinner sporadisk, noe som gjør det vanskelig å teste sjekke nye maler opp mot de gamle.

## Hva er planen neste uke

- Få ferdigstilt kNN-metoden for å fylle inn manglende data. Testet den på tilfeldig data

Tabell 4-G Timer for uke 8

Uke	8			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter	1	1	2	14
Div. admin	2		2	34,5
Hovedrapport			0	0
Dokumentasjon	7	5,5	12,5	12,5
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)	19	27,5	46,5	150,5
Maskinlæring	7	2	9	9
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	36	36		
Sum timer totalt	123	117,5		240,5



## 4.6 Uke 9

### Hva er gjort

#### Magnus

- Fullført førsteutkast visjonsdokument
- Implementert lineær interpolasjon til å fylle ut manglende måledata. Skal også brukes som del av imputering vha. kNN-regresjon, ved å fylle ut manglende målestasjonskoordinater.
- Avholdt og deltatt på veiledermøte
- Gjort refactoring av hovedloopen i programmet, slik at bare enkelte deler av pipelinen kan kjøres avhengig av behov.
- Sett på biblioteket Accord.NET for kNN-regresjon, og begynt å skrive kode for å tilpasse måledataene våre til bibliotekets grensesnitt.
- Review & planning-møte etter sprint 4.

#### Trym

- Avholdt og deltatt på veiledermøte
- Gjorde research i forhold til å sette opp nevrale nettverk for å predikere værdata på de ulike bølgestasjonene.
- Brukte ganske lang tid på å sette opp, og prøve forskjellige metoder.
- Fikk uten noen svar, men de var ikke veldig lovende.

### Hvilke problemer har eventuelt oppstått

-

### Hva er planen neste uke

- I perioden frem til og med onsdag 13.03 vil vi komme til å fokusere på å øve til eksamen i Systemtenkning med økonomi. Derfor blir det et lite opphold i arbeidet med bacheloren fremover.

Tabell 4-H Timer for uke 9

Uke	9			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter	3	1	4	18
Div. admin			0	34,5
Hovedrapport			0	0
Dokumentasjon	3	4	7	19,5
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)			0	150,5
Maskinlæring	22,5	23	45,5	54,5
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	28,5	28		
Sum timer totalt	151,5	145,5		297

## 4.7 Uke 10 og 11

### Hva er gjort

- I perioden uke 10 og uke 11 tom. onsdag 13.03 hadde vi en pause fra arbeidet med bacheloroppgaven, for å fokusere på å øve til eksamen i Systemutvikling med økonomi, som vi hadde onsdag 13.03. Denne pausen hadde vi på forhånd planlagt at vi skulle ta.

I slutten av uke 11, etter eksamen, gikk vi gjennom tilbakemeldingene fra sist veiledermøte og laget et nytt format for timelistene. Det nye formatet for timelistene er tett knyttet til Sprint-task'ene vi fører i Trello, slik at det er lettere å se konkret hvor mange timer vi har brukt på hvilken task eller user story.

### Hvilke problemer har eventuelt oppstått

- 

### Hva er planen neste uke

- Veiledermøte mandag 18.03.
- Ferdigstille kode for kNN-regresjon ved manglende måledata.
- Skrive kravdokument.
- Begynne på front-end ved å lage GUI-views.

Tabell 4-I Timer for uke 10

Uke	10			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter			0	18
Div. admin			0	34,5
Hovedrapport			0	0
Dokumentasjon			0	19,5
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	0
Klassebibliotek (E39_Lib)			0	150,5
Maskinlæring			0	54,5
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	0	0		
Sum timer totalt	151,5	145,5		297

Tabell 4-J Timer for uke 11

<b>Uke</b>	<b>11</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
<b>Møter</b>			0	18
<b>Div. admin</b>	5	3	8	42,5
<b>Hovedrapport</b>			0	0
<b>Dokumentasjon</b>			0	19,5
<b>Obligatoriske forelesninger</b>			0	20
<b>Prosjektoppsett</b>			0	0
<b>Klassebibliotek (E39_Lib)</b>			0	150,5
<b>Maskinlæring</b>			0	54,5
<b>Tjener (Backend)</b>			0	0
<b>Klient (Frontend)</b>			0	0
<b>Sum timer uke</b>	5	3		
<b>Sum timer totalt</b>	156,5	148,5		305

## 4.8 Uke 12

### Hva er gjort

#### Magnus

- Jobbet videre med kNN-regresjon
- Gjort diverse endringer og refactoring ellers i koden, der det har vært nødvendig for å få kNN-regresjon til å fungere. Blant annet tilpasset hoved-pipelinen i programmet, slik at data kan hentes ut i rett steg når manglende data skal imputeres.
- Begynt å se på bruk av appsettings.json-formatet for konfigurasjon. Dette kommer vi til å benytte videre når vi skal utvikle web-backend til programmet, siden ASP.NET core bruker dette formatet for konfigurasjon.

#### Trym

- Avholdt og deltatt på veiledermøte

### Hvilke problemer har eventuelt oppstått

- Sykdom har rammet leiren!
- Vi har vært rammet av sykdom mesteparten av denne uka. Dette har naturligvis hindret oss i arbeidet, og flesteparten av gjøremålene vi hadde planlagt for denne uka fikk vi ikke gjort. Disse må vi derfor føre videre til neste uke og neste sprint.

