



UNIVERSITETET I AGDER

Prosjektportalen Teams IS-304: 2021

Tittel:

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder	Geir Inge Hausvik
Oppdragsgiver:	Puzzlepart

Studenter:

Etternavn	Fornavn
Ertzeid Toft	Ragnhild Sofie
Islann Fremstad	Henriette
Eikaas	Tuva Cecilie
Ruud	Martin
Bjurholt Rein	Kristian

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeider uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA X	NEI ___
Kan besvarelsen brukes til undervisningsformål?	JA X	NEI ___
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA X	NEI ___

Forord

Denne oppgaven utgjør siste del av et bachelorstudium i IT og informasjonssystemer ved Universitet i Agder. Vi er en gruppe på til sammen fem personer som alle har vært med på å bidra til det endelige resultatet, der produktet gjenspeiler vår utvikling i løpet av de tre studieårene. Oppgaven vi har arbeidet med har alle hatt stor interesse for og fått stort utbytte av, noe som vi vil ta med oss videre på veien. Det har både vært en spennende og krevende prosess, der vi alle har måttet håndtere og tilrettelegge arbeidet under pandemien vi har stått ovenfor. Likevel, kan vi med stolthet se tilbake på et semester med hardt arbeid og ta med oss gode erfaringer som vil komme til nytte ved senere anledninger.

Vi ønsker i denne sammenheng å rette en stor takk til bedriften Puzzlepart. Vi vil takke for et flott samarbeid og for all hjelpen vi har fått i form av rådgivning, tilgang på arbeidsplass og fra start til slutt å være interessert og engasjert i vårt arbeid. Dette har virket motiverende og gitt oss ekstra trygghet i gjennomføringen av oppgaven.

Vi retter også en særlig stor takk til vår veileder, Geir Inge Hausvik. Takk for de gode rådene og tilbakemeldingene vi har fått i løpet av semesteret. Dette har vært til stor inspirasjon.

Avslutningsvis ønsker vi å takke alle deltakerne i vår spørreundersøkelse som ofret sin tid på vårt konsept. Deres erfaringer og meninger har vært med på å kunne gjøre gjennomføringen av bacheloroppgaven mulig.

Sofie Toft

Ragnhild Sofie Ertzeid Toft

Henriette I. Fremstad

Henriette Islann Fremstad

Tuva C. Eikaas

Tuva Cecilie Eikaas

Martin Ruud

Martin Ruud

Kristian B Rein

Kristian Bjurholt Rein

Kristiansand, 13.mai 2021

Abstrakt

Som en del av bachelorprogrammet i IT og informasjonssystemer, har gruppen i løpet av våren 2021 gjennomført prosjektet sitt hos bedriften Puzzlepart. IT bedriften er Norges ledende på Microsoft 365, og har i de siste årene opplevd en enorm vekst knyttet til etterspørsel av deres tjenester. Fra et tidlig stadium, var det en felles enighet om at gruppen var på utkikk etter et prosjekt som både var utfordrende og relevant. Gruppen hadde et stort ønske om å utvikle tekniske ferdigheter og forståelse, som er bakgrunnen for at valget falt på å jobbe sammen med Puzzlepart. Arbeidet med prosjektet begynte i januar, og et ferdig sluttprodukt skal kunne leveres og presenteres for UiA i begynnelsen av juni. I løpet av denne perioden har gruppen jobbet med å utvikle en Microsoft Teams-basert applikasjon etter forslag av Puzzlepart. Det vil derfor i rapporten bli presentert om de valgene som er foretatt underveis i prosjektet, samt begrunne disse. Gruppen har satt seg som mål om at applikasjonen skal være et rammeverk som Puzzlepart kan bygge videre på, og som i første omgang kan betraktes som et «proof of concept». Dette fordi produktet ikke er ferdiginnstilt, men er en forretningside som Puzzlepart kan benytte seg videre av.

Innholdsfortegnelse

1. Introduksjon.....	6
1.1 Om Puzzlepart.....	6
1.2 Om Prosjektportalen 365	6
2. Produktet.....	7
2.1 Produktbeskrivelse	7
2.2 Minste og mest brukbare produkt.....	7
2.3 Produktdemonstrasjon	8
2.4 Teknisk overordnet.....	8
3. Analyse og design	10
3.1 Målgruppe	10
3.2 Spørreundersøkelse.....	10
3.3 Brukerhistorier	12
3.4 Design brukergrensesnitt	13
3.4.1 Wireframes	14
3.4.2 Prototype.....	16
3.5 Arkitekturdesign	18
4. Prosjekthåndtering	20
4.1 Kommunikasjon og arbeid.....	20
4.1.1 Azure DevOps og Microsoft Planner	20
4.1.2 Microsoft Teams og Discord	21
4.1.3 Arbeidslokasjon	21
4.2 Scrum	22
4.2.1 Sprint.....	22
4.2.2 Daglige møter	23
4.2.3 Scrum Team.....	23
4.2.4 Produkt Backlog	24
4.2.5 Sprint Backlog	24
4.2.6 Sprintplanleggingsmøte	25
4.2.7 Sprintgjennomgangsmøte	25
4.2.8 Sprintretrospektiv	26
4.2.9 Sprintgjennomføring.....	26
4.3 Burndown diagram.....	30
4.4 Risikoanalyse	32
4.5 Programmeringsspråk og miljø.....	33
4.5.1 Programmeringsspråk og rammeverk.....	33
4.5.2 Versjonskontroll	34
4.5.3 Kodestandard og struktur.....	35
5. Kvalitet.....	37
5.1 Hvordan sikre kvalitet?	37
5.1.1 Oppdragsgiver sine krav om kvalitet.....	37
5.1.2 Sikre kvalitet i kildekoden.....	38
5.1.3 Arbeidsrutiner for å sikre kvalitet.....	39

5.1.4 Testing	40
6. Testing og kvalitetssjekk av produktet	41
6.1 Kvalitetssjekk	41
6.2 Testing	42
6.3 Vurdering av oppnådd kvalitet.....	43
7. Refleksjon.....	44
7.1 utfordringer og endringer	44
7.2 Bruk av erfaringer fra tidligere semester	45
7.3 Læringsutbytte	46
7.3.1 Tekniske ferdigheter	46
7.3.2 Samarbeidsferdigheter og gruppeevaluering	46
7.3.3 Egenevaluering	47
8. Uttalelse fra oppdragsgiver.....	49
9. Referanseliste.....	51
10. Vedlegg.....	54
10.1 Spørreundersøkelse med svar	54
10.2 Brukerhistorier	59
10.3 Risikomatrix.....	65

Figurliste

Figur 1 - Brukerhistorie	13
Figur 2 - Wireframe 1	15
Figur 3 - Wireframe 2	15
Figur 4 - Forside prototype	17
Figur 5 - Risikomatrix prototype.....	17
Figur 6 - Burndown chart	31
Figur 7 - Risikomatrix	32
Figur 8 - Teams tab.....	33
Figur 9 - Github webhook poster oppdateringer i kodebase.	35
Figur 10 - Funksjonelt komponent	35
Figur 11 - Interface og komponent struktur.....	36

Tabelliste

Tabell 1 – Prioritering av designkriterier	18
--	----

1.Introduksjon

I tett samarbeid med Puzzlepart, har gruppen jobbet med å utvikle en Microsoft Teams-applikasjon som skal kunne installeres i Prosjektportalen 365-prosjektet. I henhold til oppdragsgiver er tanken at applikasjonen skal kunne gi kundene til Puzzlepart en oppsummering av deres prosjekter og dets viktigste elementer som prosjekttegenskaper, statusrapportering, risiko og prosjektleveranser. Nøyaktig hvilke elementer applikasjonen skal inneholde har vi fått lov til å være med å ta en avgjørelse på, men noen kriterier er blitt fastsatt. Applikasjonen skal først og fremst være konfigurierbar gjennom app properties, den skal ta hensyn til alle brukerflater, være gratis tilgjengelig for alle, og til slutt skal den kunne settes opp med template for Microsoft teams.

1.1 Om Puzzlepart

Puzzlepart er en IT-bedrift som startet opp i 2008. Siden den gang har bedriften utviklet flere tjenester og løsninger, som baserer seg på Microsoft teknologi. Puzzlepart har både lokaler i Oslo og Kristiansand, og i 2019 passerte bedriften 35 ansatte (Puzzlepart, u.å.). Det hele begynte med et ønske om å bli anerkjente leverandører av «Disruptive Enterprise Apps» på global basis (Puzzlepart, u.å.). Bedriften fant ut at dette ikke kom til å bli en suksess, og tok en rask beslutning om å bli et konsultentselskap. Puzzlepart ble etter hvert med i Crayon, og i denne perioden startet satsingen for fullt på Office 365 (Puzzlepart, u.å.). Som følge av dette har bedriften den dag i dag, mange store kunder og opplever stadig ett stort behov av deres tjenester.

1.2 Om Prosjektportalen 365

Prosjektportalen 365 er en løsning som er godt egnet for portefølje- og prosjektstyring, og som baserer seg på Microsoft 365 (Puzzlepart, u.å.). Det er med andre ord et smart verktøy som lar deg kunne samle prosjektene dine på ett og samme sted. Verktøyet er særlig tilpasset det norske markedet, og passer både privat og offentlig sektor (Puzzlepart, u.å.). Det kan også anvendes i bygg- og anleggsbransjen (Puzzlepart, u.å.). Gjennom SharePoint har man tilgang til ulike verktøyer for dokumenthåndtering, og ved å benytte seg av Planner, sørger Prosjektportalen 365 for at det er enkelt å håndtere oppgaver.

2. Produktet

Under dette prosjektet har gruppen utviklet Prosjektportalen Teams, som er en Microsoft Teams applikasjon. Produktet skal i hovedsak gjøre prosjektplanlegging enklere for Puzzlepart sine kunder. I dette kapittelet vil det fremkomme mer informasjon om Prosjektportalen Teams, en videodemonstrasjon av produktet og informasjon om det tekniske bak produktet. Den tekniske overordnede informasjonen skal være med på å legge en grunnleggende forståelse av applikasjonen, da den er noe utradisjonell. Gruppen sitt mål for prosjektet vil fremkomme ved hjelp av en beskrivelse av minst brukbare produkt.

2.1 Produktbeskrivelse

Prosjektportalen Teams er et prosjektstyringsverktøy som skal være tilgjengelig på samarbeidsplattformen Microsoft Teams. Applikasjonen er bygget på åpen kildekode og er gratis for alle brukere av Prosjektportalen 365. Dette er en utvidelse av Prosjektportalen 365 som er skreddersydd for Microsoft Teams. Her har brukeren oversikt over blant annet generelle prosjektegenskaper, prosjektleveranser, prosjektstatus, risikoanalyse og ressursallokering. Brukerne har lese- og redigeringsmuligheter til hvert enkelt prosjekt avhengig av sine tildelte rettigheter. Prosjektportalen Teams gir en visuell oversikt over ett enkelt prosjekt i hver Teams kanal. Visse egenskaper i denne instansen kan brukeren tilpasse etter eget behov. Dette innebærer seleksjon av hvilke navigasjonselementer som ønskes, samt rekkefølgen av disse, størrelse på risikomatrise, og seleksjon av private lister som er lagt til prosjektet.

2.2 Minste og mest brukbare produkt

For å vite om målet for utviklingen av produktet er oppnådd må det defineres. Måloppnåelse går langs en skala, fra minst brukbare produkt (MBP) til mest brukbare produkt. MBP har nok funksjoner til å tilfredsstille tidlige kunder, samt kan produktet generere tilbakemeldinger for fremtidig produktutvikling (Som de Cerff et al., 2018, s. 2510). Gruppen definerer minst brukbare produkt slik; Når Prosjektportalen i Microsoft Teams kan brukes som et rammeverk som kan utvikles videre, der noe av den viktigste funksjonaliteten er implementert. Samt godt dokumentert og strukturert kode som gjør det enkelt for utviklere å forstå koden. Mest brukbare produkt er; et fullverdig produkt der flertallet av de viktigste komponentene er på plass. Produktet kan også brukes uavhengig av SharePoint Prosjektportalen 365. Gruppen sitt

mål var å nå minst brukbare produkt. Den definerte måloppnåelsen kan senere brukes for å måle vårt arbeid og produkt mot slutten av utviklingen.

2.3 Produktdemonstrasjon

Vår produktdemonstrasjon viser status 13. mai 2021. Den gjennomgår de mest sentrale egenskapene til applikasjonen med forklaring. Ved å klikke på denne linken, https://drive.google.com/file/d/1UNUEIcAFPpJ4N-T4yh_VFGFYMVU-B3_y/view vil en få tilgang til en videodemonstrasjon av Prosjektportalen Teams, hvor flere ulike plattformer blir tatt i bruk.

For å installere applikasjonen navigerer en til vårt åpne Github repository, vist med denne linken <https://github.com/Rundez/Prosjektportalen-teams>. Her vil en finne installasjonspakken under «Releases». Etter en har lastet ned nyeste utgivelse må deretter sspkg-filen legges til i organisasjonens SharePoint app-bibliotek. Dette vil publisere Prosjektportalen Teams som fane for alle brukere i organisasjonen. Alle nyopprettede prosjekter vil heretter genereres med ferdig konfigurert Teams utvidelse.

2.4 Teknisk overordnet

Prosjektportalen Teams er bygget på SharePoint som gjør at «stacken» fremstår som noe utradisjonell. Hver enkelt organisasjon har sitt eget område i Microsoft som på fagspråket kalles en Tenant. I denne tenanten eksisterer SharePoint, Teams, Office dokumenter og liknende. Selve Prosjektportalen Teams fungerer ved at det opprettes en SharePoint hub-side ved installasjon, som er en side hvor alle prosjekter samles opp og knyttes til. For å utvide standardfunksjonaliteten til SharePoint blir det lagt til tilpassede SPFx (SharePoint framework) React komponenter på utvalgte sider. Ved opprettelse av et prosjekt lages en underside av hub-siden med sine egne tilpasninger.

Denne oppbygningen av det eksisterende miljøet ga oss i utgangspunktet noen valgmuligheter. Vi kunne bygget en standard backend og frontend som hadde blitt knyttet opp mot en applikasjon i Teams. Da ville vår backend måtte ha godkjenning fra hver organisasjon sitt område til å kunne autentisere brukere, skrive og lese SharePoint miljøet. Dette ville betydd mer arbeid for brukerne da de må godkjenne API forespørsler, samt for utviklerne.

Istedenfor valgte vi å gå for en Microsoft-løsning utviklet for SharePoint, SharePoint Framework (SPFx). SharePoint framework er et rammeverk for å bygge applikasjoner som lever i en SharePoint tenant. Som oftest blir det brukt til å lage egne applikasjoner og/eller funksjonalitet som ikke kan lages med Microsoft sine ferdige løsninger. SPFx bruker SharePoint som tjener, noe som eliminerer behovet for å bygge vår egen backend løsning. Dette er en attraktiv løsning ettersom brukerne «eier» sin egen installasjon og de vil kun sende spørringer til sin tilhørende instans av programmet. Det vil med andre ord bli opprettet en tjener til hver enkelt installasjon, noe som bidrar til økt sikkerhet for brukerne samt redusert arbeidsmengde for gruppen. Dette sørger også for at data blir lagret i henhold til personvern automatisk.

3. Analyse og design

Ettersom Prosjektportalen 365 er en eksisterende plattform, ble en spørreundersøkelse utsendt til de tilhørende brukerne. Svarene ble nøye gjennomgått og analysert, og deretter brukt som grunnlag til utforming av brukerhistorier. Brukerhistoriene lister funksjoner, krav for aksept og prioritering i utviklingen. På grunnlag av denne analysen og prioriteringen ble designforslag lagt frem ved hjelp av wireframes. Gruppen så på dette som forberedende arbeid, som ble gjennomført i pre-sprinten.

3.1 Målgruppe

For å sikre at det ferdige produktet vil bli tatt i bruk må en treffe riktig målgruppe. Tanken var at målgruppen ville være eksisterende brukere av Prosjektportalen 365 til Puzzlepart. Her er det to undergrupper som begge kunne være mulige målgrupper, brukere av Prosjektportalen i SharePoint, og brukere av Prosjektportalen gjennom Microsoft Teams. Derfor vil både de ansatte i Puzzlepart og deres kunder som anvender Prosjektportalen være den ønskede målgruppen. Denne målgruppen ble dermed tildelt en spørreundersøkelse for å innhente informasjon angående deres opplevelse med Prosjektportalen 365.

3.2 Spørreundersøkelse

En kvantitativ spørreundersøkelse ble utformet og brukt for å innhente informasjon om dagens bruk av Prosjektportalen 365, eksisterende viktig funksjonalitet, mangler og hvem som bruker siden. En kvantitativ spørreundersøkelse samler inn informasjon som omkodes til tall, der tallene fungerer som symboler (Jacobsen, 2018, s. 38). Ved spørsmålet “Hvor fornøyd er du med Prosjektportalen 365?” kan svaralternativene gå på en 5-punktsskala fra “Svært misfornøyd” til “Svært fornøyd”. Her vil svaralternativene kunne omkodes til tall, der 1 tilsvarer “Svært misfornøyd” og 5 tilsvarer “Svært fornøyd”. Denne analysemetoden ble valgt på grunnlag av at gruppen ønsket å generalisere bruken av Prosjektportalen 365, slik at applikasjonen kunne inneholde de viktigste funksjonene. Samt er Prosjektportalen 365 en eksisterende plattform, som vi har god forhåndskjennskap til og dermed ønsker å gå mer i bredden (Jacobsen, 2018, s. 136).

Fordelene ved å samle inn kvantitative data er at mange enheter kan undersøkes, det har lave kostnader og man kan generalisere fra utvalg til populasjon med stor grad av sikkerhet (Jacobsen, 2018, s. 137). Ulempen med å velge bort en kvalitativ metode som intervju er

blant annet at nyanser og muligheten til å komme med oppfølgingsspørsmål faller vekk (Repstad, 2007, s.17). Nyanser og oppfølgingsspørsmål kan være med på å få frem viktige detaljer om dagens bruk av Prosjektportalen 365. På den andre siden kan det være problematisk å generalisere svarene fra en kvalitativ metode (Repstad, 2007, s.27).

Utvalget i vårt prosjekt er brukerne av Prosjektportalen 365 som gjennomfører spørreundersøkelsen, populasjonen er alle brukerne av Prosjektportalen 365. Gruppen ønsket at så mange som mulig kunne gjennomføre spørreundersøkelsen og fikk hjelp av Puzzlepart til å sende den ut til Prosjektportalen 365 sine brukere. Dessverre var det kun 13 personer som valgte å svare på den, også etter purring. Et så lavt svartall kan svekke muligheten til å generalisere svarene. Svarene som kom inn viser at dette er brukere av Prosjektportalen 365 som har ulike roller, både prosjektledere, IT-ansvarlige og prosjektdeltakere har svart på undersøkelsen. Dette gjør at svarene på spørsmålene er nyanserte, da ulike roller har ulike behov i bruk av Prosjektportalen.

Spørreundersøkelsen avdekker blant annet hvilken funksjonalitet som er viktigst for brukerne. Funksjonalitetene som er mest populære stemmer også overens med hva Puzzlepart har ytret at de mener er essensielt å ha med i applikasjonen. Enigheten mellom bedriften og brukerne av Prosjektportalen 365 var avgjørende for at gruppen valgte å bruke svarene i spørreundersøkelsen videre i prosjektet. Spørreundersøkelsen ble på bakgrunn av dette grunnpilaren ved utvikling av brukerhistorier, skrevet om i Kapittel 3.3.

