




IT1901

GRUPPE 1

Halvor Horge: 749525
Hanne Brynildsrud: 493403
Magnus Tidemann: 493433
Marius Sjøberg: 494272
Morten Skandsen: 493442
Sander Bjerklund Lindberg: 493389
Sara Haugen: 742239

Antall ord: 5904



Innhold

1 Introduksjon	5
1.1 Prosjektet	5
1.2 Faget	5
1.3 Problemstilling	5
2 Produktet	5
3 Scrum	5
3.1 Gruppens sprint planning	7
3.2 Hvordan kommunikasjon fungerte under sprinten	7
3.3 Hvordan sprint backlogger ble gjennomført	7
3.4 Hvordan arbeidsrommet ble satt opp	8
3.5 Hvordan Daily Scrums foregikk	8
3.6 Hvordan testing ble gjennomført	9
3.7 Hvordan sprint review ble gjennomført	9
3.8 Hvordan sprint retrospective ble gjennomført	9
4 Sprint gjennomføring	10
4.1 Sprint 1	10
4.2 Sprint 2	13
4.3 Sprint 3	15
5 Læringsutbytte	19
Bibliografi	20
A Product Backlog	21
B Releaseplan	22
B.1 Opprinnelig releaseplan	22
B.2 Oppdatert releaseplan	23
B.3 Siste releaseplan	24
C Tasks	25
C.1 Sprint 1	25
C.2 Sprint 2	26
C.3 Sprint 3	26
D Referater	28

D.1	Daily Scrum	28
D.2	Møtereferater	28
D.3	Møte med produkteier	33
D.4	Referat fra retrospectives	36
E	Burndown chart	42
E.1	Sprint 1	42
E.2	Sprint 2	43
E.3	Sprint 3	44
F	Testplan	45
F.1	Testplan for brukerhistoriene i sprint 1	45
F.2	Testplan for brukerhistoriene i sprint 2	45
F.3	Testplan for brukerhistoriene i sprint 3	45
G	Designutkast	47
H	Epost	48
H.1	Epostutveksling med produkteier	48
I	Brukermanual	49
I.1	Rammeverk	49
I.2	Django	49
I.3	Kode	50
I.4	Programmeringsspråk	50
I.5	Sammenheng mellom språkene	50
I.6	UML	55
I.7	Design	56
I.8	Teknisk	57
I.9	Ikke-teknisk	58
I.10	Nyttige tips til nettsiden	60

Tabeller

1	Brukerhistorier sortert etter produkteiers prioritering.	21
---	--	----

Figurer

1	Førsteutkast av releaseplanen.	22
2	Oppdatering av releaseplanen før sprint 2.	23
3	Oppdatering av releaseplanen før siste sprint.	24
4	Vår digitale daily scrum oversikt	28
5	Burndown chart for sprint 1	42
6	Burndown chart for sprint 2	43
7	Burndown chart for sprint 3	44
8	Testplan til sprint 1, ID 1, 2, 3 og 4.	45
9	Testplan til sprint 2, ID 28, 29 og 30.	45
10	Testplan til sprint 3, ID 7, 8, 20, 5, 9 og 10.	46
11	Designutkast	47
12	UML diagram av nettsiden	55
13	Forside av nettsiden	56

1 Introduksjon

1.1 Prosjektet

Denne rapporten bygger på erfaringer og refleksjoner rundt oppgaven gruppen ble tildelt i IT1901 - Informatikk Prosjektarbeid 1, høsten 2018. Prosjektet har blitt gjennomført i forbindelse med bachelorgraden til samtlige deltakere.

1.2 Faget

IT1901 er et fag der studentene blir inndelt i grupper for å gjennomføre et prosjekt som går ut på å produsere en nettside, for en fiktiv produkteier som representeres av en tildelt studentassistent. I gjennomføringen av denne oppgaven skal gruppen skaffe seg erfaringer tilknyttet arbeidsmetode og gjøre seg kjent med rammeverket Scrum.

1.3 Problemstilling

Ved oppstart fikk gruppen utdelt en rekke brukerhistorier fra produkteier. Oppgaven var å benytte Scrum for å utvikle et produkt som produkteier skulle være fornøyd med. Det er verdt å merke seg at faget har fokus på selve arbeidsprosessen og refleksjon rundt denne, i større grad enn produktet.

2 Produktet

Gruppen fikk i oppgave av produkteier å lage et Content Management System (CMS) som skulle støtte gjennomlesing av innhold før innholdet publiseres. Etter samtale med produkteier ble det bestemt at produktet skulle ligne en variant av Pinterest og typiske nettaviser eller bloggplattformer, slik som Wordpress. Produktet valgte vi å kalle Pålterest, blant annet fordi man slik som på Pinterest kan huke av hvilke kategorier man vil lese artikler om, og fordi produkteieren heter Pål.

3 Scrum

Scrum er et rammeverk for smidig programvareutvikling tilegnet små lag. I Scrum er det spesifisert tre roller: Scrum Product Owner, Scrum Master, og Scrum Development Team [Schwaber and Sutherland, 2017]. Product Owner, eller produkteier, representerer produktets interessenter, og kundene som skal bruke produktet. Produkteier har også ansvaret for backloggen, som beskrives senere, samt å maksimere verdien til det utviklingslaget produserer.

Utviklingsteamet, Scrum Development Team, er ansvarlige for å utvikle produktet, og alle andre relaterte prosesser. Dette kan være alt fra koding til testing av ferdig produkt.

Bindeleddet mellom teamet og produkteier er Scrum Master. Den overordnede oppgaven til Scrum Master er å sørge for at alle som er en del av utviklingsprosessen følger rammeverket Scrum. For eksempel er Scrum Master ansvarlig for å holde de faste møtene gjennom utviklingen [Schwaber and Sutherland, 2017].

Utviklingen deles opp i såkalte “sprinter” på 2-4 uker. Disse sprintene starter med hver sin sprintplanlegging. Planleggingen tar for seg en rekke momenter som når/hvor møter skal holdes, hvem som jobber med hvem og med hva, samt estimere hvor lang tid en brukerhistorie kommer til å ta. De viktigste møtene i løpet av sprinten er “Daily Scrums”. Dette er korte, ca. 15 minutter, daglige møter hvor alle på teamet går gjennom hva de gjorde dagen før, om de støtte på problemer, og hva de har planer om å gjøre den dagen [Rising and Janoff, 2000]. Det er hensiktsmessig at dette skjer i samme rom hver dag for å akkommodere et annet viktig aspekt av Scrum, nemlig Scrum-boardet. Vanligvis brukes en tavle til dette som er delt opp i diverse kolonner som korresponderer til de ulike stegene i utviklingsprosessen. Eksempelvis “Stories”, “ToDo”, “Doing”, “Testing” og “Done”. Backlog er noe av det første man fullfører når man starter et nytt prosjekt, dette er da listen over oppgaver som må gjøres. Disse kalles brukerhistorier i Scrum, og er en oppdeling av hele prosjektet. Dette kan representeres med post-it-lapper som har gjøremålet skrevet på seg. Når et medlem av teamet er ferdig med en oppgave kan h*n ta for seg en ny brukerhistorie fra ToDo-kolonnen og flytte den over til Doing-kolonnen. Scrum-boardet lar derfor hele teamet ha god oversikt over prosjektets fremgang og hva som er under utvikling. På Scrum-boardet har man også et Burndown-chart som er en grafisk fremstilling av prosjektets fremgang [Kniberg, 2015, kap. 6].

Videre har man i tillegg en releaseplan. Dette er en oversikt over de ulike sprintene og hva som skal gjøres fra starten av prosjektet til man har et ferdig produkt. For å vite når produktet er ferdig er det viktig med en god definisjon av hva et ferdigstilt produkt er. Dette refereres til i Kniberg [2015, s. 37] som å ha en god “Definition of Done” eller en DoD. Dette er som oftest en liste med en rekke kriterier som teamet har blitt enige om sammen med produkteier, ofte oppbevart i samme rom som Scrum-boardet. Underveis i utviklingen er det viktig å teste de ulike funksjonalitetene til produktet. Teamet følger sin egen testplan gjennom sprinten, men før man kan si at et aspekt ved produktet er ferdig må man også ha en akseptansetest med produkteier. Her får produkteier en mulighet til å godkjenne de delene av produktet som laget mener er ferdige. Blir alle interne tester, akseptansetester, og DoD godkjent er produktet ferdig. Fordi Scrum er et smidig rammeverk kan dette i utgangspunktet skje etter hver eneste sprint. Det er ikke noe i veien for at produkteier og teamet ser seg fornøyd med resultatet før alle sprintene er gjennomført [Kniberg, 2015, kap. 14].

Sprint review er i følge Kniberg [2015, kap. 9] en avsluttende del på sprinten, hvor man holder en demonstrasjon av produktet. Demonstrasjonen innebærer at teamet presenterer sprintmålet og de faktiske ferdige brukerhistoriene og får muligheten til å få tilbakemelding fra produkteier, interessenter og andre Scrum team. Det er viktig å holde denne demonstrasjonen ikke-teknisk. Altså ikke gå inn på detaljer på hvordan ting er kodet og hvordan det fungerer, men heller vise hva som er implementert og hva som virker, ikke hvordan.

Etter en sprint er fullført er det essensielt å ha et Sprint retrospective. I følge Schwaber and Sutherland [2017] bør teamet gjennomgå hvordan de synes sprinten har gått, områder ved arbeidsprosessen som kan forbedres, og hva man kan gjøre for å forbedre disse. Når Sprint Retrospective er ferdig får man et lite pust i bakken (slack) før man starter med sprintplanlegging til neste sprint. Siden man akkurat har sett på hva som gikk bra og dårlig i forrige sprint får man derfor et veldig godt utgangspunkt når man skal planlegge neste sprint. Sprint retrospectivet ledes av Scrum master og foregår gjerne i avslappende omgivelser.

3.1 Gruppens sprint planning

Lengden på sprintene falt veldig naturlig med tidsfristene på innleveringene og demonstrasjonene. Det ble derfor enighet om å ha tre sprinter, hvor hver av dem skulle ha en lengde på rundt to og en halv uke.

Planleggingsmøtene ble gjort uten produkteier, men det har likevel vært mulig å få en forklaring fra han på hva hver av de ulike brukerhistoriene innebar og hvordan han ønsket å prioritere de.

For å estimere hvor lang tid gruppen kom til å bruke på de ulike brukerhistoriene, ble det brukt en small-medium-large skala (1, 2, 3 for enkelhetens skyld). Under første planleggingsmøte gikk dette kun på magesfølelse, mens på de to neste var det mulig å bruke tilegnet erfaring fra de foregående arbeidsperiodene. Dette kalles i følge Kniberg [2015] for 'yesterday's weather', eller 'gårsdagens vær'.

3.2 Hvordan kommunikasjon fungerte under sprinten

I Kniberg [2015, kap. 5] drøftes det hvordan hele teamet skal kunne holde oversikt over sprinten i et dokument, en såkalt sprint infoside. Denne skulle ifølge Scrum inneholde målet med sprinten, sprint-backloggen for den kommende sprinten, når og hvor møtene skulle holdes, og hvem som deltok på sprinten. Dette har blitt gjort litt annerledes siden gruppen er relativt liten. Alle har hatt oversikt over sprinten og arbeidsoppgavene via GitLab og Google Drive, og siden arbeidet har blitt gjennomført sammen i møterom, hvor det har blitt gjennomført Daily Scrum på starten av hvert møte, har eventuelle misforståelser raskt blitt oppklart. I tillegg til dette ble det opprettet en gruppechat i Messenger, hvor det var mulig å gi beskjed om man for eksempel ble forhindret i å komme på et møte.