### Hva er planen neste uke

- Fra forrige uke:
- Ferdigstille kode for kNN-regresjon ved manglende måledata.
- Skrive kravdokument.
- Begynne på front-end ved å lage GUI-views.

Tabell 4-K Timer for uke 12

<b>Uke</b>	<b>12</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter		1	1	19
Div. admin	3	3	6	48,5
Hovedrapport			0	0
Dokumentasjon			0	19,5
Obligatoriske forelesninger			0	20
Prosjektoppsett	1		1	1
Klassebibliotek (E39_Lib)			0	150,5
Maskinlæring	2		2	56,5
Tjener (Backend)			0	0
Klient (Frontend)			0	0
Sum timer uke	6	4		
Sum timer totalt	162,5	152,5		315

## 4.9 Uke 13

### Hva er gjort

#### Magnus

- Sprint review & planning etter sprint 5
- Jobbet videre med kNN-regresjon. Gikk bort fra Accord.NET, implementerte kNN-regresjon selv
- Som del av kNN: Jobbet med forskjellige metoder for å finne avstand mellom datapunkter, både i tid og rom
- Forbedret hoved-pipelinen i programmet, gjort den mer konfigurierbar.
- Gjort nødvendige endringer i DataLoader og ExcelWriter for at dataene kan fungere med funksjonaliteten for imputering.
- Arbeidsmøte med produkteier for planlegging av brukergrensesnitt, og gjennomgang av "SCAT"-rapportmalen.

#### Trym

- Satt opp webapplikasjon i backend.
- Lagd wireframes for GUI.
- Skrevet førsteutkast til kravspek.
- Jobbet en del med å lage GUI, lage REST-endepunkter og koble disse delene sammen.
- En del research om MVC 5.

### Hvilke problemer har eventuelt oppstått

- Har brukt en del tid på å få på plass en DateTime picker i GUlet som fungerte som den skulle.
- .NET-biblioteket Accord.NET var tenkt å brukes til kNN-regresjon, men på grunn av dårlig dokumentasjon og begrenset tilpasningsmulighet fikk vi ikke til å bruke biblioteket med dataene vi behandler. Siden kNN-regresjonen er en ganske simpel algoritme, implementer vi den selv istedenfor.

### Hva er planen neste uke

- Fortsette med GUI, og få satt i gang backend gjennom GUI.
- Gjøre det mulig å sette i gang generering av rapporter som en jobb i bakgrunnen i backenden på web-tjenesten.

Tabell 4-L Timer for uke 13

<b>Uke</b>	<b>13</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
<b>Møter</b>	2	2	4	23
<b>Div. admin</b>	2,5	2	4,5	53
<b>Hovedrapport</b>			0	0
<b>Dokumentasjon</b>		11	11	30,5
<b>Obligatoriske forelesninger</b>			0	20
<b>Prosjektoppsett</b>		3	3	4
<b>Klassebibliotek (E39_Lib)</b>			0	150,5
<b>Maskinlæring</b>	47		47	103,5
<b>Tjener (Backend)</b>		13,5	13,5	13,5
<b>Klient (Frontend)</b>		10	10	10
<b>Sum timer uke</b>	51,5	41,5		
<b>Sum timer totalt</b>	214	194		408



#### 4.10 Uke 14

##### Hva er gjort

###### Magnus

- Ferdigstilt funksjonaliteten for kNN-regresjon.
- Endret i og skrevet om deler av hoved-pipelinen, slik at rapporter kan genereres mellom to datoer og klokkeslett istedenfor bare mellom hele måneder.
- Ryddet opp i konfigurasjon av pipelinen og paths til hvor rådatafiler, excel-maler og ferdige rapporter skal forventes å finnes og lagres.
- Arbeidsmøte med Andreas for gjennomgang av Singleton pattern, Observer pattern, SignalR og Azure AppService deployment.
- Laget slik at imputerte verdier markeres med en farge i Excel-rapportene.
- Laget infrastruktur for at en pipeline (rapportgenerering) kan startes og kjøres i bakgrunnen i webappen. Klassene Job og JobManager håndterer de enkelte rapportgenererings-jobbene, og de trigger events når en jobb endrer status eller legger til en logg-melding. Jobber startes asynkront vha. .NET Tasks.

###### Trym

- Lagde GUI for opplasting av NetCDF filer.
- Lagde rest-enderpunkter for Upload files, og koblet sammen med GUI.
- Lagde en login-side (som ikke blir brukt, ettersom at autentisering via Azure AD tar seg av alt.
- Satte opp autentisering via Azure AD.
- Skrev på visjonsdokumentet.
- Startet med å lage et buffersystem for NetCDF-filer.

##### Hvilke problemer har eventuelt oppstått

- 

##### Hva er planen neste uke

- Begynne å koble sammen frontend og backend, sånn at det begynner å bli et brukbart produkt.

