

FORPROSJEKT - RAPPORT

FOR BACHELOROPPGAVE

TITTEL:

Websystem for håndtering av release notes

KANDIDATNUMMER(E):

Lars Øyvind Ous
Markus Randa
Jan Anton Lande Pedersen

DATO:

30/01/2020

EMNEKODE: *

IE303612

EMNE:

Bacheloroppgave (Data)

DOKUMENT TILGANG:

- Åpen

STUDIUM:

DATAINGENIØR

ANT SIDER/VEDLEGG:

14/1

BIBL. NR:

- Ikke i bruk -

OPPDRAUGSGIVER(E)/VEILEDER(E):

Oppdragsgiver Harald Bjørshol
Veileder Girts Strazdins
Bi-veileder Anniken Susanne T. Karlsen

+OPPGAVE/SAMMENDRAG:

Dette er forprosjektrapporten til bacheloroppgaven «Release note system».

Denne bacheloroppgaven er gitt av Cordel, der utførende gruppe skal produsere et system som skal brukes av Cordel og deres kunder. Oppgaven beskriver at det skal lages et system som gjør det enkelt for kunder å sjekke hva som er nytt i hver utgivelse av en ny versjon. I tillegg skal det være lett for utviklere å dokumentere disse utgivelsene. Her skal det legges vekt på å lage en moderne nettside, med mulighet for kundene å enkelt finne frem og enkelt for redaktør å opprette disse release-artiklene.

I denne forprosjektrapporten er det definert at scrum blir benyttet som utviklingsmetodikk og hvordan scrum blir implementert i dette prosjektet. Verktøyet som brukes for prosjektstyring er Azure DevOps, der det finnes funksjonalitet for scrum-tavle. Til utvikling brukes verktøyene JetBrains Rider og Visual Studio Code som er utviklingsprogram.

En av de store utfordringene i dette prosjektet er at det potensielt sett kan bli mange system som skal spille på lag når systemet skal implementeres inn hos Cordel.

**Denne oppgaven er en eksamensbesvarelse utført av
student(er) ved NTNU i Ålesund.**

INNHold

INNHold	2
1 INNLEDNING	3
2 BEGREPER	3
3 PROSJEKTORGANISASJON	4
3.1 PROSJEKTGRUPPE	4
3.1.1 Oppgaver for prosjektgruppen - organisering	4
3.1.2 Roller	4
3.2 STYRINGSGRUPPE (VEILEDER OG KONTAKTPERSON OPPDRAGSGIVER)	4
4 AVTALER	4
4.1 AVTALE MED OPPDRAGSGIVER	4
4.2 ARBEIDSSTED OG RESSURSER	4
4.3 GRUPPENORMER – SAMARBEIDSREGLER – HOLDNINGER	5
5 PROSJEKTBESKRIVELSE	6
5.1 PROBLEMSTILLING	6
5.2 KRAV TIL LØSNING ELLER PROSJEKTRESULTAT – SPESIFIKASJON	6
5.2.1 Spesifikasjoner	6
5.2.2 Utvidelsesmulighet for oppgaven	7
5.3 PLANLAGT FRAMGANGSMÅTE FOR UTVIKLINGSARBEIDET – METODE	7
5.3.1 Utviklingsmetodikk	7
5.3.2 Arbeidsstruktur	7
5.4 INFORMASJONSINNSAMLING – UTFØRT OG PLANLAGT	8
5.5 VURDERING – ANALYSE AV RISIKO	9
5.6 HOVEDAKTIVITETER I VIDERE ARBEID	9
5.7 FRAMDRIFTSPLAN – STYRING AV PROSJEKTET	9
5.7.1 Hovedplan	9
5.7.2 Styringshjelpemidler	10
5.7.3 Utviklingshjelpemidler	10
5.7.4 Intern kontroll – evaluering	11
5.8 BESLUTNINGER – BESLUTNINGSPROSESS	11
6 DOKUMENTASJON	12
6.1 RAPPORTER OG TEKNISKE DOKUMENTER	12
7 PLANLAGTE MØTER OG RAPPORTER	12
7.1 MØTER	12
7.2 PERIODISKE RAPPORTER	12
7.2.1 Framdriftsrapporter	12
7.2.2 Logg	12
8 PLANLAGT AVVIKSBEHANDLING	13
9 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING	13
10 VEDLEGG	14
11 REFERANSER	15

