

# FORPROSJEKT - RAPPORT

## FOR BACHELOROPPGAVE

TITTEL:

**Web-basert notearkiv for korps og band**

KANDIDAT(ER) (NAVN):

**Gaute Hjellbakk Pettersen, studentnr. 266738**

**Sindre Sjøholt, studentnr. 130401**

**Thomas Robert Tennøy, studentnr. 130341**

DATO:

**26.01.17**

EMNEKODE:

**IE303612**

EMNE:

**Bacheloroppgave**

DOKUMENT TILGANG:

- Åpen

STUDIUM:

**BACHELOR I INGENIØRFAG - DATA**

ANT SIDER/VEDLEGG:

**20 / 0**

BIBL. NR:

- Ikke i bruk -

OPPDRAUGSGIVER(E)/VEILEDER(E):

**Arne Styve**

OPPGAVE/SAMMENDRAG:

Oppgaven for dette bachelorprosjektet, er å utvikle et web-basert system for håndtering av noter. Premissene er at dette systemet må kunne fungere på alle vanlige plattformer, inklusivt smarttelefoner og nettbrett. Dette innebærer at webapplikasjonen skal være responsiv (RWD). I tillegg til dette, skal systemet ha god tilgjengelighet gjennom å innfri kravene for universell utforming.

I denne forprosjektrapporten har vi som mål å analysere og svare på spørsmål som må være på plass før prosjektet kan starte opp. Dette går på hva som er målene for prosjektet, hvordan det skal utvikles og hvordan det skal fungere.

Vi går også inn på hvilken teknologi som skal benyttes under utviklingen av prosjektet, samt at vi gjør en liten risikoanalyse over faktorer som kan true gjennomføringen av prosjektet.

I tillegg til dette er det satt opp en foreløpig plan over prosjektets fremgang, med milepæler og estimert tidsforbruk.

# INNHold

<b>INNHold .....</b>	<b>2</b>
<b>1 INNLEDNING .....</b>	<b>3</b>
<b>2 BEGREPER .....</b>	<b>3</b>
<b>3 PROSJEKTORGANISASJON.....</b>	<b>5</b>
3.1 PROSJEKTGRUPPE .....	5
3.1.1 Oppgaver for prosjektgruppen - organisering.....	6
3.1.2 Oppgaver for Utviklingsteamet.....	6
3.1.3 Oppgaver for produkteier .....	6
3.1.4 Oppgaver for sekretær .....	7
3.1.5 Oppgaver for Scrum Master .....	7
3.2 STYRINGSGRUPPE (VEILEDER OG KONTAKTPERSON OPPDRAGSGIVER).....	7
<b>4 AVTALER.....</b>	<b>7</b>
4.1 AVTALE MED OPPDRAGSGIVER .....	7
4.2 ARBEIDSSTED OG RESSURSER .....	7
4.3 GRUPPENORMER – SAMARBEIDSREGLER – HOLDNINGER.....	8
<b>5 PROSJEKTBESKRIVELSE .....</b>	<b>9</b>
5.1 PROBLEMSTILLING - MÅLSETTING - HENSIKT.....	9
5.2 KRAV TIL LØSNING ELLER PROSJEKTRESULTAT – SPESIFIKASJON.....	10
5.3 PLANLAGT FRAMGANGSMÅTE(R) FOR UTVIKLINGSARBEIDET – METODE(R).....	11
5.4 INFORMASJONSINNSAMLING – UTFØRT OG PLANLAGT.....	12
5.5 VURDERING – ANALYSE AV RISIKO.....	13
5.6 HOVEDAKTIVITETER I VIDERE ARBEID.....	14
5.7 FRAMDRIFTSPLAN – STYRING AV PROSJEKTET.....	15
5.7.1 Hovedplan.....	15
5.7.2 Styringshjelpemidler .....	15
5.7.3 Utviklingshjelpemidler.....	15
5.7.4 Intern kontroll – evaluering.....	16
5.8 BESLUTNINGER – BESLUTNINGSPROSESS .....	16
<b>6 DOKUMENTASJON .....</b>	<b>17</b>
6.1 RAPPORTER OG TEKNISKE DOKUMENTER .....	17
<b>7 PLANLAGTE MØTER OG RAPPORTER .....</b>	<b>18</b>
7.1 MØTER.....	18
7.1.1 Møter med styringsgruppen .....	18
7.1.2 Prosjektmøter .....	18
7.2 PERIODISKE RAPPORTER .....	19
7.2.1 Framdriftsrapporter (inkl. milepæl) .....	19
<b>8 PLANLAGT AVVIKSBEHANDLING .....</b>	<b>19</b>
<b>9 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING .....</b>	<b>19</b>
<b>10 REFERANSER .....</b>	<b>20</b>

## 1 INNLEDNING

Ved valg av oppgave, hadde vi flere kriterier vi ville ha oppfylt. Dette gikk blant annet på at vi ville ha en oppgave vi kunne se på som gjennomførbar ut i fra de kunnskapene vi har opparbeidet oss gjennom studieløpet så langt, i tillegg til at vi måtte ha et prosjekt som vi anså som interessant, lærerikt og utfordrende.

Gjennom studieløpet har vi vært innom mange forskjellige fagområder. Det var viktig for oss å kunne bruke det vi har lært hittil, og derfor velge et prosjekt som inneholdt elementer fra mange fagområder. Vi har valgt å lage et web-basert notearkiv som vårt bachelorprosjekt, fordi oppgaven virker spennende og utfordrende, i tillegg til at denne oppgaven gir oss muligheten til å benytte opparbeidet kunnskap om webutvikling, databaseutvikling og håndtering, utvikling av servertjeneste for webapplikasjon, samt generell prosjektutvikling og prosjektstyring.

I dag blir samfunnet mer og mer digitalisert. Dette er som regel for å gjøre ting enklere for brukerne, og et web-basert notearkiv vil forenkle driften av et korps eller storband. Håndtering av noter kan være en stor utfordring for ethvert korps eller storband. Noter blir ofte håndtert ved å bruke noteutskrifters som deles ut til utøverne, for så å bli samlet inn igjen når de ikke skal brukes lenger. Stadig flere korps og band begynner nå å håndtere sine notearkiv elektronisk (PDF-dokumenter eller lignende), og organisert i en felles filstruktur. Dette kan gjøres mye mer effektivt og brukervennlig både for vedkommende som har ansvar for arkivet, og utøverne som skal hente ut sine noter.

I denne bacheloroppgaven skal det utvikles et web-basert system for håndtering av noter. Systemet skal kunne fungere på alle vanlige plattformer, inklusivt smarttelefoner og nettbrett. I tillegg skal systemet bygges opp med god tilgjengelighet, såkalt universell utforming.

Den grunnleggende problemstillingen for denne oppgaven, blir å besvare følgende spørsmål: Hvordan utvikle et web-basert notearkiv med tilhørende databaseløsning og serverløsning, på en måte som gjør tjenesten til et sikkert, effektivt og brukervennlig system tilpasset alle vanlige brukerplattformer, og som innfrir kravene for universell utforming?