3.3 Hvordan sprint backlogger ble gjennomført

I Scrum brukes vanligvis en stor tavle eller vegg som Sprint Backlog. Backloggen skulle inneholde en ToDo-kolonne, en Doing-kolonne og en Done-kolonne, samt et Burndown chart. Siden det ikke har vært mulig å ha et fast møterom hadde det vært særdeles problematisk å gå rundt med en stor tavle med Post-It lapper på. Av den grunn ble det i stedet brukt GitLab som en digital Backlog. Burndown charts ble gjort på PC og tok utgangspunkt i datoene hver ID ble ferdigstilt (se burndown charts i E).

3.4 Hvordan arbeidsrommet ble satt opp

Fordi samtlige medlemmer av gruppen er studenter, og ikke fast ansatte i en bedrift, har det ikke vært mulig å ha et eget arbeidsrom. Det har dermed ikke vært mulig å ha en “sprint-wall” eller “whiteboards” slik Kniberg [2015, s. 68] anbefaler. Derimot har gruppen nærmest alltid hatt et eget grupperom eller vært samlet når det har blitt arbeidet med prosjektet. På denne måten har det altså vært mulighet til å oppfylle kravet om «å være sammen» når det arbeides:

- Det har vært mulig å snakke med hverandre uten å forlate arbeidsplassen.
- Alle har kunnet se hverandre og taskboardet siden det har vært lett tilgjengelig i en felles skyløsning.
- Teamet har hatt en skjerming fra omgivelsene, slik at det har vært mulig å jobbe uforstyrret.

Siden produkteier også er student har det ikke vært mulig å ha produkteier i samme rom som teamet. Likevel har det fortsatt vært mulig å komme raskt i kontakt med han via mail og et fast ukentlig møte.

3.5 Hvordan Daily Scrums foregikk

Daily Scrum har blitt gjennomgått med de samme oppgavene og målene som beskrevet i Schwaber and Sutherland [2017]. Når dette er sagt har gruppen gjort en god del modifiseringer for å tilpasse møtene til samtliges ulike timeplaner og situasjoner. En av hovedforskjellene har vært frekvensen. Siden det kun har vært mulig for gruppe-medlemmene å møtes tre ganger i uken har det derfor kun blitt gjennomført tre Daily Scrums ansikt til ansikt. Derfor har det i tillegg blitt brukt et digitalt Scrum Board, og en digital oversikt over hva som jobbes med og om noen har støtt på problemer. Dette har blitt gjort gjennom et “Issue Board” på GitLab, og gjennom en Daily Scrum oversikt på et felles Excel-dokument (se D.1). På GitLab har “tavlen” vært oppdelt i fire kolonner. En av kolonnene har inneholdt alle brukerhistoriene for prosjektet. For hver sprint har historiene som det er planlagt å jobbe med blitt flyttet over i “ToDo”-kolonnen. Etterhvert som gruppen har tatt på seg oppgavene, har de blitt flyttet over i “Doing”-kolonnen, akkurat som på et fysisk Scrum-Board.

I Excel-dokumentet har det som nevnt tidligere vært en skriftlig versjon av Daily Scrums, se D.1. Her har hvert teammedlem hatt sin egen kolonne med ukedagene under. For hver dag er det tre rader, én for hva vedkommende gjorde dagen før, om de støtte på problemer/om noe hindret de i å gjennomføre oppgaven, og hva de skal gjøre den dagen. I tillegg til en kort skriftlig beskrivelse har dokumentet blitt fargekodet slik at man lett kan se om det har oppstått problemer for noen av de andre på teamet. En annen fordel med fargekodingen er at det blir lettere å lokalisere hvor det har oppstått problemer når “Sprint Retrospective” skal gjennomføres.

3.6 Hvordan testing ble gjennomført

Tester ble laget for hver brukerhistorie (se F for testplan), med korrekt utførelse av hvordan testen skulle gjennomføres for at den skulle bli godkjent, og en oppskrift til brukeren på hva som skulle gjøres. Før det ble gjennomført akseptansetesting med produkteier ble det gjennomført alpha-testing [Rouse, 2018] innad i teamet, og beta-testing ute hos reelle brukere (samboere i kollektiv). Dette ble gjort før det ble gjennomført akseptansetesting slik at det var mulig å avdekke eventuelle feil og mangler som mulig hadde blitt oversett underveis i utviklingen av produktet. Mot slutten av hver sprint ble det gjennomført akseptansetesting med produkteier for å få feedback, og for å sjekke om kravene slik gruppen hadde kommet til enighet med produkteier om var tilstrekkelige. Dersom akseptansetesten til en ID ikke ble godkjent, hadde produkteier ofte konkrete tilbakemeldinger om hva som burde utbedres, og dermed var det enkelt å sette seg ned igjen for å utbedre feilen før testprosessen ble gjennomført på nytt igjen.

3.7 Hvordan sprint review ble gjennomført

I følge Kniberg [2015, kap. 9] bør sprint review involvere toveiskommunikasjon, hvor teamet demonstrerer de ferdige brukerhistoriene og får tilbakemelding fra produkteier og de andre som overværer demonstrasjonen. Sprint review har blitt gjennomført ved å demonstrere produktet i form av enveiskommunikasjon for produkteier og/eller andre studentassistenter. Grunnen til dette er at denne demonstrasjonen har vært en del av vurderingsgrunnlaget i faget og tilbakemeldingene som har kommet inn i etterkant av demonstrasjonen har gjenspeilet denne og ikke selve produktet.

3.8 Hvordan sprint retrospective ble gjennomført

Etter hver sprint ble det dedikert et møte til Sprint Retrospective. Scrum-masteren tok en rask oppsummering av sprinten, og hvordan gruppen lå an med backloggen. Først ble det satt opp kolonner på en tavle med hva som gikk bra, hva som kunne gått bedre, og hvordan man kunne forbedre dette til neste sprint. Samtlige medlemmer fikk mulighet til å komme med innspill, og gi konstruktive tilbakemeldinger. Etter dette ble alle brukerhistoriene gjennomgått for å se om det hadde blitt feilestimert hvor lang tid vi hadde planlagt å bruke. Mot slutten av retrospektivet kom var det viktig å gi ros til oss selv om hva/hvem som skilte seg ut, før Scrum-masteren tok en rask oppsummering rundt hva som burde fokuseres på til neste sprint.

4 Sprint gjennomføring

4.1 Sprint 1

4.1.1 Sprint planning

Gruppen hadde lite/ingen erfaring med Django, og hvor lang tid de ulike brukerhistoriene ville ta. Første punkt på agendaen ble derfor å gjøre seg kjent med Django. Dette ble gjort fordi det er veldig viktig å ha gode estimater for å kunne lage en pålitelig releaseplan. Når samtlige på gruppa hadde gjennomført oppsett av Django var neste punkt å sette størrelsesestimater på alle brukerhistoriene. Her ble tidsbruken rangert fra 1-3, dvs. small, medium eller large (se backlog med alle brukerhistorier i A og estimatene er oppført i releaseplan i B.1).

På grunn av manglende erfaring var det vanskelig å vurdere omfanget til de ulike brukerhistoriene. Et par brukerhistorier fikk av den grunn estimater først etter møtet med produkteier, da en oppklaring var nødvendig for å sikre at produkteier ville bli fornøyd.

Deretter ble det en gjennomgang av produkteiers tolkning av hver ID for å forsikre at det ikke var noen misforståelser. Med grundigere forståelse ble de siste estimatene satt, og med det kunne releaseplanen ferdigstilles. Produkteieren var tilfreds med antallet brukerhistorier per sprint (se releaseplan i B.1). I utgangspunktet hadde han allerede rangert og bestemt rekkefølgen på brukerhistoriene, og han hadde ingen siste modifikasjoner for å inkludere noen flere brukerhistorier i sprint 1. Brukerhistoriene ble deretter brutt ned i mindre oppgaver, også kalt tasks, for å få en oversikt over hvilken funksjonalitet som måtte implementeres før brukerhistorien kunne deklarerer som ferdig. Disse finner man i oversikten over tasks C.1.

Målet med sprinten ble satt til å lage en oversiktlig nettside med mulighet for forfattere å publisere artikler.

4.1.2 Sprint iteration

Første iterasjon delte gruppen seg opp i to, to og tre personer. To av undergruppene startet med å bli enda bedre kjent med Django og henholdsvis brukerhistorie 1 og 2, mens den siste undergruppen lagde et designutkast (se G) og begynte med en CSS-fil. Utover i sprinten fortsatte hver undergruppe eller hvert gruppemedlem å ta på seg nye tasks (se oversikt over tasks i C.1) når man var ferdig med den man holdt på med. Når man ble ferdig med implementasjonen av en brukerhistorie, testet man funksjonaliteten etter testplanen (se F.1). Det var et par ganger denne ikke ble bestått og måtte forbedres før den kunne ferdigstilles.

Etter hvert som brukerhistoriene ble ferdige ble det en glidende overgang til å jobbe med demonstrasjonen av produktet samt et rapportutkast om arbeidsmetode og prosess som skulle leveres inn samtidig med demonstrasjonen.

Burndown chart for denne iterasjonen finner man i E.1. Der ser man blant annet et faresignal mellom dag 4 og dag 8 i sprinten. Grunnen til at grafen er flatet ut der er at det var problemer med GitLab og derfor gikk det litt tregt med

implementasjonen av brukerhistoriene.

4.1.3 Sprint testing

Gruppen satte opp et dokument med oversikt over hvordan hver brukerhistorie kan testes (se F). Det ble gjennomført både egen testing av hver brukerhistorie i sprinten og en akseptansetest sammen med produkteier.

Da det ble gjennomført en akseptansetest av brukerhistorie 1-4 fungerte ikke innloggingsfunksjonen som den skulle, og ID 1 og 2 kunne av den grunn ikke testes fullt ut. Dette var en av manglene produkteier fant, og noe som burde blitt oppdaget under egentesting. Produkteier hadde også to andre bemerkninger. Han ville ha mulighet til å **ikke** legge ved et bilde når han skulle lagre et utkast, noe gruppen så seg enige i. Han ønsket også mer design på siden, da det kun var brukt en helt basic template.

Til demonstrasjonen på slutten av sprinten var manglene nevnt over utbedret og brukerhistorie 1-4 tilfredstilte gruppens DoD.

4.1.4 Sprint review

Demonstrasjonen ble holdt for 3 studentassistenter. Her ble den oppdaterte releaseplanen vist, samt resultatene fra akseptansetesten med produkteier, alle de 4 implementerte brukerhistoriene, og at hver av de tilfredstilte testplanen vår. Et punkt som kunne forbedret demonstrasjonen av testingen hadde vært å ta med hvordan produktet håndterte feil, for eksempel:

- Ikke bare vise at man får laget en artikkel når man er innlogget som en author, men også vise at man ikke får laget en artikkel hvis man er innlogget som f.eks. en registered user.
- Vise at man må ha en brukerkonto for å logge inn og at man ikke kan nå siden for å lage en artikkel uten å være logget inn med en brukerkonto som har rettigheter til det.

4.1.5 Sprint retrospective

Se fullstendig referat fra sprint retrospective i D.4.1.