1 INNLEDNING

Oppgaven er en oppgave tildelt av Cordel Norge, som trenger et websystem som håndterer «release notes» for forskjellige systemer som utvikles av bedriften. Løsningen skal følge prinsippene om en trelags-arkitektur med database, API, og front-end. Cordel er en bedrift som bedriver smidig utvikling, som medfører at det utgis hyppige oppdateringer av programvare ut til kunder. De trenger med dette et system som skal forenkle prosessen å dokumentere utgivelser på en måte som både er enkelt for utviklerne, og brukervennlig for brukerne. Målet med oppgaven er å utvikle et system som forenkler og effektiviserer prosessen å utgi en programvareoppdatering med dokumentasjon over endringene.

2 BEGREPER

Begrep	Definisjon
Front-end	En samlebetegnelse på delen av systemet som blir presentert til brukeren. Dette vil være nettsiden.
Back-end	En samlebetegnelse på deler av systemet brukeren ikke direkte jobber med. I dette tilfellet er det API og database som er back-end
Mikrotjeneste	En liten tjeneste som oppnår et spesifikt mål.
Wireframe	En visuell presentasjons av innholdet til en nettside. Viser hvor felt, knapper og tekst er plassert. Indikerer også hvor sidene leder brukeren til.
MVP	Akronym for «Minimal Value Product». Definer hva som er minste kravet for produktet.
DevOps	Akronym for «development and operation». Et uttrykk for oppgaver som binder utvikling og operative systemer.
Confluence	Et verktøy for å bedrive smidig utvikling
AutoDeploy	En hosting tjeneste fra NTNU for virtuelle maskiner.

3 PROSJEKTORGANISASJON

3.1 Prosjektgruppe

Studentnummer(e)
460200 Lars Øyvind Ous 492850 Markus Randa 492828 Jan Anton Lande Pedersen

3.1.1 Oppgaver for prosjektgruppen - organisering

3.1.2 Roller

Møtereferent	Jan Anton Lande Pedersen
Scrum-master	Lars Ous
Kommunikasjonsansvarlig	Markus Randa
Utviklings-team	Markus Randa, Lars Ous, Jan Anton Lande Pedersen

3.2 Styringsgruppe (veileder og kontaktperson oppdragsgiver)

Veileder	Girts Strazdins
Bi-veileder	Anniken Susanne T. Karlsen
Kontaktperson	Harald Bjørshol
Oppdragsgiver	Cordel Norge

4 AVTALER

4.1 Avtale med oppdragsgiver

Det signeres en arbeidsavtale mellom oppdragsgiver og gruppen. Denne avtalen sier i korte trekk hva slags forpliktelser gruppen og oppdragsgiver har i dette prosjektet.

Det skal også signeres en taushetserklæring mellom oppdragsgiver og gruppen. Kontrakten som brukes vil være NTNU sin ferdigdefinerte kontrakt, som forteller i korte trekk at gruppen ikke skal utgi noe informasjon som kan skade oppdragsgiver, men at oppdragsgiver er pliktig å tydeliggjøre hva som er taushetsbelagt. Dette betyr at gruppen beholder eierskap til kildekoden, og kan offentliggjøre den.

4.2 Arbeidssted og ressurser

Arbeid vil foregå på lab, romnummer L167. Om L167 ikke er tilgjengelig blir andre lab-rom, eventuelt grupperom anvendt. Alle nødvendige ressurser er allerede sørget for av NTNU. Dette inkluderer tilgang til autodeploy og lisenser til programvare.

4.3 Gruppenormer – samarbeidsregler – holdninger

Gruppen arbeider i tiden 8-16, mandag til fredag. Hvert medlem er pliktig å melde fra til gruppen om man ikke kan møte opp. Alle større avgjørelser tas sammen med hele gruppen, der det blir gjort demokratiske valg som avgjør hvilken løsning gruppen går for.

Samarbeidet i gruppen baseres på Scrum-prinsipper. Det holdes daglige stand-up møter. I prinsippet legges det med dette opp til at gruppen har en åpen kommunikasjon med sunne diskusjoner.

Gruppen skal jobbe profesjonelt mot målet om å skape et produkt som er både er nyttig for oppdragsgiver, og som gir god erfaring og læring for veien videre.