Alle spørsmål og svar ligger under Vedlegg 10.1. Spørsmål fire i undersøkelsen avdekker hvilken kommunikasjonsplattform respondenten bruker til prosjektarbeid i dag, det mest gjentakende svaret er Microsoft Teams. Dette er positivt for utviklingen av prosjektet, da vi vet at brukerne har god kjennskap til plattformen applikasjonen skal befinne seg i, samt at de bruker det aktivt i prosjektarbeid. Dette gir også grunnlag for å basere designet i applikasjonen på Microsoft design, for bedre brukervennlighet. 76% av respondentene bruker dagens Prosjektportalen 365 to til tre ganger i uken eller mer, der halvparten av dem benytter seg av den hver dag. I kombinasjon med svarene fra spørsmål 12 viser det at en Prosjektportalen applikasjon i Microsoft Teams mest sannsynlig ville blitt brukt, da alle respondentene svarer dette. Spørsmål seks viser at 86% av respondentene er fornøyde med Prosjektportalen 365, men sammen med spørsmål 11 får gruppen innsikt i hva som kan forbedres, blant annet bedre planleggingsverktøy som Gantt diagram. I dagens samfunn er internett og teknologi med over alt, og arbeid kan gjøres på kontoret via en stasjonær datamaskin eller på mobiltelefonen. Dette gjør det nødvendig å ha universell utforming, som

gjør at sider kan brukes på ulike plattformer. Spørsmål 12 underbygger at universell utforming også er viktig når det kommer til Prosjektportalen Teams. Svarene på spørsmål åtte, ni, ti og elleve er sentrale ved utformingen av brukerhistoriene, da de gir innsikt i de hyppigst anvendte funksjonene, de viktigste funksjonene, samt forslag til nye funksjoner i Prosjektportalen Teams.

3.3 Brukerhistorier

Svarene i spørreundersøkelsen er brukt som grunnlag ved utvikling av brukerhistoriene, da svarene får frem hvilken funksjonalitet som blir hyppigst brukt og har størst verdi for brukerne av Prosjektportalen 365. Brukerhistorier anvendes for å fokusere og se et abstrakt nivå av brukere eller andre systemer som samhandler med målsystemet, i dette tilfellet Prosjektportalen 365 (Mathiassen, Munk-Madsen, Nielsen, Stage, 2018, s.121).

Brukerhistorier illustrerer et mønster for interaksjon mellom Prosjektportalen og dens brukere, der essensiell funksjonalitet blir beskrevet (Mathiassen, et al., 2018, s.122). Gruppen valgte å anvende brukerhistorier av flere grunner. For det første gir brukerhistorier utviklerne rom for å se Prosjektportalen Teams fra brukerens perspektiv og deres behov. For det andre gir de en god oversikt over ønsket funksjonalitet, når den er nådd og hvorfor den er viktig for brukeren. For det tredje gir brukerhistorier sammen med MoSCoW-metoden en ryddig oversikt og prioritering av oppgaver, som gjør fremgangen i utviklingen målbar.

For å kunne prioritere hvilke krav i brukerhistoriene som er av størst betydning for prosjektet anvendes MoSCoW-metoden. MoSCoW er et akronym for Must have, Should have, Could Have og Would/Won't have, og er en agil tilnærming for prioritering av krav (Hatton, 2008, s. 517-526). Bruk av en slik prioritering gir et godt utgangspunkt i utviklingen, der funksjonalitet vil bli utviklet på grunnlag av prioriteringen av brukerhistoriene. Dette gjør også at vi sikrer at den viktigste funksjonaliteten for brukerne er på plass når prosjektet skal leveres. Prioriteringen av brukerhistoriene er basert på svarene i spørreundersøkelsen.

Brukerhistoriene er inndelt i fire hoveddeler, selve brukerhistorie, beskrivelse av brukerhistorie, vilkår for aksept og argumentasjon. Den første hoveddelen, selve brukerhistorien, er satt opp etter malen: «Som en [bruker] ønsker jeg en [funksjon] slik at jeg [oppnår en verdi]» (Entur, u.å.). Her får vi frem hvem som bruker systemet, hvilken funksjonalitet de ønsker og hvorfor de ønsker den spesifikke funksjonaliteten. Videre blir brukerhistorien beskrevet, dette er for å beskrive den ønskede funksjonaliteten. Den tredje delen er vilkår for aksept, som stadfester når brukerhistorien er akseptert, det vil si når den

regnes som fullført. Argumentasjon er den fjerde og siste delen av brukerhistorien. Den har som hensikt å argumentere for plassering på prioriteringslisten av brukerhistorien, der gruppen bruker MoSCoW-prioritering. Eksempel på brukerhistoriene utformet i dette prosjektet er illustrert i Figur 1. Alle brukerhistoriene ligger i Vedlegg 10.2, der de er sortert etter MoSCoW-prioriteringen.

Nr. 1		Prioritet: Must have
Brukerhistorie	Som en prosjektleder ønsker jeg en oversikt over tilstanden til prosjekter slik at jeg enklere kan få oversikt over prosjektets status.	
Beskrivelse	Ettersom det er mange som jobber parallelt med prosjekter vil det være nødvendig å skaffe seg et oversiktlig bilde av fremdriften.	
Vilkår for aksept	Kriteriet er akseptert når brukeren enkelt kan få oversiktlig oppsummering av prosjektet.	
Argumentasjon	Dette er en «must have» fordi synliggjøring av prosjektstatus er med på å visualisere progresjonen innad i prosjektet. Dette fører til at prosjektlederen kan gjøre enda mer produktive valg på vegne av prosjektet.	

Figur 1 - Brukerhistorie

3.4 Design brukergrensesnitt

Før utviklingsteamet kunne starte var det viktig å utarbeide et godt grunnlag for forståelse om hvordan applikasjonen skulle se ut og fungere. Brukerhistoriene var et godt utgangspunkt for dette, da gruppen ut ifra disse utformet noen krav om funksjonalitet. Videre under et møte angående design, ble gruppemedlemmene enige om at grunndesignet skulle ligne det grunnleggende designet som Microsoft Teams fører. Tanken med dette var at applikasjonen og dens funksjoner skulle være lett gjenkjennelig for brukere av Teams, og derfor være enklere å forstå seg på. Dette fører til at innføringen av nye programmer og programvarer krever mindre arbeidskraft, og det blir dermed lavere terskel for å oppgradere nødvendig teknologi. Dette handler om brukervennlighet og bygger på kjente designprinsipper som omhandler lærbarhet og gjenkjennelighet. Brukervennlighet er viktig da det refererer til hvor enkelt og behagelig et design er å bruke (Benyon, 2013, s. 81).

Gruppen har valgt å anvende noen av de 12 designprinsippene til Benyon. Disse er delt opp i fire underkategorier, hvor den første kategorien omhandler gjenkjennelighet (Benyon, 2013, s. 86-87). De tre andre kategoriene handler om hvor enkel applikasjonen er å bruke,

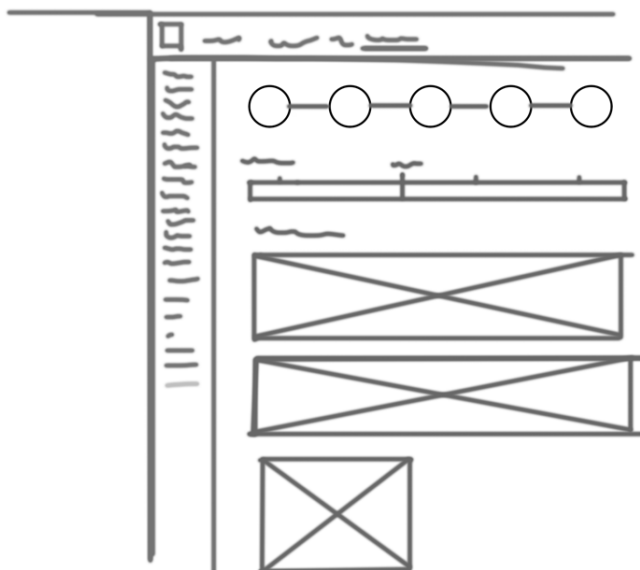
sikkerhet, og tilpasning (Benyon, 2013, s. 86-87). Alle designprinsippene er viktige på hver sin måte. Benyon's designprinsipper er ment til å hjelpe designere gjennom designprosessen, samt kan de brukes til å evaluere og kritisere ideer og tanker rundt designet til prototypen (Benyon, 2013, s. 86).

I følge Benyon er det fire designprinsipper som omhandler lærbarhet (Benyon, 2013, s. 86). Dette er prinsipper som er opptatt av tilgang, enkel læring, og ikke minst det å huske hvordan alt fungerer. Et av designprinsippene i denne kategorien er gjenkjennelighet. Det er et prinsipp som går ut på at brukerne skal være velkjent med språk og symboler som blir brukt (Benyon, 2013, s. 86). De andre designprinsippene under samme kategori handler om at designet skal være forståelig, og at funksjoner skal være synlige (Benyon, 2013, s. 86). Et eksempel på dette er at en knapp skal se ut som en knapp.

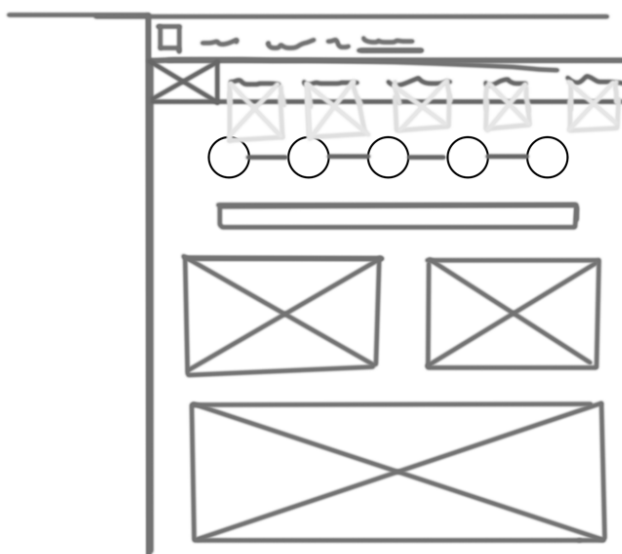
Som tidligere nevnt bestemte gruppen seg for å følge designstandardene til Microsoft Teams og Prosjektportalen 365. For å få til dette ble Fluent UI biblioteket tatt i bruk. Dette er et bibliotek som inneholder flere Microsoft ikoner. I tillegg mottok gruppemedlemmene et dokument fra Puzzlepart under sprintgjennomgangsmøtet i Sprint 1. Dokumentet inneholdt Puzzlepart sitt designsystem «Pins». Dette skal være et dynamisk, åpent og gjenkjennelig system for deres tidligere prosjekter, Prosjektportalen 365, og fremtidige Puzzlepart produkter (Puzzlepart, 2021, s. 2).

3.4.1 Wireframes

For å få en felles forståelse om hvordan hver enkelt funksjon og side i applikasjonen skal fungere, var det viktig å utarbeide skisser og wireframes. Gruppemedlemmene utarbeidet seg alle noen ideer om hvordan applikasjonen skulle se ut og fungere. Det ble laget flere wireframes, som er en vanlig teknikk innenfor brukeropplevelsesdesign (UX) (Benyon, 2013, s. 173). Det er en modell som i hovedsak fokuserer på generelle elementer i designet til applikasjonen, uten å måtte fokusere på detaljer (Benyon, 2013, s. 173). For å utforme wireframes og skisser, ble verktøyet Figma benyttet av gruppemedlemmene. Dette er et nettbasert prototypeverktøy som egner seg godt til å utvikle forskjellige designmodeller. Noen wireframes ble også tegnet for hånd, men Figur 2 og 3 viser to ulike wireframes for hvordan gruppemedlemmene tenkte forsiden skulle se ut.



Figur 2 - Wireframe 1



Figur 3 - Wireframe 2

Gruppen tok i bruk av wireframes for å få en samlet forståelse om hvordan hovedelementene i applikasjonen skulle operere. Ved å gjennomføre slike skisser kan en enklere visualisere seg de generelle funksjonalitetene til applikasjonen. Basert på wireframene var det mulig å lage mer realistiske skisser som igjen ble til stor hjelp under utviklingen av en prototype.

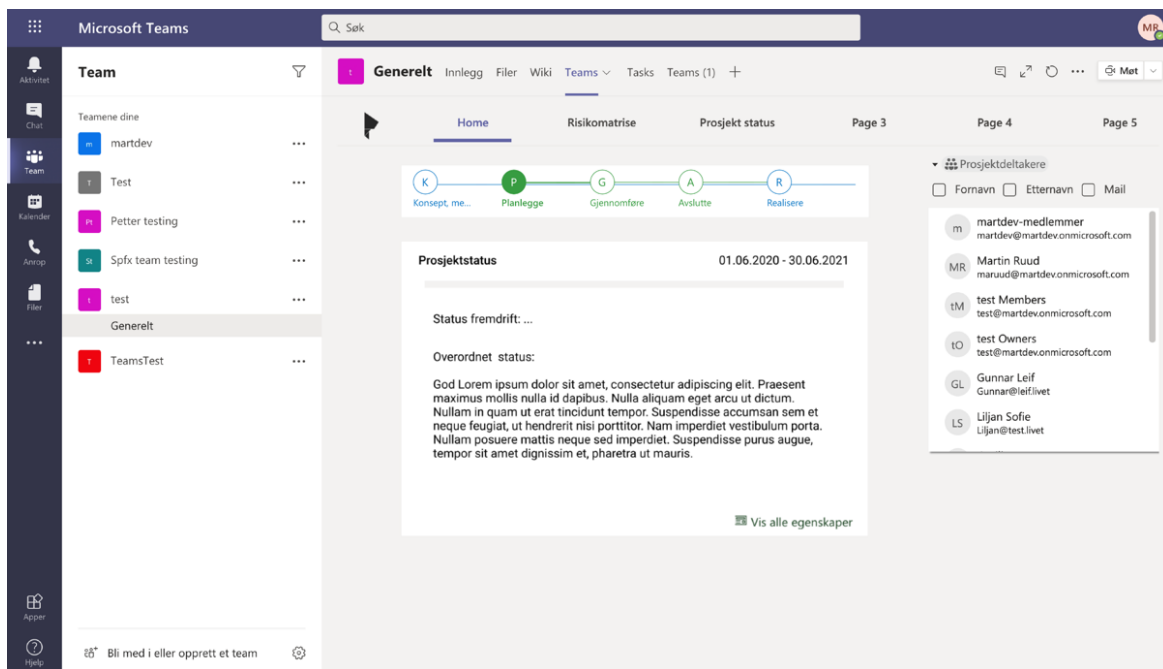
3.4.2 Prototype

En prototype er en konkret, men delvis representasjon eller implementasjon av et systemdesign (Benyon, 2013, s. 175). Prototyper brukes ofte i designsammenhenger, og kan brukes til å demonstrere et konsept (Benyon, 2013, s. 175). Gruppen utviklet en prototype basert på skissene og de wireframes som ble laget tidligere. Det var viktig for gruppen å ha en velfungerende prototype, hvor alle viktige funksjoner og knapper skulle fungere, slik at prototypen skulle ligne det tenkte ferdige produktet. Ut ifra en velfungerende og god prototype er det enklere å utvikle det som skal bli den ferdige applikasjonen. Dette er fordi en da vet hvordan produktet skal se ut, men også hvordan det skal fungere og hvilke oppgaver de forskjellige komponentene skal utføre. Gruppen sin prototype ble laget med verktøyet Figma, og er et resultat av flere skisser satt sammen og tilkoblet via kommandoer for å illustrere gruppen sitt tenkte produkt.

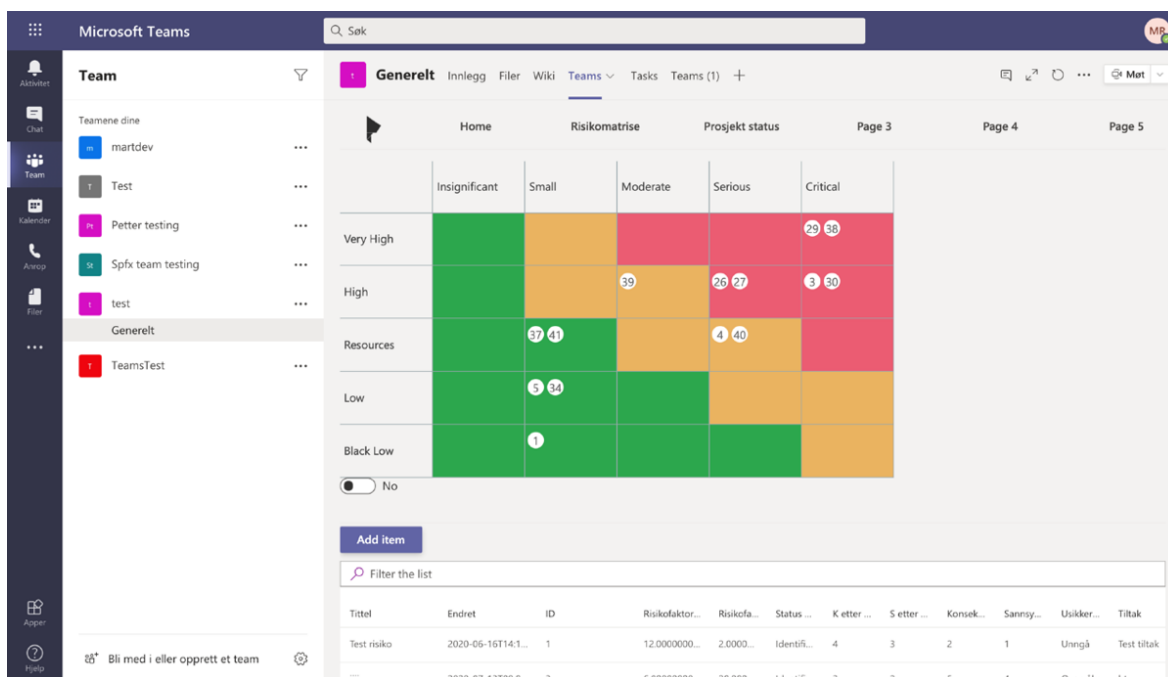
Forsiden blir illustrert med Figur 4. Dette er slik gruppen i første omgang tenkte seg en forside ville se ut, med en dynamisk navigasjonsbar på toppen, og en fasevelger under denne. En liste med deltagende medlemmer til høyre virket praktisk, samt en oppsummering av prosjektstatus på midten, mens Figur 5 viser risikomatriksen til det gitte prosjektet.

Prototypen ble utviklet tidlig i prosjektet for å sette noen klare føringer på hvordan designet skulle se ut, og hvilke komponenter som skulle være med. Figur 4 illustrerer hvordan navigasjonsbaren antagelig ville se ut med hjemknapp, risikomatrikse og lignende, som følger designstandarden til Microsoft Teams. Figur 5 illustrerer oppsettet av risikomatriksen og hvordan gruppen tenkte å implementere denne slik at den er i henhold til designstandarden til Microsoft, samt holder på designet til Puzzlepart.

Under utformingen av produktet endret designet seg noe. Flere komponenter og funksjoner ble implementert, som gjorde at det endelige produktet og prototypen derfor ble ulike. Likevel forble essensen fra prototypen og de viktigste komponentene, da det var grunnsteinen i prosjektet.



Figur 4 - Forside prototype



Figur 5 - Risikomatrise prototype

3.5 Arkitekturdesign

For å oppnå en god teknisk arkitektur har vi valgt å anvende et utvalg av design kriteriene til boken Object Oriented Analysis & Design. I Tabell 1 fremvises de kriteriene som har vært mest utslagsgivende for vårt prosjekt.