Det som gikk bra med sprinten var:

- Vi har hatt møter tre ganger i uka, slik at vi har hatt muligheten til å parprogrammere og stille spørsmål til de andre på gruppa om det skulle være noe man lurte på.
- Vi har vært fleksible, fornuftige og gode på å diskutere, slik at vi har kommet frem til gode løsninger, og det har ikke vært noe krise om det er noen som ikke har mulighet til å møte en dag. Vi har lagt egne interesser til side for å skape et best mulig produkt, og for å samarbeide om et felles mål.

- Vi er så langt fornøyde med produktet.
- Arbeidskontrakten har blitt fulgt til en tilfredsstillende grad, og vi har hevet sanksjonene ved kontraktbrudd ved et smil.
- Vi har vært fornøyde med estimatene da vi ble ferdige med alle brukerhistoriene i perfekt tid før sprinten var over.
- Bra med booking av rom slik at vi alltid har hatt et sted å arbeide i stillhet sammen med gruppa.

Det som kunne gått bedre med sprinten var:

- Vi var ikke fullt så drevne på Git og GitLab, og det gikk mye til til pushing og merge conflicts, og det oppstod forvirring rundt de forskjellige branchene.
- Scrumboardet gikk dårlig, vi flyttet ikke ting frem og tilbake når vi skulle. Det var ikke alltid så lett å vite når funksjonene var ferdig, og ofte ble de jobbet med i flere dager i strekk.
- Vi har vært dårlige på referering fra tidligere møter, så dette må vi bli flinkere på, da rapporten avhenger svært mye av dokumentasjon.
- Vi var ikke så veldig flinke på å kommunisere internt om hvem som gjorde hva på møtene, og det tok litt tid før vi kom igang.
- Vi har ikke vært fornøyde med Daily Standup, fordi vi har brukt mye tid på å komme igang.

Forbedringer til neste sprint:

- Lære oss mer Git og GitLab, slik at det ikke går unødvendig tid til dette.
- Bryte ned brukerhistoriene ID-ene i mindre oppgaver, og legge dem inn i GitLab, slik at man kan se hva som må gjøres og hva som jobbes med.
- Lage en ny testing-liste i Scrumboardet, som skal inneholde testene til hver brukerhistorie.
- Daily Scrum kan bli bedre oppdatert, så om vi kan bruke ToDo-tasks kan dette kanskje hjelpe med at gruppen til enhver tid har noe å gjøre. Daily Scrum-boardet er viktig å få oppdatert før hver arbeidsdag, så vi kan pushe på for å fylle inn dette ved starten av hvert møte da vi møtes. Om vi finner ut av at dette mot formodning ikke fungerer, kan vi ta en avstemning på hva vi skal gjøre neste.
- Vi kan lage en oversikt over hvem som refererer under hvert møte, slik at alle får være referent og slik at vi er sikre på at hvert møte blir dokumentert.
- Et eget dokument til møtet med produkteier hver onsdag.

4.2 Sprint 2

4.2.1 Sprint planning

Til sprint 2 kom produkteier med ytterligere 13 brukerhistorier, hvorav tre stykk skulle ha maks prioritet. Sprinten startet med en gjennomgang av brukerhistoriene sammen med produkteier for å komme fram til en felles forståelse av hva hver av de innebar. Deretter satte gruppen seg ned uten produkteier og gjorde estimer for hver av brukerhistoriene.

I følge releaseplanen (B.1) skulle sprint 2 ta for seg brukerhistorie 7, 8, 20 og 5, med et total estimat på 5. Siden brukerhistorie 28, 29 og 30 skulle ha førsteprioritet og de til sammen hadde et estimat på 6, ble gruppen nødt til å endre estimatene og sprint 2 til å ta for seg brukerhistorie 28, 29 og 30 i stedet for 7, 8, 20 og 5 (se B.2).

Målet for denne sprinten var å implementere en ny rolle på nettsiden.

I retrospektiven etter første sprint var et av punktene som kunne vært utført bedre å bryte ned brukerhistoriene i mindre oppgaver (tasks) på scrum-boardet. GitLab ble derfor oppdatert med tasks fra C.2, og samtlige syntes med en gang at de hadde bedre oversikt over hva som skulle gjøres.

4.2.2 Sprint iteration

Før sprinten hadde gruppen levert et rapportutkast som de fikk tilbakemelding på under møte med produkteier (kommentarene kan ses under D.3.2 Møte med produkteier 17. oktober). Gruppen hadde god erfaring med par-programmering og det å dele opp i par innad i gruppen. Én person brøt ned brukerhistorie 28, 29 og 30 i mindre oppgaver/tasks for deretter å starte med koding, to gruppemedlemmer startet med implementasjon av brukerhistorie 28, to stykk fortsatte med CSS og to personer startet å forbedre rapportutkastet.

Sprint 2 skulle opprinnelig være ferdig mandag 22. oktober, men siden møtet med produkteier for en akseptansetest først skulle holdes onsdag 24. oktober, ble gruppen enige om å utsette fristen til onsdagen. Da en akseptansetest uansett var essensiell for å fullføre sprinten. Dette i tandem med at produkteier måtte konsulteres om hvilke brukerhistorier som skulle prioriteres, resulterte i at sprinten ble startet noe sent. Dette kan også ses i burndown chart i E.2, da sprinten kun besto av 6 dager på grunn av den forsinkede planleggingen av sprinten.

4.2.3 Sprint testing

Akseptansetesten med produkteier ble tatt på møtet 24. oktober (se D.3.2). Han hadde noen generelle tilbakemeldinger:

- Vi burde navngi ting tydeligere og lage en bedre substruktur på siden, det vil gjøre det klarere hvor man skal trykke for å komme seg inn på de ulike sidene.

- Hadde vært mer logisk dersom “executive links”-siden ikke hadde så flat struktur.
- “Home”-knappen bør ikke føre til der man lager artikkel, men heller der man ser alle artiklene.
- Ha en knapp med nye tags inne på lag-artikkel-siden.
- Mer design.

Oppsummert så godkjente han brukerhistorie 28 og 30, men ikke 29 da det var en viktig funksjon han ikke kunne se at eksisterte (en manglenede link-kobling). Generelt så ønsket han bedre navngiving og en dypere struktur samt mer design, men funksjonaliteten var god.

4.2.4 Sprint review

Det ble ikke holdt en egen demonstrasjon for hverken produkteier eller andre scrum teams. Gruppen sørget selv for at produktet tilfredstilte definisjon av ferdig som ble satt. Dette ble gjort ved å blant annet fikse den funksjonen som produkteier påpekte at manglet for at ID 29 skulle bli godkjent.

4.2.5 Sprint retrospective

Se fullstendig referat fra sprint retrospective i D.4.2.

Det som gikk bra med sprinten var:

- Vi føler selv at vi har jobbet bra, siden vi har grupperom på alle møtene og alle vet hva de skal gjøre når de kommer på møtene. Scrumboardet vårt ble bedre brukt enn det det ble forrige sprint, både det å bryte ned IDene i mindre oppgaver.
- Vi har blitt enda bedre til å oppdatere Daily Scrumboard.
- Vi har tatt tak i punktene til forbedringer fra forrige sprint, og faktisk forbedret disse på denne sprinten, slik at referent har blitt fastsatt, arbeidsoppgavene er bedre fordelt og folk har alltid ting de skal gjøre.
- Gjennomgangen med produkteier har også forbedret seg.

Det som kunne gått bedre med sprinten var:

- Vi bruker fortsatt mye unødvendig tid på Git og GitLab. Dette er tid som kunne ha blitt brukt til andre ting. Vi prøver å gjøre det så godt vi kan, men det er tidkrevende og tar en del tid fra å faktisk kunne jobbe med nettsiden.
- Oppmøtetidene er ikke helt ideelle, siden det er tidlig mandag og sent fredag. Det høres jo bra ut å ha møte annenhver dag, men dersom man er bortreist over en helg blir det vanskelig å møte på noe annet enn onsdagsmøtet. Vi får ikke gjort noe med dette nå men er noe å tenke på til fremtidige prosjekt.

- Selv om vi har blitt flinke til å oppdatere Daily Scrumboardet, så har vi ikke vært like flinke til å ha oversikt over hva andre har problemer med eller jobber med. Det er uansett vanskelig å hjelpe de på gruppa dersom de har problemer med f.eks. Git, da ingen av oss er veldig flinke på dette.
- Vi fikk litt dårlig tid opp mot deadline, siden vi glemte litt av den første delen av Scrum, fordi vi kun har fokusert på arbeidet. Gikk dermed litt tilbake i tradisjonell prosjektjobbing.
- Kommunikasjonen med veileder kan forbedres. Den trenger ikke nødvendigvis kun være hver onsdag, og den kan også tas over mail.

Forbedringer til neste sprint:

- Vi burde øke kunnskapen og kompetansen rundt Git og GitLab, men vi har ikke mulighet eller tid til å gjennomføre dette før prosjektet er over. Så lenge vi får det til å funke sånn at vi får vist det ved demonstrasjonen sir vi oss fornøye.
- Vi bør ha et tidsrom på 5 minutter på starten av møter der hver person sier sin Daily Scrum, slik at alle får en oversikt over problemene og arbeidsoppgavene til hele gruppa.

Ros:

- Veldig bra at vi har tatt til oss forbedringene fra forrige gang, vi føler at arbeidsflyten går mye jevnere. Møteromsansvarlig gjør en fantastisk jobb! Vi er løsningsorienterte, konstruktive og ser fremover fremfor å se bakover.
- Morten og Sander har gjort en fantastisk jobb med Django-problemer, og det har hjulpet på at vi har fått en fast referent som referer på møter. Det gjør det mye lettere å ha oversikt over prosjektet og fremgangen.

4.3 Sprint 3

4.3.1 Sprint planning

Tredje og siste sprint skulle være ferdig til 7. november, hvor siste akseptansetest med produkteier skulle finne sted. Demonstrasjonen og rapportinnlevering skal foregå 14. november.

Siden det ble endring i prioritering av brukerhistoriene under forrige sprint, vil også denne sprinten inneholde andre brukerhistorier enn hva som var opprinnelig planlagt i følge releaseplanen (se releaseplanene i B). De brukerhistoriene som nå sto for tur var de som opprinnelig skulle blitt gjort i sprint 2, nemlig ID 7, 8, 20 og 5.

Etter en diskusjon av brukerhistoriene ble gruppen enige om at ID 7 og 8 så å si var gjort. Brukerhistoriene ble derfor reestimert da gruppen kunne bruke "gårsdagens vær" (bruke erfaringer fra tidligere sprinter) samt det faktum at det hadde blitt implementert en del funksjonalitet som var gjennomgående for brukerhistoriene. Fordi to av fire brukerhistorier var delvis implementert så gruppen at det var mulig å ta med en ID til. Neste på prioriteringslisten

var ID 6, men den hadde et stort størrelsesestimat. Med bakgrunn i at ID 9 allerede var delvis implementert og ID 10 hadde et passende størrelsesestimat, ble det derfor sendt en mail til produkteier (se H.1) for å spørre om han sa seg enig i å ta med et par ekstra brukerhistorier selv om de hadde noe lavere prioritet.

Sprint 3 endte dermed med brukerhistorie 7, 8, 20, 5, 9, 10 (se oppdatert releaseplan B.3), hvorav ID 7, 8 og 9 bare trengte et par justeringer og testing før de kunne bli sett på som ferdige. ID 9 og 10 ble prioritert over ID 6 på grunn av størrelsesestimatet. Målet for denne sprinten var å gjøre produktet mer brukervennlig.