5 PROSJEKTBESKRIVELSE

5.1 Problemstilling

Cordel har de siste årene arbeidet med å automatisere og effektivisere samtlige utviklings-prosesser, etter prinsipper som Continuous Integration og Continuous Deployment. Dette medfører at antallet oppdateringer som blir pushet ut til produksjon øker betydelig. Det er ikke uvanlig at det blir publisert flere oppdateringer per dag, med så mange som tusenvis oppdateringer per uke for større bedrifter.

Et av leddene i denne prosessen er å dokumentere til kundene hva som er nytt i hver utgivelse. Per dags dato gjøres dette i Cordel gjennom en ineffektiv, manuell prosess.

Gruppens mål for prosjektet er gitt i tabellen under.

Effektmål	Skape et system som forenkler og effektivisere prosessen å dokumentere og publisere utgivelser
Prosessmål	Gjennom prosjektet skal gruppen også oppnå og vise en god forståelse for programvaredesign, prosjektstyring og bruk av smidig utviklingsteknikk.
Resultatmål	Løsningen skal tilfredsstill krav stilt av vår oppdragsgiver, og gir et funksjonelt sluttprodukt som er bedre enn det som brukes av Cordel i dag. Målet deles opp i flere deler: <ul style="list-style-type: none">• <i>MVP</i>; Et minimum for hva som utgjør et funksjonelt system• <i>Automatisert system</i>; Integrering med Cordels versjonskontroll-systemer.• <i>Dypere integrasjon</i>; Integrasjon med flere av Cordels interne systemer. Eks. Cordels support-system.

5.2 Krav til løsning eller prosjekresultat – spesifikasjon

Oppgaven er gitt med beskrivelse av ønsket funksjonalitet, samt forslag til teknologi som brukes i løsningen. Her har gruppen valgt å bruke de foreslåtte teknologiene og rammeverkene.

5.2.1 Spesifikasjoner

- Systemet skal benytte prinsippet om REST-API for kommunikasjon mellom backend og front-end.
- Systemet skal benytte prinsippet om en trelagsarkitektur, med database, API og front end.
- Alle nettsider skal være responsive.
- Systemet skal kjøres i form av mikrotjenester oppnådd med Docker.
- Applikasjonsserver skal skrives i programmeringsspråket C#.
- Systemet skal bruke en PostgreSQL-database.
- Web-app skal skrives i React.

Funksjonelle krav

- Administrasjonsside som er ment for utviklerne i Cordel Norge. Denne siden trenger følgende funksjonalitet:
 - Opprette og redigere sider for nye systemer som Cordel Norge utvikler
 - Opprette og redigere release notes for valgt system
 - Opprette og redigere brukere
- Egen side uten innlogging. Denne siden er ment for Cordel Norge sine ansatte og kunder. Her skal man kun ha lesetilgang, og kunne se release notes for de forskjellige systemene.

Kravene over utgjør basisen for hva som anses som et fullført prosjekt. Det er også gitt utvidelsesmuligheter for oppgaven, om det skulle bli rom for dette.

5.2.2 Utvidelsesmulighet for oppgaven

Utvide nettsiden slik at den kan eksponere data fra Cordels support system.» Dette vil innebære følgende utvidelser:

- Må kunne hente og sende data fra Cordels «ticket system»
- Integrering med Cordels versjonskontroll
- GUI for en tredje side som skal vise supportsaker
- Kunder skal få tildelt brukere. Når de logger inn vil de få oversikt og status på supportsaker som er relevante for dem.

5.3 Planlagt framgangsmåte for utviklingsarbeidet – metode

5.3.1 Utviklingsmetodikk

Dette prosjektet trengte en utviklingsmetodikk som sikret oss en strukturert kontinuerlig dialog med oppdragsgiver, fleksibel planlegging, oversikt over arbeidsmengde og en måte å estimere arbeidet på. Det ble gjennomført en utredning av kjente utviklingsmetodikker, der gruppen så på fossefallsmetoden, scrum og kanban. Det gruppen fant ut etter utredningen var at scrum passet best de kravene som ble stilt til utviklingsmetodikk.