Tabell 4-M Timer for uke 14

<b>Uke</b>	<b>14</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter	3	3	6	29
Div. admin	0,5		0,5	53,5
Hovedrapport			0	0
Dokumentasjon		5	5	35,5
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	4
Klassebibliotek (E39_Lib)	19		19	169,5
Maskinlæring	6		6	109,5
Tjener (Backend)	17	30	47	60,5
Klient (Frontend)		12	12	22
Sum timer uke	45,5	50		
Sum timer totalt	259,5	244		503,5

## 4.11 Uke 15

### Hva er gjort

#### Magnus

- Laget en SignalR-hub for å sende jobbstatus til brukeren i realtime. En JobHub i backenden lytter til et SignalR-enderpunkt. Klienter mottar meldinger fra denne kanalen i frontend javascript i Create Report view. En JobHubEventPropagator lytter til Job-events fra JobHub, og videresender disse til JobHub slik at de kommer frem til klientene.
- Jobbet med Create Report view og tilhørende javascript, slik at GUIet oppdateres i henhold til den kjørende jobbens status, fremgang og logg. Bruker Redux som state container for view state.
- Optimalisert ExcelWriter-klassen og fått ned kjøretiden dens. Nå åpner og leser den inn Excel-malene bare en gang til å begynne med, og lagrer de ferdig utfylte rapportene bare en gang per stasjon.
- Avholdt og deltatt på veiledermøte
- Fikset og ryddet opp i git-repositoriet som følge av problemer med store filer.
- Fikset at Download Reports-knappen i Create Reports-viewet laster ned en zip-fil med alle rapporter når genereringen er ferdig.
- Lagt inn knapp for å avbryte kjørende jobb i Create Reports-viewet.
- Jobbet med og fikset problemer i pipeline som kjøres i en Job.
- Lagt inn filsti til Python-programmet i config, slik at vi kan konfigurere hvor Python finnes i App Servicen i Azure.

#### Trym

- Ferdigstilt buffersystem for mellomlagring av rådata NetCDF-filer og konverterte CSV-filer.
- Satt opp logging med Log4net.
- Lag en pipeline som fungerer i Job.cs slik at programmet kan kjøres.
- Satt opp blob storage for filer framfor å lagre dem på serveren.

### Hvilke problemer har eventuelt oppstått

- Det hadde på et tidspunkt med uhell blitt lagt til mange store datafiler i git-repositoriet, slik at pushing og pulling av repoet tok svært lang tid. Siden disse filene var blitt en del av git-historikken var det ikke helt enkelt å få de fjernet, men vi fikk løst problemet ved hjelp av et program som rensket opp i historikken.
- Det oppsto problem med samtidig skriving og lesing fra filer når flere brukere brukte systemet. Løste dette med å lagre filene i blob storage.

### Hva er planen neste uke

- Ferdigstille Stored Reports- og Upload Files-views.

Tabell 4-N Timer for uke 15

<b>Uke</b>	<b>15</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
<b>Møter</b>	3	3	6	35
<b>Div. admin</b>	2	1	3	56,5
<b>Hovedrapport</b>			0	0
<b>Dokumentasjon</b>			0	35,5
<b>Obligatoriske forelesninger</b>			0	20
<b>Prosjektoppsett</b>	8		8	12
<b>Klassebibliotek (E39_Lib)</b>	4		4	173,5
<b>Maskinlæring</b>			0	109,5
<b>Tjener (Backend)</b>	15,5	44,5	60	120,5
<b>Klient (Frontend)</b>	19		19	41
<b>Sum timer uke</b>	51,5	48,5		
<b>Sum timer totalt</b>	311	292,5		603,5

## 4.12 Uke 16

### Hva er gjort

#### Magnus

- Satt meg inn i EF Core, jobbet med å sette dette opp i prosjektet vårt, og laget modeller som svarer til rådata- og rapport-tabellene vi bruker i databasen.
- Jobbet med å vise frem tidligere genererte rapporter i Stored reports-viewet vha. EF Core, ut i fra tabellen med rapporter i databasen.
- Jobbet med å liste opp tilgjengelige rådatafiler i systemet i Upload Files-viewet vha. EF Core, ut i fra tabellen med rådatafiler i databasen.

#### Trym

- For å unngå at samtidig skrijving og lesing fra filer når flere brukere forsøker å bruke systemet samtidig, valgte vi å flytte filene over i Azure Blob Storage.
- La til funksjonalitet for å lese antall tomme datapunkter i en rapport.
- La til funksjonalitet for å legge til disse dataene i selve rapporten.
- Lagde en ny metode som så etter modifiserte rådatafiler.
- Etter omskriving av noen metoder var ikke estimert tid korrekt lengre, så den ble omskrevet.
- Skrev om kode, og rapportmaler sånn at de fungerer for over 1 år. Dette resulterte også i at malene ble veldig små og gikk fra å være på samlet 30 MB til 160 KB.

### Hvilke problemer har eventuelt oppstått

- Oppdaget at funksjonen som så etter om en rådatafil hos Met hadde blitt modifisert i det siste, ikke gjorde det den skulle.
- Oppdaget at rapportmalene til prosjektet bare fungerte for perioder på 6 måneder eller mindre.

### Hva er planen neste uke

- Ferdigstille funksjonalitet for nedlasting, sletting og opplasting av filer til/fra blobstorage i Stored Reports- og Upload Files-viewene.
- Ferdigstille øvrig funksjonalitet i systemet.
- La produkteier og Andreas Ravnstad bruke og teste systemet, og fikse eventuelle feil de skulle finne.