Tabell 1 – Prioritering av designkriterier

Kriteria	Veldig viktig	Viktig	Mindre viktig	Irrelevant	Enkelt å gjennomføre
Brukbarhet	X				
Sikkerhet	X				X
Effektiv			X		
Korrekt			X		
Pålitelig	X				
Vedlikehold		X			
Testbar			X		
Fleksibel	X				
Forståelig		X			
Gjenbrukbar		X			
Portabel				X	
Interoperabilitet				X	

Det første kriteriet som er prioritert høyt er brukbarhet. Kriteriet er i hovedsak delt i to synspunkter, brukernes behov og den tekniske plattformens behov (Mathiassen, et al., 2018, s. 182). I applikasjonen prioriteres brukeren sitt behov, da vi mener det er en selvfølge at det tekniske skal tilpasse seg brukeren sin opplevelse av systemet. Et eksempel på dette er alle listevisninger og former som tilpasser seg etter organisasjonens egendefinerte og forhåndsdefinerte lister i SharePoint. Dette gir muligheten for organisasjonen å skreddersy applikasjonen etter sine behov. Ved å prioritere brukerne som i dette eksempelet, vil en få en mer kompleks kodebase enn om en hadde prioritert i motsatt rekkefølge. Da ville brukerne ikke hatt muligheten til å skreddersy Prosjektportalen Teams, og de ville endt opp med å kun ha tilgang til standard oppsett.

Det andre kriteriet som er høyt prioritert er sikkerhet, og omhandler hvem som har autorisering til å lese og skrive data i programmet (Mathiassen, et al., 2018, s. 182). Dette er kategorisert som veldig viktig og enkelt å gjennomføre. I enhver applikasjon med sensitiv data på Internett burde sikkerhet være høyt prioritert. Konsekvensene er store ved datalekkasje, spesielt når organisasjoner lagrer personlig data. Grunnen til at dette er enkelt å gjennomføre er til dels grunnet SharePoint Framework og Web Parts sin natur. I dette miljøet ligger tjeneren i organisasjonen sin SharePoint Tenant, noe som betyr at den er adskilt fra ekstern tilgang. Dette sikkerhetsrammeverket blir vedlikeholdt og driftes av Microsoft som er en aktør med høyt fokus på sikkerhet (Microsoft, 2018).

Det tredje kriteriet med høy prioritering er pålitelighet, som er at systemet er til å stole på (Mathiassen, et al., 2018, s. 182). En bruker skal aldri være bekymret for at informasjonen en legger til i en liste ikke blir synkronisert med databasen. En skal heller ikke være usikker på om dataen som blir lastet inn i brukergrensesnittet er feil. Dette kan føre til at beslutninger i store prosjekter blir tatt på feil grunnlag, som igjen resulterer til store økonomiske og ressurskrevende kostnader.

Det siste kriteriet som er høyt prioritert er fleksibilitet. Kriteriet omhandler innsatsen en må gjøre for å utføre endringer i kodebasen uten for store kostnader (Mathiassen, et al., 2018, s. 182). Dette er høyt prioritert ettersom vi har hatt som mål å utvikle et rammeverk som senere skal kunne videreutvikles. Om det ikke hadde vært fleksibelt ville det vært krevende å utføre endringer uten å påvirke fungerende funksjonalitet. Derfor har vi valgt å prioritere løs coupling og høy cohesion, som er løst med blant annet elementer fra funksjonell programmering, som baserer seg på isolerte funksjoner uten side effekter (Stokke, 2019).

Vi har valgt å sette to av kriteriene som irrelevant, portabilitet og interoperabilitet. Felles for disse er at de omhandler at applikasjonen kan kobles og migreres til eksterne plattformer (Mathiassen, et al., 2018, s. 182). Dette vektlegges som irrelevant ettersom applikasjonen er dypt integrert i Microsoft økosystemet og består av en totalt annerledes arkitektur enn de fleste andre plattformer. Ulempen med dette er at Microsoft dikterer store deler av miljøet og kan gjøre endringer som vi må tilpasse oss etter.

4. Prosjekthåndtering

For å jobbe effektivt gjennom utviklingen av Prosjektportalen Teams er det nødvendig å definere rammene for arbeidet, hvordan prosjektet skal håndteres. Disse rammene er vesentlig å definere tidlig i prosessen, og store deler ble fastsatt under planleggingen av prosjektet. Gruppen valgte å bruke det agile rammeverket Scrum som er svært hyppig brukt i utviklERMILJØER. Ved bruk av Scrum får gruppen sikret fremdrift i prosjektet gjennom hele semesteret, da rammeverket sørger for kontinuerlig møter internt, samt med veileder og bedrift, god planlegging og refleksjon. Det er også nødvendig med god kommunikasjon mellom gruppemedlemmene for å sikre fremdriften, det falt dermed naturlig for medlemmene å bruke de to kommunikasjonsplattformene Microsoft Teams og Discord. For å arbeide mest effektivt for å oppnå kvalitet, er det også viktig å avdekke risiko og hvordan det kan håndteres ved hjelp av en risikoanalyse.

4.1 Kommunikasjon og arbeid

Under utviklingen av dette prosjektet har gruppen brukt flere ulike verktøy for å optimalisere arbeidsfordelingen, kommunikasjonen og samarbeidet i gruppen. Verktøy som Microsoft Planner, Microsoft Teams og Discord ble alle brukt for å sikre en arbeidsflyt i gruppen, noe som er viktig for å oppnå ønsket utfall i prosjektet.

4.1.1 Azure DevOps og Microsoft Planner

Under pre-sprinten ble gruppen enig med samarbeidspartneren Puzzlepart at Azure DevOps skulle bli tatt i bruk, slikt at også bedriften kunne følge med på arbeidet i sprintene. Gruppen valgte senere å gå bort i fra dette planleggingsverktøyet da rettighetene vi hadde fått tildelt begrenset bruken. Begrensningene gjorde det blant annet vanskelig å timeføre og få oversikt over fremgangen i arbeidet.

Gruppen valgte dermed å ta i bruk Microsoft sin planleggingstjeneste Planner for å sette utviklingen i system og utarbeide Produkt backlog og Sprint backlog. Microsoft Planner ble valgt, da den leverer den funksjonaliteten som ønskes og gruppen får sentralisert sitt arbeid på færre plattformer. Dette er en tjeneste som en kan legge inn som en fane inne i en gruppe på Teams. Et slikt planleggingsverktøy ble tatt i bruk for å sikre fremdrift i prosjektet, der gruppemedlemmene kan estimere tid per oppgave i backlogen, og timeregistrere arbeidet.

4.1.2 Microsoft Teams og Discord

For enkel dokumenthåndtering og mulighet for videokommunikasjon valgte gruppen å benytte seg av kommunikasjonsplattformen Microsoft Teams. Dette er en kommunikasjonsplattform som alle medlemmene var kjent med fra tidligere, samt en plattform som Puzzlepart aktivt bruker. I dagens samfunn er det essensielt å kunne gjennomføre møter digitalt, noe Microsoft Teams tilbyr. Plattformen ble derfor brukt til møter for gruppen, men også møter med Puzzlepart. For enkel dokumenthåndtering ble alle dokumenter opprettet og lagret i Microsoft Teams, dette gjorde det også mulig for flere medlemmer å jobbe parallelt med samme oppgave. Microsoft Teams er med på å tilrettelegge for et godt samarbeid, gir god oversikt over dokumenter, samt bidrar til å skape enkel kommunikasjon internt i gruppen og med Puzzlepart.

For en mer daglig kommunikasjon over chat, valgte gruppen å ta i bruk Discord, som er svært kjent i utviklingsmiljøer. Discord er en kommunikasjonsplattform som tilbyr muligheten til å snakke med andre både gjennom tekst og video. Gruppen valgte å ta i bruk plattformen da den er passende til daglig kommunikasjon, og den har funksjonalitet som gjør at ulike samtaleemner kan inndeles i ulike kanaler i gruppen. Dette gjør at gruppen blant annet har en «skole»-kanal og en «fritids»-kanal der all kommunikasjon utenfor skole kan foregå. Discord kan også kobles opp mot Github, gruppen har derfor en kanal med «Githubpushes», som gir alle medlemmene en god oversikt over nye endringer i koden.

4.1.3 Arbeidslokasjon

For å jobbe så produktivt som mulig var det viktig for gruppen å finne et sted der vi kunne møtes fysisk og arbeide sammen. Pandemien gjorde i midlertidig dette til en krevende oppgave, og flere lokasjoner måtte brukes gjennom prosjektet, der Microsoft Teams ble en viktig plattform. Utenom videomøter møttes gruppen blant annet på skolen og hos Puzzlepart for å jobbe sammen. Covid-19 begrenset også hvor ofte vi kunne møtes på skolen, da antall grupperom var redusert som førte til stor konkurranse om dem. Arbeid på kontorene til Puzzlepart tilrettela for et tettere bånd med bedriften, der gruppen fikk mulighet til å bli kjent med flere av de ansatte. Gruppen prøvde å gjennomføre så mange møter som mulig fysisk, slik at en blant annet enkelt kunne drive med parprogrammering.

4.2 Scrum

For å kunne optimalisere produktutviklingen på best mulig vis valgte gruppen å innføre deler av det anerkjente prosessrammeverket Scrum under gjennomføringen av prosjektet. Denne måten å jobbe med et prosjekt på har gruppen tidligere erfaringer med, og mener er et rammeverk som egner seg godt for slike typer prosjekter. Grunnen til dette er at arbeidsformen åpner opp for å oppnå raske resultater, jevnlig tilbakemelding på produktet, og til slutt bedre produktkvalitet da modellen innebærer kontinuerlig integrasjon og testing (Microsoft, 2019). Dette var faktorer som hadde stor innvirkning på valg av arbeidsmodell. I tillegg har gruppen bare hatt positive opplevelser av denne måten å håndtere prosjekter på, som også var en betydelig faktor for valget.

Scrum er en agil tilnærming utviklet for å kunne støtte komplekse tjenester og produkter (Nes, 2019). Den baserer seg på samtlige produktutviklingsprinsipper introdusert av Ikujiro Nonoka og Hirotaka Takeuchi i 1986 (Rubin, 2012, s.3). De sammenlignet denne måten å jobbe på med sporten rugby, der spillerne jobbet oppimot et felles mål, og hvor det er mulighet for å kunne starte om og gjøre endringer underveis (Rubin, 2012, s.3). I stedet for langsiktige planer, forutsetter Scrum at man jobber i små og selvstendige grupper over korte iterasjoner der kontroll over resultater kommer på løpende bånd (Högstrand, 2019). Det er med andre ord en arbeidsmodell som er fleksibel, i motsetning til fossefallmodellen som krever nøye planlegging, og der planene er rigide.

4.2.1 Sprint

En Sprint er i Scrum en iterasjon eller en syklus der teamet jobber med forhåndsbestemte oppgaver valgt i sprintplanleggingsmøtet (Rubin, 2012, s.20). Arbeidet som blir gjort under hver Sprint skal ha en konkret verdi for enten brukeren eller kunden (Rubin, 2012, s.20). Hver Sprint er estimert på tid slik at de alltid skal ha en fast dato for start og slutt, og dette skal i utgangspunktet være det samme for alle sprintene (Rubin, 2012, s.20). Tidlig i prosjektet ble gruppen enig om at hver Sprint skulle ha en timeboxing-lengde på tre uker. Gjennom erfaring fra tidligere prosjekter mener vi dette er tilstrekkelig med tid til å kunne ferdigstille enhver oppgave i sprintene. Ved at hver Sprint har en definert tidsbegrensning, hjelper det gruppen til å fokusere på å få unnagjort oppgaver med verdi. Det har også den fordel at det hjelper oss med å visualisere progresjon i den gjeldende sprinten. Vi har alle på gruppen erfart fra tidligere emner at oppgaver som har en spesifisert sluttdato, har en høyere sannsynlighet til å bli gjennomført. Frister er med på å oppfordre hvert enkelt gruppemedlem

til å fullføre arbeidet sitt i tide. Uten en fastsatt sluttdato, har en lett for å miste motivasjon til å gjennomføre oppgavene, og oppgavene blir gjerne utsatt. Det var derfor viktig for gruppen å tidsbegrense enhver Sprint, for å kontinuerlig sikre fremdrift i prosjektet.

4.2.2 Daglige møter

Vanligvis skal det i løpet av en Sprint holdes daglige møter mellom medlemmene i Scrum Teamet. Møtet er normalt satt til femten minutter eller mindre, der gruppemedlemmene informerer og gir en status på hva hvert enkelt medlem har gjort siden sist daglig møte, og hva de tenker å gjøre frem til neste. (Rubin, 2012, s.23). Dette er noe gruppen har valgt å ikke følge til punkt og prikke. Vi ble enige om at et møte annenhverdag holdt og at daglige møter ikke var av nødvendighet. Dette fordi møter for ofte vil kunne gå utover gruppen sin produktivitet, da prosjektet er tidkrevende og det enkelte medlem av gruppen trenger tid for å ha ferdig noe konstruktivt til gruppemøtene. Det gruppen kan risikere ved å ikke ha daglige møter er at gruppemedlemmene ikke er koordinerte, samt at problemer og hindringer som oppstår hos den enkelte ikke blir kommunisert til resten av gruppen. Selv om de korte daglige møtene er i mange tilfeller å foretrekke, var gruppen ikke redd for at valget om møter annenhverdag ville påvirke kommunikasjons- og arbeidsflyten innad i gruppen.

Vi lot oss derimot inspirere av å stille spørsmål som går ut på fremgang, progresjon og videre plan for prosjektet. Vi starter hvert møte med at enhver på gruppen svarer på spørsmål som hva vedkommende oppnådd siden sist møte, hva planlegger personen å gjøre frem til neste gang, og til slutt om vedkommende har støtt på noen problemer eller hindringer under arbeidet. Bakgrunnen for disse spørsmålene er at gruppen skal kunne få en felles forståelse. Ved å legge til slike spørsmål hjelper det alle gruppemedlemmene til å kunne se det store bildet. Å holde regelmessige møter er helt nødvendig for å kunne hjelpe utviklingsteamet med å beherske arbeidsflyten i en Sprint.

4.2.3 Scrum Team

Et Scrum Team består av tre roller: produkteieren, utviklere og Scrum Master. Hver en av disse har forskjellige oppgaver og ansvarsområder. Førstnevnte er ene og alene ansvarlig for å bestemme hvilken funksjonalitet og egenskaper som skal implementeres, og i hvilken rekkefølge (Rubin, 2012, s.15). Med utviklere menes det blant annet ulike jobbtyper som arkitekt, database administrator, tester, programmerer og designer (Rubin, 2012, s.16).

Utvikler teamet er en kryss-funksjonell og selvorganisert gruppe som er ansvarlig for å bygge, designe og teste produktet (Rubin, 2012, s.16). Den sistnevnte rollen har som oppgave å opptre som en støttespiller, der vedkommende motiverer teamet til å kunne utnytte ferdighetene sine til det ytterste (Rubin, 2012, s.16). I denne sammenheng er det Puzzlepart som har rollen som produkteier. Det å være en Scrum Master vil si at vedkommende skal holde informasjon synlig for alle parter i prosjektet, og skal være til hjelp på de områdene det trengs. I vårt tilfelle var det naturlig at valget på Scrum Master ble tildelt Martin. Vi mener han har egenskaper som kvalifiserer han til denne rollen, blant annet gode kommunikasjons- og forhandlings evner, og ikke minst gode lederegenskaper. Sammen med resten av gruppen, er alle tildelt rollen som utviklere. Som utviklere har vi ikke valgt å tildele noen spesifikke roller, men alle er med å bidra når det kommer til programmering, testing og design. Dette valget ble tatt bevisst, da vi ønsket at alle skulle kunne lære og ikke minst sørge for at gruppen fikk en felles forståelse av prosjektet.

4.2.4 Produkt Backlog

En Produkt backlog er en liste over funksjonaliteter som er prioritert og estimert (Rubin, 2012, s.18). Det er som regel produkteieren og resten av Scrum Teamet som avgjør hvilke funksjoner som skal være under utvikling til enhver tid (Rubin, 2012, s.18). I listen skal det fremkomme hva slags funksjonaliteter som er av høy prioritering, men også de av mindre betydning. Gruppen så på det som nødvendig å utforme en Produkt backlog. Dette fordi en slik liste hjelper gruppen med å få et overordnet bilde på prosjektet på en ryddig og kontrollert måte, samt tydeliggjør for de viktigste elementene som skal bli unnagjort i prosjektet. Produkt backloggen er basert på brukerhistoriene noe som gjør at gruppen får innsikt i brukerne sine behov knyttet til Prosjektportalen Teams.

4.2.5 Sprint Backlog

En Sprint backlog er en liste som inneholder ett sett med detaljerte oppgaver, utarbeidet under sprintplanleggingsmøte (Rubin, 2012, s.18). Listen skal beskrive hvordan gruppen har tenkt å bygge, teste, designe og integrere de ulike delene valgt ut ifra Produkt backloggen for den gjeldende sprinten (Rubin, 2012, s.18). Til forskjell fra Produkt backloggen, er en Sprint backlog av mindre omfang, samt mindre dynamisk. Når gruppen har ferdigstilt en Sprint backlog, kan den ikke endres på. Oppgavene som ikke fullføres under denne Sprinten, flyttes tilbake til Produkt backloggen.

Gruppen utarbeidet en Sprint backlog på starten til den påfølgende sprinten. Det ble hentet elementer fra Produkt backloggen, som ble brutt ned til mindre deloppgaver. Dersom noen oppgaver ikke ble fullført, håndterte gruppen dette ved å videreføre dem til neste Sprint. Om gruppen hadde resterende tid til overs, hentet vi elementer fra Produkt backloggen istedenfor å avslutte sprinten tidlig. Dette gjorde det enklere å planlegge sprintgjennomgangsmøter med Puzzlepart.

4.2.6 Sprintplanleggingsmøte

I sprintplanleggingsmøtet skal utviklingsteamet, produkteier og Scrum Master definere hva de viktigste underpunktene skal være i en Sprint backlog for den kommende sprinten (Rubin, 2012, s.21-22). Målet med møtet er å kunne reflektere over hva man ønsker å oppnå for sprinten. Under dette møtet skal utviklingsteamet bestemme seg over hva som skal være høyst prioritert i backloggen, og hva som er mest realistisk å gjennomføre (Rubin, 2012, s.21-22). Gruppen har på starten av hver Sprint valgt ut elementer fra Produkt backloggen, som videre igjen ble brutt ned til mindre deloppgaver. Disse deloppgavene ble plassert i Sprint backloggen og tidsestimert.

4.2.7 Sprintgjennomgangsmøte

Sprintgjennomgangsmøtet er et møte der Scrum Teamet sammen med produkteieren, ledelsen og andre interessenter gjennomgår et fungerende produktinkrement. Målet med dette møtet er å sørge for at alle interessenters forståelse for produktet eller systemet har økt, og for å kunne forsikre om at gruppen er på samme bølgelengde (Nilsen, 2008, s.11). Gruppen gjennomførte Sprintgjennomgangsmøtet i slutten av hver Sprint sammen med Puzzlepart. Her fikk vi gode råd og tilbakemeldinger som gruppen kunne dra nytte av. I tillegg hadde vi et separat møte med vår veileder, der vi gikk gjennom arbeidet som ble utført i den gjeldende sprinten. Under disse møtene holdt vi en produktdemonstrasjon for å vise hvordan produktet fungerte på en innbyende måte. På denne måten sørget vi for at både produkteier og veileder kunne komme med tilbakemeldinger på produktet, samt var dette med på å bygge tillit mellom gruppen og de ulike partene.