## 2 BEGREPER

Her følger en kort forklaring og definering av noen sentrale uttrykk og begreper som er brukt i denne rapporten.

- **Web**  
Forkortet versjon av World Wide Web, verdensveven/veven. Et globalt informasjonsrom som utgjør tekstdokumenter, bilder, multimedia og mange andre typer informasjon tilgjengelig over Internett. [1]
- **Database**  
En database er strukturert samling av relaterte data, begrepet er vanligvis knyttet til elektroniske datasystemer. Et databasesystem skal kunne behandle store mengder data effektivt, motsetningsfritt og sikkert. Databasesystemet skal sikre at endringer og tilføyelser ikke fører til motsetninger eller feil, og skal kunne levere riktig informasjon effektivt til brukere og programvare. [2]
- **Server**  
En server, også kjent som en tjener, er en programvare som tilbyr («serverer») en eller flere tjenester til andre datamaskiner (klienter) over et datanettverk. Begrepet *tjener/server* er også ofte brukt om maskinvaren som programmet (eller programmene) kjøres fra. For eksempel kan én og samme datamaskin tilby flere tjenester samtidig, så fremt maskinen har kapasitet til å utføre alle oppgavene. [3]

- **Webapplikasjon**  
En webapplikasjon er et program som kan kjøres i en nettleser. Det er programmert i et språk som nettlesere støtter, som for eksempel HTML, JavaScript og/eller CSS. [4]
- **PDF**  
Portable Document Format (PDF) er et digitalt dokumentformat som er utviklet av Adobe Systems og introdusert i 1993. Filene vises på skjerm i samme form som de har når de blir skrevet ut. Det kan betraktes som en form for digitalt papir. [5]
- **Smarttelefon**  
Smarttelefon er mobiltelefoner som drives av et operativsystem som lar brukeren installere og kjøre avansert programvare. I tillegg til å kunne ringe og motta samtaler har de en kombinasjon av en eller flere funksjoner som avspilling av musikk, foto, video, internettleser, kart og GPS. [6]
- **Nettbrett**  
Nettbrett (engelsk: tablet eller tablet computer), er en komplett datamaskin bestående kun av en flat berøringsskjerm (touch screen) hvor pekepenn, digital penn eller fingertuppen brukes til å styre/jobbe med enheten i stedet for et tastatur eller mus. [7]
- **Universell utforming**  
Med universell utforming menes utforming eller tilrettelegging av hovedløsningen i de fysiske forholdene, herunder informasjons- og kommunikasjonsteknologi (IKT), slik at virksomhetens alminnelige funksjon kan benyttes av flest mulig. [8]
- **Scrum**  
Scrum er en smidig utviklingsmetode, laget med tanke på å utvikle komplekse systemer. Brukes i hovedsak til å utvikle programvarebaserte systemer. Scrum går ut på at utviklingen av systemet foregår gjennom at man jobber iterativt og inkrementelt, og at jobben utføres av tverrfaglige, selvstyrte team. Hvert team jobber mot sine mål, ved å utvikle konkrete, små produktinkremer over tidsperioder på 1-4 uker (vanligvis). En slik periodene kalles Sprint.
- **Product Backlog**  
Product Backlog, eller produktkø, er en liste over alle funksjoner, tjenester og endringer sluttproduktet kan ha. Denne listen kan inneholde elementer som en bør ha med, er ønskelig å ha med, og elementer som en kunne hatt med, men som ikke er nødvendig. Derfor foreligger denne listen i prioritert rekkefølge, men er grov-estimert. Listen er dynamisk, så elementer kan legges til og fjernes så lenge Backloggen eksisterer, og det er så lenge produktet selv eksisterer. En vesentlig ting med elementene som ligger i Backlog, er at de kan implementeres selvstendig, og ingen er avhengig av noen andre. Derfor kan man avvike fra den prioriterte rekkefølgen, og strengt tatt velge ut de funksjonene man til en hver tid vil implementere.
- **Sprint Backlog**  
Produktkøen med alle oppgavene som utviklingsteamet skal utvikle og implementere i gjeldende Sprint.
- **REST – Representational State Transfer**  
Representational state transfer (REST) eller RESTful webtjenester er en måte å gi inter-operabilitet mellom datasystemer på internett. REST-kompatible webtjenester gir systemer tilgang til, og muligheten til å manipulere, tekstbaserte representasjoner av nettressurser ved hjelp av en enhetlig og et forhåndsdefinert sett av tilstandsløse operasjoner. [10]
- **Git**  
Git er et versjonskontrollsystem, en type programvare. Det kjennetegnes ved å være distribuert og ikke mappehierarki-basert. [11]

**Diverse forkortelser brukt i denne rapporten:**

- **API**  
Application Programming Interface
- **CSS**  
Cascading Style Sheet
- **GPS**  
Global Positioning System
- **HTML**  
HyperText Markup Language
- **IDE**  
Integrated Development Environment
- **IKT**  
Informasjons- og kommunikasjonsteknologi
- **JDBC**  
Java Database Connectivity
- **RWD**  
Responsive Web Design
- **SQL**  
Structured Query Language

### 3 PROSJEKTORGANISASJON

I denne delen av rapporten presenterer vi hvordan prosjektorganisasjonen er bygget opp. Hvert medlem i prosjektgruppen vil ha forskjellig roller, med spesifikke arbeidsoppgaver og ansvarsområder. I tillegg til disse medlemmene, har prosjektet en oppdragsgiver og veileder, som blir presentert senere i dette kapittelet.

#### 3.1 Prosjektgruppe

Studentnummer	Navn
266738	Gaute Hjellbakk Pettersen
130301	Sindre Sjøholt
130341	Thomas Robert Tennøy

*Tabell 1: Oversikt over medlemmene i prosjektgruppen, med navn og studentnummer*

Nevnte studenter er alle med i prosjektgruppen, og vedlagte studentnummer gjelder for NTNU i Ålesund. Denne prosjektgruppen er ansvarlig både for å levere denne forprosjektrapporten for kurset **IF300114 Ingeniørfaglig systemteknikk og systemutvikling**, samt gjennomføring av **IE303612 Bacheloroppgave**.

### 3.1.1 Oppgaver for prosjektgruppen - organisering

I dette prosjektet vil utviklingsmetoden Scrum bli benyttet. Denne metoden er beskrevet i punkt 5.3.

Noe av det viktigste med Scrum, er hvordan rollefordelingen er. Hele konseptet består av 3 typer roller; Produkteier, Utviklingsteamet og Scrum Master.

Siden vi i prosjektgruppen kun er 3 personer, og alle inngår i utviklingsteamet, må vi dessverre også til tider ha dobbeltrolle. Det skal ikke by på noen problemer, slik vi anser det.