Tabell 4-O Timer for uke 16

<b>Uke</b>	<b>16</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
<b>Møter</b>			0	35
<b>Div. admin</b>		1	1	57,5
<b>Hovedrapport</b>			0	0
<b>Dokumentasjon</b>			0	35,5
<b>Obligatoriske forelesninger</b>			0	20
<b>Prosjektoppsett</b>			0	12
<b>Klassebibliotek (E39_Lib)</b>		12	12	185,5
<b>Maskinlæring</b>			0	109,5
<b>Tjener (Backend)</b>	13,5	11,5	25	145,5
<b>Klient (Frontend)</b>	9		9	50
<b>Sum timer uke</b>	22,5	24,5		
<b>Sum timer totalt</b>	333,5	317		650,5

#### 4.13 Uke 17

##### Hva er gjort

###### Magnus

- Byttet ut manuelle spørringer mot SQL-databasen i Job og FileCacher med kall til EF Core database-konteksten.
- Lagt inn funksjon for å laste ned en tidligere generert rapport.
- Lagt inn funksjon for å slette en tidligere generert rapport.
- Lagt inn funksjon for å kunne laste opp rådatafiler manuelt.
- Lagt inn funksjon for å slette en rådatafil. Statusen til rapporter som er generert ut i fra denne rådatafilen settes til "Unknown".
- Gjort at FileCacher logger mer til loggfil.
- Satt opp EF Core Migrations, som er et rammeverk som oppdaterer databasestrukturen til å stemme overens med modellene ved eventuelle endringer.
- Fikset at planlagte Quartz-jobber får nødvendige objekter vha. dependency injection, og lagt inn at sletting av gamle jobb-filer kjøres periodisk.
- Testet å lese inn NetCDF-filer direkte i C# vha. SDSLite, som tidligere hadde blitt satt på vent. Vi ønsker å prøve å få dette til å fungere, da det vil redusere kjøretiden og eliminere Python-avhengigheten til applikasjonen.
- Rensket opp i Config-parametre som ikke lengre brukes, samt diverse refactoring.

###### Trym

- Fikset ferdig sånn at rapportmalene fungerer for uansett lengde på filer.
- Ordnet GUI sånn at det var mere praktisk, og fungerte bedre for skjermer med lavere oppløsning.
- Satte opp en jobb for å slette en mappe med midlertidige filer, periodisk.

### Hvilke problemer har eventuelt oppstått

- App Service i Azure kræsjer noen ganger under generering av rapport, når Excel-maler lastes inn. Muligens forårsaket av høyt minne- og CPU-bruk.
- File Cacher kjører ikke når den skal hver dag kl 02:00. Det er muligens pga. at Always On ikke var påslått for App Servicen. Prøver å slå den på og sjekker om det hjelper.
- GUI var ikke optimalisert og fungerte veldig dårlig for skjermer med lavere oppløsning.

### Hva er planen neste uke

- Jobbe med systemdokumentasjon og hovedrapport

Tabell 4-P Timer for uke 17

Uke	17			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter	1		1	36
Div. admin	1,5		1,5	59
Hovedrapport			0	0
Dokumentasjon			0	35,5
Obligatoriske forelesninger			0	20
Prosjektoppsett	12		12	24
Klassebibliotek (E39_Lib)		12,5	12,5	198
Maskinlæring			0	109,5
Tjener (Backend)	24,5	2	26,5	172
Klient (Frontend)	7	10	17	67
Sum timer uke	46	24,5		
Sum timer totalt	379,5	341,5		721



#### 4.14 Uke 18

##### **Hva er gjort**

###### **Magnus**

- Gjort diverse små feilrettinger og finpuss, bl.a. presentering av estimert gjenstående tid for en jobb i GUI, og vise rapporter og rådatafiler i sortert rekkefølge.
- Videre testing av innlesing av NetCDF-filer vha. SDSLite. På grunn av en feil i programvarebiblioteket eller i rådatafilene feilet derimot innlesingen av enkelte filer. Denne funksjonaliteten ble derfor lagt på is, da vi ikke ville få tid til å se nærmere på årsaken til feilen. Den opprinnelige metoden ved å først konvertere NetCDF-filene til CSV vha. Python-skriptet blir derfor stående.
- Skrevet og rettet opp i API-dokumentasjon i koden

###### **Trym**

- Ordnet GUI sånn at det fungerer bedre for flere oppløsninger.
- Ordnet listene for Update Raw Data Files og Stored Reports sånn at de tar mindre plass.
- Skrevet systemdokumentasjon.
- Skrevet på hovedrapporten.

##### **Hvilke problemer har eventuelt oppstått**

- Innlesing av NetCDF-filer i C# med SDSLite fungerte ikke for alle filer, og måtte derfor legges på is.

##### **Hva er planen neste uke**

- Jobbe med systemdokumentasjon og hovedrapport.