4.2.8 Sprintretrospektiv

Sprintretrospektiv er et møte som foregår innad i gruppen, og er det siste en gjør i en Sprint. Det finner sted etter sprintgjennomgangsmøtet og før neste Sprint. I motsetning til sprintgjennomgangsmøtet, blir det under dette møtet fokusert på å inspisere hele prosessen, og ikke bare produktet. (Rubin, 2012, s.27-28). Målet med møtet er at i slutten av sprinten skal gruppen identifisere forbedringer som vil bli implementert til neste Sprint (Rubin, 2012, s.28). Gruppen syntes det var lurt å holde sprintretrospektmøter, ettersom at slike møter er med på å lage en oversiktlig plan der både kvalitet og effektivitet økes. Her blir også sprinten grundig inspisert og evaluert av gruppen, og eventuelle feil vil dermed kunne oppdages raskt. Under dette møtet var det viktig for gruppen å diskutere punkter som, hva var det som gikk bra i sprinten, hva kan bli forbedret og hva kan vi fokusere enda mer på til neste Sprint.

4.2.9 Sprintgjennomføring

Pre-sprint

I denne fasen jobbet gruppen med å forberede seg når det kom til anskaffelse av data, design, sette opp prosjektet, og finne teknisk arkitektur som skulle ligge til grunn for prosjektet. Som metode for datainnsamling valgte gruppen å benytte seg av spørreundersøkelse. Valget falt på denne datainnsamlingsmetoden ettersom en slik tilnærming egner seg godt for å få et stort antall svar fra mange enheter (Jacobsen, 2018, s. 137). Vi ønsket å få et bredt bilde på prosjektet, fremfor å gå i dybden. Dette fordi Prosjektportalen 365 er en eksisterende plattform. Risikoen med å velge bort en kvalitativ metode som intervju er at nyanser bli begrenset av satte spørsmål og svaralternativer, da nærheten med undersøkelsesobjektet forsvinner (Repstad, 2007, s.17). En annen ulempe er at spørreundersøkelsen er svært lite fleksibel sammenliknet med et intervju, og spørsmålene stilt i skjemaet kan bli misforstått og føre til at vi ikke får samlet inn den informasjonen vi ønsker (Repstad, 2007, s.18). Når vi hadde fått inn nok informasjon, utformet vi flere brukerhistorier. Brukerhistorier er et format som har til hensikt å beskrive funksjonalitet (Mathiassen, et al., 2018, s.122). Formatet bidrar med å gi en felles forståelse til alle partene i prosjektet, og er et hjelpemiddel mye brukt i smidig utvikling. Sammen med brukerhistoriene, valgte vi å anvende prioriteringsteknikken «MoSCoW». Dette er en teknikk som brukes for å prioritere krav, og er med på å sikre at de mest kritiske funksjonene blir høyt prioritert (Varma, 2015, s.97).

Etter utforming av både brukerhistorier og «MoSCoW» valgte gruppen å lage flere mockups, som er en interaktiv prototype. Dette gjorde vi for å kunne visualisere designet på prosjektet. Da vi skulle sette opp prosjektet startet vi med å migrere komponenter fra SharePoint-versjonen av Prosjektportalen 365. Etter mye frem og tilbake kom vi deretter frem til at komponentene kunne lastes opp som bibliotekpakker. Dette for å kunne redusere sjansen for kodeduplisering, og for å kunne holde tritt med versjonene. I denne perioden valgte vi også å bytte fra planleggingsverktøyet Clickup til Azure DevOps, etter anbefaling av Puzzlepart. Dette på grunnlag av at de selv bruker tjenesten og ønsket å ha mulighet til innsyn i gruppen sin fremgang.

Sprint 1

Under Sprint 1 var hovedfokus å starte å utvikle funksjonalitet, derav navigeringsbaren og listefremvisninger. Timeestimering under denne sprinten var ikke optimal, mye på grunn av skiftet fra Clickup til Azure DevOps. Dette var hovedsakelig fordi gruppen ikke fant noen gode løsninger for timeføring, men også at vi var ikke konsekvente nok på akkurat det området. Gruppen ble dermed enig om at timeestimering var noe som måtte forbedres til den påfølgende sprinten. Sprinten startet med at gruppen satte opp oppgaver i Sprint backloggen, men ikke alle oppgavene ble gjennomført. Estimeringen av hvor mye gruppen har kapasitet til å fullføre på en Sprint ble kraftig overvurdert, samt møtte gruppen på noen hindringer. En av grunnene til denne overvurderingen er at gruppemedlemmene ikke hadde tidligere erfaring med å utvikle i SharePoint. For flere av medlemmene var utviklingsmiljøet med Typescript og React helt nytt, dette førte til at tid måtte brukes for å bli kjent med miljøet. Utviklingen tok lengre tid ettersom at flere av funksjonene vi konsentrerte oss om under sprinten ble bestemt at skulle være dynamiske. Utvikling av dynamiske funksjoner er tidkrevende å utvikle, da de er mer komplekse. Oppgavene som ikke ble gjennomført i denne sprinten, ble videreført til neste Sprint. Før vi kunne avslutte Sprint 1, ble et sprintgjennomgangsmøte utført sammen med Puzzlepart. Vår veileder var ikke til stede under dette møtet, og derfor holdt vi et separat Sprintgjennomgangsmøte med vedkommende.

Under sprintgjennomgangsmøtet med Puzzlepart ble kravene til prosjektet klarert enda tydeligere, både for oss og for samarbeidspartner. Et Sprint retrospektmøte ble til slutt gjennomført, der gruppen ble enig om at timeestimering måtte være på plass til neste Sprint. Tiltaket for å få dette til var å begynne å bruke planleggingsverktøyet Microsoft Planner, da dette verktøyet ikke hadde de samme brukerbegrensningene som Azure DevOps. Gruppen

avsluttet møtet med å se på risikoanalysen utviklet under pre-sprinten, og det ble tatt en vurdering om flere risikoelementer skulle legges til. Gruppen så det ikke som nødvendig å legge til flere risikoelementer på dette tidspunktet.

Sprint 2

Under den andre sprinten ønsket gruppen å få på plass mer funksjonalitet. For å gjøre dette mulig måtte mye tid brukes til å få opp installasjonen av Puzzlepart sin kode. Det å få installasjonene på plass og å få det til å fungere var tidkrevende. Store deler av grunnen til dette var at ulike bibliotekspakker ikke var kompatible med hverandres versjoner. Dette ble løst ved å finne ut av hvilke versjoner som var kompatible.

Selv om gruppen møtte på flere store utfordringer under Sprinten vil vi anslå at 25% av «Must have» funksjonalitetene i Produkt backloggen var på plass etter denne Sprinten. Bytte til Microsoft Planner ga positivt utslag, og førte til at timeestimeringen forbedret seg i denne sprinten. Et sprintgjennomgangsmøte med gjennomgang av produktet ble gjennomført på slutten av Sprint 2 sammen med Puzzlepart. Bedriften var positive til det vi hadde fått til og ønsket å stille seg mer tilgjengelig til å hjelpe de neste sprintene. Til neste Sprint ønsket de samt at hjem-siden til applikasjonen skulle bli satt mer fokus på. Det ble også gjennomført et møte med veileder for å få innspill til prosjektet, samt holde han oppdatert. Under retrospektmøtet ble gruppen enig om å starte neste Sprint med å lage en skisse av hjem-siden til Prosjektportalen Teams, dermed utvikle den og flere av «Must have» funksjonaliteten under neste Sprint. Gruppen avsluttet møtet med å se på risikoanalysen utviklet under pre-sprinten, og la til flere risikoelementer. Risikoelementene som ble lagt til var «Gruppemedlemmene mister motivasjon» og «Dårlig kommunikasjon med Puzzlepart». Disse ble lagt til da gruppen så det som en potensiell fare for fremgangen hvis motivasjonen sank, og gruppen så at det var viktig å opprettholde god kommunikasjon med Puzzlepart for å få konstruktive tilbakemeldinger.

Sprint 3

Målet som gruppen satte seg for denne sprinten, var å få på plass de fleste «must have» og «should have» funksjonene, samt planlegge hvordan hjem-siden til applikasjonen skulle se ut. Gruppen tegnet opp en skisse av hjem-siden, der den ble brukt som en mal på hvordan siden skulle se ut. Skissene ble også sendt til Puzzlepart for å holde dem oppdatert på vårt forslag,

og for å kunne gi dem mulighet til å komme med innspill. For å kunne legge til mer funksjonalitet måtte gruppen gå igjennom kildekoden til Prosjektportalen 365. Her møtte gruppen på lite dokumentert kode, som gjorde at feilsøkingen ble mer krevende enn forventet. Feilsøkingen resulterte i at det ble funnet flere bugs i koden som ble rapportert tilbake til bedriften. Ved å rette opp feilene fikk gruppen mye funksjonalitet på plass i applikasjonen, og var på god vei til å bli ferdige med funksjonalitetene som skulle implementeres. Vi vil anslå at ca. 41% av alle brukerhistoriene i Produkt backloggen ble gjennomført i denne perioden. Vi har også kommet godt i gang med å få til navigering på alle sider med lister, slik det er i eksisterende Prosjektportalen 365.

Under denne sprinten møtte vi på problemer ved implementasjon av typescript komponenter produsert av Puzzlepart. Disse komponentene er opprinnelig laget for å leve i et SharePoint miljø. Dermed kan de oppføre seg uforutsigbart når de kjører i Teams, noe vi fikk erfare.

Videre hadde gruppen et ønske når det kom til finpussing av applikasjonen. For eksempel det å gi brukeren flere tilpasningsmuligheter ved bruk av tabeller. Gruppen ønsket fremover å vektlegge fokus på design og bugs. Slik som det er nå kan for eksempel det å navigere fra en side til en annen føre til at alle spørringer går til feil URL, og i verste fall føre til at programmet krasjer. Under Sprint retrospektmøtet ble vi enige om at dersom det er tilstrekkelig med tid igjen, hadde det vært lurt å gjennomføre brukertesting med bedriften sine kunder. Gruppen ønsket også å fikse bugs for å forbedre brukeropplevelsen av applikasjon.

Sprint 4

I denne sprinten hadde vi som hovedfokus å fikse bugs, refaktorere og skrive på rapporten. Vi holdt på med feilhåndtering, som forhindrer at programmet krasjer i uforutsette bruksmønster. Dette går under defensiv programmering, som sikrer kvalitet, og som er med på å skape en trygghet for brukeren. Under denne sprinten møtte vi på flere småfeil, som for eksempel at visse input-verdier ikke ble korrekt satt inn i databasen. Gruppen hadde stor omtanke for refaktorering, for å gjøre kodebasen mer lesbar og håndterbar i fremtiden. Vi har også gjort en del små endringer når det kommer til henting av data. Tidligere ble dette gjort fra et spesifikt prosjekt, men etter endringer kan programmet innhente data fra alle prosjekter i organisasjonen sin portefølje. Somt nevnt i Sprint 3 ønsket gruppen å gjennomføre brukertesting. Gruppen så ikke på dette som nødvendig på et så tidlig tidspunkt. Dette fordi prosjektet ikke var utviklet nok for at brukertesting hadde vært lønnsomt. Under Sprint retrospektiv ble gruppen enig om å fokusere på design i den kommende Sprinten.

Hovedsakelig fordi det tidligere designet ikke la stor nok vekt på de designprinsippene som fremmer brukervennlighet, skrevet om i Kapittel 3.4. Vi vil anslå at ca. 17% av alle brukerhistoriene i Produkt backloggen ble gjennomført i denne perioden.

Sprint 5

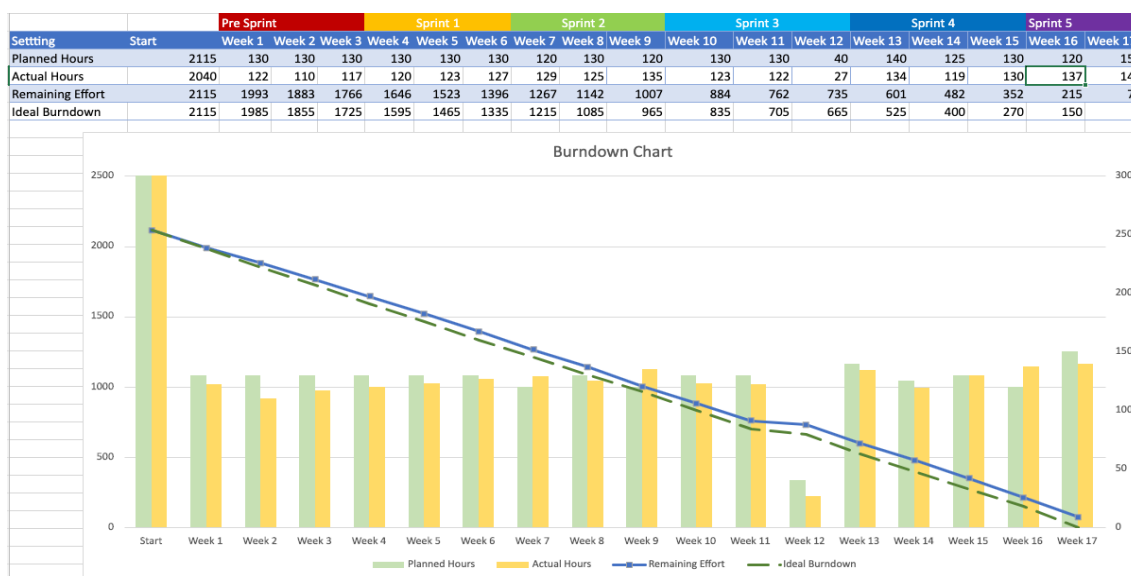
I Sprint 5 la gruppen hovedvekt på design, universell utforming og ferdigstille rapporten. I dette tidsrommet arbeidet vi med å utforme designet, samt implementere det inn i prosjektet. Gruppen hadde et stort fokus på at forsiden skulle være brukervennlig og tiltalende. Med dette menes at forsiden inneholder nå mer prosjektinformasjon som oppsummerer prosjektet på en ryddig og oversiktlig måte. I en av brukerhistoriene fremkommer det et krav om universell utforming. Dette vil si at applikasjonen skal fungere optimalt på flere bruksflater som for eksempel mobil, nettbrett og PC. Etersom kravet var et «must have» ble det gjennomført i denne Sprinten. I denne perioden har også ferdigstilling av rapporten vært hovedfokuset. I sammen med veileder, har vi gjennomgått og fått tilbakemeldinger angående forbedringer i rapporten. Noen av de viktigste endringene som er blitt foretatt omhandler struktur og flyt, begrunnelser av valg, og refleksjon. Sprinten ble avsluttet med et sprintgjennomgangsmøte med oppdragsgiver, der produktet ble presentert for både ledelsen, mellomledelsen, utviklere, og CEO. Vi fikk svært gode tilbakemeldinger, som vi sier oss fornøyd med. Puzzlepart var særlig fornøyd med utviklingen av den universelle utformingen, og hvordan gruppen hadde tilpasset prosjektet til å være dynamisk. Bedriften uttrykte et ønske om å videreføre prosjektet, samt å vise det frem i deres brukerforum. I Sprint retrospektmøtet konstaterte gruppen at dette var den siste fullverdige Sprinten. Gruppen ble enige om at vi ønsket å fortsette utviklingen frem til muntlig eksamen, samt videreføre prosjekthåndteringspraksisen. Vi vil anslå at ca. 8% av alle brukerhistoriene i Produkt backloggen ble gjennomført i denne perioden

4.3 Burndown diagram

Et Burndown diagram er en grafisk fremstilling som viser fremgangen for de ulike sprintene i et prosjekt. Ideelt sett skal diagrammet ha en nedadgående trend, der grafen skal treffe nullpunktet mot slutten av prosjektet (Agarwal & Majumdar, 2012, s. 98-99). Dersom dette ikke er tilfellet, gir det en indikasjon på at enten arbeidsintensiteten må økes, eller at teamet må foreta noen endringer (Agarwal & Majumdar, 2012, s. 98-99). Diagrammet skal representere gjenstående arbeid i mengde, sett i sammenheng med tid i en Sprint.

Gruppen valgte å utforme diagrammet i Excel der medlemmene for hver uke estimerte den antatte tidsbruken. Som vist ved Figur 6 blir timeestimeringen bedre mot slutten da gruppen etter hvert fikk en ytterligere forståelse av prosjektet, noe som gjorde det enklere å beregne nødvendig tidsbruk.

Figur 6 viser en nedadgående trend, noe som vil tilsi at arbeidsintensiteten er tilnærmet lik den estimerte tiden. I slutten av Sprint 3 stiger grafen minimalt ettersom gruppen i uke 12 la opp til påskeferie. Vi ser at dette har hatt en effekt på de planlagte timene, noe som gjør at grafen ikke treffer nullpunktet på aksene. En annen faktor for at grafen ikke treffer nullpunktet skyldes underestimering. Diagrammet har hjulpet gruppen gjennom prosjektet med å gi et klart bilde på hvor mye tid til rådighet til de resterende sprintene. Samt har det hjulpet medlemmene med å reflektere og forbedre estimering av tid.



Figur 6 - Burndown chart

4.4 Risikoanalyse

For å kartlegge og beskrive risiko som kan påvirke prosjektarbeidet, samt risiko som kan være avgjørende for å hindre måloppnåelse, ble det i pre-sprinten gjennomført en risikoanalyse. En risiko er sammensatt av komponentene usikkerhet og konsekvens, der en aktivitet kan lede til en fremtidig usikker konsekvens (Aven, 2015, s.13). Det følger flere fordeler i prosjektarbeidet ved å gjennomføre en risikoanalyse, blant annet får gruppen et bilde av hendelser rundt prosjektet, deres konsekvenser og en kan planlegge hvordan en eventuell hendelse skal håndteres (Aven, 2015, s. 3). Gruppen utførte derfor en risikoanalyse for å være forberedt og forebygge mot risiko som kan fremtre under prosjektet. Risikoanalysen gir gruppen en ryddig oversikt over eventuelle hendelser, der alvorlighetsgrad, sannsynlighet og risikonivå blir vurdert, som vist i Figur 7. Aktuelle tiltak for hver hendelse blir utformet for best mulig risikohåndtering.

RISIKO	ALVORLIGHETSGRAD	SANNSYNLIGHET	RISIKOPÅVIRKNING	ANBEFALTE HANDLINGER
En av gruppe medlemmene får corona	Tolerable	Possible	Medium	For å unngå dette kan gruppen være nøye med avtand på fysiske møter, bruke antibac og ikke møte opp ved eventuelle symptomer. Hvi en får corona tar vi møter på Teams, og møtes ikke på en stund for å unngå økt smitte.
En av medlemmene smitter gruppen med corona	Generally unacceptable	Possible	High	Samme som den over. Det er viktig å ikke komme på fysiske møter ved symptomer.
Prosjektet blir slettet	Generally unacceptable	Not likely	High	Ha prosjektet lagret på flere maskiner, i sky og på github.
En av gruppe medlemmene blir syk	Acceptable	Possible	Low	Resten av gruppen jobber som normalt. Den syke blir hjemme og gjør det den orker av arbeid.
Gruppe medlemmene mister motivasjonen	Tolerable	Possible	Medium	Å prøve å ha fysiske møter, være sosiale sammen også utenfor skole.

Figur 7 - Risikomatrise

Gruppen så det som en nødvendighet å utføre en risikoanalyse når verden står ovenfor en pandemi. Covid-19 og dens smittebølger gjør at samfunnet hele tiden er i endring, der alt fra besøksforbud til nedstegning kan bli innført på en og samme dag. En stor risiko gruppe medlemmene også står ovenfor er smitte, som kan føre til alvorlig sykdom. Risikomatrisen vist ved Figur 7 er en oversiktlig fremvisning av risiko funnet ved risikoanalysen, dets risikonivå og tiltak for å unngå og håndtere risikoen. Hele risikomatrisen er fremvist i Vedlegg 10.3. Risikomatrisen vil på slutten av hver Sprint bli vurdert. Eventuell risiko som dukker opp underveis i prosjektet vil kunne tilføyes matrisen under Sprint retrospektmøtene.