I fossefallsmetoden så gjennomfører man en lengre planleggingsprosess i starten av et prosjekt, som blir den eneste planleggingsprosessen og er en veldig rigid måte å styre prosjekt på. Denne metodikken tilbyr dessverre ingen av de teknikkene som dette prosjektet hadde behov for.ⁱ

Kanban inneholder flere teknikker som er nyttige for dette prosjektet. Der den sikrer en god oversikt over gjenstående arbeidsmengde og fleksibilitet i hvordan prosjekter planlegges. Det kanban manglet var en strukturert kontinuerlig dialog, og en måte å estimere arbeidet på.ⁱⁱ

Scrum var den metodikken som inneholdt flest nyttige teknikker. Der scrum tilbyr en strukturert kontinuerlig dialog med både teamet, oppdragsgiver og veileder. Andre teknikker scrum tilbyr er fleksibel planlegging, oversikt over arbeidsmengde og en måte å estimere arbeidsmengde i form av story points. Det story points i all hovedsak er, er en måte å abstrakt og kjapt definere hvor mye innsats teamet må gi for å få gjennomført et funksjonelt krav.ⁱⁱⁱ

Det finnes også noen svakheter med Scrum, der utfordringen ligger i at hele Scrum-teamet må bestå av personer med god erfaring med denne metodikken. En annen ting er at det ikke er like enkelt å regne kostnad på et prosjekt, der det ofte blir slik at man fortsetter å legge til funksjoner som skal utvikles.

Selv om det finnes noen utfordringer med Scrum så overveier fordelene ulempene kraftig, så lenge man som et team fokuserer på å forbedre seg fortløpende og integrere retningslinjene fra Scrum til å passe Scrum-teamet best mulig så fungerer denne metodikken veldig bra.

5.3.2 Arbeidsstruktur

Dette prosjektet anvender flere teknikker fra scrum-metodikken, disse teknikker er user stories, sprint review, sprint planning og retrospective.

Den første teknikken som skal anvendes er måten man definerer funksjonelle krav i form av user stories. Det denne teknikken sørger for er at vi får en måte å definere funksjonelle krav på som skjuler hvordan kravet skal implementeres, en måte å abstrakt avgjøre arbeidsmengde, og gjør det mulig for gruppen å diskutere hva slags funksjonelle krav som er behov for uten teknisk forvirring. Hver user story blir også gitt et antall

poeng som sier noe om den forventede arbeidsmengden til oppgaven. Dette poengsystemet brukes til tids-estimering. En siste egenskap prosjektet blir å utnytte for å vurdere om kravet er ferdig, er en liste med akseptansekrav som må være oppfylt for at funksjonskravet skal regnes som ferdigstilt.

Den siste dagen av sprinten vil bestå av en sprint- review og planning sammen med oppdragsgiver og veileder. Dette møtet vil bestå av en kort gjennomgang av arbeidet gjort så langt, og en tilbakemelding fra oppdragsgiver på arbeidet. Til slutt vil oppdragsgiver være med på å plukke backlog items fra product backlog innenfor en poenggrense gruppen er komfortabel med.

Etter hvert fullførte sprint review, vil gruppen gjennomføre en sprint retrospective som består av en gjennomgang av alt som har foregått i forrige sprint. Denne gjennomgangen gjennomføres for å finne ut hva som har gått godt i denne sprinten, og hva som kunne vært gjort annerledes. Formålet med denne sprint retrospektives er å sørge for at gruppen til enhver tid forbedrer oss eller fortsetter med å gjøre en god jobb.

5.4 Informasjonsinnsamling – utført og planlagt

Gruppen har møtt med Cordel og fått innsikt i deres interne systemer og behov som er relevante for oppgaven. Videre informasjon som skulle bli nødvendig vil bli anskaffet gjennom oppdragsgiver, veileder eller internett, alt etter behov.

Planlegging av programvare-arkitekturen, er gjort ved å se på kombinasjoner av eksisterende populære biblioteker, i tillegg til å lene seg på tidligere erfaringer, og referere til anbefalinger gitt i dokumentasjonen til rammeverkene som brukes. Det vil også benyttes kilder fra populære kuraterte ressurs-lister.

I tabellen under er det lagt ressurser som er innsamlet gjennom forprosjektet så langt.