Tabell 4-Q Timer for uke 18

<b>Uke</b>	<b>18</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter	2	2	4	40
Div. admin	3		3	62
Hovedrapport	4,5	20	24,5	24,5
Dokumentasjon	12		12	47,5
Obligatoriske forelesninger			0	20
Prosjektoppsett		19	19	43
Klassebibliotek (E39_Lib)	3		3	201
Maskinlæring			0	109,5
Tjener (Backend)	8		8	180
Klient (Frontend)	7	7,5	14,5	81,5
Sum timer uke	39,5	48,5		
Sum timer totalt	419	390		809

#### 4.15 Uke 19

##### **Hva er gjort**

###### **Magnus**

- Jobbet med å kryssvalidere kNN-regresjonen, for å avgjøre hvor nøyaktige verdier den estimerer
- Skrevet teori om imputering og kNN i hovedrapporten.
- kNN-regresjon: Lagt inn utregning av modulært gjennomsnitt for features som måles i grader. Vanlig, aritmetisk gjennomsnitt gir feil resultat for slike sirkulære mål.

###### **Trym**

- Skrevet sammendrag og forord til hovedrapport.
- Skrevet om ingeniørfaglige resultater og tilhørende diskusjon.
- Startet på konklusjon.
- Laget klassesdiagram i systemdokumentasjonen.

##### **Hvilke problemer har eventuelt oppstått**

- 

##### **Hva er planen neste uke**

- Ferdigstille hovedrapporten
- Ferdigstille systemdokumentasjon
- Ferdigstille øvrige dokumenter og prosjekthåndbok
- Klargjøre alle dokumenter og kildekode for innlevering

Tabell 4-R Timer for uke 19

<b>Uke</b>	<b>19</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter			0	40
Div. admin			0	62
Hovedrapport	54	49,5	103,5	128
Dokumentasjon		7,5	7,5	55
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	43
Klassebibliotek (E39_Lib)			0	201
Maskinlæring			0	109,5
Tjener (Backend)	3		3	183
Klient (Frontend)			0	81,5
Sum timer uke	57	57		
Sum timer totalt	476	447		923

#### 4.16 Uke 20 og 21

##### Hva er gjort

###### Magnus

- Ryddet og korrigert enkelte timeføringer i timelistene, samt fylt ut manglende statusrapporter
- Skrevet videre på teori om og valg av imputering og kNN i hovedrapporten
- Skrevet om valg av designmønstre i hovedrapporten

###### Trym

- Ferdigstilt på hovedrapporten.
- Skrevet om vitenskapelige resultater i forhold til redusering av tidsbruk, samt diskusjon for dette.
- Skrevet om vitenskapelige resultater i forhold til eliminering av feilkilder som følge av menneskelig interaksjon, og diskusjon for dette.
- Gjort om dokumentene fra Google Docs til Word.
- Ordnet prosjekthåndboka.

##### Hvilke problemer har eventuelt oppstått

- 

##### Hva er planen neste uke

- Forberede presentasjon av produktet

Tabell 4-5 Timer for uke 20

Uke	20			
Aktivitet	Magnus	Trym	Sum aktivitet uke	Sum aktivitet totalt
Møter	2,5	2,5	5	45
Div. admin	3		3	65
Hovedrapport	17,5	15,5	33	161
Dokumentasjon		12,5	12,5	67,5
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	43
Klassebibliotek (E39_Lib)			0	201
Maskinlæring			0	109,5
Tjener (Backend)			0	183
Klient (Frontend)			0	81,5
Sum timer uke	23	30,5		

<b>Sum timer totalt</b>	499	477,5		976,5
-------------------------	-----	-------	--	-------

Tabell 4-T Timer for uke 21

<b>Uke</b>	<b>21</b>			
<b>Aktivitet</b>	<b>Magnus</b>	<b>Trym</b>	<b>Sum aktivitet uke</b>	<b>Sum aktivitet totalt</b>
Møter			0	45
Div. admin	2		2	67
Hovedrapport	10	13	23	239,5
Dokumentasjon	2	2	4	74,5
Obligatoriske forelesninger			0	20
Prosjektoppsett			0	43
Klassebibliotek (E39_Lib)			0	201
Maskinlæring			0	109,5
Tjener (Backend)			0	183
Klient (Frontend)			0	81,5
<b>Sum timer uke</b>	14	15		
<b>Sum timer totalt</b>	541,5	522,5		1064

## 5 Sprint Boards

For å organisere og dokumentere sprintene i prosjektet brukte vi verktøyet Trello. Før hver sprint ble det tatt bilde av Sprint Backlog. Etter hver sprint ble det tatt bilde av listen «Done», som inneholdt tasks/kort vi hadde fullført i løpet av sprinten.

For de fleste sprintene kan du se at «Review – done» inneholder flere kort enn «Planning - backlog». Dette er enten fordi, et kort på «Planning – backlog» ble splittet opp i flere små lettere håndterbare kort, eller fordi kort fra forrige sprint henger igjen og ble fullført. Fra sprint 6 og ut har kortene fått løpenummer. Dette ble brukt for å føre timeforbruk på de forskjellige kortene i timelisten.