Organiseringen for prosjektet blir derfor som følger:

Rolle	Navn
Produkteier	Arne Styve
Utviklingsteam	Sindre Sjøholt
Utviklingsteam	Thomas Robert Tennøy
Utviklingsteam	Gaute Hjelbakk Pettersen
Scrum Master*	Thomas Robert Tennøy
Sekretær**	Sindre Sjøholt

Tabell 2: Oversikt over rollene i prosjektet

\* Scrum Master er som regel en egen rolle. Det er ikke nok folk knyttet til dette prosjektet til at vi kan oppfylle det kriteriet. Derfor må vi benytte oss av dobbeltrolle i dette tilfellet. Vi mener det ikke skal by på noen problemer, og at vedkommende skal kunne klare å skille på sine roller.

\*\* Selv om det Agile Manifesto [13] sier at det er viktigere med programvare som virker, fremfor omfattende dokumentasjon, så vil det uansett være behov for noe dokumentasjonsarbeid gjennom prosjektets livsløp. Derfor har vi opprettet en rolle som har det overordnede ansvar for dette, som beskrevet i punkt 3.1.3.

### 3.1.2 Oppgaver for Utviklingsteamet

- Ansvar for selve utviklingen av produktet
- Forsøke å levere et "ferdig" produktinkrement for demonstrasjon i hver sprint
- Planlegge og estimere hvilke elementer fra Backlog som skal utvikles og implementeres i neste sprint

### 3.1.3 Oppgaver for produkteier

- Skal sikre at Scrum-teamet til en hver tid jobber med de rette tingene sett fra et forretningsperspektiv
- Ansvarlig for at Product Backlog til en hver tid er estimert, prioritert og tilgjengelig for utviklingsteamet
- Ansvar for å omrokere og organisere Backlog

- Skal fokusere på "hva", ikke "hvordan"
- Beslutningsansvar for prioriteringer
- Kan være en del av Scrum-teamet, men kan aldri være Scrum Master
- Fungerer **ikke** som prosjektleder

### 3.1.4 Oppgaver for sekretær

- Overordnet ansvar for dokumentasjon, både interne dokumenter for prosjektgruppen og dokumenter ment for styringsgruppe og/eller universitet
- Ansvar for logging av prosjektarbeid og fremskritt
- Skrive møtereferat og annen dokumentasjon fra møter mellom styringsgruppen og prosjektgruppen
- Kvalitetssikring av all skriftlig dokumentasjon
- Sekretæren deltar ellers i planlegging og utvikling på samme nivå som resten av prosjektgruppen

### 3.1.5 Oppgaver for Scrum Master

- Ansvarlig for det smidige rammeverket Scrum, og at dette rammeverket blir brukt i henhold til dets standarder
- Ansvarlig for at utviklingsteamet lærer seg å selvorganisere
- Ansvarlig for at man hele tiden forbedrer arbeidsformen og at refleksjonsmøtene fører til forbedring
- Ansvarlig for å sikre at Scrum-teamet fungerer som planlagt, for å møte den nødvendige mengden av arbeid som er beskrevet i sprinten
- Scrum Master deltar ellers i planlegging og utvikling på samme nivå som resten av prosjektgruppen

## 3.2 *Styringsgruppe (veileder og kontaktperson oppdragsgiver)*

### Oppdragsgiver og veileder \*:

Arne Styve, universitetslektor hos NTNU i Ålesund  
Mobil: 464 74 059  
E-post: [asty@ntnu.no](mailto:asty@ntnu.no)

\* Arne Styve vil i dette prosjektet påta seg en dobbeltrolle som både oppdragsgiver og veileder. Vi forventer ikke at dette vil skape noen form for problemer. I spesifikke tilfeller vil andre personer fra fagmiljøet på NTNU i Ålesund kunne bistå med veiledningen i prosjektet, men dette er ikke planlagt per nå.

## 4 AVTALER

### 4.1 *Avtale med oppdragsgiver*

Det foreligger per i dag ingen signerte avtaler med oppdragsgiver.  
Dette blir eventuelt gjennomført på et senere tidspunkt.

### 4.2 *Arbeidssted og ressurser*

Prosjektgruppen sitt arbeidssted vil i hovedsak være i NTNU sine lokaler i Larsgårdsvegen 2, Ålesund. Rom L167 i lab-bygget er avsatt til arbeid med bacheloroppgavene hele

vårsemesteret 2017, og dette vil være det primære arbeidsted. Medlemmene i prosjektgruppen kan også benytte seg av hjemmekontor, dersom dette er planlagt og avtalt på forhånd.

I tillegg til dette vil forelesere og andre eksperter være tilgjengelig i NTNU sine lokaler, for eventuelle konsultasjoner.

Når det kommer til ressurser prosjektgruppen vil benytte seg av under utviklingen av prosjektet, er alle medlemmene i gruppen utstyrte med private bærbare datamaskiner. Disse vil være det primære verktøyet når det kommer til maskinvare. I tillegg til dette, vil NTNU ha tilgjengelig dedikerte virtuelle servere for databasehåndtering, serverløsning og webapplikasjon.

Det vil bli brukt flere forskjellige utviklingsmiljøer i dette prosjektet, da det inneholder flere forskjellige deler (som database, server og webapplikasjon). En foreløpig liste over program- og maskinvare som er planlagt brukt gjennom dette prosjektet finnes i punkt 5.7.2 og 5.7.3.

Når det kommer økonomiske ressurser, så vil det primært bli brukt maskinvare som allerede er innkjøpt, samt programvare som enten er gratis tilgjengelig, eller tilgjengelig gjennom NTNU sine biblioteker og tjenester. Vi vil likevel måtte gå til innkjøp av programvaren Bootstrap Studio 2.5. Dette har en kostnadsramme på \$50 (NOK 415,73 per 26.01.17), og vil gi muligheten til å benytte denne programvaren på inntil 6 maskiner, og ha tilgang på programvareoppdateringer i en periode på 12 måneder.

Møter mellom prosjektgruppen og styringsgruppen vil bli forsøkt avholdt annenhver uke. Disse møtene vil holde sted i NTNU sine lokaler.

### **4.3 Gruppenormer – samarbeidsregler – holdninger**

Gruppenormer og samarbeidsregler som vil være fundamentet i prosjektarbeidet inkluderer følgende punkter:

- Møter i prosjektgruppen skal begynne, og slutte, på avtalt klokkeslett
- Møter mellom prosjektgruppen og styringsgruppen skal begynne, og slutte, på avtalt klokkeslett
- Alle medlemmene i prosjektgruppen skal kunne delta i bestemmelser og avgjørelser relatert til prosjektet
- Det skal oppfordres til samarbeid og åpenhet i løpet av prosjektets levetid
- En positiv holdning til prosjektet og håndheving av god oppførsel skal utgjøre grunnlaget for arbeidet
- Det skal være stort fokus på informasjonssikkerhet og sikker drift av systemet i sin helhet, i samsvar med gjeldende lover, regler og normer
- Fravær fra avtalt arbeidstid skal meldes inn i så god tid i forveien som mulig, slik at det kan håndteres på best mulig måte av prosjektgruppen i sin helhet