Videre ble brukerhistoriene brutt ned i tasks som ble lagt til på GitLab (se C.3). Testplanen ble oppdatert med blant annet riktig utfall dersom tester/brukeren gjør noe annet enn det som er riktig (se for eksempel test av ID 7 i testplanen (F)), noe som var savnet i demonstrasjonen av sprint 1 (4.1.4).

4.3.2 Sprint iteration

Første møte startet med en oppdatering av daily standup skjemaet (se D.1) og en rask muntlig fremstilling av hva hver person hadde tenkt til å gjøre denne dagen, noe som kunne ha blitt gjort bedre i forrige sprint.

Siden parprogrammering hadde fungert så bra, var dette en arbeidsmetode som ble opprettholdt også i siste sprint. Deretter delte gruppen seg i tre mindre grupper. En delgruppe begynte med å gjøre ferdig brukerhistorie 7 og 8 som allerede hadde mye funksjonalitet implementert. En delgruppe startet å gå gjennom kommentarene på rapportutkastet en annen gruppe hadde skrevet. Den siste delgruppen fortsatte med CSS. Når en person eller delgruppe ble ferdig med en oppgave kunne de lett finne nye oppgaver i scrum-boardet vårt på GitLab, fordi tasks til brukerhistoriene lå der (se C.3 for en oversikt over tasks).

Burndown chart for sprint 3 finner man i E.3. Dette var den sprinten hvor arbeidet med brukerhistoriene gikk mest uforstyrret og konstant sammenlignet med de to andre sprintene. En av grunnene til dette var at gruppen hadde tilegnet seg erfaring fra de foregående iterasjonene.

På slutten av sprinten ble det klart at noen brukerhistorier hadde blitt gjort indirekte. De hadde falt naturlig på plass da de planlagte brukerhistoriene i releaseplanen ble implementert, siden de var ganske like i funksjonalitet eller de var som ID 23 og handlet om rettigheter (se produkt backloggen i A). Siden mange av brukerhistoriene overlapper hverandre, ble brukerhistorie 11, 14, 20, 23 gjennomført indirekte via implementasjon av de opprinnelige brukerhistoriene for sprinten.

4.3.3 Sprint testing

Gruppen utførte også egen testing, hvor et gruppemedlem testet ut siden og all funksjonalitet mens han var innlogget som ulike brukere. Problemer oppdaget under testing:

- Back på draft linket til feil side

- Kunne ikke save draft/publisere draft
- Author-links - create post linket til feil
- Next knapp på forsiden fungerte ikke

Disse ble rettet opp i før akseptansetesten ble gjennomført.

Det ble gjennomført en akseptansetest på to ulike møter med produkteier. Den første akseptansetesten involverte brukerhistorie 7 og 8. ID 7 ble godkjent, men det var en del bugs som burde bli fikset. Produkteier godtok ikke ID 8, fordi meldingen “You are not authenticated” kom når man prøvde å legge til en tag innlogget som en admin. Produkteier hadde i tillegg en kommentar relatert til poster som er skrevet av flere brukere. Han anbefalte å ha en liste med alle authors og editors med rettigheter slik at det vil være lettere å velge hvilke brukere som har jobbet på samme post (slik at man slipper å huske et helt brukernavn).

Den andre akseptansetesten involverte brukerhistorie 29 (igjen), 8 (igjen), 20, 5, 9 og 10. ID 8, 20, 5 og 9 ble godkjent med en gang. ID 29 ble ikke godkjent på akseptansetesten for sprint 2 (se 4.2.3), men funksjonaliteten ble fikset slik at den var “ferdig” ut i fra vår definisjon. Men etter som det har blitt implementert ny funksjonalitet senere, har dette påvirket ID 29 sin funksjonalitet negativt og det gikk ikke an å lagre en artikkel lenger, derfor ble ikke denne IDen godkjent. ID 10 hadde samme problem. Disse problemene ble fikset senere samme dag og dette gjorde det mulig å kjøre en ny akseptansetest med produkteier senere på dagen. Denne gangen ble ID 29 og 10 godkjent uten påpekninger.

4.3.4 Sprint review

Demonstrasjon av siste sprint og dermed det ferdige produktet vil foregå etter innleveringen av denne rapporten.

Fokuset på siste demonstrasjon vil være en oppdatert releaseplan, med en oversikt over hvilke brukerhistorier som inngikk i de tre sprintene, og en gjennomgang av de viktigste brukerhistoriene implementert samt vise at de tilfredsstiller vår plan for testing av løsningen.

Fra forrige demonstrasjon var det et forbedringspotensiale når det kom til å vise hvordan produktet håndterte feil, det vil si vise hva som skjer hvis f.eks. en bruker uten innloggingsdetaljer prøver å logge seg inn (se 4.1.4). Dette er noe som vil bli fokusert på i den siste demonstrasjonen.

4.3.5 Sprint retrospective

I alt består produkt backloggen av 33 brukerhistorier, hvorav 13 planlagte brukerhistorier har blitt implementert og 4 ikke-planlagte brukerhistorier ble implementert indirekte ved implementasjon av de planlagte. Dette medfører at det er 16 brukerhistorier med tilhørende funksjonalitet som ikke vil bli implementert i produktet. Produktet vil dermed ha mindre funksjonalitet enn hva produkteier har forespurt. Grunnen til at ikke alle brukerhistoriene har

kunnet blitt gjort er i stor grad på grunn av kapasiteten til teamet. Dersom all funksjonalitet skulle blitt utviklet, ville teamet sannsynligvis trengt minst tre sprinter til, det vil si dobbelt så lang tid. Gruppen føler ikke at det å følge scrum-rammeverket har vært til hinder for å gjennomføre flere brukerhistorier, det har heller vært i stor grad en hjelp, på grunn av møtemetodikken, iterasjonsplanlegging, den dynamiske gruppedynamikken og den smidige utviklingsprosessen.

I og med at denne sprinten er den siste i dette prosjektet, blir forbedringer til neste sprint heller forbedringer gruppen kan ta med seg til nye prosjekter i fremtiden. Det gjelder for så vidt alle sprint retrospektiver.

Se fullstendig referat fra sprint retrospective i D.4.3.

Det som gikk bra med sprinten var:

- Fin produktivitet i gruppen, noe som Daily Standups har bidratt mye til.
- God stemning med kake og burger-onsdag.
- Vi har alltid hatt rom.
- Vi har fylt ut Daily Standups ved hvert møte.
- Vi har forbedret mange av forbedringspunktene fra forrige sprint.
- Vi har blitt bedre på egen testing slik at produkteier ikke blir første tester. Dermed har vi oppdaget egne feil mye tidligere i arbeidsprosessen.
- Kommunikasjonen med veileder har blitt forbedret.

Det som kunne gått bedre med sprinten var:

- Vi har ikke alltid hatt konkrete mål til arbeidsdagene, noe som har gått utover produktiviteten.
- Dårlig oppfølging av kontrakten på PowerKing- og New Energy-straffene. Men når dette er sagt har vi hatt veldig god gjennomføring av kake-straffene.
- Git og GitLab
- Scrumboardet i GitLab har ikke blitt utnyttet til sitt fulle potensiale.

Hvordan dette kunne blitt forbedret dersom det hadde hatt en sprint til:

- Daily Standups burde være mer konkret. Som for eksempel å ikke skrive "jobbet med brukerhistorie 29", men heller konkret hva man gjorde med brukerhistorie 29 og hvordan.
- Fokus på håndheving av kontrakten.
- Få gruppa til å melde seg på kurs i Git og GitLab, eller få en i gruppa til å kun fokusere på dette slik at den personen kan håndtere problemene som oppstår.

5 Læringsutbytte

Når gruppen nå er ferdig med prosjektet, koden er levert inn og rapporten begynner å bli ferdigskrevet kan samtlige medlemmer endelig senke skuldrene litt. Selv om Scrum som rammeverk virket stort å sette seg inn i ved starten av semesteret, har gruppen erfart at denne arbeidsmetoden gjør kommunikasjonen og produktutviklingen innad, og opp i mot produkteier, mye smidigere enn hva det har vært ved tidligere prosjekter. Gruppen har erfart nyttigheten av å dokumentere arbeidet underveis og å ha en gjennomgående kommunikasjon med gruppen fremfor slik vi tidligere har gjennomført prosjekter med en typisk “hver mann for seg selv” tilnærming.

Retrospektivene er noe samtlige har fått utnyttet godt. Det å ha en oppsummering etter hver sprint der alle kan reflektere rundt eventuelle utfordringer og problemer underveis har kraftig økt gruppens effektivitet i sprintene. I tillegg gjør retrospektivene det veldig lett å lære av sine feil, og ikke minst utbedre dem til kommende sprinter. Det at Scrum i tillegg er et smidig rammeverk og ikke en metodikk som tvinger fram en spesifikk fremgangsmåte gjør det lett å tilpasse den til hverdagen som studenter der det ikke alltid passer å jobbe åtte til fire eller ha Daily Scrum hver dag. Gruppen har møtt på en rekke utfordringer når det kommer til Git og GitLab, på grunn av manglende forkunnskap om det, men har erfart nytten av å bruke det som et verktøy og samtlige medlemmer er ivrige etter å lære mer om systemet. Alt i alt har dette prosjektet gitt oss en nyttig innføring til effektivt teamarbeid innenfor systemutvikling.

Referanser

H. Kniberg. *Scrum and XP from the Trenches*. Lulu. com, 2015.

L. Rising and N. S. Janoff. The scrum software development process for small teams. *IEEE software*, 17(4):26–32, 2000.

M. Rouse. alpha testing. URL: <https://searchsoftwarequality.techtarget.com/definition/alpha-testing>. [12.11.2018], 2018.

K. Schwaber and J. Sutherland. Scrumguiden, den definitive guiden til scrum: Spillereglene. URL: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Norwegian.pdf>. [09.11.2018], 2017.

A Product Backlog

Tabell 1: Brukerhistorier sortert etter produkteiers prioritering.

ID	User story
1	As an author, I want to create content, so that I can add it to the site.
2	As an author, I want to submit my content, so that editors can read and publish it.
3	As an unregistered visitor, I want to view content, so that I can consume content I am interested in.
4	As an unregistered visitor, I want to browse content, so I can choose what to consume.
28	As an executive editor, I want to create categories, so that the content can be organized.
29	As an executive editor, I want to assign unpublished content to copy editors so that they can proofread it.
30	As an executive editor, I want to unpublish content, so that it can be taken off the site.
7	As an admin, I want to assign and manage the roles of registered users, so that I can control what they can do on the site (what rights they have).
8	As an admin, I want to do everything the other roles are able to do.
20	As a registered user, I want to update my profile, so that I can keep my information up to date.
5	As a copy editor, I want to give feedback on the authors content by commenting on the pages, so that I can review content.
6	As an unregistered visitor, I want to register and log in to the page, so that I can create a personalized feed.
9	As a registered user, I want to be able to do everything the unregistered visitors can do.
10	As a copy editor, I want to see a list of all content that needs proofreading and assign the work to myself, so that all copy editors know what work is in progress.
11	As a copy editor, I want to edit contents I have editor permission to, so that I can edit if necessary.
12	As a registered user, I want to save content, so I can find it later.
13	As an admin, I want to manage and/or customize the look of the website.
14	As an author, I want to give content categories, so they can be organized.
15	As a registered user, I want to be able to subscribe to both categories and authors, so I can see all content that match a category or content.
16	As a registered user, I want to get a view of new contents from my subscriptions, so I can see what is new.
17	As an author, I want to edit content, so that I can keep the page updated.
18	As an author, I want to delete unpublished content drafts, so unnecessary content is not stored.
19	As authors, we want to be able to make content together, so that all of us are credited.
21	As an executive editor, I want to decide if content can be published or not, so that all published contents are well suited for the page.
22	As a copy editor, I want to mark content as complete or incomplete, so that it is ready to be published.
23	As an executive editor, I want to do everything the copy editor can do.
24	As a registered user, I want to be able to request the admin to be a different role, so I can have different permissions.
25	As an author, I want to add images to my content.
26	As an author, I want to choose the layout of my page based on a set predefined layouts, so that I can adjust the look of the content.
27	As a registered user, I want to comment on and rate content, so that I can give feedback.
31	As an executive editor, I want to analyze user profile data and what they like, comment, and subscribe to, so I can analyze statistics to tailor future content to target groups.
32	The site should include search functionality.
33	As an unregistered visitor, I want to share content on social media, so that I can share with friends and family.