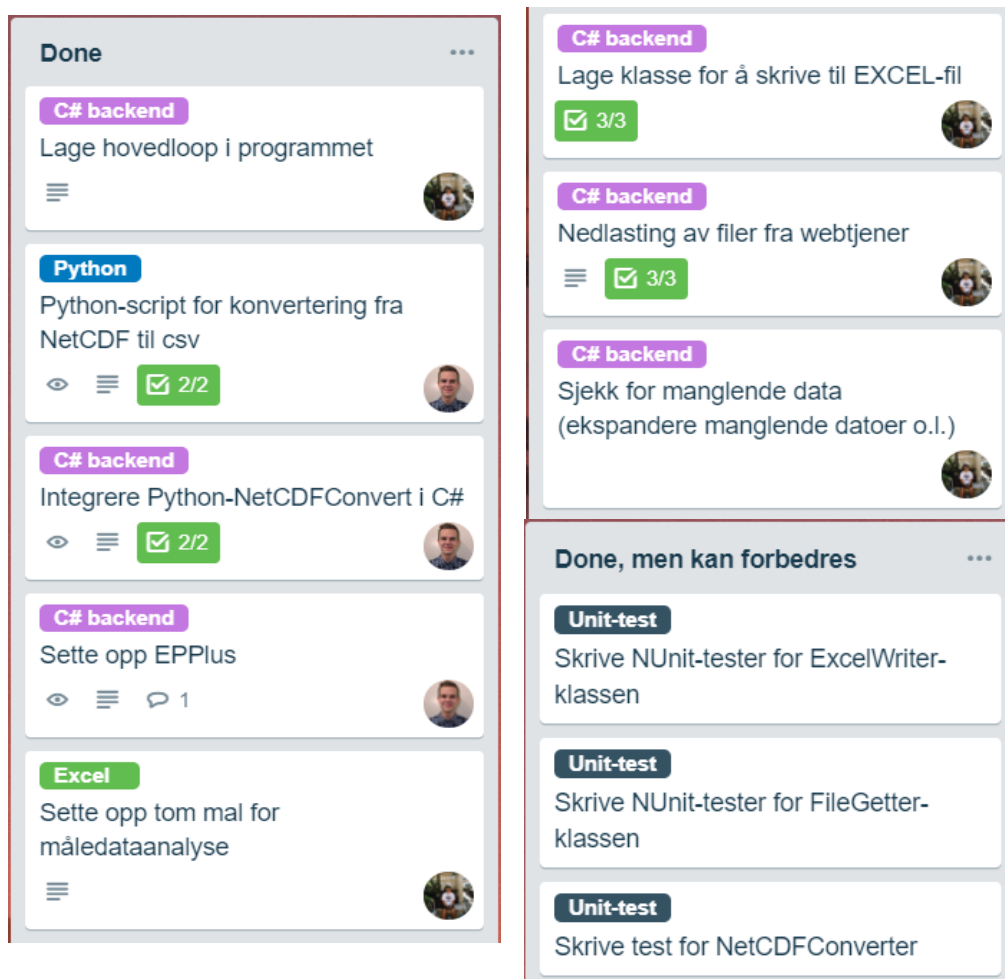
### 5.1 Sprint 1

tor. 24.01.2019 (uke 4) - fre. 08.02.2019 (uke 6)

#### Planning – backlog:

- Lage funksjonalitet for nedlasting av måledatafiler fra webtjener
- Lage funksjonalitet for å lese inn NetCDF-filer i programmet
- Sette opp mal og funksjonalitet for å skrive måledataanalyse til Excel-fil

#### Review – done:



## 5.2 Sprint 2

fre. 08.01.2019 (uke 6) - fre. 15.02.2019 (uke 7)

### Planning – backlog:



### Review – done:

På grunn av lite tid til bachelorprosjektet denne uka, ble det ikke nok tid til å fullføre denne sprintens gjøremål. Denne føres derfor over videre til neste sprint.



### 5.3 Sprint 3

fre. 15.02.2019 (uke 7) - tor. 21.02.2019 (uke 8)

#### Planning – backlog:

**Sprint Backlog** ...

**C# backend** **Python**  
Sette opp parallellisering for konvertering av NetCDF-filene


**C# backend** **Python**  
Undersøke en passende maskinlæringsteknikk for imputering




**BUG**  
Sørge for at data blir vist fram i grafene når det skrives til tom mal





**Dokumentering**  
Førsteutkast visjonsdokument


#### Review – done:





**Done - Sprint 3** ...




**Excel** **C# backend** **Forbedring**  
Skrive om Excel-mal sånn at den ikke bruker "Samlet"-fanen. Endre ExcelWriter-klassen sånn at den slipper å slette overflødige formler.  


**C# backend** **Python**  
Sette opp parallellisering for konvertering av NetCDF-filene  
  2/2 

**C# backend**  
Lage felles, cross-platform metoder for tilgang til data-mapper  
   1/1 

**BUG**  
Sørge for at data blir vist fram i grafene når det skrives til tom mal  


**BUG** **C# backend**  
Fikse programkrasj i hovedloop  
  1/1  

**Dokumentering**  
[Uferdig] Førsteutkast visjonsdokument  
  

## 5.4 Sprint 4

tor. 21.02.2019 (uke 8) - fre. 01.03.2019 (uke 9)