Holdninger til prosjektet og ingeniørfaget som profesjon:

- Vi skal vise respekt for kolleger og samarbeidspartnere uavhengig av kjønn, livssituasjon, kultur og etnisk opprinnelse. Dette innebærer å:
  - vise respekt for menneskets iboende verdighet
  - respektere tradisjoner og kulturelle verdier
  - ha respekt for andres fag- og ansvarsområde og gi anerkjennelse for deres arbeid
  - bidra til et godt arbeidsmiljø gjennom åpenhet, ærlighet og utvist toleranse
- Vi skal vedkjenne vårt faglige ansvar og utføre vårt arbeid i henhold til anerkjente kvalitetsnormer. Dette innebærer å:
  - ajourføre og videreutvikle egen kompetanse for å sikre kvaliteten i profesjonsutøvelsen
  - utføre pålagte oppgaver på en faglig forsvarlig måte



- dele sin kunnskap med kolleger slik at resultatet blir best mulig
- gi anbefalinger og velge faglige løsninger med fokus på kundens behov
- være årvåken overfor de etiske sidene ved arbeidet som utføres
- opptre lojalt overfor arbeidsgiver så lenge det samsvarer med våre etiske retningslinjer
- avklare forhold som kan påvirke egen habilitet, med dem det kan ha betydning for
- unngå forhold som kan påvirke egen objektivitet i relasjon til kunder og leverandører
- Vi skal vise respekt for samspillet mellom teknologi og menneskelige verdier og bidra til åpenhet om konsekvenser av teknologiske løsninger for miljø og enkeltindivid. Dette innebærer å:
  - arbeide for å utvikle tekniske løsninger som vektlegger forholdet til natur, miljø og enkeltindivid
  - bidra til teknologiske løsninger som styrker utviklingen og sikrer våre felles ressurser
  - synliggjøre miljømessige konsekvenser av valgte teknologiske løsninger
  - bidra med egen kompetanse i samfunnsdebatten i spørsmål knyttet til naturvitenskap og teknologi  
(De ovennevnte punktene er inspirert av NITO (Norges ingeniør og teknologorganisasjon) sine etiske retningslinjer [9])
- Vi skal være ærlig om egen kompetanse, og ikke påta oss oppgaver vi ikke er kvalifisert til
- Vi skal ikke overta en kollega sine oppgaver uten samtykke fra vedkommende

## 5 PROSJEKTBEKRIVELSE

### 5.1 Problemstilling - målsetting - hensikt

Den grunnleggende problemstillingen for denne oppgaven, blir å besvare følgende spørsmål: Hvordan utvikle et web-basert notearkiv med tilhørende databaseløsning og serverløsning, på en måte som gjør tjenesten til et sikkert, effektivt og brukervennlig system tilpasset alle vanlige brukerplattformer, og som innfrir kravene for universell utforming?

Hensikten og målsettingen er derfor å kunne utvikle et komplett system som innfrir alle de implisitte krav som ligger i problemstillingen. Dette innebærer å designe og konstruere en komplett og funksjonell databaseløsning som inneholder all den informasjon vi måtte trenge for å utføre prosjektet videre, det innebærer å designe og utvikle en REST-basert serverløsning som skal fungere som mellomledd for database og webapplikasjon.

Hovedmålet er å kunne utvikle, og levere, en løsning for digitalisering og håndtering av noter til utøvere i korps og/eller band. Å levere en løsning som er bygget opp slik at den fungerer på en sikker og effektiv måte, og at brukervennligheten er i høysetet. All funksjonalitet som er ønsket og foreslått av oppdragsgiver skal være implementert, og skal fungere på en hensiktsmessig og tilfredsstillende måte.

Det komplette systemet, også beskrevet som prosjektet, er i praksis delt i 3 deler; databasen, serveren og webapplikasjonen. Disse tre delene avhenger av hverandre, og systemet vil ikke fungere som ønsket før alt er på plass. Derfor vil det å få alle disse tre komponentene av systemet til å fungere både hver for seg, samt som del av det komplette systemet, være resultatmålet.

Effekt målet for prosjektet er å skape en enklere situasjon for brukerne av systemet, i forhold til sortering, lagring og håndtering av noter. Om systemet gir økt effektivitet og brukervennlighet, er effekt målet oppnådd.

## 5.2 Krav til løsning eller prosjektresultat – spesifikasjon

Kravet til løsningen vår kan ses fra to forskjellige perspektiver. Løsningen skal fungere på en tilfredsstillende måte for en bruker av systemet, i henhold til de krav og spesifikasjoner som er satt på forhånd. Dette innebærer at all funksjonalitet og krav om brukervennlighet skal være på plass, og krav om responsiv design og universell utforming skal være innfridd.

En bruker av systemet skal kunne gjøre avanserte søk i notearkivet, og finne noter basert på søk med flere kriterier, og åpne eller laste ned notene for ønsket stemme eller instrument. I tillegg til dette, så skal det være en funksjon for å organisere settlister til et spesifikt arrangement, for eksempel en konsert. Administrator kan legge til låter/melodier/sanger i en spesifikk settliste, og gi brukerne mulighet til å hente ut notene for hele settlisten i en samlet PDF. Gerne basert på ønsket stemme eller instrument.

Det andre perspektivet er fra administrators side. Og det er den viktigste siden. Det skal utvikles et system som gjør jobben med organisering og håndtering av noter enklere enn dagens situasjon. Også her skal alle krav og spesifikasjoner innfris. Administrator må kunne ha full oversikt over registrerte brukere i systemet, og kunne håndtere disse. Den viktigste funksjonaliteten administrator skal ha, er selve håndteringen av notene (PDF-dokumentene). Det skal implementeres en solid funksjonalitet for sortering og organisering av notene, basert på mange kriterier (artist, komponist, tittel, arrangør, vanskelighetsgrad, varighet, arkivnummer, kommentar osv.). I tillegg skal en kunne dele opp PDF-filene, for eksempel en for hver stemme eller instrument.

Som nevnt i et tidligere kapittel i denne rapporten, så kan systemet som skal utvikles i dette prosjektet deles i tre. De underliggende delsystemene er databasen, den tilstandsløse serverløsningen og webapplikasjonen.