Beskrivelse	Link
Samling informasjon om agil utvikling	https://www.atlassian.com/agile
Opplæring C# og ASP.NET Core.	https://www.freecodecamp.org/news/an-awesome-guide-on-how-to-build-restful-apis-with-asp-net-core-87b818123e28/
Hvordan effektivt utføre code reviews	https://blog.digitalocean.com/how-to-conduct-effective-code-reviews/
Stor samling informasjon om React-Redux emner og teknikker	https://github.com/markerikson/react-redux-links
React-Redux dokumentasjon	https://react-redux.js.org/using-react-redux/connect-mapstate
React-router	https://reacttraining.com/react-router/web/guides/quick-start
Opplæring Redux / Thunk	https://youtu.be/93p3LxR9xfM
Opplæring React	https://youtu.be/sBws8MSXN7A
Opplæring React.	https://youtu.be/Ke90Tje7VS0
Diverse dokumentasjon om oppsett og distribusjon av frontend	https://create-react-app.dev/docs/documentation-intro
Backend informasjon	https://github.com/fpereiro/backendlore
React + redux eksempler	https://spapas.github.io/2016/03/02/react-redux-tutorial/

5.5 Vurdering – analyse av risiko

Alle medlemmer i gruppen har gjennom utdanning fått tilstrekkelig erfaring med programvareprosjekter til å kunne realisere prosjektet innenfor tidsrammen. Mulige utfordringer er å jobbe med nye teknologier i alle deler av prosjektet, som medlemmene har begrenset erfaring med. Det er også litt usikkerhet i hvor komplekst det blir å integrere med Cordels interne systemer.

Det vil være viktig at gruppen tidlig i arbeidet setter opp et solid rammeverk for videre utvikling. Dette inkluderer CI løsning og solid programvare-arkitektur.

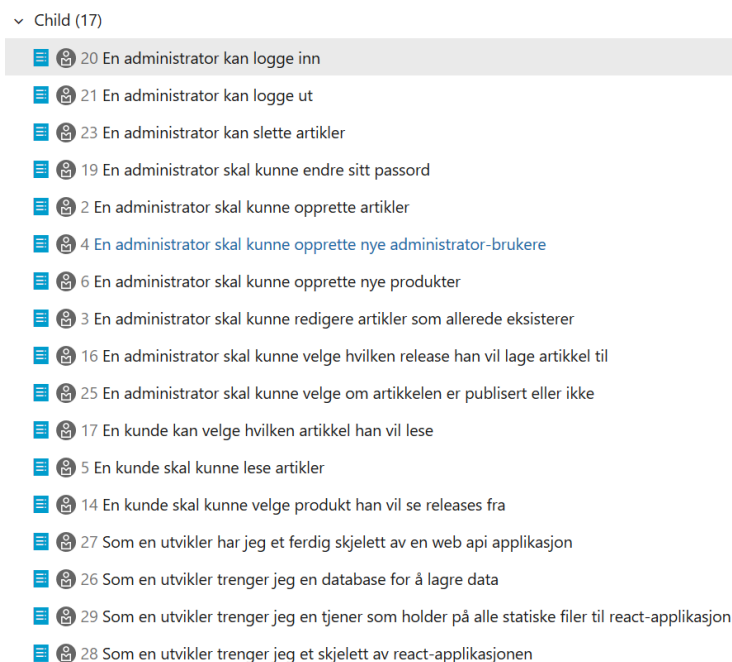
5.6 Hovedaktiviteter i videre arbeid

Prosjektet er delt inn i tre hovedmål, som nevnt i 5.1. Hvert mål har en egen liste med delmål i form av user stories for å beskrive funksjonalitet som må være på plass.

De tre hovedmålene er som følger:

- *MVP*; Et minimum for hva som utgjør et funksjonelt system. Vil være en enkel, mer manuell løsning. Et skjelett for all videreutvikling.
- *Automatisert system*; Integrering med Cordels versjonskontroll-systemer. Automatisk uthenting av release-notes fra Cordel, samt implementering av nye funksjoner som forenkler bruken av systemet.
- *Dypere integrasjon*; Integrasjon med flere av Cordels interne systemer. Hver release note på kundesiden har referanser til support-saker i Cordel.

Til å starte med, er det kun definert user stories for *MVP* målet. Disse er vist i bildet under.