4.5 Programmeringsspråk og miljø

Puzzlepart er en bedrift som baserer store deler av sitt arbeid innenfor Microsoft miljøet. Dette innebærer at produktene er utviklet ved hjelp av visse rammeverk og standarder for å hjelpe utviklere med raskere og tryggere utvikling. I gruppen sitt tilfelle er det miljøet innenfor SharePoint, Teams og web utvikling som er relevant og kommer til å dekket i denne rapporten. Standard Teams applikasjoner omtales som «Tabs», og befinner seg som regel i kanaler eller grupper som vist med Figur 8. En tab er i hovedsak ikke mer enn en «Teams-bevisst» nettside som blir vist gjennom en iframe (Microsoft, 2021). Dette vil si at nesten alle nettsider i teorien kan fungere i en Teams tab. Men selv om nettsider eksisterer i en Teams tab betyr det ikke nødvendigvis at de er optimalisert for dette.



Figur 8 - Teams tab

4.5.1 Programmeringsspråk og rammeverk

Programmeringsspråket som blir brukt i dette prosjektet er Typescript. Typescript er et superset av JavaScript, som anses å være mer skalerbart enn Javascript i store applikasjoner og arbeidsgrupper (Typescript, u.å.). Typescript introduserer typer, i likhet med andre statisk-skrevet programmeringsspråk som Java, og C++. I motsetning til JavaScript har du ikke lov til å endre typen til en variabel etter den er satt. Ved å tilføre en slik begrensning i kodebasen reduserer vi mange feil som kunne oppstått under utvikling. Ettersom vi er en gruppe på fem personer, vil ikke alle gruppemedlemmene kjenne til den fullstendige kodebasen. Dette er et kjent begrep innenfor programmering som kalles abstraksjon. Abstraksjon går i korte trekk ut på at man ikke behøver å vite *hvordan* den spesifikke klassen eller funksjonen virker på innsiden, men heller forstå utfallet, og hvordan man skal benytte den i sin egen kode (Barnes & Kölling, 2012, s.63). Typescript forenkler abstraksjon ved at det er obligatorisk å skrive hva slags type parameteret er. Dermed vil et gruppemedlem kunne skrive en metode og et annet gruppemedlem forstå hva slags typer metoden tar imot uten å se på selve innholdet i metoden.

I tillegg til selve språket Typescript, brukes rammeverket React. Det er diskutert om React *egentlig* er et rammeverk eller et bibliotek, men vi velger å kalle det et rammeverk ettersom det fundamentalt endrer strukturen en skriver Typescript frontend kode. React er et verktøy

for å bygge brukergrensesnitt og baserer seg i stor grad på deklarativ syntaks og gjenbrukbare komponenter. Kun komponenter som har endret tilstand, for eksempel et input-felt når en skriver, blir rendret på nytt. Dette fremfor den tradisjonelle måten hvor hele siden blir lastet mellom hver spørring (React, 2021). Som igjen gir muligheten til å skape et grafisk brukergrensesnitt som er raskt og effektivt i bruk.

Som sagt baserer React seg på gjenbrukbare komponenter. Det er med på å skape en skalerbar kodebase som kan vokse uten tett coupling. En komponent kan leve i flere sider, og flere steder på hver av disse sidene. Dette gjør ikke bare at koden er lettere å forstå, men en vil også kunne være agil i større grad enn om man har en tett koblet kodebase. En komponent vil enkelt kunne bli byttet ut med et annet uten at andre komponenter berøres.

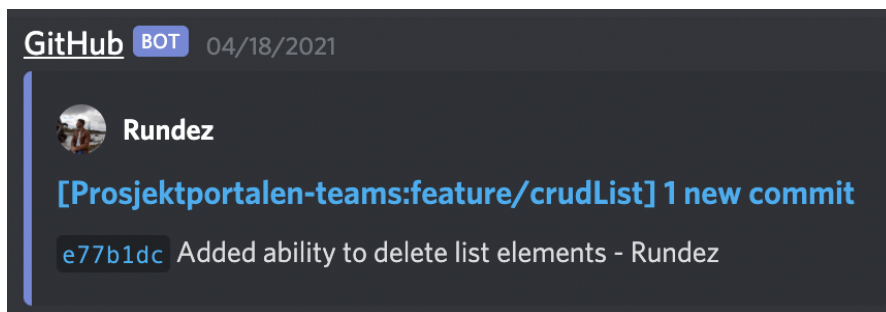
4.5.2 Versjonskontroll

Versjonskontroll har blitt en viktig del av moderne programvare utvikling, spesielt når det er flere utviklere på samme prosjekt. I vårt tilfelle brukes Git og Github for å gjøre det gjennomførbart at fem personer arbeider på samme prosjekt. Flyten vi bruker er «Github flow», som innebærer retningslinjer for hvordan Git og Github skal brukes trygt og strukturert. I praksis vil dette si at vi har en branch som brukes til utvikling, en branch til stabil kode og flere branches til utvikling av funksjonalitet. Om et gruppemedlem skal utvikle funksjonalitet vil personen branche ut av utviklingsbranchen og lage en «feature» branch. På denne branchen legges til nødvendig kode med forklaring. Når medlemmet er fornøyd med sine bidrag lages det en pull request hvor et eller flere andre gruppemedlemmer ser over koden. Hvis koden godkjennes, blir den slått sammen med den opprinnelige branchen.

Ved hjelp av Github Flow metodikken sikrer man kvalitet av koden som implementeres ettersom den tvinger flere av gruppemedlemmene til å se over og godkjenne koden.

Git brukes gjerne via kommandolinjeverktøy som Terminal, men alle gruppemedlemmene har valgt å ta i bruk grafisk brukergrensesnitt. Dette gir en visuell fremvisning av hvordan koden er distribuert blant gruppemedlemmene som er oversiktlig og gjør det enkelt å holde kontroll over fremgang.

Vi har også implementert en webhook i Discord gruppen som er koblet til Github repositoret vårt. Denne gir beskjed og viser commit meldingen når medlemmene pusher og merger kode i kodebasen, slik som Figur 9 viser. Dette gjør at alle får god kontroll over hva som blir jobbet med og får fulgt med selv om det ikke er foretatt noen nylige statusmøter.



Figur 9 - Github webhook poster oppdateringer i kodebase.

4.5.3 Kodestandard og struktur

Mennesker er forskjellige og gjør sjeldent ting på samme måte i dagliglivet. Det samme gjelder i kode, og det er ikke alltid enkelt å forstå hva funksjoner skrevet av andre gjør. En måte vi har håndtert dette på er å innføre kodestandarder. Kodestandarder er visse «regler» en følger når en skriver kode, og som gjør det enklere for andre å tolke det en har skrevet. Vi har først og fremst valgt å ha kodestandarder på relativt enkle «snutter» med kode som å for eksempel skrive funksjonelle komponenter fremfor klassebaserte komponenter, som vist på Figur 10. Dette vil si at en komponent alltid skal lages på denne måten om mulig.

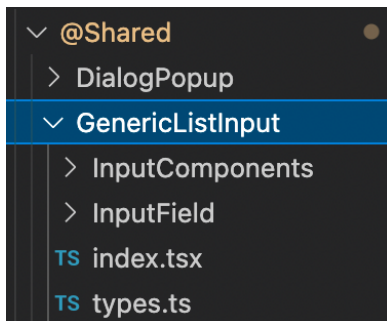
```
/**
 * Et eksempel på funksjonelt komponent
 */
const functionComponent = (props) => {
  return <div>Some code here</div>;
};
```

Figur 10 - Funksjonelt komponent

Vi har også valgt å bruke en «kode formatter» som formatterer kode etter et sett regler. Denne leser koden man skriver og setter den opp på nytt etter ønsket format. Det er viktig å presisere at output ikke vil bli endret, kun hvordan det ser ut. Vi har for eksempel satt opp så koden vil bli indentert med to mellomrom, maksimum linjelengde på ca. 50 karakterer etc.

En kodestandard som er verdt å nevne er bruken av interfaces og mappestruktur. I TypeScript benyttes det som sagt typer, og disse typene må ofte defineres ved hjelp av interfaces. Noen utviklere velger å ha disse i tsx filen som returnerer selve komponenten, og andre velger å ha interfaces og typer i egne filer. Uten kodestandarder ville dette vært et godt eksempel hvor man så at gruppe medlemmene implementerte interfaces på hver sin måte, og det ville endt med unødvendig «leting» for å finne koden man ønsket. Vi benytter oss dermed av

bedriftsstandarden i Puzzlepart for å standardisere bruken av interfaces. Det betyr at en fil er dedikert til typene i hver enkelt mappe som vist ved Figur 11.



Figur 11 - Interface og komponent struktur

5. Kvalitet

For å forsikre at oppdragsgiver får produktet de ønsker er det viktig å konstatere hva som er kvalitet for Puzzlepart i forhold til Prosjektportalen Teams. Basert på kravene er det mulig å optimalisere arbeidsrutiner for å sikre kvalitet. Kravene for kvalitet kan senere i prosjektet brukes som måleenhet ved kvalitetssjekk av produktet. I dette kapittelet vil det fremkomme informasjon om hva kvalitet er, og hvordan gruppen kan sikre kvalitet i prosjektet.

5.1 Hvordan sikre kvalitet?

Kvalitet er et relativt begrep, der krav internt i gruppen og krav fra oppdragsgiver kan utgjøre kvalitet. Det er flere internasjonale organisasjoner som har definert kvalitet, men vi ønsker å ta utgangspunkt i ANSI (American National Standards Institute) sin definisjon: «Kvalitet er totalen av funksjoner og egenskaper til et produkt eller en tjeneste som innehar evnen til å tilfredsstille de gitte behovene.» (Fitzpatrick, 1996, s.6). På grunnlag av denne definisjonen kan vi si at Prosjektportalen Teams har oppnådd god kvalitet, når den har funksjoner og egenskaper som tilfredstiller oppdragsgiver og brukerens krav og behov.

5.1.1 Oppdragsgiver sine krav om kvalitet

Gruppen ønsket å levere et produkt med høy grad av kvalitet, og ble enig om dette allerede fra start. Det er svært viktig for gruppen å utvikle et produkt med høy kvalitet. Dette på grunn av at produktet muligens skal brukes eller videreutvikles av Puzzlepart etter at vårt sluttprodukt blir levert. Ettersom oppdragsgiver investerer så mye tid i gruppen og utviklingen av Prosjektportalen Teams ble det også viktig å kunne gi noe nyttig tilbake.

For å sikre kvalitet tok gruppen flere grep. Gruppen forsto tidlig at det var nødvendig å ha en tett dialog med oppdragsgiver, da de under prosjektet kunne komme med forslag, kunnskap og konkrete tips om endringer. Ved å ha en slik tett dialog kunne gruppen forsikre seg om at det ble utviklet et produkt som stod til Puzzlepart sine forventninger, et produkt med de funksjonaliteter de ønsket.

Det ble tidlig kjent at Puzzlepart hadde visse krav til produktet gruppen skulle utvikle. Under et møte med bedriften fikk gruppemedlemmene lov til å komme med egne forslag og innspill, og til slutt ble alle enige om visse krav for applikasjonen. Kravene satte noen rammer for prosjektet, men holdt likevel oppgaven relativt åpen i starten. Det første kravet var at prosjektet skulle legges i et eget «repository» på Github. De ønsket at gruppen skulle jobbe

med åpen kildekode. Det neste kravet var at applikasjonen skulle bli laget som en Microsoft Teams fane, og gi en oppsummering av de viktigste elementene på prosjekter fra Prosjektportalen 365. Applikasjonen skal også være gratis tilgjengelig, og fungere på flere flater, som på desktop, nettbrett og mobil.

Etter gjennomføring av en spørreundersøkelse rettet mot brukere av Prosjektportalen 365, kom det frem hva brukerne selv ønsket at applikasjonen skulle inneholde. De funksjonalitetene som viste seg å være mest populære, samsvarte godt med hva Puzzlepart ønsket. Denne enigheten mellom Puzzlepart og brukerne av Prosjektportalen 365 gjorde det enklere for gruppen å sette klare rammer for prosjektet. Basert på svarene fra spørreundersøkelsen og kravene fra Puzzlepart tok gruppen i bruk av MoSCoW teknikken for å prioritere hvilke komponenter som var viktigst å få implementert. Denne prioriteringsteknikken hjelper med å sikre kvalitet på den måten at gruppen har god oversikt over hvilke komponenter som er nødvendige, og hvilke som burde være med. Dette legger til grunne for god planlegging og strukturering av produktet som legger et godt grunnlag for videre utvikling og kvalitetssikring.

5.1.2 Sikre kvalitet i kildekoden

For å sikre kvalitet i kildekoden var det noen spesifikke punkter gruppemedlemmene så på som viktige. Etersom Prosjektportalen 365 er et eksisterende produkt, som Puzzlepart ønsker å kunne bruke gjennom Microsoft Teams, vil det være nødvendig å følge deres kodenstandard. Dette gjør det enklere for bedriften å videreutvikle applikasjonen senere.

For at kildekoden skal være mest mulig brukbar, var det viktig at gruppens utviklere tenkte på både coupling og cohesion for å sikre kvalitet. Hele programmet er komponentbasert, og det er dermed så løs coupling mellom komponentene at de blir gjenbrukbare. Slike uavhengige komponenter kan dermed gjenbrukes opp til flere steder uten at det oppstår problemer.

For å utvikle en Microsoft Teams applikasjon som fungerer sammen med Puzzlepart sin Prosjektportalen 365, var det være nødvendig å gjenbruke en god del av deres kildekode. På grunn av dette ble minst mulig i koden hardkodet, og målet var å holde det så dynamisk som mulig. Dette vil si at hvis det blir gjort en endring i den originale koden til Prosjektportalen 365, vil denne endringen bli gjenspeilet i gruppen sin kode uten at noen endringer må bli gjort.

5.1.3 Arbeidsrutiner for å sikre kvalitet

Gruppen opprettet et felles repository på Github hvor all kode ble lastet opp og lagret, og er tilgjengelig for alle medlemmene. Dette var et viktig grunnsteg for å sikre kvalitet i utviklingsprosjektet. Github er en plattform som bidrar til et felleskap hvor åpen kildekode er i fokus. Åpen kildekode vil si at programvaren er gjort tilgjengelig for alle (Fuggetta, 2003, s. 79). De generelle reglene for åpen kildekode tilsier at alle program publisert med åpen kildekode vil være fullt tilgjengelige for hvem som helst (Fuggetta, 2003, s. 78- 79). Dette innebærer også at andre har mulighet til å se på koden, gjøre endringer, og å komme med forslag til forbedringer for å fikse eventuelle bugs (Fuggetta, 2003, s. 78-79). Et av kravene til Puzzlepart var at gruppen skulle utvikle med åpen kildekode. Dette bidro til at de lettere kunne følge med på hvordan gruppen jobbet, samt hadde lettere tilgang til å hjelpe, eller komme med konstruktiv kritikk.

Github ble brukt av gruppen til flere ting. En annen måte Github ble brukt til å sikre kvalitet, var ved at hvert gruppemedlem hadde muligheten til å enkelt kunne se gjennom hverandres kode før eventuelle kodesammenslåinger. Github tilbyr en funksjon hvor et gruppemedlem kan tildele et annet medlem oppgaven med å se gjennom koden når den blir lastet opp fra en branch, før den blir slått sammen med resten av koden. På denne måten sikret gruppen kvalitet, og forsikret seg om at det kun var funksjonell og brukbar kode som ble lastet opp og slått sammen med resten av kildekoden.

For versjonskontroll ble programmet GitKraken brukt. Dette programmet ble flittig tatt i bruk av gruppemedlemmene, og var et viktig aspekt når det kom til å sikre kvalitet. Programmet tilbyr en enklere måte å visualisere hva av kode som er lastet opp til Github, og av hvem. Dette gjorde det lettere for gruppemedlemmene å forstå hva de andre medlemmene jobbet med, samt at en kan visualisere de forskjellige branchene som ble jobbet på parallelt. Hvis en skulle være så uheldig å slette koden sin eller gjøre en enorm feil, lar GitKraken deg returnere til din tidligere opplastede versjon enkelt.

Under programmeringen var det flere gruppemedlemmer som foretrakk å benytte seg av parprogrammering. Dette er en utviklingsteknikk hvor to utviklere jobber sammen på en gitt oppgave (Williams & Kessler, 2003, s. 3-4). Ofte er det en person som utfører selve programmeringen mens den andre parten observerer. Under observasjonen leter en etter strategiske feil og mangler for å forsikre seg at koden blir riktig og dermed sikre kvalitet (Williams & Kessler, 2003, s. 3-4). Dette kan være syntax error, skrivefeil, feil metodekall, og lignende (Williams & Kessler, 2003, s. 3-4).

På grunn av Covid-19 har det vært noen utfordringer med parprogrammeringen da det i perioder ikke har vært mulighet til å møtes fysisk. En løsning har vært å dele skjerm over videosamtale, selv om dette ikke har vært helt optimalt. Gruppen fant heldigvis en mer egnet løsning, hvor Microsoft tilbyr en utvidelse i Visual Studio Code kalt «Live Share». Dette er en utvidelse som lar en utvikler direkte sende sin kode til andre, mens programmeringen foregår. Under direktestrømmingen kan alle deltakere både endre, modifisere og lagre kildekode slik de måtte ønske. Det kan minne om Microsoft Teams sin direkte redigering i alle deres programmer, som «Word» og «Powerpoint».

Under utviklingen har det vært viktig for gruppen å dokumentere koden ordentlig. Dette gjøres ved å legge til kommentarer i kildekode for å forklare hva forskjellige deler av koden gjør. Dette er spesielt viktig da Puzzlepart skal bruke koden senere, og ved god kodedokumentasjon blir det enklere for dem å forstå koden, og det sparer dem for mye tid når de skal gjennomgå kildekode. God kodedokumentasjon er også noe gruppen brukte for å sikre kvalitet i prosjektet. Det ble gjort kontinuerlig gjennom hele programmeringsprosessen, og det bidro til at alle gruppe medlemmene forsto kildekode. Det var også viktig hvis en i senere anledning måtte endre koden, da det gir bedre oversikt over hva de forskjellige delene gjør. Ved å vite dette er det enklere å unngå å gjøre feil, eller eventuelt få fjernet viktig kode.

5.1.4 Testing

For å sikre kvalitet i koden valgte gruppen å gjennomføre kontinuerlig testing. Testene ble gjennomført manuelt, og enhetstesting ble nedprioritert. Dette er fordi enhetstesting er veldig tidkrevende og ikke like anerkjent med tanke på frontend programmering. Gruppen sørget for løs coupling i kildekode som gjør at enhetstesting ikke er like nødvendig. I stedet for valgte gruppe medlemmene å legge hovedfokus på manuell testing. Slik testing ble gjennomført for å sikre at applikasjonen fungerte som ønsket, og at eventuelle feil ble fikset tidlig. Dette førte til færre følgefeil ut i prosessen.

6. Testing og kvalitetssjekk av produktet

For å sikre kvalitet i prosjektet var det viktig for gruppen å følge kravene fra Puzzlepart og brukerne av Prosjektportalen 365. Gruppen definerte ut ifra dette kvalitetskrav tidlig i prosjektet, utformet i Kapittel 5. I dette kapittelet vil vi se nærmere på testingen gjennomført for å sikre kvalitet i koden, risikoen ved den valgte testmetoden, samt vil prosjektet bli kvalitetssjekket opp mot kvalitetskravene.