Uten å foreløpig ha spikret alt når det kommer til teknologi og fremgangsmåter, kan vi antyde at dette vil bli utviklet på følgende måte:

### Database

- Utarbeides i Apache Derby [12]
- Derby er et underprosjekt fra Apache Software Foundation, og er en relasjonsdatabase implementert i Java
- Derby har et lite fotavtrykk; databasemotoren og integrert JDBC-driver er på tilsammen ca. 2 megabyte
- Derby er bygd på Java-, JDBC- og SQL-standardene
- Derby tilbyr en integrert («embedded») JDBC-driver som lar deg integrere Derby i en hvilken som helst Java-applikasjon
- Derby er en integrert tjeneste i applikasjonsserveren Payara (Glassfish)

### Server

- Det skal skrives en REST API for håndtering av databasespøringer
- REST står for Representational State Transfer, og fungerer tilstandsløs i kommunikasjon mellom server og klient, som regel gjennom HTTP-protokollen
- Skal fungere som link mellom database og webapplikasjon
- Skal foreta spørringer mot databasen, og levere resultater på disse til webapplikasjonen
- Serverløsning skal være skrevet i JAX-RS, som er et Java API som tilstandsløse webtjenester
- Server skal utvikles gjennom Payara Server, som er en applikasjonsserver i åpen kildekode som nedstammer fra Glassfish Server
- Payara Server fungerer som applikasjonsserver for utvikling av Java Enterprise Edition applikasjoner
- Payara integreres i utviklingsmiljøet Netbeans IDE

### Webapplikasjon

- Webapplikasjon skal utvikles i henhold til forskrift om universell utforming av IKT-løsninger. Det vil si at nettsider må oppfylle 35 av 61 suksesskriterier i standarden Retningslinjer for tilgjengelig webinnhold (WCAG 2.0)

- Applikasjonen skal være tilgjengelig for alle vanlige plattformer, det vil si at den må tilpasses for bruk på smarttelefoner og nettbrett, i tillegg til vanlige personlige datamaskiner. Med andre ord skal den kunne brukes på de fleste skjermstørrelser.
- Applikasjonen skal kunne fungere i alle populære nettlesere, som Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, Microsoft Internet Explorer og Opera
- Applikasjonen skal utvikles ved hjelp av HTML5 (HyperText Markup Language er et markeringsspråk for formatering av nettsider med hypertekst og annen informasjon som kan vises i en nettleser)
- CSS – Cascading Style Sheets – vil bli brukt til å definere utseende på filer skrevet i HTML. Oppsett, farger og annen stilinformasjon skal beskrives ved hjelp av eksterne og linkede CSS-filer
- Det vil bli brukt JavaScript for utvikling av funksjonalitet
- Webapplikasjonen skal utvikles ved hjelp av et spesifikt rammeverk. Her har valgt vårt falt på front-end rammeverket Bootstrap [14]:
  - Gratis og åpen kildekode
  - Inneholder HTML- og CSS-baserte maler for typografi, skjema, knapper, navigasjon og andre elementer i grensesnittet
  - Støtter utvikling av responsiv webdesign
  - Har utvidelser for JavaScript
  - Konsentrerer seg kun om front-end utvikling
  - Originalt kalt Blueprint, og er utviklet av Twitter
  - Kompatibel med alle populære nettlesere
  - Kan benyttes sammen med en kompatibel WYSIWYG-editor (**What You See Is What You Get**), som for eksempel Bootstrap Studio 2.5

### **5.3 Planlagt framgangsmåte(r) for utviklingsarbeidet – metode(r)**

Vi har for dette prosjektet valgt å bruke utviklingsmetoden Scrum. Scrum er en såkalt smidig (agile) utviklingsmetode, laget med tanke på å utvikle komplekse systemer. Brukes i hovedsak til å utvikle programvarebaserte systemer.

Scrum går ut på at utviklingen av systemet foregår gjennom at man jobber iterativt og inkrementalt, og at jobben utføres av tverrfaglige, selvstyrte team.

Hvert team jobber mot sine mål, ved å utvikle konkrete, små produktinkremer over tidsperioder på 1-4 uker (vanligvis). Disse tidsperiodene kalles Sprints. Vi har bestemt at våre sprints skal vare i 2 uker.

Bare en sprint blir planlagt i gangen. Denne planleggingen er det produkteieren og Scrum-teamet som står for. Her tas elementer fra Product Backlog og legges inn i Sprint Backlog. Det er det selvstyrte teamet som bestemmer hvor mange elementer som skal inn i Sprinten, men teamet kan ikke forandre prioriteringene som er satt av produkteier. Deretter blir oppgavene delt ned i mindre arbeidsoppgaver, og estimert i forhold til ressursbruk (tid, som regel).

I slutten av hver Sprint, holdes et såkalt Sprint Review (Sprintrefleksjon), der en fungerende/kjørende versjon av arbeidet blir demonstrert for produkteieren og oppdragsgiveren. Produkteieren vil da gi en tilbakemelding på det som er gjort, på det som var planlagt men ikke ble gjennomført og den generelle statusen for prosjektet. Oppdragsgiveren vil også gi tilbakemelding på om arbeidet godkjennes eller skrotes. Dette er også møtet hvor oppdragsgiveren kan komme med nye ideer, eller tanker rundt den neste sprinten.

Det siste møtet i Sprinten vil være et Sprint Retrospective Meeting, her ser vi oss tilbake. Vi gir feedback til hverandre, slik at vi kan forbedre prosessen. Hva gikk bra i denne sprinten, hva kan bli bedre? Dette skal ikke være en rapport til Daglig leder, men en samtale hvor målet er å forbedre prosessen samt skape et eierforhold til selve utviklingsprosessen.

**Personer og samspill** fremfor prosesser og verktøy  
**Programvare som virker** fremfor omfattende dokumentasjon  
**Samarbeid med kunden** fremfor kontraktsforhandlinger  
**Å reagere på endringer** fremfor å følge en plan

Tabell 3: Det Agile Manifestet [13]

Scrum anses som en fornuftig metode å bruke om det forventes å kunne dukke opp endringer i prosjektet, for eksempel i krav eller spesifikasjoner. Ved andre metodikker kan endringer skape problemer, på grunn av en mer låst måte å håndtere prosjektet og dets oppbygging.



Figur 1: Grafisk oversikt over hvordan Scrum fungerer i praksis. (Kilde: scrumalliance.org)

Utviklingsteamet vil ha daglige møter kalt Daily Scrum, som skal vare maksimalt 15 minutter. Hvert medlem i det selvstendige Scrum-teamet skal redegjøre for hva medlemmet gjorde i går, hva medlemmet skal gjøre i dag og eventuelle problemstillinger medlemmet har møtt på eller kan møte på i arbeidet. Disse møtene er satt opp hver dag for at den selvstyrte gruppen skal ha en god og konstruktiv kommunikasjon gjennom hele sprinten.

Vår oppdragsgiver og veileder, Arne Styve, vil figurere som Product Owner (produkteier) i dette prosjektet.

## 5.4 Informasjonsinnsamling – utført og planlagt

Vi vil under utarbeidelse av dette prosjektet trenge å samle inn en hel del med informasjon og kunnskap om flere forskjellige tema. Noe av det vi sannsynligvis vil bruke, er:

- Dokumentasjon om Bootstrap front-end webrammeverk
- Dokumentasjon om Apache Derby database
- Dokumentasjon om alle planlagt brukte programmeringsspråk, som Java, HTML5, SQL, CSS
- Dokumentasjon om Java Enterprise Edition og Java API for RESTful Web Services
- Dokumentasjon om utviklingsverktøyer som Netbeans IDE, Payara Server, Bootstrap Studio etc.