### Planning - backlog:

**Sprint Backlog** ...

**C# backend** **Python**  
Undersøke passende maskinlæringsteknikker for imputering



**BUG** **C# backend**  
Fylle data inn på rett rad i Excel-mal i forhold til tidssone


**Dokumentering**  
Fullføre førsteutkast visjonsdokument




**Dokumentering**  
Førsteutkast kravdokument




### Review - done:


**Done - Sprint 4** ...


**C# backend** **Imputering**  
Implementere lineær-interpolasjon til å fylle manglende data  
 

**C# backend** **Python** **Imputering** ✎  
Lage en modell for generering av manglende måledata  


**C# backend** **Python** **Imputering**  
Undersøke passende maskinlæringsteknikker for imputering  
  

**Dokumentering**  
Fullføre førsteutkast visjonsdokument  
  26. feb 

**C# backend** **Forbedring**  
Separere måledata-objekt fra ExcelWriter  


**Excel** **C# backend**  
Sørge for at Excel-Writer også kan skrive og opprette SCAT-filer  


## 5.5 Sprint 5

fre. 15.03.2019 (uke 11) - man. 25.03.2019 (uke 13)

### Planning - backlog:



**Sprint Backlog** ...

- GUI**  
GUI
- C# backend**  
Sette opp logging
- C# backend** **Forbedring**  
Konfigurasjonsfil (appsettings.json)
- C# backend** **Imputering**  
Implementere kNN-regresjon for generering av manglende måldata
- Deployment**  
Sette opp Continuous Integration med automatisk testing
- BUG** **C# backend**  
Fylle data inn på rett rad i Excel-mal i forhold til tidssone
- Dokumentering**  
Førsteutkast kravdokument

### Review - done:

På grunn av sykdom i teamet hele uke 12, fikk vi ikke til å fullføre sprintens tasks. På sprint planning-møte ble derfor flere av task'ene ført over videre til neste sprint.

## 5.6 Sprint 6

man. 25.03.2019 (uke 13) - man. 01.04.2019 (uke 14)

### Planning – backlog:

#### Sprint Backlog

**C# backend** **Imputering**  
#5.1 Implementere kNN-regresjon for generering av manglende måledata

**C# backend** **Forbedring**  
#5.2 Konfigurasjonsfil (appsettings.json)

**GUI**  
#6.1 Lage wireframe for GUI

**Dokumentering**  
#6.2 Førstekast kravdokument

**Excel**  
#6.3 Se på SCAT-excel-filene, og hør hva ting er inni der

**BUG** **C# backend**  
#6.4 Fylle data inn på rett rad i Excel-mal i forhold til tidssone

**C# backend**  
#6.5 Legg til ASP.NET Core-prosjekt i solution

**C# backend**  
#6.6 Sette opp nødvendige klasser for å få opp en web-backend (ASP.NET Core)

### Review – done:

#### Done - Sprint 6

**C# backend** **Imputering**  
#5.1 Implementere kNN-regresjon for generering av manglende måledata  
12/14

**C# backend** **Forbedring**  
#5.2 Konfigurasjonsfil (appsettings.json)

**Excel**  
#6.3 Se på SCAT-excel-filene, og hør hva ting er inni der

**C# backend**  
#6.5 Legg til ASP.NET Core-prosjekt i solution

**C# backend**  
#6.6 Sette opp nødvendige klasser for å få opp en web-backend (ASP.NET Core)

**BUG** **C# backend**  
#6.4 Fylle data inn på rett rad i Excel-mal i forhold til tidssone  
1/2

**GUI**  
#6.1 Lage wireframe for GUI  
4/4

**Dokumentering**  
#6.2 Førstekast kravdokument  
3/3

**C# backend** **JavaScript** **AJAX**  
#6.8 Koble Create Report opp med REST-endepunktene i backend  
3/3

**GUI**  
#6.7 GUI: Create Report  
5/5

**GUI**  
#6.9 GUI: Saved Reports  
3/3

**C# backend** **JavaScript** **AJAX**  
#6.10 Koble Saved Reports opp med REST-endepunktene i backend  
3/3

## 5.7 Sprint 7

man. 01.04.2019 (uke 14) - man. 08.04.2019 (uke 15)

### Planning – backlog:

#### Sprint Backlog

**C# backend** **JavaScript** **AJAX**

#7.1 Koble Upload Files opp med REST-endepunktene i backend  
☑ 0/4

**BUG** **C# backend**

#7.2 Fikse at DataLoader slutter å lese inn en csv-fil når den møter en rad med bare 0-er  
☑ 0/1

**C# backend**

#7.3 Lage Tasks for å generere rapporter i parallell  
☑ 0/6

**GUI**

#7.4 GUI: Login page

**GUI** **C# backend**

#7.5 Autentisering  
☑ 0/2

**Dokumentering**

#7.6 Ferdigstille siste versjon visjonsdokument  
☑ 0/0

**C# backend** **Forbedring**

#7.7 Fikse opp i hvor programmet forventer å lagre filer og finne maler  
☑ 0/0

### Review – done:

#### Done - Sprint 7

**C# backend** **Forbedring**

#7.7 Fikse opp i hvor programmet forventer å lagre filer og finne maler  
☑ 4/4

**C# backend**

#7.10 Lage et generelt Config-objekt som Pipeline bruker  
☑ 3/3

**BUG** **C# backend**

#7.2 Fikse at DataLoader slutter å lese inn en csv-fil når den møter en rad med bare 0-er  
☑ 0/1

**Excel** **C# backend** **Imputering**

#6.12 Markere imputerte verdier i rapport  
☑ 3/4

**C# backend**

#7.8 Endre backend sånn at rapporter kan lages mellom 2 bestemte klokkeslett  
☑ 0/0