B Releaseplan

B.1 Opprinnelig releaseplan

Importance	ID:	Estimate
Sprint 1		
200	1	2
190	2	1
180	3	1
170	4	2
Ferdig: 07.10.2018		
Sprint 2		
160	7	2
150	8	1
140	20	1
130	5	1
Ferdig: 22.10.2018		
Sprint 3		
120	6	2
110	9	1
100	10	2
90	11	2
Ferdig: 05.11.2018		

Figur 1: Førsteutkast av releaseplanen.

B.2 Oppdatert releaseplan

Importance	ID:	Estimate
Sprint 1		
200	1	2
190	2	1
180	3	1
170	4	2
Ferdig: 07.10.2018		
Sprint 2		
168	28	2
166	29	3
164	30	1
Ferdig: 24.10.2018		
Sprint 3		
160	7	2
150	8	1
140	20	1
130	5	1
Ferdig: 07.11.2018		

Figur 2: Oppdatering av releaseplanen for sprint 2.

B.3 Siste releaseplan

Importance	ID:	Estimate
Sprint 1		
200	1	2
190	2	1
180	3	1
170	4	2
Ferdig: 07.10.2018		
Sprint 2		
168	28	2
166	29	3
164	30	1
Ferdig: 24.10.2018		
Sprint 3		
160	7	1
150	8	1
140	20	1
130	5	1
110	9	1
100	10	1
Ferdig: 07.11.2018		

Figur 3: Oppdatering av releaseplanen for siste sprint.

C Tasks

Tasks er brukerhistorier brutt ned i mindre oppgaver. Dette gjøres for å skaffe en oversikt og dypere forståelse for hva som inngår i brukerhistoriene. Under er brukerhistoriene med tilhørende tasks, oppgitt punktvis, gitt for hver sprint iterasjon.

C.1 Sprint 1

ID 1: As an author, I want to create content, so that I can add it to the site.

- Skal kun være mulig for authors eller for de som har “høyere” rang å lage content
- Vi lager et tekstfelt, og en knapp for å legge til bilder
- Lag et tittelfelt
- En publish knapp.

ID 2: As an author, I want to submit my content, so that editors can read and publish it.

- Skal kunne lagre utkast
- Skal kunne hente ut gamle utkast og redigere dem
- Gjør innholdet i en tekst tilgjengelig for flere enn en bruker.
- Pass på så kun editors kan lese innhold som ikke er offentliggjort.
- Opprett en mulighet for publisering slik at alle skal kunne se innholdet.
- Knapper der man kan lagre eller hente ut utkast

ID 3: As an unregistered visitor, I want to view content, so that I can consume content I am interested in.

- Lage en homepage/forside med ulike artikler innenfor forskjellige kategorier, som lages senere.
- Legge inn linker til artikkelsidene som redirecter brukeren til artikkelen.
- Lage bokser for hver artikkel nedover på siden.

ID 4: As an unregistered visitor, I want to browse content, so I can choose what to consume.

- Gjøre det mulig å trykke på artikkel for å lese den.
- Lage HTML/CSS struktur for å forme hvordan artikkel skal se ut

C.2 Sprint 2

ID 28: As an executive editor, I want to create categories, so that the content can be organized.

- Innlegg skal kunne bli merket med en kategori.
- Lag et kategori-felt i publiseringssiden.
- Kun for executive editor: Lag et tekstfelt der navnet på kategorien skal skrives og en submit-knapp.
- Kategori-feltet skal være en drop-down meny med de tilgjengelige kategoriene som executive editor har laget.

ID 29: As an executive editor, I want to assign unpublished content to copy editors so that they can proofread it.

- Lage modell for unpublished content
- Lage view for unpublished content
- Lage html for unpublished content
- Lage en dropdown med alle copy editors
- Gi de valgte copy editors tilgang til contentet

ID 30: As an executive editor, I want to unpublish content, so that it can be taken off the site. Tasks:

- Kun for executive editor: Lag en slett-knapp på hvert innlegg.
- Innlegg som 'fjernes' skal dukke opp i drafts.
- Innlegg som fjernes skal fjernes fra forsiden.

C.3 Sprint 3

ID 7: As an admin, I want to assign and manage the roles of registered users, so that I can control what they can do on the site (what rights they have)

- Degradere brukere
- Oppgradere brukere
- Tilegne roller
- Fjerne roller

ID 8: As an admin, I want to do everything the other roles are able to do.

- Admin skal gjøre alt alle de andre kan
- Lag admin gruppe på django, som har alle tilganger

ID 9: As a registered user, I want to be able to do everything the unregistered visitors can do.

- Lag reg side
- Lag view for reg side
- lag reg form

ID 20: As a registered user, I want to update my profile, so that I can keep my information up to date.

- Lag profil side
- Lag view

ID 5: As a copy editor, I want to give feedback on the authors content by commenting on the pages, so that I can review content.

- Oppdater unpublished modell så den får kommentarer
- Legg til så den vises på unpublished siden
- Gi authors tilgang til sine egne unpublished posts

ID 10: As a copy editor, I want to see a list of all content that needs proofreading and assign the work to myself, so that all copy editors know what work is in progress.

- Lag knapp for "Assign to me" "In process"

D Referater

D.1 Daily Scrum

	A	B	C	D	E	F	G	H	I
1			Alles gut	Noch einmal	Kennst nicht				
2			Sander	Halvor	Morten	Magnus	Hanne	Marius	Sara
11									
12		Sprint 1	Sprint Planning						
13		Hva gjorde jeg i går	Ingenting	Ingenting	Ingenting	Ingenting	Ingenting, har vært bortreist	Ingenting	Ingenting
14	mandag 24.09	Problemer?	Nei.)	Nei	Nei	Nei	Nei	Nei	
15		Hva skal jeg gjøre i dag	Lære meg git	Se video om Django	Lære meg Django	Se video om Django	Er på tur, så har ikke mulighet til å jobbe idag	Lære meg git	se video om Django
16		Hva gjorde jeg i går	Lærte meg git	Begynnte med oppsett av Django	Fikk startet litt i Django, og lekt meg i Git Bash	Så halve videoen om Django	Ingenting, har vært bortreist	Oppstart Gitlab	så halve Django video
17	tirsdag 25.09	Problemer?	Pushet en del til Master ved en feil, fikk slettet og fikset det.)	Litt tulling med oppsett, men fikk ordnet det til slutt	Nei	Nei	Nei	Nei	databas trøbbel (fant ut at man kan bruke den oppgitte)
18		Hva skal jeg gjøre i dag	Ingenting	Ingenting	Ingenting	Ingenting	Begynne å tenke på hvordan implementere felles styling til siden.	Prøve å pushe	se resten
19		Hva gjorde jeg i går	Ingenting	Ingenting	Ingenting	Ingenting	Ingenting	Gjorde meg kjent med django-prosjekt	Så ferdig Django video
20	onsdag 26.09	Problemer?	Nope	Nei	Nei	Nei	Nei	Nei	
21		Hva skal jeg gjøre i dag	Fikse hjemmeside/bruker kan legge ut poster	Lagde homepage-filer med Masgy T. Se resten av video om Django	Å få pushet prosjektet til GitLab, sånn at vi har noe å jobbe med	Så ferdig videoen om Django og lagde homepage-filer med Halvor	Lage utkast til stylingen av siden vår.	Opprette mitt eget Django-prosjekt	Parprogrammere med Sander og Morten, mulig å legge ut poster
22		Hva gjorde jeg i går	Fikset hjemmeside/bruker kan legge ut poster	Lagde homepage-filer og så resten av Django-videoen	Fikk pushet startprosjektet "Pålterest", og skrev kode med Sara og Sander til homepageen til nettsiden	Så ferdig videoen om Django og en underside i illustratør.	Lagde utkast til hovedsiden og en underside i illustratør.	Kartlegging av design.	Parprogrammerte med Sander og Morten, fikset at bruker kan legge ut poster
23	torsdag 27.09	Problemer?	Nei.)	Klarte til slutt å pushe etter mye om og men	Hadde en del problemer med å pushe til Gitlab, merge conflict bla. Men det ordnet seg tilslutt	Litt problemer med Python og Pip	Nei	Antallet sider, og tidsbruk	
24		Hva skal jeg gjøre i dag	Ingenting	Ingenting	Lærte meg litt mer om Django	Ingenting	Diskuterte designoppsettet på siden med Marius	Sitte med Hanne og utvikle design	google git
25		Hva gjorde jeg i går	Ingenting	Ingenting	Så flere video-er om Django	Ingenting	Satte opp og la inn draft til stylingen	Satte opp sider	Ingenting
26	fredag 28.09	Problemer?	Nope	Nei	Nei	Nei	Hvordan fikse css til django	Hvordan implementere design i HTML	tidsproblem
27		Hva skal jeg gjøre i dag	Fikse drafts	Jobbe lage drafts med Tidemannnnnn	Jobbe videre med ID #2	Jobbe med drafts sammen med Halvor	fortsette med sideutforming	fortsette med sideutforming	Parprogrammere med Sander og Morten, prøv å fikse draft
28		Hva gjorde jeg på fredag?		Fikk gjort ferdig og pushet drafts	Pushet ID #2 til Gitlab	Jobbet med drafts sammen med Halvor	Design til siden	Ingenting	Parprogrammerte med Sander og Morten, fikset drafts
29	mandag 01.10	Problemer?		Litt problemer med å få pushet, men det ordnet seg.	Problemer igjen med merge conflicts	Litt problemer med å få pushet/commitet i Git	Litt problemer med pushing og pulling til Git	Nei	Problemer med ssh og gitlab
30		Hva skal jeg gjøre i dag			Fikse en bug som gjør at drafts ikke får image-filer	Jobbet med views og fikset urls	Starte på style og felles html til siden og pushe til git	Hjelp Hanne med utvikling av CSS dokument	Jobbe med en oversikt over flere artikler

Figur 4: Vår digitale daily scrum oversikt

D.2 Møtereferater

Det ble ikke ført referat ved alle møter, under foreligger de som eksisterer.

D.2.1 Før igangsetting av sprintene

Møtereferat 10. september: Skaffe oversikt:

Sprint goal: Lage en oversiktlig nettside med mulighet for forfattere å publisere artikler. Demo date: 09.10.2018 (Innlevering av endelig versjon 7.11.2018, 13.11.2018).

Vi skal nå fastsette tidsbruken til de forskjellige ID'ene til Storiesene i backloggen. Vi undrer over at vi mulig er noe optimistiske når det

Spørsmål til Pål: Skal admin redigere i Django eller via funksjonalitet på websiden?