- Eventuelle lignende og konkurrerende tjenester/produkter vil bli analysert
- Informasjons og data til testing, i hovedsak noter, vil bli hentet inn fra oppdragsgiver Arne Styve
- Informasjon, tips og hjelp vil bli hentet inn fra veileder hos NTNU, Arne Styve
- Eventuelt vil også andre eksperter hos NTNU bli kontaktet dersom det skulle bli behov for informasjon om spesifikke tema

## 5.5 Vurdering – analyse av risiko

Med risiko menes sannsynligheten for, og konsekvensen av, at noe uønsket skal skje eller utvikle seg. I dette avsnittet skal vi gjennomgå en risikoanalyse, før arbeidet på prosjektet starter for alvor. Dette gjør vi for å kunne finne eventuelle risikoelementer i prosjektet vårt, og iverksette passende tiltak for å øke kontrollen over forhold som må ivaretas under gjennomføringen av prosessen.

Etter en diskusjon internt i prosjektgruppen, ser vi på det som sannsynlig at det er mulig å realisere prosjektet innenfor de rammene som er gitt, dersom kravene nevnt under blir møtt:

- Effektiv og konstruktiv kommunikasjon innad i prosjektgruppen, og mellom prosjektgruppen og styringsgruppen
- At forprosjektering og vurderinger underveis er gjort på en god måte
  - God planlegging gir god programvare
- Gode dokumentasjonsrutiner, ved å følge gitte dokumentasjonsstandarder
- God programmeringsskikk og programmeringsstandarder
- Grundige tester av applikasjonen, med påfølgende analyse av testene
- At prosjektgruppen har valgt rett utviklingsmodell for programvare

Vi ser på det som særlig viktig at kommunikasjonen innad i gruppen er god, samt kommunikasjonen mellom prosjektgruppen og styringsgruppe. Dette er fordi vi tror at vi kan takle uforutsette hendelser og andre risikoelementer ved et godt samarbeid innad i prosjektgruppen. I tillegg er det særlig viktig med en tydelig kommunikasjon mellom prosjektgruppen og oppdragsgiver, siden vi antar at det er sannsynlig at oppdragsgiver kan endre på noen av sine krav og ønsker for sluttproduktet underveis.

Prosjektgruppen vurderer de følgende elementene som risikoelementer, som kan utgjøre en risiko for prosjektets sluttprodukt. Disse elementene kan forlenge eller gjøre arbeidet med planlegging, gjennomføring og testing vanskeligere:

- For høy belastning på medlemmene i prosjektgruppen
  - Prosjektgruppen kan ta på seg for mange arbeidsoppgaver ved å utvide prosjektoppgaven, og dermed skape et større tidspress og en følelse av man ikke klarer å gjennomføre prosjektet på en tilstrekkelig måte
- Støy og forstyrrelser i utviklingsmiljøet
  - Kan skape et dårlig utviklingsmiljø, og dermed en mindre effektiv prosjektgruppe
- For lite ressurser og tid lagt ned i forberedelser og planlegging
  - Kan skape problemer under utvikling, ved at det dukker opp for mange uventede faktorer
- Manglende kommunikasjon mellom gruppemedlemmer
  - Kan føre til at arbeidsoppgaver blir glemt, eller gjort dobbelt opp. Dette kan bety at prosjektgruppen ikke klarer å holde tidsplan
- Manglende eller utydelig kommunikasjon mellom prosjektgruppe og oppdragsgiver

- Kan sette prosjektgruppen på feil vei i forhold til retningen oppdragsgiver ønsker, og dermed kan sluttproduktet ikke bli av god nok kvalitet
- Ikke standardisert eller dårlig kodeskikk
  - Kan gjøre at prosjektet blir umulig å vedlikeholde
- Utilstrekkelig med testing og analysering
  - Kan gjøre at prosjektgruppen gir ut programvare som ikke fungerer godt nok, og dermed skape et dårlig sluttprodukt

Vi i prosjektgruppen har laget en plan for å minimere risikoelementene.  
Denne planen innebærer å:

1. gjøre et grundig forarbeid av prosjektet, gjøre nødvendige analyser og lage en tidsplan
2. ha minst 3 gruppemøter hver uke, og bruke gode verktøy for å håndtere kommunikasjon og systemutviklingsmetoden
3. lage prosedyrer for dokumentasjon, og gi ansvar til prosjektmedlemmer for utførelse av disse prosedyrene
4. gå igjennom hverandres kode, og etter beste evne bruke «beste praksis» innen programvareutvikling. Konstruktive tilbakemeldinger blir viktig gjennom hele prosjektets livsløp
5. ha prosedyrer for testing og analysering av utført arbeid

## 5.6 Hovedaktiviteter i videre arbeid

#	Aktivitet	Ansvar	Tidsperiode
<b>A1</b>	<b>Prosjektplanlegging og oppsett av utviklingsmiljø</b>	<b>Prosjektgruppe</b>	<b>2 uke</b>
A1.1	Forprosjekteringsrapport ferdig	Prosjektgruppe	N/A
A1.2	Utarbeide en kravspesifikasjon (kan kanskje ses bort fra, dersom en slik rapport ikke er nødvendig)	Prosjektgruppe	2 dager
A1.3	Design og skissering av database	Prosjektgruppe	2 dager
A1.4	Design og konfigurasjon av server	Prosjektgruppe	2 dager
A1.5	Valg og konfigurasjon av utviklingsmiljø	Prosjektgruppe	1 dag
A1.6	Innsamling av data til arkiv-systemet	Prosjektgruppe	1 dag
A1.7	Design og skissering av brukergrensesnitt til webapplikasjon	Prosjektgruppe	2 dager
<b>A2</b>	<b>Utvikling av databasesystem</b>	<b>Prosjektgruppe</b>	<b>5 uker</b>
A2.1	Generere database med innhold	Prosjektgruppe	1 uker
A2.2	Initialisere og konfigurere server mot database	Prosjektgruppe	2 uker
A2.3	Lage funksjonalitet for innsetting/uthenting av data i database og server	Prosjektgruppe	2 uker
<b>A3</b>	<b>Utvikling av en fungerende prototype av webapplikasjon</b>	<b>Prosjektgruppe</b>	<b>5 uker</b>
A3.1	Utvikle back-end for administrator	Prosjektgruppe	1 uker
A3.2	Utvikle brukergrensesnittet til applikasjonen	Prosjektgruppe	2 uker
A3.3	Utvikle funksjonalitet som håndterer samhandlingen mellom applikasjon og informasjonssystem	Prosjektgruppe	2 uker

A4	Testing, oppgradering og debugging	Prosjektgruppe	2 uker
A4.1	Testing av webapplikasjon på flere plattformer	Prosjektgruppe	1 uke
A4.2	Analysering av kode	Prosjektgruppe	1 uke
A5	Dokumentasjon: sluttrapport	Prosjektgruppe	1 uke

Tabell 4: Oversikt over hovedaktivitetene i prosjektet

Denne oversikten er bare et anslag på når ting skal gjøres, og når det skal være ferdig. Arbeidsflyten kommer til å bli styrt av hva som senere planlegges for de respektive sprintene.