**GUI**

#6.11 GUI: Upload files  
☑ 3/3

**C# backend** **JavaScript** **AJAX**

#7.1 Koble Upload Files opp med REST-endepunktene i backend  
☑ 0/4

**GUI**

#7.4 GUI: Login page  
☑ 4/4

**GUI** **C# backend**

#7.5 Autentisering  
☑ 1/2

## 5.8 Sprint 8

man. 08.04.2019 (uke 15) - fre. 12.04.2019 (uke 15)

### Planning – backlog:

**Sprint Backlog**

**C# backend**  
#8.1 Sette opp logging

**C# backend**  
#8.2 Lage SignalR-hub for logg og status for jobber  
0/4

**C# backend**  
#8.3 Lage klasse/service som persisterer rapportdata til DB når jobb er ferdig  
0/3

### Review – done:

**Done - Sprint 8**

**C# backend**  
#8.3 Lage klasse/service som persisterer rapportdata til DB når jobb er ferdig

**BUG**  
Scatter plot i vanlig rapport er ikke som den skal være

**C# backend**  
#8.5 Lag en fungerende pipeline

**GUI** **JavaScript**  
#8.4 Create report view: Logikk og SignalR for jobb-status og logg

**C# backend** **Forbedring**  
#8.6 Optimalisere ExcelWriter

**C# backend**  
#7.3 Lage og kjøre en Task når en Job settes i gang  
4/4

**C# backend**  
#7.9 Lage Job og JobManager-klasser  
3/3

**C# backend**  
#8.2 Lage SignalR-hub for logg og status for jobber  
3/3

**C# backend** **Database**  
#7.11 Lag buffersystem for NetCDF/rådatafiler  
4/4

**C# backend**  
#8.1 Sette opp logging

## 5.9 Sprint 9

fre. 12.04.2019 (uke 15) - ons. 24.04.2019 (uke 17)

### Planning – backlog:

**Sprint Backlog** ...

**GUI** **C# backend** **Database**  
#9.1 Stored reports: Liste over ferdige rapporter fra DB  
☑ 0/2

**GUI** **C# backend** **Database**  
#9.2 Uploaded raw data files: Liste over eksisterende rådatafiler fra DB

**C# backend** **Database**  
#9.3 Uploaded raw data files: Last opp og lagre rådatafil

**C# backend**  
#9.4 Telle antallet tomme, imputerte og totalt datapunkter

**Excel** **C# backend**  
#9.5 Skrive metadata og antall manglende datapunkter til Excel-rapporter

### Review – done:

**Done - Sprint 9**

**GUI** **C# backend** **Database**  
#9.2 Uploaded raw data files: Liste over eksisterende rådatafiler fra DB  
👁 ☑ 3/3

**GUI** **C# backend** **Database**  
#9.1 Stored reports: Liste over ferdige rapporter fra DB  
👁 ☑ 4/4

**C# backend** **Database**  
#9.9 Bruke EF Core istedenfor manuelle SQL-spørringer  
👁 ☑ 3/3

**C# backend** **Python**  
#9.8 Legge path til Python i config  
👁

**GUI** **C# backend**  
#9.7 Create report: Knapp for å avbryte jobb, fjerne ferdige jobber  
👁

**GUI** **C# backend**  
#8.7 Create report: Last ned zip-fil med ferdige rapporter ved knappetrykk  
👁

**C# backend** **Forbedring**  
#9.6 Sette opp blobstorage for filer

**BUG**  
#9.10 Fikse sånn at file-cacher ser etter rett dato for modifisering.

**Excel** **C# backend**  
#9.5 Skrive metadata og antall manglende datapunkter til Excel-rapporter

**C# backend**  
#9.4 Telle antallet tomme, imputerte og totalt datapunkter

## 5.10 Sprint 10

ons. 24.04.2019 (uke 17) - 14.05.2019 (uke 20)

### Planning – backlog:

**Sprint Backlog**

C# backend Database

#9.3 Uploaded raw data files: Last opp og lagre rådatafil  
0/3

GUI

#10.1 Finpusse GUI for forskjellige oppløsninger

C# backend

#10.2 Slett temp-mappe for zip-filer når download-knappen har blitt trykket på

C# backend

#10.3 FileCacher: Endre status på rapporter når rådatafiler erstattes

Excel BUG

#10.4 Rapportmalene fungerer bare for 6 måneder

GUI C# backend

#10.5 Finpusse presentering av data i GUI  
0/4

### Review – done:

**Done - Sprint 10**

GUI C# backend

#10.4 Finpusse presentering av data i GUI  
10/10

C# backend

#10.2 Slett temp-mappe for zip-filer når download-knappen har blitt trykket på

C# backend

#10.3 FileCacher: Endre status på rapporter når rådatafiler erstattes

C# backend Database

#9.3 Uploaded raw data files: Last opp og lagre rådatafil  
3/3

Excel C# backend

#9.12 Gjøre sånn at rapportmalene fungerer for uansett lengde på tidsperiodene

Excel BUG

#9.12 Rapportmalene fungerer bare for 6 måneder

GUI

#10.1 Finpusse GUI for forskjellige oppløsninger

GUI C# backend

#9.11 Fikse oppdatering av estimert tid og progressbar.