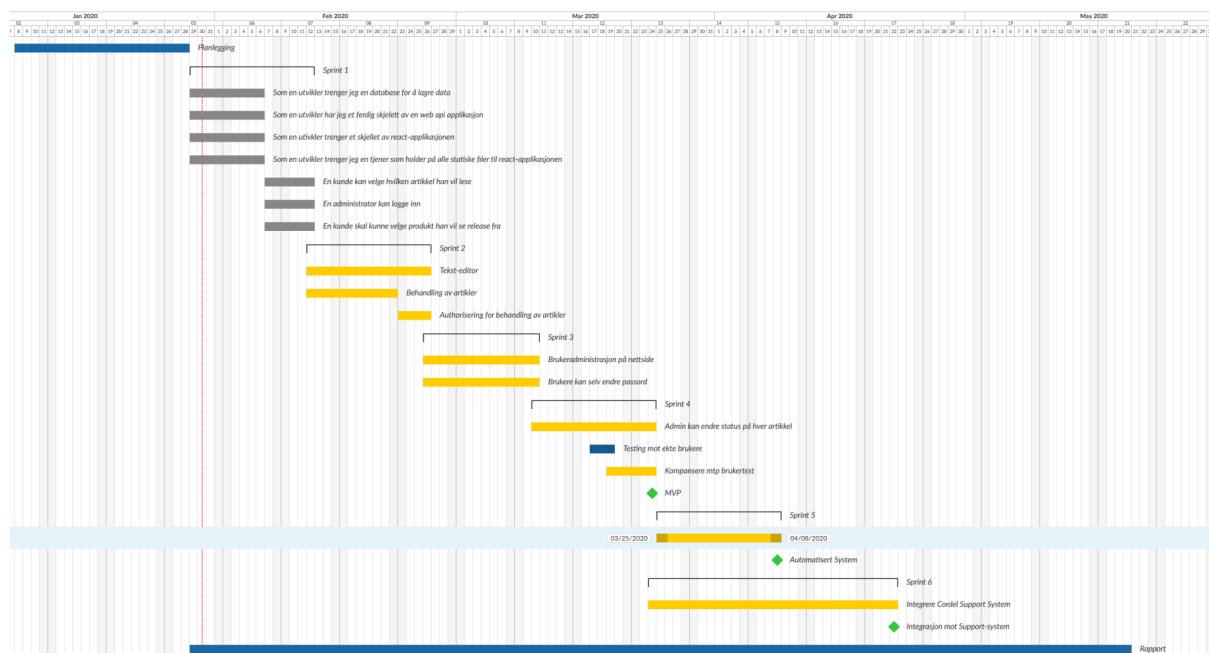
5.7 Framdriftsplan – styring av prosjektet

5.7.1 Hovedplan

Det blir først fokus på å oppfylle MVP. Deretter vil gruppen utrede user stories for ytterlige hovedmål. Selv om gruppen har et bilde av hvordan hvert hovedmål skal utføres, ventes det med utredning av detaljer til et tidspunkt der det er tydeligere hvilke behov som gjenstår å fylles, og hva som bør prioriteres. Det vil også kontinuerlig oppstå nye ideer som må vurderes å inkludere. Slike vurdering vil tas i de nevnte sprint review

og sprint planning møtene. Her er det viktig at gruppen ved hvert møte har et fungerende system ved hvert møte, slik at nye endringer kan demonstreres for oppdragsgiver og drøftes.

Vedlagt finnes det et Gantt-diagram som viser hvordan gruppen tenker å gjennomføre prosjektet. I diagrammet er det gjort estimat for hvordan funksjoner som skal jobbes med i løpet av prosjektet, og når de kan være ferdige. Dette diagrammet er ment for å lage seg en ide om hvordan prosjektet kan gjennomføres, der det strider mot prinsipper i agil-utvikling^{iv}. En større utgave av bildet nedenfor finnes i kapittelet Vedlegg Figur 1 – Plan for gjennomføring.



Det vil legges vekt på å starte hovedrapporten tidlig, og arbeide med den underveis. Det er også satt en absolutt slutt dato for all programmering. Etter denne datoen vil det kun fokuseres på hovedrapporten.

5.7.2 Styringshjelpemidler

For å hjelpe prosjektet med å holde oversikt over alle diagrammer, møtereferat og andre relevante dokumenter, blir Confluence brukt.

Til prosjektstyring blir Microsofts Azure DevOps brukt. Dette inkluderer flere hjelpemidler for å hjelpe med styring av prosjektet:

- Prosjekt-backlog for å se og opprette arbeidsoppgaver. Hver arbeidsoppgave beskrives med et sett akseptansekrav, og gis story points.
- Verktøy for å utføre sprints
- Scrum-tavle, som vil brukes til å organisere arbeidsoppgaver.
- Synkronisering med GitHub

I tillegg vil Gliffy bli brukt til å lage wireframes til å planlegge brukergrensesnitt, samt tekniske diagrammer.