## 5.7 Framdriftsplan – styring av prosjektet

### 5.7.1 Hovedplan

En foreløpig plan over prosjektet er beskrevet i punkt 5.6. Denne planen er ikke fastlåst, og det vil forekomme endringer på et senere tidspunkt. Når prosjektet kommer i gang, vil en Product Backlog (produktkø) opprettes, der arbeidsoppgaver vil bli lagt inn i de respektive sprintene, og fordelt på medlemmene i utviklingsteamet. Denne Backlog'en vil bli styrt og håndtert av produkteier.

### 5.7.2 Styringshjelpemidler

Versjonskontroll er et system som holder styr på forandringer i en fil, eller et sett av filer, over tid, slik at du kan finne tilbake til spesifikke versjoner senere. Under utviklingen av dette prosjektet, vil følgende produkter og løsninger bli brukt for versjonsstyring og versjonskontroll:



**Atlassian Jira** – et av verdens mest utbredte systemer for oppgavehåndtering



**Atlassian Confluence** – organiser og diskuter arbeidet innad i teamet



**Atlassian Bitbucket** – Git-håndtering



**Atlassian SourceTree** – Git desktop-applikasjon

### 5.7.3 Utviklingshjelpemidler

Maskinvare som vil bli benyttet til utvikling og testing av dette prosjektet:

- **PC** med Microsoft Windows operativsystem
- **Apple MacBook Pro** med macOS Sierra operativsystem
- **Apple iPhone 6S** med iOS 10 operativsystem
- **Apple iPad Mini** med iOS 9 operativsystem
- **Huawei Honor 7** med Android 6.0 operativsystem
- **Samsung Galaxy S7 Edge** med Android 6.0 operativsystem
- **Samsung Galaxy Tab** med Android 6.0 operativsystem



iOS 10



ANDROID

Figur 2: De forskjellige operativsystemene som skal brukes under utvikling og testing

**Programvare som vil bli benyttet til utvikling og testing av dette prosjektet:**



Bootstrap Studio 2.5



Microsoft Office 365 (Microsoft Word, Microsoft Powerpoint)



Payara Server 164



NetBeans IDE 8.1



Adobe Photoshop CC 2017



Google Chrome 55.0



Apple Safari 10.0



Mozilla Firefox 50.1



Atom 1.12.3



Adobe Acrobat Reader 11



PDF Reader X 2.2.0



Sublime Text 3



EndNote X8

#### **5.7.4 Intern kontroll – evaluering**

Intern kontroll og evaluering av arbeidet som er gjort av prosjektgruppen vil bli utført av prosjektgruppen sammen med oppdragsgiver hver sprintstart. Prosjektgruppen skal bruke et verktøy som heter Atlassian JIRA til å organisere arbeidet.

Oppdragsgiver vil også være en del av denne organiseringen, og kan dermed følge utviklingen av prosjektet når han måtte ønske. Ved å bruke dette verktøyet, samt god kommunikasjon mellom gruppemedlemmene vil vi holde kontroll over arbeidet, og evaluere hvert enkelt delmål underveis i prosessen.

Hver fjortende dag vil prosjektgruppen og styringsgruppen holde Sprint Review-møter, hvor vi vil gå igjennom hva som er gjort, hva som kan forbedres og hva som er delmålet/målet for neste sprint i utviklingsmetoden.

I tillegg til dette, vil vi benytte oss av verktøyet Atlassian Confluence for organisering av arbeid, samt kommunikasjon mellom styringsgruppe og prosjektgruppe.

#### **5.8 Beslutninger – beslutningsprosess**

I prosjekter vil det hele tiden oppstå nye utfordringer og problemer, det ligger i prosjektet natur. Dette også basert på prosjektgruppens valg av prosjektutviklingsmodell der kontinuerlig



testing med prøving og feiling er sentralt. Prosjektarbeidet vil sannsynligvis forandres underveis i prosjektets livssyklus.

Alt av drastiske beslutninger vedrørende prosjektet skal tas i samråd med styringsgruppen, som består av veileder og oppdragsgiver.

Ved utfordringer/problemer vil prosjektgruppen prioritere denne oppgaven, så lenge den er av vesentlig grad. Slik kan vi som prosjektgruppe identifisere ulike løsninger som vil kunne løse årsaken til problemet. Løsningen bør være mulig å implementere, samt at det bør forhindre at problemet oppstår igjen.

Deretter vil prosjektgruppen ta de ulike løsningene til styringsgruppen slik at vi kan fatte en beslutning sammen. Deretter vil prosjektgruppen gjennomføre beslutningen. Prosjektgruppen vil da fokusere på å få denne beslutningen implementert.

Ved mindre utfordringer/problemer som ikke vil drastisk endre prosjektet vil prosjektgruppen selv ta en avgjørelse, og i neste styringsgruppemøte informere om hva som har skjedd og hvilke tiltak som er iverksatt for det spesifikke problemet/utfordringen. Alle små som store problemer/utfordringer vil påvirke prosjektet på en eller annen måte, og løsningen bør gjennomføres så kjapt som mulig, før neste utfordring måtte dukke opp.

## 6 DOKUMENTASJON

### 6.1 *Rapporter og tekniske dokumenter*

Gjennom dokumentasjon skal vi få fram det vi har observert og utført av arbeid gjennom hele arbeidet av prosjektet. Vi skal dokumentere fordi dokumentasjonen skal brukes til noe. Enten som grunnlag for å forstå, for videre arbeid, som grunnlag for å få tatt en beslutning eller fordi det skal leveres til oppdragsgiver som en del av sluttproduktet.

Prosjektgruppen skal legge vekt på følgende i dokumentasjonen:

- At innholdet i dokumentasjonen skal gi en tilstrekkelig beskrivelse av det som dokumenteres
- At all viktig informasjon er lett å få øye på for en ny leser
- At vi skal skrive så kort og presist som mulig
- At vi skal bruke et godt språk, og minimere antallet skrivefeil
- At vi skal ha god ryddighet og organisering av dokumentet
- At dokumentasjonen skal være lettlest, men seriøs
- At det skal legges stor vekt på nøyaktig bruk av referanser
- At vi ellers skal følge god skikk og struktur, som beskrevet av Nils Olsson [15]

Følgende dokumentasjon skal utarbeides av prosjektgruppen gjennom livssyklusen til prosjektet:

- Applikasjonsutviklingsrutiner
- Distribusjon og kopiering av materialer, samt dokumentasjon om opphavsrettigheter
- Fullstendig kravspesifikasjon av systemet
- Periodiske utviklingsrapporter
- Test- og analysedokumentasjon
- Rapporter fra styringsmøte
- Rapporter om avvik gjennom prosjektets livssyklus
- Komplett sluttrapport for bachelorprosjektet

## 7 PLANLAGTE MØTER OG RAPPORTER

### 7.1 Møter

#### 7.1.1 Møter med styringsgruppen

Planlagte møter mellom prosjektgruppen og styringsgruppen vil primært bli avholdt i NTNU sine lokaler i Ålesund.