Møtereferat 11. september: Sist: Sagt hva vi mener om arbeidsmengde Skal snakke med Pål om hva punktene betyr og utdype om hva prosjektplanen skal inneholde osv.. (Veileder Pål) Spør Pål om tester også.

I dag: Finne ut av arbeidsprosess. (Hannes hjertesak). Hva vi fikk til med arbeidsprosess: Se “Arbeidsprosess” i driven.

Møtereferat 17. september: Møtestart 10:15 - Samtlige er tilstede

Sara jobber med å få releaseplanen inn på presentasjonen. Marius og Hanne jobber med arbeidsprosess Halvor og Morten jobber med testing av løsning, og Sander og Magnus jobber med planen for første iterasjon.

Vi lager hver våre slides på PP, og går gjennom punkt for punkt hva vi synes skal stå på slidesene. Deretter legges det til forelesningsnotater. Bruker oppgavearket vi lagde på fredag for å ferdigstille presentasjonen vår.

D.2.2 Sprint 1

Møtereferat 21. september: Sprint 1 første møte: Sprint 1 består av brukerhistorie 1, 2, 3 og 4 Vi har gjort oss godt kjent med mange av kravene allerede, men må fikse design, teste i følge testplan. Har laget et forslag til et daily standup excel dokument

Har strukturert google disk bedre, nå har vi mapper for dokumentering/møter, presentasjon av prosjektplan, sprint 1, sprint 2 og sprint 3.

Møtereferat 25. september: Forrige møte gikk mye ut på å sette opp github og skaffe seg oversikt.

Møtereferat 26. september: Gjort siden sist: Vi har alle testet opp mot gitlab og fått til en push og pull. Vi måtte fjerne test branchen og leste oss opp på hvordan Django fungerte. Vi har også kartlagt hvordan Django-adm funker og snakket med produkteier om hvordan administrasjonspanelet skal fungere.

Gjøre mål: Etter å ha snakket med produkteier satte vi oss ned for å sette opp en django-nettside. Vi gjorde oss alle kjent med oppsettet og bestemte oss for at den som fikk satt opp siden først skulle starte med opprettelsen av siden vår. Deretter delte vi opp ID: 1 Slik at Magnus og Halvor startet med oppsett av hjemmesiden, mens Sander, Morten og Sara satte opp en “Light-sql” og få den opp på serveren.

Marius og Hanne satte seg ned for å kartlegge designet på siden, samt antall sider. Der tok de utgangspunkt i Storiesene.

problemer:

- Vi fikk ikke til å operere med mySql. Løsning: Vi løste oppsett av server og Database med SQL-light.
- Antall branches. Løsning: Vi setter opp branches etter hvilke “Funksjoner” vi skal bruke.

- Var feil i datagenerert kode. Løsning: Oppdaterte programvaren og prøvde opprettelsen av django-prosjektet på nytt.
- Uvisst antall sider og hvordan utformingen av disse vil bli. Løsning: Vente til vi har fått utviklet mer slik at vi vet hva
- Sliter med å pushe opp koden til gitLabprosjektet.
- Bruker mye tid på å orientere oss rundt hvordan filstrukturen skal være.
- Design - Vi er litt usikre på hvilken del av funksjonaliteten til siden vi skal implementere i Django og hva vi skal designe selv. Derfor er det vanskelig å lage et utkast for sidene vi er usikre på om skal ha noe design i det hele tatt.
- Pusher kun opp filstruktur, men filene er tomme. Skjønner ikke hvorfor det funka, med det funker.

Møtereferat 1. oktober: Oppdukkende spm: Høre med Pål tilknyttet Testplan. Hvordan skal man teste? Skal vi begynne å skrive rapport? Hva menes med “Vise releaseplan for produktet”? Gi en kjapp update om det er noen endringer. Hva menes med utkast på rapport? Skrive så mye vi klarer.

Hva må gjøres til 10 oktober?!?!? Utvikle en releaseplan for produktet. Vi viser den vi allerede har laget. Første sprint kommer til å være over, og vi må vise hvor langt vi har kommet i produktet. (Vise hva vi har gjort hittil.) Vi skal vise frem alle brukerhistoriene som er implementert. Vi viser frem hva vi har gjort og hvordan vi har gjort det. 3. Vise hvordan det er i henhold til testplan. Høre med Pål om hvordan testing vi burde forholde oss til. Når vi skal demonstrere det, så viser vi hvordan det skal bli testet.

Hva gjøres i dag? Sander, Magnus, Morten, Sara, Halvor, Hanne, Marius.

Problemer: Hanne slet med å opprette en ny branch, Marius hjalp henne.

Møtereferat 3. oktober: Sara: Release plan og starter på powerpoint Marius, Magnus, Halvor: Begynner på rapporten ved å gå gjennom boka for å sammenligne vår arbeidsprosess med hvordan den er satt opp i Scrum Hanne: Jobber med CSS til siden Morten, Sander: Jobber med siden/Django

Problemer: Hvis man lagrer et draft på siden, og så blir ikke bildet med når man henter opp draftet igjen.

D.2.3 Sprint 2

Møtereferat 17. oktober: Innser at vi bare har 5 dager å bli ferdig med Sprint 2, og vi har ikke begynt Sprint 2 planning ennå på grunn av de nye brukerhistoriene. Akkurat 3 uker til kode-frys.

Vi må bli flinkere til å bruke branches. Og deretter merge de. Vi må bryte ned brukerhistoriene til tasks, og legge det ut på gitlab. Vi må få med omrokkering av sprint planning. Vi må fraksjonere user stories til mindre tasks bedre

enn før Halvor foreslår at vi deler arbeidsoppgavene opp til enkelte personer. At det blir mer effektivt arbeid da, enn om vi jobber sammen om ting. Det blir foreslått at vi bare gjennomfører 3 user stories denne sprinten (28,29,30?) Magnus tar oppgaven om å bryte ned stories til tasks. Diskusjon om hvordan vi skal presentere tasks på gitlab. (Todo, Doing, testing)

Hanne tar opp at vi har mindre tid på sprint 2 (teknisk sett 1,5 uke, men pga. nye stories får vi bare 5 dager (Vi følte at vi måtte vente med å starte sprint 2 planning etter møte med Pål den. 17. Oktober)) enn vi har på sprint 3 (2 uker). Dette påvirker tidsestimatene.

Trøbbel med å pushe til git. Hanne foreslår git rebase. Vi leser litt mer om rebase og finner ut at den hjelper til med å forhindre merge conflict:

- Puller først med rebase
- Commit
- Pusher

Arbeidsoppgaver ble utdelt: Django og programmering: Sander og Morten CSS/Design: Hanne og Marius Dele stories til tasks: Magnus (og Sander) Halvor jobber med presenteringen av user stories/tasks på gitlab Sara jobber med rapporten

Bestemmer oss for å ha tasks på gitlab som vi kan flytte fra “ToDo” til “Doing” til “testing” osv. Det diskuteres hvordan syntaxen til tasks på gitlab skal være.

15:45 Hva skjer? User stories ferdig brutt ned til tasks Halvor ferdig med gitlab representasjonen av tasks Sander og Morten begynner med Django/programmering av de nye brukerhistoriene for Sprint 2 (ID: 28, 29, 30) Andre jobber med allerede fastsatte arbeidsoppgaver Ikke alle har skrevet ned daily-standup oppdatering (Hva man gjorde i går, og hva man planlegger å gjøre i dag) 5 minutter pauser tas for at alle skal gjøre dette Hanne og Marius jobber bra med CSS/Design, men har problemer med git push og merge conflicts Sara og Halvor har tatt en “executive decision” og har bestemt at vi lager rapporten i LaTeX. Arbeidet går bra. Må legge til: Innholdsfortegnelse, Vedlegg, Referanser

Møtereferat 19. oktober: Sara og Halvor jobber på rapport. Morten og Sander jobber med ID 28 og 29. Marius jobber med CSS. Magnus er referent og leser seg opp på rapporten.

Møtereferat 22. oktober: Magnus, Sara, Halvor: Parprogrammerer, finner løsning på ID 30 Sander, Hanne: Fikk fikset Git Sander, Morten: Startet arbeid med ID 29 Hanne, Marius: Jobber med CSS

Det oppstod store problemer med Git da vi benyttet oss av checkout-funksjonen.

D.2.4 Sprint 3

Møtereferat 26. oktober: Marius startet møtet ved å skrive opp IDer planlagt i sprint 3. Det ble deretter opplesning av disse IDene og vi snakket om de. ID 7 og 8 var allerede implementert, mens 20 var ikke. Det ble snakket om hva som måtte gjøres i ID 20. Det ble også snakket om hva som skal gjøres for ID 5.

Vi tenker å legge til flere IDer til, siden ID 7 og 8 allerede var implementert. Det diskuteres om hva som menes med ID 6. Det vurderes om vi skal sende epost til Pål og høre om vi skal droppe 6 og heller ta inn en annen ID. Det diskuteres om ID 9. Går videre til høytlesning av ID 10. Diskuteres over ID 10 vs ID 29. Tar med ID 10. Det konkluderes med at vi sender en epost til Pål om at vi dropper ID 6 og tar ID 10 i stedet. Vi sier til Pål at vi reestimerte og fikk for stort estimat til ID 6, men kan ta med ID 10. Hanne skriver mail. Det diskuteres om hvordan mailen skal skrives. Det lekes med persienne på rommet. Det tegnes sinnafiges på tavlen.

Marius foreslår vi også lager tasks til ID 7, 8, 9 i tillegg. Vi fortsetter med breakdown av IDene på GitLab.

Møtereferat 29. oktober: Morten og Sander har jobba med ID 7 og 8 og bugs. De har også prøvd på nytt i gitlab og det funka! Marius har jobba med css. Magnus og Sara har jobba med rapporten. Magnus har jobba med sprint 2 retrospectives og Sara har jobbet med sprint 3 planning og sprint 2 iterasjon/testing/demo (ferdigstille det)

Vi fikk en god rapport-tilbakemelding av en annen gruppe på mail fra Pål, viktig å gå gjennom den før rapportinnlevering.

Møtereferat 02. november: Marius og Hanne skal prøve å få overført cssen. Sander jobber med ID 20. Halvor jobber på rapporten.

Møtereferat 05. november: Mål for dagen: Bli ferdig med de resterende ID-ene: ID 20, ID 5.

Ikke nødvendig at vi blir ferdig, da blir vi en erfaring rikere.

Halvor, Sara og Magnus jobber videre med rapporten Marius går over til rapport-teamet. Hanne jobber videre med CSS. Sander og Morten prøver å gjøre ferdig ID 20 og 5.

10:30 Hanne fikk problemer med git pull/push, men ved bruk av rebase så ble det i orden. 10:43 ID 20 er ferdig. Må nok bug testes litt mer. Arbeidet på ID 5 begynner

10:53 Marius spør: Hvilke ID-er skal vi vise på 14.11 demonstrasjonen?

ID 5 er ferdig. Sander og Morten prøver nå å fikse ekle bugs.

Halvor, Magnus og Marius begynner på Power-pointen til 14.11 demonstrasjonen

D.3 Møte med produkteier

D.3.1 Sprint 1

Første møte med studass 4. september:

- Bruk Django
- Les scrum boka. (Vær oppmerksom på notater der det kan fremkomme at kapitlet/teksten er unødvendig.)
- Innleveringene så vil Scrum være essensiell.
- Gjør dere kjent med hva en releaseplan er?
- Sett av 2-3 dager i uka. (gjør etter møte med kunde) (Mandag(1 time), onsdag, torsdag(2-6, Timer)
- Inviter hverandre til GitLab prosjektet.
- Skriv rapport fortløpende.