5.7.3 Utviklingshjelpemidler

- *JetBrains Rider*, til å utvikle back-end og database.
- *Visual Studio Code*, til å utvikle front-end

- *Azure DevOps Pipelines*, for å håndtere Continuous Integration og Continuous Development.

5.7.4 Intern kontroll – evaluering

Evaluering blir fortløpende gitt gjennom kundemøter som blir holdt etter hver sprint. I kundemøtene presenterer gruppen fremfor kunden framgangen. Under presentasjonen vil det bli notert mulige endringer på det som er gjort. Forslag tatt opp av kunden vil bli lagt til som oppgaver i backlog, og det vurderes hva skal som utføres frem til neste kundemøte. Om møte blir kansellert vil det bli sendt mail. Mailen vil inneholde skjermbilder av fremgang og eventuelle konkrete spørsmål som har oppstått.

Gruppen vil også ha egen intern vurdering på hvordan de siste to ukene har gått. Denne vurderingen blir gjort i det som kalles en sprint retrospective. Sprint retrospective vil handle om hva som fungerte bra, og hva som ikke fungerte bra de siste to ukene. Hensikten bak dette er å unngå dårlige arbeidsvaner, og forbedre eller danne gode arbeidsvaner.

5.8 Beslutninger – beslutningsprosess

Code review er en gjennomgang der hvert medlem må godkjenne en innføring av endring eller tillegg til kildekode. Design review blir gjennomført på samme måte som en code review, bare at det er innføring eller endring av design som skal vurderes.

Teknologivalget er tatt med hensyn for oppdragsgiver. Når systemet er ferdig overleveres systemet til oppdragsgiver, og derfor er det behov for at det benyttes enkelte teknologier.

Videre beslutninger vil diskuteres innad i gruppen og eventuelt med kunde. På grunn av fleksibiliteten til scrum kan slike beslutninger tas fortløpende, og arbeidet kan tilpasses deretter.

6 DOKUMENTASJON

6.1 *Rapporter og tekniske dokumenter*

- Arbeidstimer; lagres som Excel-dokument i Teams.
- Avtale med firma; skannes og lagres i Confluence.
- Taushetserklæring; skannes og lagres i Confluence.
- Retrospective report; lagres i Confluence.
- Møtereferat; lagres i Confluence.
- Alle typer diagram; lagres i Confluence.
- Arbeidslogg

7 PLANLAGTE MØTER OG RAPPORTER

7.1 *Møter*

De møtene gruppen kommer til å holde gjennom prosjektet er sprint review, sprint planning, retrospective og stand-up. Møtene sprint review, sprint planning og retrospective blir kjørt annenhver uke i sprint-slutt, mens stand-up foregår hver dag.

Sprint-slutt er annenhver onsdag klokken 13:00.

7.2 *Periodiske rapporter*

7.2.1 *Framdriftsrapporter*

Måten fremdrift blir dokumentert i dette prosjektet er gjennom Azure DevOps sitt scrum-tavle, der hvert funksjonelle krav har en status som blir oppdatert fortløpende ved fullført arbeid. Kravene er inndelt i milepæler som kan sees i vedlagt Gantt diagram

7.2.2 *Logg*

Fra og med første sprint, så kommer gruppen til å skrive en liten logg hver fredag, over hva som er gjort den siste uken. Denne er bare for å få en mer detaljrik oppsummering over hva som har skjedd hver uke i løpet av hele prosjektet.