6.1 Kvalitetssjekk

For å sikre kvalitet i alle aspekter av prosjektet ble det tidlig i prosessen definert hva som er kvalitet, og gruppen fulgte prinsippene beskrevet i Kapittel 5 for å oppnå god kvalitet. Gruppen har blant annet god kvalitet i koden, ettersom den er godt dokumentert og følger kodestandarder ønsket av bedriften. Som nevnt i Kapittel 5.1.2 ønsket gruppen også at kodebasen skulle ha løs coupling og høy grad av cohesion, dette har gruppen oppnådd med mange komponenter som er svært selvstendige. Prosjektet ble også manuelt testet jevnlig gjennom prosjektet, noe som sikret god kvalitet i kodebasen. Arbeidsrutiner nevnt i Kapittel 5.1.3 ble fulgt, der bruk av Github og Gitkraken var svært sentralt for å sikre at koden ble lagret, delt og godkjent etter endringer i kodebasen.

Gruppen fikk også gode føringer fra Puzzlepart, hva de forventet og krav de hadde til prosjektet. Som tidligere nevnt ble brukerhistorier formet på bakgrunn av brukerne og bedriften sine uttalelser om krav til Prosjektportalen Teams. Brukerhistoriene ble prioritert ved bruk av MoSCoW-metodikken, som har gitt gruppen en god oversikt over fremtidige oppgaver og fremgang. Brukerhistoriene har gjort at Prosjektportalen Teams har god kvalitet, da flertallet av dem er gjennomført, som vil si at kravene fra brukere og Puzzlepart er nådd.

En av de viktigste valgene som har ført til kvalitetssikring og fremdrift i arbeidet er bruk av Scrum metodikken. Ved bruk av dette rammeverket ble kvaliteten i arbeidet og utviklingen evaluert kontinuerlig. Sprint gjennomganger med samarbeidsbedriften ble gjennomført hver tredje uke, på denne måten fikk gruppen validert sitt arbeid og forsikret seg om at utviklingen hadde den kvaliteten Puzzlepart ønsket. Slike møter ble også gjennomført med veileder, der gruppen kunne få tilbakemelding og bekreftelse eller avkreftelse på forventet fremgang. Ved bruk av en Scrum Master opprettholdt gruppen en tett dialog med bedriften gjennom hele Sprinten, der vi hele tiden kunne stille spørsmål. Denne dialogen var med på å oppnå god kvalitet da gruppen fikk kontinuerlig tilbakemeldinger på arbeidet. Sprint retrospektmøtene

som avsluttet hver Sprint gjorde at de delene av arbeidsmetoden som hadde god kvalitet ble tatt med videre, samt at eventuelle trekk ved arbeidet som ikke fungerte ble fjernet.

6.2 Testing

Testing er med på å sikre kvalitet i produktet, og er dermed en viktig del av utvikling. Det er mange former for testing, blant annet manuell testing, brukertesting, enhetstesting og integrasjonstesting. Under utviklingen har gruppen gjort en vurdering om fordelene av testing veier opp mot kostnadene. I vårt tilfelle har vi benyttet oss av manuell testing av komponenter og logikk gjennom prosjektet. Gruppen valgte å ta i bruk manuell testing ettersom dette er en effektiv og tidsbesparende form for testing i vårt prosjekt. Generelle fordeler med manuell testing er at det egner seg godt til å avdekke feil og mangler, da disse testene tar hensyn til oppfinnsomhet og menneskelig kreativitet (Sjoli, 2019). I dette prosjektet er det tilstrekkelig med slik testing da vår løse coupling i kodebasen tilrettelegger for at det enkelt å manuelt identifisere og lokalisere feil som oppstår. Løs coupling gjør det enklere å teste koden, da modulene ikke avhenger av hverandre. Dersom problemer skulle oppstå, vil disse befinne seg isolert på små og selvstendige enheter av koden. Risikoen ved dette valget er at Prosjektportalen Teams kan ende med å ha betydelige feil i koden. Blir disse feilene oppdaget sent kan de ha vokst seg store som kan kreve enorme mengder arbeid for å løse, og da faller tidsfordelen med vår valgte testmetode vekk.

Manuell integrasjonstesting gjennomføres etter en har implementert endringer på en «feature branch» før en sender pull request. Etter branchen har blitt merget gjennomføres igjen manuell integrasjonstesting for å se at ingen feil har dukket opp. Testingen er basert på en forhåndsbestemt liste med punkter som skal gjennomgås. Sjekklisten består av handlinger som for eksempel å legge til nye listeelementer, se til at data blir hentet og formatert rett, sjekke at data er korrekt synkronisert med SharePoint.

Med jevne mellomrom blir det også utført adhoc testing som innebærer å teste funksjonalitet tilfeldig. Det er en type test som gjennomføres uten et scenario eller struktur, noe som vil si at det er store variasjoner fra individ til individ (Chhabra, 2012, s. 66-67). Dette gjør at en dekker en stor flate og skaper mange ulike testscenarioer som igjen fører til at en kan avdekke flere feil.

Gruppen planla å gjennomføre brukertesting hvis tiden strakk til, noe den ikke gjorde. Testingen vil ha omtanke for brukervennlighet og kan i tillegg avdekke tekniske feil og

svakheter. Det er ofte slik at utviklingsteamet har et annet perspektiv på systemet enn sluttbrukeren, noe som kan føre til at systemet oppleves på en annen måte enn tiltenkt (Foss-Pedersen, 2017). Ved å observere en bruker uten kjennskap til systemet vil en kunne avdekke bruksmønstre og tanker rundt systemet en selv ikke har tenkt på. Ved brukertesting er det vanlig å gi brukeren et sett med relativt åpne oppgaver en skal løse. Om oppgavene er for detaljstyrende vil det være vanskelig å avdekke andre bruksmønstre enn de en selv har satt opp (Foss-Pedersen, 2017). Testobjektene burde helst være reelle sluttbrukere av systemet, med andre ord, eksisterende Prosjektportalen brukere, ettersom systemet er spesielt designet for disse (Foss-Pedersen, 2017). For å få størst mulig utbytte av denne testingen ville det vært ønskelig at testobjektene innehar ulike roller og kompetanse, for å dekke et større spekter av brukerbasen. Informasjonen vi tilegner oss gjennom disse observasjonene vil bli brukt for å forbedre systemet for brukerne.

6.3 Vurdering av oppnådd kvalitet

Gruppen har oppnådd og overgått den forhåndsdefinerte minst brukbare produkt (MBP) beskrevet i Kapittel 2.2. Produktet har med mer av den viktigste funksjonaliteten implementert enn MBP forventer, koden er svært godt dokumentert og har en oversiktlig struktur som legger til rette for videre utvikling av samarbeidsbedriften. For å sikre at koden og brukergrensesnittet opprettholder den gode kvaliteten under faktisk bruk hadde det vært nødvendig med brukertesting. Dessverre fikk ikke gruppen tid til dette, noe som kan svekke kvaliteten til produktet. Oppdragsgiver Puzzlepart har begynt å vise frem prosjektet til noen kunder, dette ser gruppen som indirekte bekreftelse på at de er fornøyde med vårt arbeid. Dette er enda en bekreftelse på at Prosjektportalen Teams har god kvalitet og møter de gitte krav.

7. Refleksjon

Under utviklingen av Prosjektportalen Teams møtte gruppen på flere utfordringer som måtte løses. For å gjennomføre prosjektet og løse utfordringene gruppen møtte, ble erfaring fra tidligere semester brukt og ny kunnskap ble tilegnet. Prosjektet har ført til stort læringsutbytte hos samtlige gruppemedlemmer, der de tekniske ferdighetene har økt hver uke. Gruppen har hatt et godt samarbeid gjennom hele prosjektet, noe resultatet gjenspeiler. Medlemmene har jobbet jevnt, og hatt en god arbeidsfordeling gjennom hele utviklingen, der medlemmene sine styrker har blitt utnyttet til det fulle.

7.1 Utfordringer og endringer

En utfordring som fulgte gruppen gjennom pre-sprinten og Sprint 1 var dårlig timeføring. Dette problemet var tett knyttet til valg av planleggingsverktøy. Gruppen startet tidlig med planleggingsverktøyet Azure DevOps etter anbefaling og ønske fra Puzzlepart. Gruppemedlemmene ble tildelt gjestebbrukere av Puzzlepart for å kunne bruke Azure DevOps til å utforme backloggene og holde oversikt over fremdriften i prosjektet. Gjestebbrukernes rettigheter var begrenset, noe som gjorde det utfordrende å føre timer. I slutten av Sprint 1 ble det derfor bestemt at Microsoft Planner skulle bli brukt videre i prosjektet, da fikk også gruppen sentralisert arbeidet på færre plattformer. Dette var en god avgjørelse da timeføringen ble bedre og det ble det enklere for gruppemedlemmene å se fremgangen i prosjektet.

Under Sprint 2 hadde gruppen som mål å implementere to komponenter utviklet av Puzzlepart. Komponentene som skulle bli brukt var prosjektstatus og fasevelgeren som inneholder noen av de mest brukte funksjonene i prosjektet. Under implementeringsfasen møtte gruppen på flere utfordringer. Den største utfordringen var lite dokumentert kode. Dette gjorde at gruppen brukte feil properties under implementering som førte til at komponenten ikke fungerte som tiltenkt. Det at flere av de nødvendige properties var satt som valgfrie, gjorde at man så andre steder i komponentene for å løse problemet. Dette førte til at gruppen brukte mye tid på feilsøking i løpet av Sprint 2 som kunne blitt brukt på implementering av annen funksjonalitet. For å løse flere av utfordringene anvendte vi erfaringer fra tidligere semestre.

7.2 Bruk av erfaringer fra tidligere semester

Tidligere emner i bachelorgraden IT og informasjonssystemer har gitt oss kunnskap og erfaringer som vi benyttet oss av i denne bacheloroppgaven. Studiet har gitt oss økt forståelse av hvordan digitalisering kan være med på å bidra til videreutvikling av organisasjoner og samfunn. Gruppen har under arbeid med bachelorprosjektet erfart hvordan det er å arbeide med aktuelle problemstillinger i samarbeid med ulike oppdragsgivere.

Gruppemedlemmene har fordypet seg i ulike temaer, alt fra IT-sikkerhet, database og full-stack programmering, til digitalisering. Medlemmene sine ulike fordypninger skaper et bredt spekter med kunnskap og erfaringer. Dette bidrar til å utfylle hvert aspekt i et team, samt at medlemmene hjelper og lærer av hverandre.

Under studiet sitt første semester lærte medlemmene om objektorientert programmering og godt design, dette gjennom emnene IS-104 og IS-109. Sammen med emnet IS-110 fra andre semester, ga IS-109 flere av medlemmene en grunnleggende forståelse av programmering som de har tatt med seg videre gjennom studiet, samt i møte med nye programmeringsspråk. Kunnskapen om digital interaksjonsdesign har medlemmene tatt i bruk når det kommer til design av Prosjektportalen Teams. Benyon's 12 prinsipper for godt design sto sentralt i emnet IS-104, og gruppen har brukt dette som utgangspunkt for å kunne skape et godt og brukervennlig design i applikasjonen.

Høstsemesteret i 2019 omhandlet én stor oppgave, fordelt på tre emner, IS-200, IS-201 og IS-202. Dette semesteret omhandlet praktisk erfaring med å jobbe i team og opp mot en prosjekteier. Det ble lagt stort fokus på å jobbe agilt, og medlemmene lærte derfor å ta i bruk den agile metodikken Scrum. Denne metodikken valgte gruppen å bruke under utviklingen av Prosjektportalen Teams. Kunnskapen om hvordan strukturere et team var betydelig erfaring både for fremtidige kurs og bacheloroppgaven. Hvordan strukturere et utviklingsprosjekt, samt rapporter, var også viktig kunnskap gruppemedlemmene tilegnet seg dette semesteret. Våren 2020 deltok alle medlemmene i kurset IS-112-1, Tjenstedesign og forretningsmodeller. Dette emnet fokuserte på å gi studentene en plattform hvor en praktiserer planlegging av digitaliseringsprosjekter. Her ble alle grupper koblet opp mot et eksternt selskap som fungerte som prosjekteier. Dette ga medlemmene erfaring om hvordan skape en god relasjon og kommunikasjon med en ekstern bedrift. God kommunikasjon og relasjon med Puzzlepart har vært fundamentalt for et fungerende samarbeid og utvikling av et produkt med god kvalitet.

7.3 Læringsutbytte

Bachelorprosjektet har ført til stort læringsutbytte, blant annet når det kommer til tekniske ferdigheter, samarbeid og personlig utvikling. Dette delkapittelet tar opp hvilke tekniske ferdigheter medlemmene har tilegnet seg under prosjektet, hvorfor gruppen hadde et godt samarbeid og egenevaluering av individuell innsats.

7.3.1 Tekniske ferdigheter

Samtlige gruppe-medlemmer har under utviklingen av Prosjektportalen Teams tilegnet seg nye tekniske ferdigheter. Medlemmene har utviklet sin første Microsoft Teams applikasjon, dette har ført til stort læringsutbytte, der nytt språk og rammeverk har blitt tatt i bruk.

Gruppen har lært seg ett nytt programmeringsspråk, Typescript, og hvordan det kan brukes sammen med rammeverket React. Dette er læringsutbytte medlemmene kommer til å ta med seg videre etter endt studieløp, da denne kunnskapen kan gi fortrinn i eventuelle søknadsprosesser.

7.3.2 Samarbeidsferdigheter og gruppeevaluering

For å arbeide effektivt og godt gjennom bachelorprosjektet var det essensielt at gruppe-medlemmene samarbeidet. For å legge til rette for et godt samarbeid ble gruppen tidlig enig om at kommunikasjonen måtte være optimal. Dette ble gjort ved å opprette gruppesamtale på kommunikasjonskanalen Discord, samt ble Microsoft Teams anvendt for å kunne arrangere møter og arkivere filer. Et viktig element for å oppnå optimal kommunikasjon var at medlemmene ble enig om å være transparente, der ingen spørsmål er dumme og medlemmenes styrker måtte brukes til det fulle. Dette har gjort at gruppen har samarbeidet godt gjennom prosjektet, der alle medlemmene vet hva de andre arbeider med og hvor sine styrker kan anvendes på best mulig måte.

Gruppen er svært fornøyd med sitt samarbeid under dette prosjektet, spesielt med tanke på at medlemmene ikke har arbeidet sammen i tidligere semester. Medlemmene hadde en naturlig god kommunikasjon fra start, som gjorde samarbeidet enklere. Rollene de ulike medlemmene ble tildelt falt naturlig og fungerte godt. Som tidligere nevnt fikk medlemmene brukt sine styrker, men de fikk også prøvd seg på områder de var mindre komfortable med, dette for å øke læringsutbyttet.

Den største utfordringen gruppen møtte på var at coronatiltakene ble hyppig endret, noe som begrenset vår mulighet til å møtes fysisk. Etersom unntakstilstand har vært til stede over en

lengre periode greide gruppen enkelt å tilpasse seg tiltakene og var fleksible. Scrummetodikken fikk på grunn av denne situasjonen en mer betydelig rolle enn antatt før semesterstart. Daglig gjennomgang gjorde det enklere å arrangere møter, backloggene ga alle medlemmene en oversikt over de ulike oppgavene som skulle gjennomføres, og sprintgjennomgang gjorde at gruppen kunne få tilbakemelding fra bedrift og veileder.

7.3.3 Egenevaluering

Tuva Cecilie Eikaas

I løpet av bachelorprosjektet har jeg hatt flere roller, og brukt mine styrker der det trengs. Mine hovedoppgaver har handlet om design av applikasjonen og rapportskriving. Jeg har bidratt med å lage flere designforslag til Prosjektportalen Teams, samt hatt hovedansvar for utformingen av prototypen vår. Når det kommer til utvikling, har jeg i hovedsak fokusert på design og frontend. Jeg har også aktivt deltatt i parprogrammering, noe som har ført til at jeg har økt min kompetanse innen utvikling. Hovedansvaret mitt i rapporten har vært kvalitet og hvordan vi kan sikre dette gjennom prosjektet. På grunn av dette har jeg hele tiden vært oppmerksom på hvordan vi som gruppe har måttet jobbe for å sikre best mulig kvalitet i vårt produkt. Jeg har jobbet jevnt og godt gjennom hele semesteret, og lært utrolig mye av mine medstudenter.

Martin Ruud

Under dette prosjektet har min rolle primært vært å ha ansvar for det tekniske, samtidig som jeg har vært knutepunktet mellom Puzzlepart og gruppen. I prosjektet har jeg fungert som en Scrum Master, der hovedoppgavene mine gikk ut på å opptre som en støttespiller, samt være disponibel på de områdene det trengtes hjelp. Det tekniske innebærer implementasjon og planlegging av arkitektur og kode. Ettersom jeg fra tidligere har arbeidet med Prosjektportalen 365 gjorde dette at jeg hadde relativt god kjennskap til miljøet, og kunne ta gode tekniske beslutninger. I tillegg har jeg vært ansvarlig for henting og innsetting av data i prosjektet, sørget for mye av kjernefunksjonaliteten, samt deltatt aktivt i beslutningen om hvordan løsningen på produktet skulle være. Avslutningsvis, har jeg styrt mesteparten av versjonskontrollen, og satt opp git-struktur.

Ragnhild Sofie Ertzeid Toft

I dette semesteret har jeg hatt flere roller innad prosjektet. Jeg har tilegnet meg ny kunnskap og erfaringer som jeg vil ta med videre ut i arbeidslivet. Hovedsakelig, har jeg arbeidet med

rapportskriving og design av applikasjon. Jeg har bidratt når det kommer til utforming av flere av designforslagene, samt deltatt i utviklingsprosessen. I rapporten har jeg hatt som hovedansvar å skrive om prosjekthåndtering. Når det kommer til utvikling har jeg først og fremst fokusert på frontend, og har aktivt deltatt i parprogrammering. Dette har bidratt til faktorer som økt jobbtilfredshet, kunnskapsoverføring mellom gruppemedlemmene, og bedre kvalitet i koden. Semesteret har vært svært lærerikt, og jeg føler personlig at jeg har vært en bidragsyter til prosjektet, på lik linje som mine gruppemedlemmer. Jeg har jobbet jevnt og trutt igjennom hele semesteret, og opplevdd en bratt læringskurve på samtlige områder.

Kristian Bjurholt Rein

I løpet av bachelor prosjektet har jeg jobbet med flere forskjellige oppgaver. Hovedsakelig har jeg jobbet med utvikling, men også design og administrative oppgaver. Jeg har bidratt med å lage wireframes til design og holde orden på azure devops/microsoft planner. Når det gjelder utvikling har jeg parprogrammert med de andre i gruppen samt lagd navigasjonssystemet i produktet. Jeg har lært mye av mine gruppemedlemmer når det gjelder teamarbeid. Denne kompetansen har gjort at jeg nå i større grad forstår hvordan man kan skape stødig progresjon, samtidig som man opprettholder et godt arbeidsmiljø som sørger for kvalitet.

Henriette Islann Fremstad

Under utviklingen av bachelor prosjektet har jeg hatt flere roller, og lært utrolig mye av mine medstudenter. Hovedoppgavene mine har dreid seg om design og rapportskriving. Jeg har vært med på å lage flere forslag til design til Prosjektportalen Teams, samt vært med på å utvikle det. I rapporten har jeg hatt ansvar for å få med og utforme relevante kapitler til prosjektet, dette har gjort at jeg hele tiden har hatt god oversikt over hva som må gjøres videre. Jeg har også hatt hovedansvar for risikoanalysen, og har etter hver Sprint implementert nye risikoelementer hvis nødvendig. Ved å parprogrammere har jeg lært mye av de andre gruppemedlemmene, og fått utviklet min kompetanse. Jeg har fått brukt mine styrker og utviklet mine svakheter til noe bedre under dette prosjektet. Dette semesteret har gjort at jeg har fått en bratt læringskurve innen utvikling, samt har jeg jobbet godt og jevnt gjennom hele prosjektet og fungert som en ressurs der jeg kan.