Vi har møte med kunde i morgen : 14:15-15:15 rom: EL6 (Onsdag 5. september). Møter: (starter fra uke 37), Mandager: 10:15-12:00, Onsdag: 14:00-∞, Fredag: 14:00-16:00.

Viktig å huske på ved møte med kunde: Være kjent med vår "kapasitet" som gruppe.

Første møte med kunde 5. september: Innlevering 1 praktisk info: Innlevering 18. september, og presentasjon 19. september (Tar 5 min.) Ikke ta med unødvendig info.

Det viktigste er det kunden sier.

Prioriteringsliste: Kunden vil at ID nr 1, 2 har førsteprioritet. (Man skal kunne lage noe og "Submitte"). Videre ønsker kunden at ID nr 3, 4, 5

ENDRINGER: Kunde ønsker at ID 7 og 8 skal komme etter ID 4. (12347856).ID 20 skal komme etter 8. (123478(20)56...osv).

Møte med produkteier 26. september:

- Burndowncharten trenger ikke se så bra ut
- Del dere opp i: Database, design og prosjektet
- Prøv å få opp django-siden.
- Pushe prosjektet opp asap.
- Rapporter og presentasjoner er viktig.
- Referer daily standup, (Vedlegg i rapport) Poenget er å se prosessen fremover.

- Sprint: Husk å dokumenter!
- Alle bør ha en egen branch når de jobber med noe nytt. Vil helst ha en branch for hver individuell funksjon som lages. Login f.eks. bør ha en egen branch. Branchene kan være for hvert av "issues-ene".

Møte med produkteier 3. oktober: Møte med veileder og produkteier

- En uke etter kodefrys skal rapporten være ferdig.
- Bør ikke dokumentere at vi jobber med rapporten. Bør ikke være noe tap av produktet til fordel for rapporten.
- På demonstrasjonen viser vi den forrige produktplanen. Skal vise hva vi har gjort hittil. Om vi har noen endringer i releaseplanen fremover i tid, så kan vi også ta med dette. Vise hvor langt vi har kommet i forhold til planen.
- Når vi demonstrerer skal vi vise testplanen og vise hvordan vi tester. Vise at det som er implementert så langt er i henhold til testplanen.
- Skal ta utgangspunkt i ID-ene som er tildelt av produkteier.
- Kan slenge ved akseptansetest og si "produkteier er fornøyd med dette, ikke fornøyd med dette", og teste resterende der og da.
- Kan endre på testplanen etterhvert om vi finner ut at måten vi implementerte funksjonen på ikke var bra nok.

PRODUKTEIER: Ønsker å legge ut artikkel uten å måtte legge ut bilde, han gleder seg til CSS.

D.3.2 Sprint 2

Møte med produkteier 17. oktober: Produktet burde håndtere feil, f.eks. hva som skjer dersom man forsøker å logge inn med en bruker som ikke eksisterer.

Kommentarer fra Pål til rapporten:

- Skriver backstory istedenfor user story, det er feil begrepsbruk.
- Snakke om hva vi gjør i introduksjonen, altså introdusere hva scrum er. Bør redegjøre litt hva scrum er og hvordan vi gjør det.
- Prosessbeskrivelsen er litt rart strukturert.
- Savner litt teori om selve scrum og scrumprosessene. Hva scrum planning er f.eks.
- To do: redegjøre og deretter beskrive hvordan vi gjør det. Skal helst skrive en rapport som kan leses av en person som ikke er så rutinert på scrum.
- Gjerne et eget punkt med oppsummering av våre modifikasjoner av scrum.

- Det vi har skrevet er bra, trenger bare mer struktur.
- Bruk vedlegg for hva det er verdt. (Pål sin gruppe hadde 46 sider med figurer og bilder).
- Referer til vedlegg som vi har.
- Husk å ta med testing og retrospective i rapporten.

Pål vil at vi skal prioritere ID 28, 29 og 30 øverst til sprint 2.

Møte med produkteier 24. oktober: Vår oppsummering til produkteier er at vi så og si ferdig å skrive om arbeidsmetode og første iterasjon. Vi sørger for at vi har mye å ta opp på reviews etc til retrospective. Deadlinene våre ble litt korte, så vi har utsatt andre iterasjon til i dag.

Akseptansetest med Pål:

- Skeptisk til å logge inn på.djangoadmin.
- Vi burde navngi ting tydeligere, eller lage en bedre substruktur på siden. Er litt vanskelig å vite hva som ligger på allposts, admin osv. Gjør det klarere og mer intuitivt hvor man skal trykke for å komme seg inn på de forskjellige sidene.
- Hadde vært mer logisk dersom det på executive links siden ikke hadde så flat struktur, kanskje navngi dem på en annen måte.
- Ha en knapp med nye tags, homeknappen bør ikke føre til der man lager artikler, men heller der man ser alle artiklene.
- 28 er good, men 29 kan ikke godtas siden han ikke kan se den, 30 er også nice.

D.3.3 Sprint 3

Møte med produkteier 31. oktober: Vi skal egentlig ikke jobbe med prosjektdelen men dersom vi ikke har mer arbeid å gjøre, og arbeidsmengden er såpass minimal kan man jobbe med prosjektet.

Bør kjøre test av produktet på noen i kollektivet eller noe. Skal jobbe på for å ferdigstille produktet mest mulig.

Skal lage dokumentasjon som skal leveres zippet sammen med rapporten. Dokumentasjonen vil gjerne være hvordan man initialiserer produktet for noen, hvordan man kloner det ned, og bruk av produktet som kanskje ikke er helt åpenbar.

Kjører akseptansetest på ID 7 og 8 på dette møtet:

- ID 7 får såvidt pass, men er en del bugs vi må få rettet opp i.
- Fungerte ikke når vi kjørte testen. Godtar ikke ID 8, prøver å lage tag, men får admission forbiddne ups.

- Når du skriver inn hvilke brukere som har jobbet på en post hadde det vært fint med en liste med alle author og editors med rettigheter, slik at man kan velge.

Møte med produkteier 7. november: Akseptansetest med produkteier av ID 8, 20, 5, 9, 10 og 29:

- Error ved saving av en artikkel. Ikke godkjent ID 29 og 10, men det er det samme problemet, så om vi fikser dette så er det good.
- Ikke ønskelig å automatisk bli author når man registrerer seg, så ID 9 kan godkjennes.
- ID 8, 5 og 20 kan godkjennes.

ID 29 fungerte fint da vi fikset problemene fra akseptansetesten 24. oktober. Det må ha oppstått problemer/konflikter når vi har implementert inn nye brukerhistorier.

D.4 Referat fra retrospectives

D.4.1 Sprint 1

Oppsummering av sprinten: Vi har hatt en god dialog med produkteier fra starten, noe som har blitt opprettholdt gjennom hele sprinten. Vi hadde ikke nok arbeidsoppgaver til at i kunne jobbe individuelt, så parprogrammering fungerte veldig bra. Ikke alle parene kunne jobbe med IDene, så noen par startet på CSSen, mens andre ble bedre kjent med Django og startet på IDene. På denne måten ble alle satt i arbeid, og vi fikk alle en god helhetsforståelse av produktet. Det ble fraksjonert inn i flere tasks, og vi satte oss ned og diskuterte hva disse taskene innebar med innspill fra produkteieren. Vi forberedte oss før møtene med produkteier slik at forståelsen for produktet som ble etterspurt fra produkteier, samsvarte med det vi har laget.

Da vi da skulle fortsette arbeidet, delte vi opp taskene slik at alle hadde noe å jobbe med. Da en ble ferdig med sin egen fortsatte man på den neste tasken. Vi begynte å bli ferdig med IDene litt før tiden, og da vi kjørte gjennom en akseptansetest med produkteier oppdaget vi at IDer vi hadde markert som ferdig, fortsatt egentlig var pågående. Derfor satt vi disse IDene til pågående igjen, og implementerte disse manglende funksjonene etter beste evne. Etter at de gjenværende IDene ble implementert i produktet, gjensto det bare å få produktet til å se mer “gjennomført” ut, ved å implementere bedre CSS på siden, etter at produkteier ga tilbakemelding om at han ønsket at nettsiden skulle se bedre ut.

Mot slutten av sprinten ble det en glidende overgang fra å jobbe med det bakenforliggende ved produktet til å faktisk jobbe med selve presentasjonen. Da denne var ferdigstilt, gikk vi på et felles rom og kjørte gjennomgang av demonstrasjonen, slik at alle på gruppa kunne stille like forberedt, og for å få en god oversikt over hva som måtte forbedres.

Etter presentasjonen stusset vi litt over spørsmål fra publikum, da sikkerheten til produktet ikke var presentert som et krav fra produkteier. Det ble heller ikke nevnt av produkteier da vi gjennomførte akseptansetest, og vi hadde derfor ikke implementert mer sikkerhet enn det som allerede er forventet implementert på en slik nettside (innlogging, artikkelvalidering).

Det vi synes gikk bra med sprinten var:

- Vi har hatt møter tre ganger i uka, slik at vi har hatt muligheten til å parprogrammere og stille spørsmål til de andre på gruppa om det skulle være noe man lurte på.
- Vi har vært fleksible, fornuftige og gode på å diskutere, slik at vi har kommet frem til gode løsninger, og det har ikke vært noe krise om det er noen som ikke har mulighet til å møte en dag. Vi har lagt egne interesser til side for å skape et best mulig produkt, og for å samarbeide om et felles mål.
- Vi er så langt fornøyde med produktet.
- Arbeidskontrakten har blitt fulgt til en tilfredsstillende grad, og vi har hevet sanksjonene ved kontraktbrudd ved et smil.
- Vi har vært fornøyde med estimatene da vi ble ferdige med alle brukerhistoriene i perfekt tid før sprinten var over.
- Bra med booking av rom slik at vi alltid har hatt et sted å arbeide i stillhet sammen med gruppa.

Det vi syntes kunne gått bedre med sprinten var:

- Vi var ikke fullt så drevne på Git og GitLab, og det gikk mye til til pushing og merge conflicts, og det oppstod forvirring rundt de forskjellige branchene.
- Scrumboardet gikk dårlig, vi flyttet ikke ting frem og tilbake når vi skulle. Det var ikke alltid så lett å vite når funksjonene var ferdig, og ofte ble de jobbet med i flere dager i strekk.
- Vi har vært dårlige på referering fra tidligere møter, så dette må vi bli flinkere på, da rapporten avhenger svært mye av dokumentasjon.
- Vi var ikke så veldig flinke på å kommunisere internt om hvem som gjorde hva på møtene, og det tok litt tid før vi kom igang.
- Vi har ikke vært fornøyde med Daily Standup, fordi vi har brukt mye tid på å komme igang.

Forbedringer til neste sprint:

- Lære oss mer Git og GitLab, slik at det ikke går unødvendig tid til dette.
- Bryte ned brukerhistoriene ID-ene i mindre oppgaver, og legge dem inn i GitLab, slik at man kan se hva som må gjøres og hva som jobbes med.
- Lage en ny testing-liste i Scrumboardet, som skal inneholde testene til hver brukerhistorie.