8 PLANLAGT AVVIKSBEHANDLING

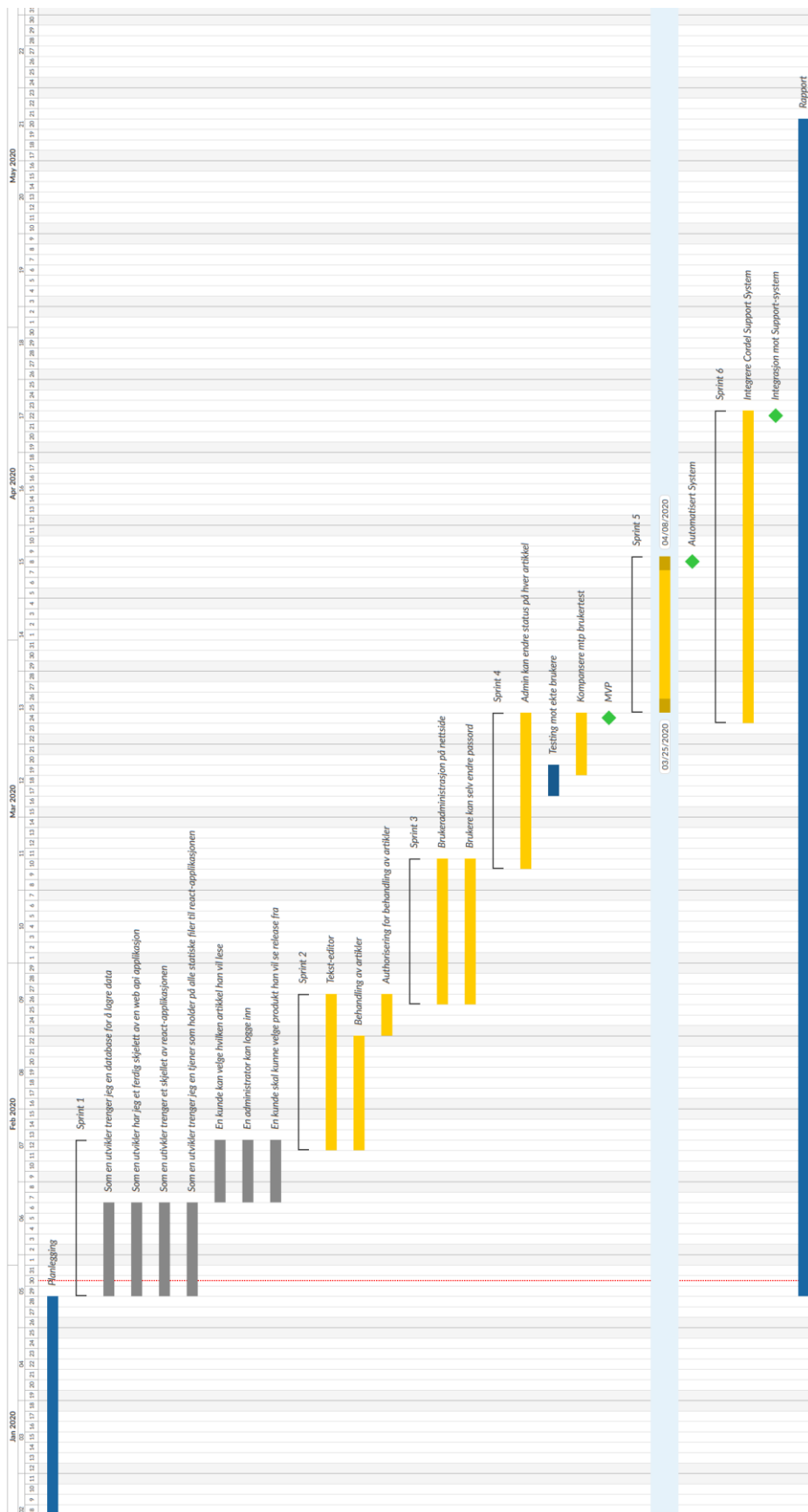
Hvis det skulle oppstå forhold som gjør at det blir lite tid til å bli ferdig med arbeid, som for eksempel om gruppen ser at det ikke er mulig å bli ferdig med en user story innen sprintslutt, så skal gruppen avgjøre om sprinten avsluttes med uferdig arbeid, eller om det må arbeides overtid.

Hvis det oppstår behov for å endre på funksjoner eller teknologivalg underveis i prosjektet skal oppdragsgiver være med i beslutningsprosessen, for å finne den mest egnede løsning. Det er viktig at løsningen gruppen forplikter seg til må være en løsning både oppdragsgiver går med på, samtidig som at den må være mulig for gruppen å utføre.

9 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING

- Tilgang til egen Confluence space hos NTNU i Ålesund.
- AutoDeploy fungerer som den skal.

10 VEDLEGG



Figur 1 – Plan for gjennomføring

11 REFERANSER

-
- ⁱ https://en.wikipedia.org/wiki/Waterfall_model
 - ⁱⁱ <https://www.atlassian.com/agile/kanban>
 - ⁱⁱⁱ <https://www.scrum.org/resources/what-is-scrum>
 - ^{iv} <https://www.scrumexpert.com/knowledge/why-you-should-not-estimate-in-hours-or-days/>