Dato	Klokkeslett	Sted	Agenda
18.01.17	10.00	Rom B315	Informasjonsmøte. Planlegging og forprosjekt.
03.02.17	10.00	Rom B315	Møte med fokus på oppstart av prosjektet
13.02.17	10.00	Rom B315	Systemutviklingsmøte med styringsgruppen
03.03.17	10.00	Rom B315	Systemutviklingsmøte med styringsgruppen
17.03.17	10.00	Rom B315	Systemutviklingsmøte med styringsgruppen
31.03.17	10.00	Rom B315	Systemutviklingsmøte med styringsgruppen
14.04.17	10.00	Rom B315	Systemutviklingsmøte med styringsgruppen
28.04.17	10.00	Rom B315	Systemutviklingsmøte med styringsgruppen
12.05.17	10.00	Rom B315	Systemutviklingsmøte med styringsgruppen
28.05.17	10.00	Rom B315	Avsluttende møte med styringsgruppen

Tabell 5: Oversikt over planlagte møter

Alle planlagte møter er tentative, og endringer kan forekomme på kort varsel.

#### 7.1.2 Prosjektmøter

Prosjektgruppen vil holde daglige Scrum-møter, kl. 08.45-09.00.

I de daglige Scrum-møtene, vil følgende spørsmål bli stilt (og presumptivt besvart):

- Hva fikk du gjort i går?
- Hva skal du gjøre i dag?
- Er det noe som hindrer din fremdrift?

I tillegg til disse forhåndsbestemte daglige Scrum-møtene, så vil det hele tiden foregå kommunikasjon i diverse kanaler mellom deltakerne i prosjektgruppen, det være seg kommunikasjon ansikt til ansikt, meldingsutveksling over SMS eller Facebook Messenger, eller e-post.

## **7.2 Periodiske rapporter**

### **7.2.1 Framdriftsrapporter (inkl. milepæl)**

All planlagt og ikke planlagt form for fremdrifts- og milepælrapportering vil skje gjennom verktøyet Atlassian Confluence. Dette vil være rapportering mellom prosjektgruppen og styringsgruppen.

## **8 PLANLAGT AVVIKSBEHANDLING**

Håndtering av avvik vil inkludere alle stegene som skal til for finne et avvik, og for å sette inn korrigerende og forebyggende tiltak for problemet. Med avvik mener vi mangelen på oppfyllelse av spesifiserte krav.

Avviksbehandling inkluderer alle steg som må gjøres om fremdriften av prosjektet ikke går som planlagt, eller at innholdet i prosjektet ikke er det som er forespeilet for produktet. Alle avvik skal dokumenteres, samt alle tiltak som er iverksatt i forbindelse med avviket. Mindre tiltak kan iverksettes umiddelbart, og har dermed ikke noen store tidsmessige konsekvenser.

For avvik av alvorlig grad kan det være naturlig å sette opp et korrigerende tiltak. Det kan være basert på produktet eller ikke. Det kan være forbedringstiltak basert på prosess, trivsel eller helse og miljø.

Håndtering av avvik og endringer er avgjørende for prosjektets suksess. Avvik trenger ikke å være av det vonde, det kan faktisk være avgjørende for prosjektets suksess på lang sikt at nødvendige endringer i prosjektet blir gjort. Prosjektet må kunne tilpasse seg en dynamisk verden hvor det er umulig å forutse og beskrive alle forutsetninger før prosjektet starter opp.

Vi ønsker å ha et aktivt forhold til hvordan avvik og endringer håndteres, samt at vi skal arbeide målbevisst med å fange opp tidlige varslings signaler og fatte tiltak basert på dette. I 5.8 står det om beslutningsprosessen som prosjektgruppen forholder seg til. Alle avvik vil også bli presentert for styringsgruppen, samt beslutninger rundt alvorlige avvik vil tas sammen med styringsgruppen. Ved mindre avvik vil prosjektgruppen selv fatte tiltak og ta en beslutning vedrørende korrigerende tiltak.

Hele prosjektgruppen vil ha et ansvar over avviksbehandlingen, men prosjektlederen vil ha et overordnet ansvar for håndteringen.

Avvik kan behandles slik:

- Omarbeidet for å tilfredsstille spesifiserte krav
- Godkjent med avvikstillatelse
- Omklassifisert til annet bruk
- Avvist
- Vraket
- Endring i kravspesifikasjon

## **9 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING**

Se punkt 5.7.3 for liste over utstyr (maskinvare, programvare etc.).

Det vil ikke være behov for noe annet utstyr utover dette, slik vi anser det per nå.

Endringer i utstyrliste kan forekomme underveis.

## 10 REFERANSER

[1]

Wikipedia Foundation. *World Wide Web*. 2016.

URL: [https://no.wikipedia.org/wiki/World\\_Wide\\_Web](https://no.wikipedia.org/wiki/World_Wide_Web)

[2]

Wikipedia Foundation. *Database*. 2016. URL: <https://no.wikipedia.org/wiki/Database>

[3]

Wikipedia Foundation. *Server*. 2017. URL: <https://no.wikipedia.org/wiki/Server>

[4]

Wikipedia Foundation. *Webapplikasjon*. 2016.

URL: <https://no.wikipedia.org/wiki/Webapplikasjon>

[5]

Wikipedia Foundation. *Portable Document Format*. 2016.

URL: [https://no.wikipedia.org/wiki/Portable\\_Document\\_Format](https://no.wikipedia.org/wiki/Portable_Document_Format)

[6]

Wikipedia Foundation. *Smarttelefon*. 2016.

URL: <https://no.wikipedia.org/wiki/Smarttelefon>

[7]

Wikipedia Foundation. *Nettbrett*. 2015. URL: <https://no.wikipedia.org/wiki/Nettbrett>

[8]

Difi – Direktoratet for forvaltning og IKT. *Hva er universell utforming?* 2017.

URL: <https://uu.difi.no/kva-er-universell-utforming#definsjon>

[9]

NITO – Norges Ingeniør- og Teknologorganisasjon. *Etikk i NITO*. 2016.

URL: <https://www.nito.no/organisasjon/om-nito/etikk-i-nito/>

[10]

Wikipedia Foundation. *Representational state transfer*. 2016.

URL: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)

[11]

Wikipedia Foundation. *Git*. 2016. URL: <https://no.wikipedia.org/wiki/Git>

[12]

Wikipedia Foundation. *Apache Derby*. 2016.

URL: [https://no.wikipedia.org/wiki/Apache\\_Derby](https://no.wikipedia.org/wiki/Apache_Derby)

[13]

Agile Manifesto. *Manifestet for smidig programvareutvikling*. 2001.

URL: <http://agilemanifesto.org/iso/no/manifesto.html>

[14]

Wikipedia Foundation. *Bootstrap (front-end framework)*. 2017.

URL: [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

[15]

Olsson, N. Praktisk rapportskriving. Trondheim: Tapir akademisk forlag; 2011.