8. Uttalelse fra oppdragsgiver

Prosjektportalen er en løsning utviklet og forvaltet av Puzzlepart som åpen kildekode.

Løsningen er laget for å støtte opp under prosjektsamhandling for prosjekter i hovedsakelig offentlig sektor, men kan også brukes til annen type prosjektsamhandling og scenarioer.

Løsningen er laget for Microsoft 365, kan lastes ned, installeres og brukes uten kostnad, og har flere tusen daglige brukere.

Prosjekter i Prosjektportalen opprettes som Microsoft 365 grupper med et tilpasset SharePoint-område. Prosjekter i Prosjektportalen kan benytte Teams til samhandling i dagens løsning ved å manuelt legge på et Team for prosjektet. Tilbakemelding fra flere kunder er at Teams-delen og SharePoint-delen av prosjektene oppleves som adskilte områder, og det er et behov for å forenkle dette samspillet. Microsoft Teams brukes av flere og flere brukere, og har økt kraftig i bruk siste året. Vi ser at flere av kundene våre ønsker Teams sterkere inn som en del av Prosjektportalen.

I dialog med prosjektgruppen (Henriette, Kristian, Martin, Ragnhild Sofie og Tuva) fant vi ut at nettopp behovet knyttet til Teams-samhandling i prosjekter hadde vært nyttig å få adressert. Basert på en overordnet kravspesifikasjon har gruppen jobbet med å bryte ned kravene i oppgaver, utarbeidet teknisk arkitektur og løsningsskisse, samlet inn behov fra brukerforumet til Prosjektportalen (i regi av Puzzlepart) og utviklet løsningen. Status og progresjon har blitt delt med Puzzlepart månedlig gjennom sprint demoer, og i prosessen har vi fått sett hvordan løsningen har reist seg frem over tid. Løsningen er en Teams-app for prosjekter i Prosjektportalen, som i kontekst av Teamet brukerne står i viser relevant informasjon om prosjektet, og gir brukeren tilgang til å enkelt arbeide med de ulike arbeidsoppgavene i prosjektet.

Gruppen har møtt på flere tekniske utfordringer gjennom prosessen, som de i hovedsak har løst selv. Gruppen har jobbet selvstendig og dedikert, har vist kontinuerlig progresjon og har hatt en god dialog med oss i Puzzlepart underveis. Løsningen per 11. mai 2021 oppleves som god, og nærmer seg en løsning som kan piloteres av brukere. Puzzlepart forventer ikke at et slikt prosjekt kan slippes direkte til brukergruppen uten å kvalitetssikres av oss. En nærmere gjennomgang av design, kodekvalitet og arkitektur må planlegges før løsningen kan anses som helt ferdig. Dette kan heller ikke forventes i et slik prosjekt. Prosjektgruppen har løst oppgaven på en tilfredsstillende måte, der mye av funksjonaliteten som er blitt levert har overgått våre forventninger. At gruppen uoppfordret også har hatt søkelys på at

brukergrensesnittet også skal kunne fungere på mobil og nettbrett har skapt engasjement og begeistring hos oss i Puzzlepart. Vi vil også fremheve at gruppen har vist gjennomgående fokus på å ta fremtidsrettede og generiske arkitekturvalg underveis i prosjektet. Vi er meget godt fornøyde med leveransen fra Prosjektgruppen.

For Puzzlepart AS,

Espen Olsen, IT Operations Manager, leder Puzzlepart Kristiansand

Tarjei E. Ormestøyl, COO, leder for Utvikling i Puzzlepart

9. Referanseliste

Aven, T. (2015). *Risk Analysis* (3.utg.). West Sussex: John Wiley & Sons.

Barnes, D., J. & Kölling, M. (2012). *Objects First with Java* (5.utg.). New Jersey: Pearson Education.

Benyon, D. (2014). *Designing Interactive Systems* (3. utg.). Edinburgh Gate: Pearson Education.

Chhabra, N. (2012). Introduction To Adhoc Testing. International Journal of Scientific & Technology Research, 1(7), 66-67. Hentet fra <https://www.ijstr.org/final-print/August2012/Introduction-To-Adhoc-Testing.pdf>

Entur. (u.å.). *Brukerhistorier*. Hentet 10.mai 2021 fra <https://design.entur.org/kom-i-gang/for-designere/brukerhistorier>

Fitzpatrick, R. (1996). Software quality: definitions and strategic issues. *Arrow-Dublin Institute of Technology*, 1-36. Hentet fra <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.169.9300&rep=rep1&type=pdf>

Foss-Pedersen, R. J. (2017, 24. august). Fem tips til deg som skal gjennomføre brukertester [Blogginlegg]. Hentet fra <https://www.usit.uio.no/om/organisasjon/bnt/web/ux/blogg/2017/brukertest.html>

Fuggetta, A. (2003). Open Source Software – An Evaluation. *Journals of Systems and Software*, 66(1), 77-90. Hentet fra: [https://doi.org/10.1016/S0164-1212\(02\)00065-1](https://doi.org/10.1016/S0164-1212(02)00065-1)

Hatton, S. (2008). *Choosing the “Right” Prioritisation Method*. Innlegg presentert ved 19th Australian Conference on Software Engineering, Crawley, Western Australia. Hentet fra <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4483241>

Högstrand, J. (2019, 29.mai). Scrum - et enkelt rammeverk for komplekst arbeid [Blogginnlegg]. Hentet fra <https://www.prosjektbloggen.no/scrum-et-enkelt-rammeverk-for-komplekst-arbeid>

Jacobsen, D., I. (2018). *Hvordan gjennomføre undersøkelser?* (3.utg.). Oslo: Cappelen Damm Akademisk.

Mathiassen, L., Munk-Madsen, A., Nielsen, P.A., Stage, J. (2018). *Object Oriented Analysis & Design* (utg. 2). Hadsund: Metodica ApS.

Microsoft. (2021, 26. april). What are Microsoft Teams Tabs? Hentet fra <https://docs.microsoft.com/en-us/microsoftteams/platform/tabs/what-are-tabs>

Microsoft. (2019, 8. august). Velge den beste Agile-metoden for prosjektet ditt. Hentet fra <https://www.microsoft.com/nb-no/microsoft-365/business-insights-ideas/resources/how-to-choose-the-best-agile-methodology-for-your-project>

Microsoft. (2018, 25. april). How SharePoint and OneDrive safeguard your data in the cloud. Hentet fra <https://docs.microsoft.com/en-us/sharepoint/safeguarding-your-data>

Nes, S., M. (2019, 16.mai). En kort introduksjon til Scrum [Blogginnlegg]. Hentet fra <https://www.visma.no/blogg/en-kort-introduksjon-til-scrum/>

Nilsen, T. (2008) *Smidig utvikling i en mellomstor bedrift på Sørlandet* [Masteroppgave] Universitet i Agder. https://uia.brage.unit.no/uia-xmlui/bitstream/handle/11250/136073/master_infosys_2008_nilsen.pdf?sequence=1&isAllowed=y

Puzzlepart. (u.å.) *Om oss*. Puzzlepart. Hentet 5.mai 2021 fra <https://puzzlepart.com/om-oss/>

Puzzlepart. (2021). *Pins: Designsystem for Puzzlepart Apps & Services*. Oslo: Puzzlepart.

Puzzlepart. (u.å.) *Prosjektportalen 365*. Puzzlepart. Hentet 5.mai 2021 fra <https://puzzlepart.com/losninger/prosjektportalen/>

React. (u. å.). *A JavaScript Library for building user interfaces*. Reactjs. Hentet 14. April 2021 fra <https://reactjs.org/>

Repstad, P. (2007). *Mellom nærhet og distanse* (4.utg.). Oslo: Universitetsforlaget.

Rubin, K., S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process* (1.utg.). Michigan: Addison-Wesley.

Sjoli, S. (2019, 24. februar). Automatiserte tester – hva og hvorfor. Hentet fra <https://www.bouvet.no/bouvet-deler/automatiserte-tester-hva-og-hvorfor>

Som de Cerff, W., van de Vegte, J., Boers, R., Brandsma, T., de Haij, M., van Moosel, W., ...van der Schrier, G. (2018). Agile Development in Meteorological R&D: Achieving a Minimum Viable Produkt in a ScrumWork Setting. *American Meteorological Society*, s. 2507-2518. Hentet fra <https://doi.org/10.1175/BAMS-D-17-0273.1>

Stokke, O., P., B. (2019, 21. januar). Hva er funksjonell programmering? *Kode24*. Hentet fra <https://www.kode24.no/guider/hva-er-funksjonell-programmering/70656838>

Typescript. (u.å.). *What is Typescript?* Typescript. Hentet 12. April 2021 fra <https://www.typescriptlang.org>

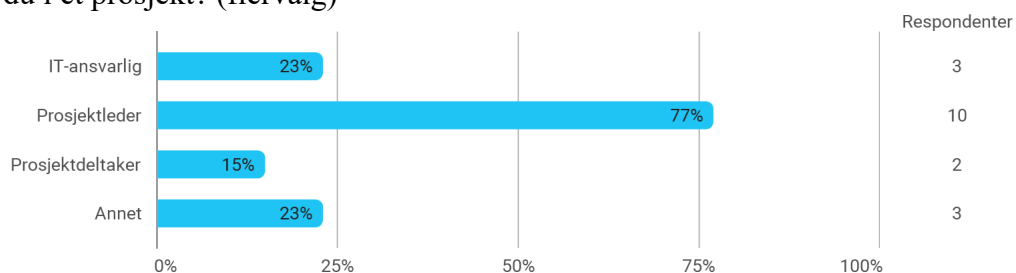
Varma, T. (2015). *Agile Product Development – How to Design Innovative Products That Create Customer Value*. New York: Apress.

Williams, L. & Kessler, R.R. (2003). *Pair Programming Illuminated*. Boston: Pearson Education, Inc.

10. Vedlegg

10.1 Spørreundersøkelse med svar

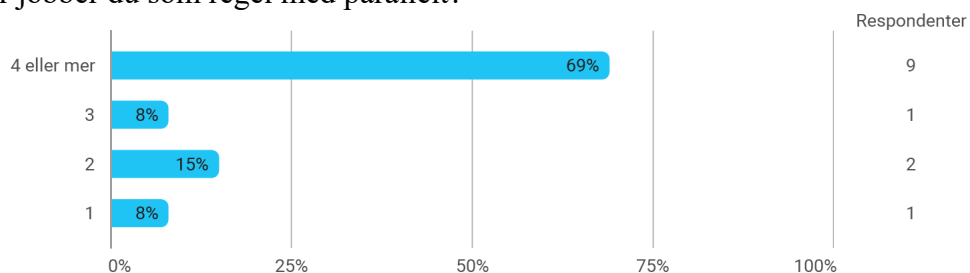
1. Hvilke rolle(r) har du i et prosjekt? (flervalg)



2. Hvilke rolle(r) har du i et prosjekt? (flervalg) - Annet

- Prosjekteier (burde vært et valg i skjemaet)
- PIL-administrator
- Programleder

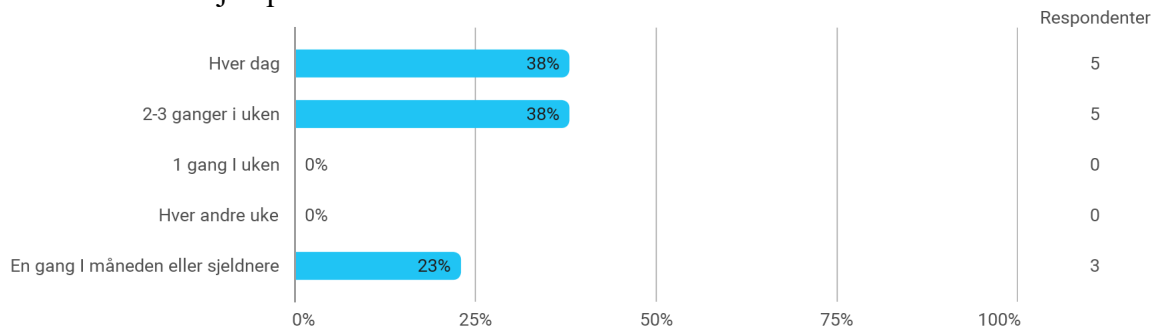
3. Hvor mange prosjekter jobber du som regel med parallelt?



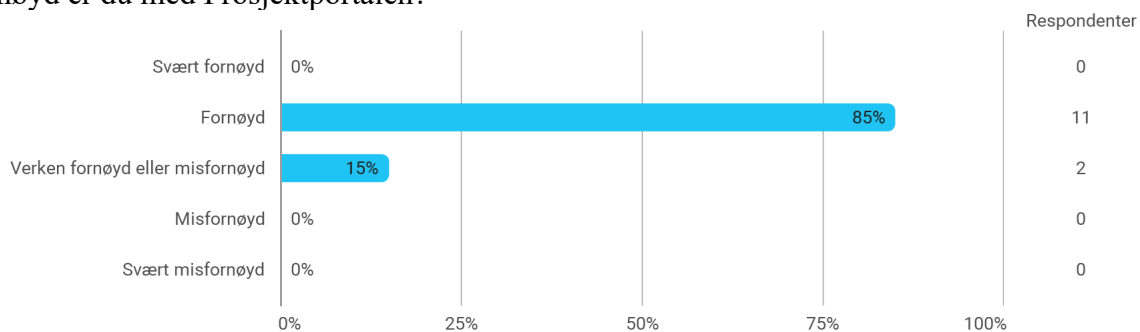
4. Hvilken kommunikasjonsplattform bruker dere for å samhandle under prosjektarbeid (f.eks. Microsoft Teams, Slack)?

- Teams intern
- MS Teams
- Teams (chat)
- Mail
- Microsoft Teams
- teams
- Microsoft Teams
- Teams og Workplace
- Microsoft Teams
- E-post
- Teams
- Hovedsaklig Prosjektportalen, Teams, Outlook, Sharepoint, MS Project, intranet(SP)
- Webex
- Mail
- Teams

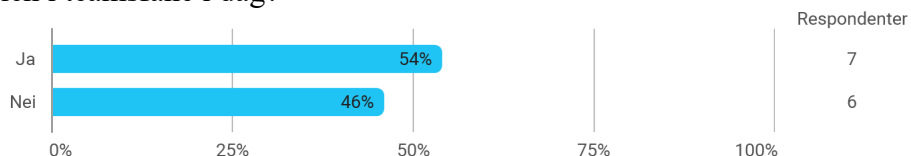
5. Hvor ofte bruker du Prosjektportalen?



6. Hvor fornøyd er du med Prosjektportalen?



7. Anvender du Prosjektportalen i teamsfane i dag?



8. Hvilke funksjoner anvender du hyppigst i Prosjektportalen?

- Oppgaveliste og fremviser andre relevante lister og sider fra portalen. Samt at all kommunikasjon i prosjektet går i Teams-kanaler
- Prosjektdelen med dokumenter og maler, logg, Planner, OneNote, ressursallokering, kommunikasjonsaktiviteter.
- Prosjektstatus, dokumenthåndtering
- Prosjektstatus
- NA
- Prosjektstatus, Prosjektlogg, Porteføljeoversikt og Porteføljeinnsikt.
- De fleste
- Knytter opp de mest aktuelle listene/verktøyene for hvert prosjekt som faner i Teams. F.eks Hvis et er veldig relevant å starte med en Interessentanalyse så lager vi en fane på det. Medlemmene forholder seg da til Teams. Det er bare prosjektledere og andre ledere som trenger å bruke selve Portalen.
- Venter med å bruke Prosjektportalen til bygg- og anleggsmodulen er på plass.
- Er i oppstartsfase i bruk av Prosjektportalen
- Porteføljesiden, se på status på ulike porteføljer, ta ut rapporter, kontrollere porteføljer og metadata på prosjekter. Om Prosjektet i det enkelte prosjektrum. Her utføres

prosjektleders rapportering for det ulike prosjekt, og dette aggregeres til porteføljesiden. Tilgangsstyring. (Tilgangsstyring bør være et av hovedpunktene dere ser på i deres videre arbeid) for en vanlig bruker, ville svaret hos oss være: dokumenter, rapportering i "Om prosjektet", milepæler/oppgaver, interensnter og kommunikasjon, usikkerheter.

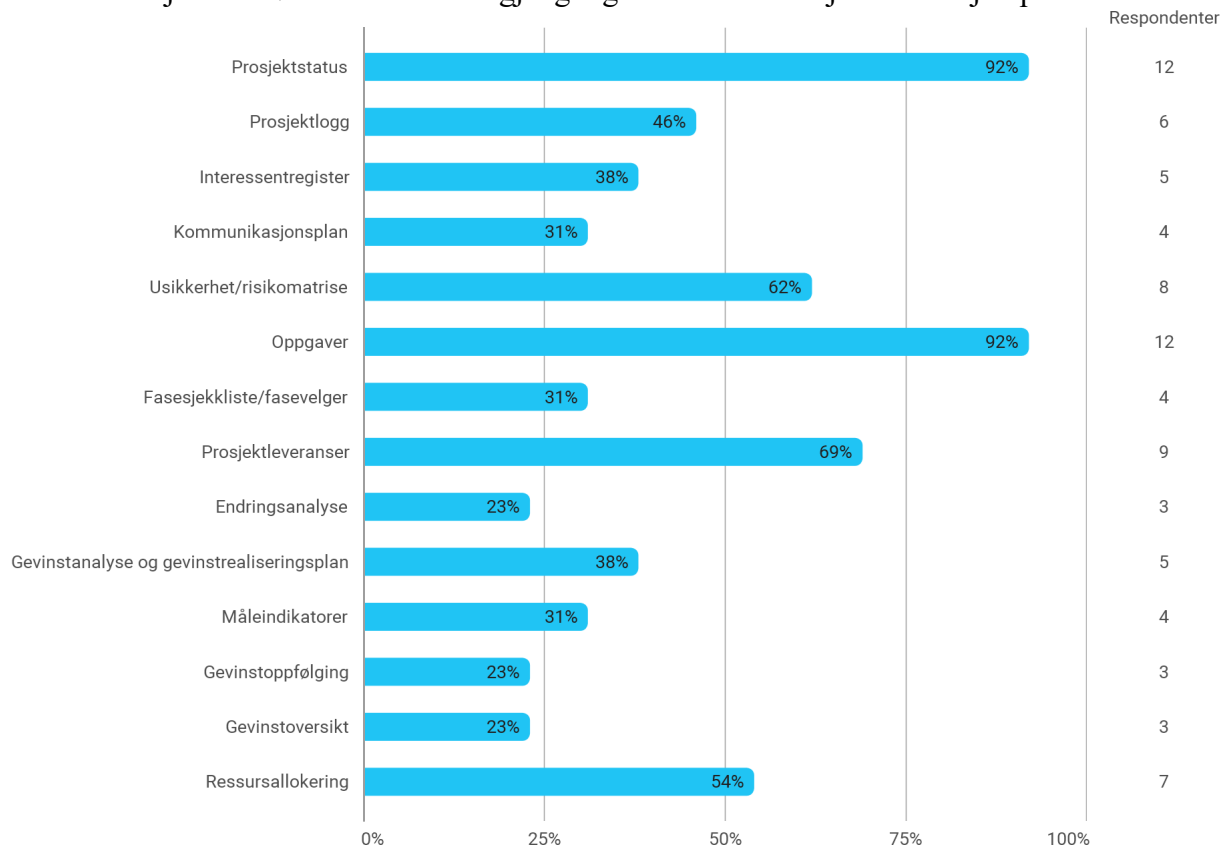
- Programrom, Dokumentbibliotek m/standard dokumenter, Oppgaver, Sjekkliste, Prosjektleveranser, Gevinst
- Statusrapportering, dokumentbibliotek, tidslinje (knyttet til gammel oppgaveliste), Usikkerhet

9. Hvilke funksjoner anser du som viktigst for ditt arbeid med prosjekter i Prosjektportalen?

- Portalen struktur, men Teams/Planner er de mest benyttete integrasjonene for oss.
- Se forrige svar, men tror at porteføljedelen vil kunne bli viktigere. Vi vil prøve å knytte den til styring.
- Gode rapporter
- Portofølje
- status, logg, planlegging
- Muligheten for at medlemmene kan oppdatere i Teams.
- Mange
 - At metoden støttes
- Planner maler er viktige, vi lager våre egne varianter til ulike typer prosjekter, men det er ofte basert på en variant av malen i prosjektportalen.

De ulike verktøyene i portalen håper jeg skal bli nyttige nå når vi forsøker å endre kultur for prosjekt og porteføljestyring i vår kommune. Viktig at det er enkelt å bruke.
- Venter med å bruke Prosjektportalen til bygg- og anleggsmodulen er på plass.
- Dokumentbibliotek
 - Prosjektstatus
 - Risikomatrise
- Prosjektlederens rapportering, at den utføres ett sted, og at PL rapportering kan brukes i ulike aggregerte sammenhenger. Gode portefølje og innsiktsflater. God og ensartet dokumenthåndtering. Tilgangsstyring, enkelt og intuitivt samtidig som det er sikkert og uten bugs.
- Som programleiar er det å få oversikt over alle prosjekt i porteføljen til programmet, og å kunne dokumentere gevinster så tidlig som i konseptfasen, og gevinstmåling over tid etter prosjektet er avslutta. Elles er det samhandling og dokumentdeling som er viktig, i tillegg til oppgåvestyring
- Statusrapportering, leveranser og oppgaver

10. Hvilken funksjonalitet ønsker du å ha tilgjengelig i en Teams versjon av Prosjektportalen?



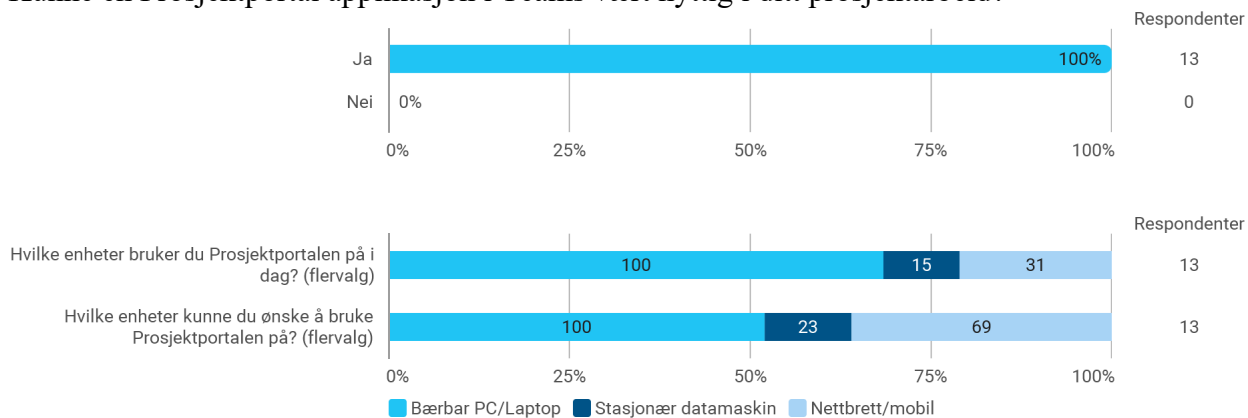
11. Finnes det ønsket funksjonalitet som ikke eksisterer i dagens Prosjektportalen?

- Konvertere opprettede Teams til portalen når det blir prosjekter
Fokus på innovasjon og førprosjektfasene (ide-fasen)
- Det er dårlig kobling til en prosjektplan. Planneroppgavene er ikke koblet til f.eks en wbs eller gannt diagram. Det må løses andre steder idag. Kunne sikkert vært en bedre kobling mellom informasjon som er lagt inn et sted og som skal brukes et annet sted.
- Fremdriftsplan
- Bedre planleggingsverktøy (gantt, MS project)
- Bedre ressursoversikter på tvers av prosjekter.
- Hvis man kan ha en enklere visning rett i Teams hadde det vært fint. Men vi skreddersyr dette selv nå ved å legge på faner i kanalene. Vi vurderer om vi burde ha egen kanal for gevinstarbeid f.eks for det kan bli mange faner i en kanal. Et bedre oversiktsbilde hadde nok vært lurt.
- Venter med å bruke Prosjektportalen til bygg- og anleggsmodulen er på plass.
- Styring av tilganger ett sted, og via AD-grupper er såvidt jeg skjønner ikke mulig i TEAMS, noe som kan gjøre det unødvendig/nyttig å ha en PP-app inne i TEAMS. i alle fra fra et porteføljesynspunkt. I prosjektportalen hos hos styres de fleste tilganger automatisk via AD-grupper, for å oppfylle krav til åpenhet og innsyn blant ansatte, men også for å slippe å gi en og en tilgang. Om man snur på det, så kunne det kanskje være en ide å fikse en chattemulighet i Prosjektportalen? :-) dette sammen med møte og

kalender fordelene vi har i teams, ville kanskje vært enda bedre enn å forsøke å putte PP inn iteams?

- Tidslinje (med mindre man bruker "gammel" oppgaveliste)

12. Kunne en Prosjektportal applikasjon i Teams vært nyttig i ditt prosjektarbeid?



10.2 Brukerhistorier

Nr. 1		Prioritet: Must have
Brukerhistorie	Som en prosjektleder ønsker jeg en oversikt over tilstanden til prosjekter slik at jeg enklere kan få oversikt over prosjektets status.	
Beskrivelse	Ettersom det er mange som jobber parallelt med prosjekter vil det være nødvendig å skaffe seg et oversiktlig bilde av fremdriften.	
Vilkår for aksept	Kriteriet er akseptert når brukeren enkelt kan få oversiktlig oppsummering av prosjektet.	
Argumentasjon	Dette er en «must have» fordi synliggjøring av prosjektstatus er med på å visualisere progresjonen innad i prosjektet. Dette fører til at prosjektlederen kan gjøre enda mer produktive valg på vegne av prosjektet.	

Nr. 2		Prioritet: Must have
Brukerhistorie	Som en prosjektdeltaker ønsker jeg en mulighet for å kunne se tildelte oppgaver, samt oppgavene til mine kollegaer, slik at man får en oversikt over alle oppgavene og dermed unngår dobbeltarbeid.	
Beskrivelse	Muligheten til å se oppgaver tildelt til hverandre innad i et prosjekt vil være med på å effektivisere og organisere teamet, samt å avverge dobbeltarbeid.	
Vilkår for aksept	Kriteriet er akseptert når prosjektdeltakerne har mulighet til å se oppgaver tildelt til seg selv, samt oppgaver tildelt til sine meddeltakere.	
Argumentasjon	Dette er en «must have» fordi det bidrar til å organisere prosjektteamet samtidig som det hjelper for å unngå dobbeltarbeid. Dette kan bidra til å styrke effektiviteten blant prosjektdeltakerne.	

Nr. 3		Prioritet: Must have
Brukerhistorie	Som en prosjektdeltaker ønsker jeg å kunne ha tilgang til prosjektportalen fra flere plattformer slik at jeg har muligheten til å kunne logge meg inn fra flere enheter.	
Beskrivelse	En app som åpner opp for muligheten til å kunne ha adgang til prosjektportalen fra flere plattformer.	
Vilkår for aksept	Kriteriet er akseptert når produktet er tilpasset flere plattformer samtidig, som nettbrett/PC, mobil og stasjonær datamaskin.	
Argumentasjon	Dette er en «must have» fordi ved å tilpasse prosjektportalen flere plattformer gir det mulighet for å kunne nå så mange brukere som mulig, og man sitter igjen med et bredere utvalg av brukere.	

Nr. 4		Prioritet: Must hav
Brukerhistorie	Som en prosjektdeltaker ønsker jeg å kunne sette opp en risikomatrise slik at jeg kan få et godt overblikk over prosjektets usikkerhet og risiko.	
Beskrivelse	En fremstilling av risikoene i et prosjekt som fokuserer på konsekvens for en gitt hendelse og sannsynligheten for dette.	
Vilkår for aksept	Kriteriet er akseptert når man har muligheten til å sette inn/endre/slette elementer fra en tilhørende liste. Ut ifra denne listen skal en risikomatrise rendres med tilhørende risikoelementer.	
Argumentasjon	Dette er en «must have» fordi en risikomatrise har som hensikt å gi en oppsummering og beskrivelse av den risikoen som foreligger. En viktig faktor med risikoanalyse er at den gir en god visuell fremstilling av risikoen, selv om det er en veldig forenkling av situasjonen.	

Nr. 5		Prioritet: Should have
Brukerhistorie	Som en prosjekteier ønsker jeg å kunne ha muligheten for å rapportere prosjektstatus slik at jeg enkelt kan formidle prosjektets fremdrift til eksterne eller interne samarbeidspartnere.	
Beskrivelse	Å rapportere prosjektstatus til samarbeidspartnere kan være med på å inkludere dem i prosjektet og oppdatere dem på fremgangen. Dette kan igjen føre til engasjement rundt prosjektet.	
Vilkår for aksept	Kriteriet er akseptert når prosjektstatus enkelt kan overføres til en PDF- og/eller Excel fil.	
Argumentasjon	Dette er en «Should have» fordi å engasjere og oppdatere samarbeidspartnere er viktig i et prosjekt, da det kan være disse som ønsker prosjektet gjennomført og sitter på ressursene. Mer engasjement rundt et prosjekt kan føre til at mer ressurser blir satt på det.	

Nr. 6		Prioritet: Should have
Brukerhistorie	Som en prosjektleder ønsker jeg en liste over aktuelle interessenter slik at jeg kan skaffe meg en oversikt over dem og deres betydning for prosjektet.	
Beskrivelse	En fremvisning av interessentene i det aktuelle prosjektet.	
Vilkår for aksept	Kriteriet er akseptert når bruker enkelt kan få en oversikt over interessentene, deres mulige påvirkningskraft og deres interesse i prosjektet.	
Argumentasjon	Dette er en «Should have» da interessenter kan ha stor påvirkningskraft på et prosjekt og hvordan det gjennomføres. Det vil være viktig for brukerne å enkelt kunne se hvem interessentene i prosjektet er, da de kan være med på å påvirke avgjørelser relatert til prosjektet.	

Nr. 7		Prioritet: Should have
Brukerhistorie	Som en prosjektleder ønsker jeg en oversikt over prosjektleveranser og dens status, slik at jeg enklere kan levere et produkt med god kvalitet.	
Beskrivelse	En fremvisning av prosjektleveranser og den status som skal bidra til å kunne forenkle arbeidsprosessen for en prosjektleder.	
Vilkår for aksept	Kriteriet er akseptert når prosjektleder kan på enkel og oversiktlig måte skaffe seg informasjon om prosjektleveranser og dens status.	
Argumentasjon	Dette er en «should have» ettersom en slik oversikt kan være nyttig fordi en prosjektleveranse vil kunne gi informasjon om fremdriften og om hvilken effekt prosjektleveransen gir. Prosjektleveranser vil også kunne gi tilbakemeldinger på de mål som er satt for leveransen.	

Nr. 8		Prioritet: Should have
Brukerhistorie	Som en prosjektdeltaker ønsker jeg å kunne ha tilgang på et gantt diagram slik at jeg kan skaffe meg en visuell oversikt over aktivitetene i et prosjekt.	
Beskrivelse	Et slik diagram har som hensikt å vise fremdrift i forhold til tid, og den viser også hvem som gjør hva, samt oppstart og avslutning av arbeid som må gjøres.	
Vilkår for aksept	Kriteriet er akseptert når en prosjektdeltaker har muligheten til å skaffe seg informasjon om aktiviteter i et prosjekt via et gantt diagram.	
Argumentasjon	Dette er en «should have» fordi et slikt diagram gir et raskt overblikk over prosjektet som det er mulig for den enkelte involverte å forstå.	

Nr. 9		Prioritet: Could have	
Brukerhistorie	Som en prosjektdeltaker ønsker jeg å kunne se listedata fra hvilken som helst liste i et prosjekt slik at jeg kan se egne opprettede lister som ikke følger med prosjektportalen.		
Beskrivelse	Prosjektdeltakere oppretter ofte egne lister for å dekke visse behov i et prosjekt. Dette kan være så spesifikt at det ikke går under noe av den medfølgende funksjonaliteten i Prosjektportalen. En generisk listefremviser vil dermed kunne dekke alle typer lister, og prosjektdeltakeren vil selv bestemme hvilke listedata h*n ønsker å se.		
Vilkår for aksept	Kriteriet er akseptert når en prosjektdeltaker kan velge hvilken som helst liste fra prosjektet og vise dette i et eget vindu. Prosjektdeltakeren burde/skal kunne oppdatere og fjerne elementer fra kolonnene som listen er bygget opp av.		
Argumentasjon	Dette er en «could have» ettersom funksjonaliteten går utenfor selve scopet og kjernefunksjonaliteten til Prosjektportalen. Men ettersom det er en funksjon som prosjektdeltakere mest sannsynlig ville hatt bruk for, vil det være mulighet til implementasjon ved nok tid og ressurser til overs.		

Nr. 10		Prioritet: Could have	
Brukerhistorie	Som en prosjektdeltaker ønsker jeg å kunne lage og bruke dokumentmaler slik at jeg og mine kollegaer kan spare tid, og unngå unødvendig dobbeltarbeid.		
Beskrivelse	Mange dokumenter brukt i et prosjekt har standard oppsett, dermed kan disse dokumentene være lagt inn som maler i prosjektportalen. Hvis et ikkeeksisterende oppsett skal brukes jevnlig kan dette også bli opprettet som en mal.		
Vilkår for aksept	Kriteriet er akseptert når bruker kan laste ned og bruke en mal, samt at nye maler kan bli opprettet og brukes på samme måte.		
Argumentasjon	Dette er en «Could have» fordi dokumentmaler ikke er en nødvendighet for å kunne gjennomføre et prosjekt. Dette er en løsning som forenkler opprettelsen av standarddokumenter, men bruker er ikke avhengig av denne funksjonaliteten for å gjennomføre sitt arbeid.		

Nr. 11		Prioritet: Could have
Brukerhistorie	Som en prosjekteier ønsker jeg å sette opp en kommunikasjonsplan for å ha oversikt over kommunikasjon med aktuelle interessenter.	
Beskrivelse	En fremvisning av kommunikasjon med interessentene i det aktuelle prosjektet.	
Vilkår for aksept	Kriteriet er akseptert når en har mulighet til å legge til/endre/slette kommunikasjonslementer tilknyttet en målgruppe (for eksempel interessenter) med metadata som status, fase, dato, beskrivelse osv.	
Argumentasjon	Dette er en could have ettersom våre undersøkelser viste at dette ikke er en av funksjonene som er mest i bruk, eller er en av de funksjonene brukerne anser som mest sentral i prosjektportalene. Det er for øvrig mulighet for implementasjon ved nok tid og ressurser til overs.	

Nr. 12		Prioritet: Would have
Brukerhistorie	Som en prosjektleder ønsker jeg en oversikt over alle prosjektene i porteføljen slik at jeg kan se status uten å måtte gå inn i hvert enkelt prosjekt.	
Beskrivelse	En forside med oversikt over alle prosjektene man deltar i. Mulighet for å få en generell oversikt uten å trykke seg inn i et spesielt prosjekt.	
Vilkår for aksept	Kriteriet er akseptert når prosjektledere/prosjektdeltakere har muligheten til å se en oversikt over alle prosjektene på en egen side.	
Argumentasjon	Dette er en «would have» fordi status over hvert enkelt prosjekt prioriteres, og en portefølje over alle prosjektene ikke er en nødvendighet for å tilegne seg informasjon om prosjekter. Likevel er det en forespørsel blant prosjektdeltakere, og en porteføljeoversikt kan vurderes etter prioriterte funksjoner er implementert.	

10.3 Risikomatrixe

RISIKO	ALVORLIGHETSGRAD	SANNSYNELIGHET	RISIKOPÅVIRKNING	ANBEFALTE HANDLINGER
En av gruppe medlemmene får corona	Tolerable	Possible	Medium	For å unngå dette kan gruppen være nøye med avtand på fysiske møter, bruke antibac og ikke møte opp ved eventuelle symptomer. Hvi en får corona tar vi møter på Teams, og møtes ikke på en stund for å unngå økt smitte.
En av medlemmene smitter gruppen med corona	Generally unacceptable	Possible	High	Samme som den over. Det er viktig å ikke komme på fysiske møter ved symptomer.
Prosjektet blir slettet	Generally unacceptable	Not likely	High	Ha prosjektet lagret på flere maskiner, i sky og på github.
En av gruppe medlemmene blir syk	Acceptable	Possible	Low	Resten av gruppen jobber som normalt. Den syke blir hjemme og gjør det den orker av arbeid.
Gruppe medlemmene mister motivasjonen	Tolerable	Possible	Medium	Å prøve å ha fysiske møter, være sosiale sammen også utenfor skole.
Puzzlepart får lite tid til å bistå oss i arbeidet ved oppgaven.	Tolerable	Possible	Medium	Prøve å holde en kontinuerlig dialog og ha en sprint review med dem hver tredje uke
Får fysiske plager av for mye arbeid foran skjerm	Acceptable	Possible	Low	Opprettholde god fysisk form
Blir store uenigheter i gruppa	Acceptable	Possible	Medium	Kunne ha en åpen dialog, diskutere og ha gruppekontrakt
Lockdown	Acceptable	Possible	Medium	Være godt kjent med digitale kommunikasjonsplattformer og utnytte alle nødvendig funksjonalitet
Prosjektportalen gjør store endringer, som påvirker vårt arbeid	Tolerable	Not likely	High	Opprettholde kommunikasjon med Puzzlepart om deres planer med prosjektportalen
Dårlig kommunikasjon med Puzzlepart	Generally unacceptable	Possible	High	Får å unngå dårlig kommunikasjon med Puzzlepart blir det opprettet et felles rom i MS Teams der vi kan ha åpen dialaog med bedriften gjennom hele Sprinten. Samt har scrum master ansvar for å opprettholde god kommunikasjon.
Dårlig kommunikasjon internt	Generally unacceptable	Possible	High	For å opprettholde god kommunikasjon internt skal gruppen anvende flere digitale kommunikasjonsplattformer der medlemmene kan kommunisere om arbeidet.
Dårlig kommunikasjon med veileder	Tolerable	Possible	Medium	For å opprettholde god kommunikasjon med veileder skal gruppen inkludere han i sprintmøter hver tredje uke. På disse møtene vil han bli oppdatert på vår fremgang, gruppen stiller spørsmål og får konstruktiv kritikk.
Dårlig planlagte møter med Puzzlepart og veileder	Generally unacceptable	Not likely	Medium	For å unngå å være dårlig forberedt til møter med Puzzlepart setter gruppen møtedata god tid i forveien. Gruppen vil også gjennomføre et internt møtet for å forberede seg.
Dårlig planlagte Sprinter	Generally unacceptable	Possible	Medium	Avslutte hver sprint med sprint retrospective for å se hva som kan forbedres til neste Sprint og hva vi vil ta med videre. Lære av den tidligere timeestimeringen, samt starte sprinten med å planlegge og utarbeide sprintbacklog.