- Daily Scrum kan bli bedre oppdatert, så om vi kan bruke ToDo-tasks kan dette kanskje hjelpe med at gruppen til enhver tid har noe å gjøre. Daily Scrum-boardet er viktig å få oppdatert før hver arbeidsdag, så vi kan pushe på for å fylle inn dette ved starten av hvert møte da vi møtes. Om vi finner ut av at dette mot formodning ikke fungerer, kan vi ta en avstemning på hva vi skal gjøre neste.
- Vi kan lage en oversikt over hvem som refererer under hvert møte, slik at alle får være referent og slik at vi er sikre på at hvert møte blir dokumentert.
- Et eget dokument til møtet med Pål hver onsdag.

D.4.2 Sprint 2

Oppsummering av sprinten: Vi føler at det er kort tid siden forrige sprint-retrospective, og det har blitt litt mindre produktivitet siden vi ikke har vært fullt antall på møtene. Vi har også måttet vente med å starte fordi vi fikk nye ID-er, som vi gjerne ville ta opp med produkteier før vi startet på dem. Tidsmessig har lengden på sprinten vært grei, men oppgavefordelingen kunne muligens vært litt bedre. Siden rapporten er en stor del av prosjektet har vi valgt å prioritere denne mer enn i den forrige sprinten.

Vi føler ikke vi har vært i en programmeringsfase, men at vi bare har løst problemer vi har støtt på underveis, noe som har gått relativt knirkefritt sett bort ifra problemer med Git. Kommunikasjonsproblemene om hvem som jobber med hva har også hengt litt igjen ifra forrige sprint.

Om vi ser på det vi skulle innføre fra forrige sprint, så har vi blitt bedre på å ta på oss oppgaver dersom vi ikke har noe å gjøre, om man ikke har noe spesifikt å jobbe med på på selve nettsiden kan man sette seg ned og jobbe med rapporten.

Ifølge burndowncharten vår har estimatene våre fungert greit nok. Vi klarte å fullføre alle IDene våre på halve tidsrommet vi hadde planlagt for, fordi vi faktisk måtte gjøre det. Vi reestimerte forsåvidt estimatene på det første møte med produkteier denne sprinen. Vi har ikke satt opp noe estimat for de som jobber med CSS og rapporten, men det har vi ikke ansett som noe problem.

Alle punktene vi ønsket å forbedre fra forrige gang har blitt forbedret, noe som vi er veldig fornlyde med. Vi har erfart at Git er tidkrevende når man ikke kan bruke det, men vi har likevel klart å gjennomføre ID-ene og oppgavene vi har satt opp for oss selv. Sprint 2 blir sprint 3 pga. de nye prioriterte brukerhistoriene.

Det vi synes gikk bra med sprinten var:

- Vi føler selv at vi har jobbet bra, siden vi har grupperom på alle møtene og alle vet hva de skal gjøre når de kommer på møtene. Scrumboardet vårt ble bedre brukt enn det det ble forrige sprint, både det å bryte ned IDene i mindre oppgaver.
- Vi har blitt enda bedre til å oppdatere Daily Scrumboard.

- Vi har tatt tak i punktene til forbedringer fra forrige sprint, og faktisk forbedret disse på denne sprinten, slik at referent har blitt fastsatt, arbeidsoppgavene er bedre fordelt og folk har alltid ting de skal gjøre.
- Gjennomgangen med produkteier har også forbedret seg.

Det vi syntes kunne gått bedre med sprinten var:

- Vi bruker fortsatt mye unødvendig tid på Git og GitLab. Dette er tid som kunne ha blitt brukt til andre ting. Vi prøver å gjøre det så godt vi kan, men det er tidkrevende og tar en del tid fra å faktisk kunne jobbe med nettsiden.
- Oppmøtetidene er ikke helt ideelle, siden det er tidlig mandag og sent fredag. Det høres jo bra ut å ha møte annenhver dag, men dersom man er bortreist over en helg blir det vanskelig å møte på noe annet enn onsdagsmøtet. Vi får ikke gjort noe med dette nå men er noe å tenke på til fremtidige prosjekt.
- Selv om vi har blitt flinke til å oppdatere Daily Scrumboardet, så har vi ikke vært like flinke til å ha oversikt over hva andre har problemer med eller jobber med. Det er uansett vanskelig å hjelpe de på gruppa dersom de har problemer med f.eks. Git, da ingen av oss er veldig flinke på dette.
- Vi fikk litt dårlig tid opp mot deadline, siden vi glemte litt av den første delen av Scrum, fordi vi kun har fokusert på arbeidet. Gikk dermed litt tilbake i tradisjonell prosjektjobbing.
- Kommunikasjonen med veileder kan forbedres. Den trenger ikke nødvendigvis kun være hver onsdag, og den kan også tas over mail.

Forbedringer til neste sprint:

- Vi burde øke kunnskapen og kompetansen rundt Git og GitLab, men vi har ikke mulighet eller tid til å gjennomføre dette før prosjektet er over. Så lenge vi får det til å funke sånn at vi får vist det ved demonstrasjonen sir vi oss fornøyde.
- Vi bør ha et tidsrom på 5 minutter på starten av møter der hver person sier sin Daily Scrum, slik at alle får en oversikt over problemene og arbeidsoppgavene til hele gruppa.

Ros:

- Veldig bra at vi har tatt til oss forbedringene fra forrige gang, vi føler at arbeidsflyten går mye jevnere. Møteromsansvarlig gjør en fantastisk jobb! Vi er løsningsorienterte, konstruktive og ser fremover fremfor å se bakover.
- Morten og Sander har gjort en fantastisk jobb med Django-problemer, og det har hjulpet på at vi har fått en fast referent som referer på møteter. Det gjør det mye lettere å ha oversikt over prosjektet og fremgangen.

D.4.3 Sprint 3

Oppsummering av sprinten: Vi føler at sprinten ble ferdig raskt. Brukerhistoriene gikk fint, men det ble litt rot i slutten av sprinten da sprinten plutselig var over og akseptansetestene ikke gikk helt som planlagt. Men når det er sagt fikk vi fikset dem til slutt. Vi føler at arbeidsflyten har vært forbedret fra forrige sprint, og vi er fremdeles fornøyde med booking av rom. Rapporten har blitt mye fyldigere. Vi har fortsatt problemer med GitLab, men vi klarer fortsatt å fullføre oppgavene, da vi har pushet alt til master-branchen. Brukerhistorie 29 fungerte fint da vi fikset problemene fra akseptansetesten 24. oktober. Det må ha oppstått problemer/konflikter når vi har implementert inn nye brukerhistorier.

Det vi synes gikk bra med sprinten var:

- Fin produktivitet i gruppen, noe som Daily Standups har bidratt mye til.
- God stemning med kake og burger-onsdag.
- Vi har alltid hatt rom.
- Vi har fylt ut Daily Standups ved hvert møte.
- Vi har forbedret mange av forbedringspunktene fra forrige sprint.
- Vi har blitt bedre på egen testing slik at produkteier ikke blir første tester. Dermed har vi oppdaget egne feil mye tidligere i arbeidsprosessen.
- Kommunikasjonen med veileder har blitt forbedret.

Det vi syntes kunne gått bedre med sprinten var:

- Vi har ikke alltid hatt konkrete mål til arbeidsdagene, noe som har gått utover produktiviteten.
- Dårlig oppfølging av kontrakten på PowerKing- og New Energy-straffene. Men når dette er sagt har vi hatt veldig god gjennomføring av kake-straffene.
- Vi burde forberede oss bedre til Sprint Retrospective, ved å tenke mer over hva som går bra og dårlig underveis i sprinten.
- Git og GitLab
- Scrumboardet i GitLab har ikke blitt utnyttet til sitt fulle potensiale.

Forbedringer til neste sprint:

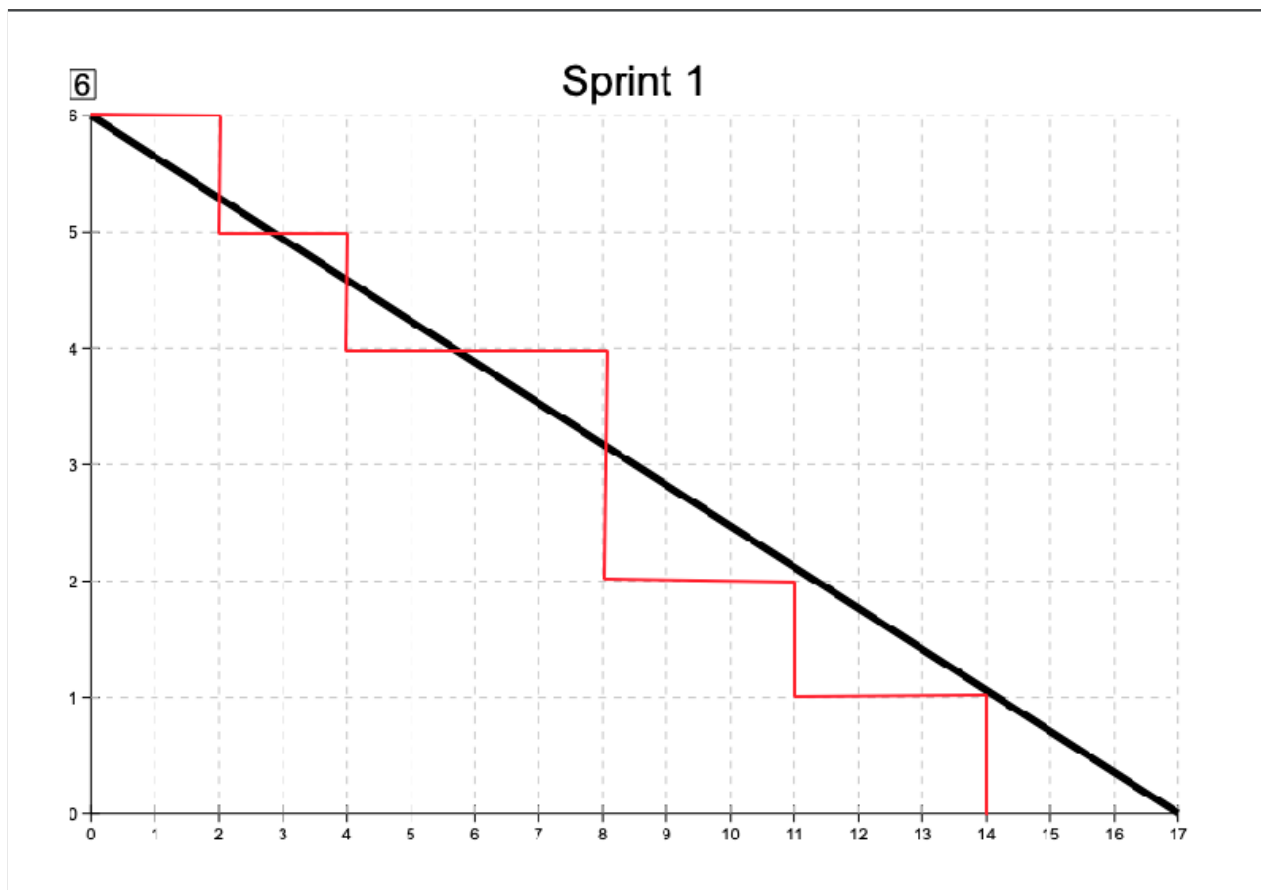
- Daily Standups burde være mer konkret. Som for eksempel å ikke skrive "jobbet med brukerhistorie 29", men heller konkret hva man gjorde og hvordan.
- Fokus på håndheving av kontrakten.

- Få gruppa til å melde seg på kurs i Git og GitLab, eller få en i gruppa til å kun fokusere på dette slik at den personen kan håndtere problemene som oppstår.

E Burndown chart

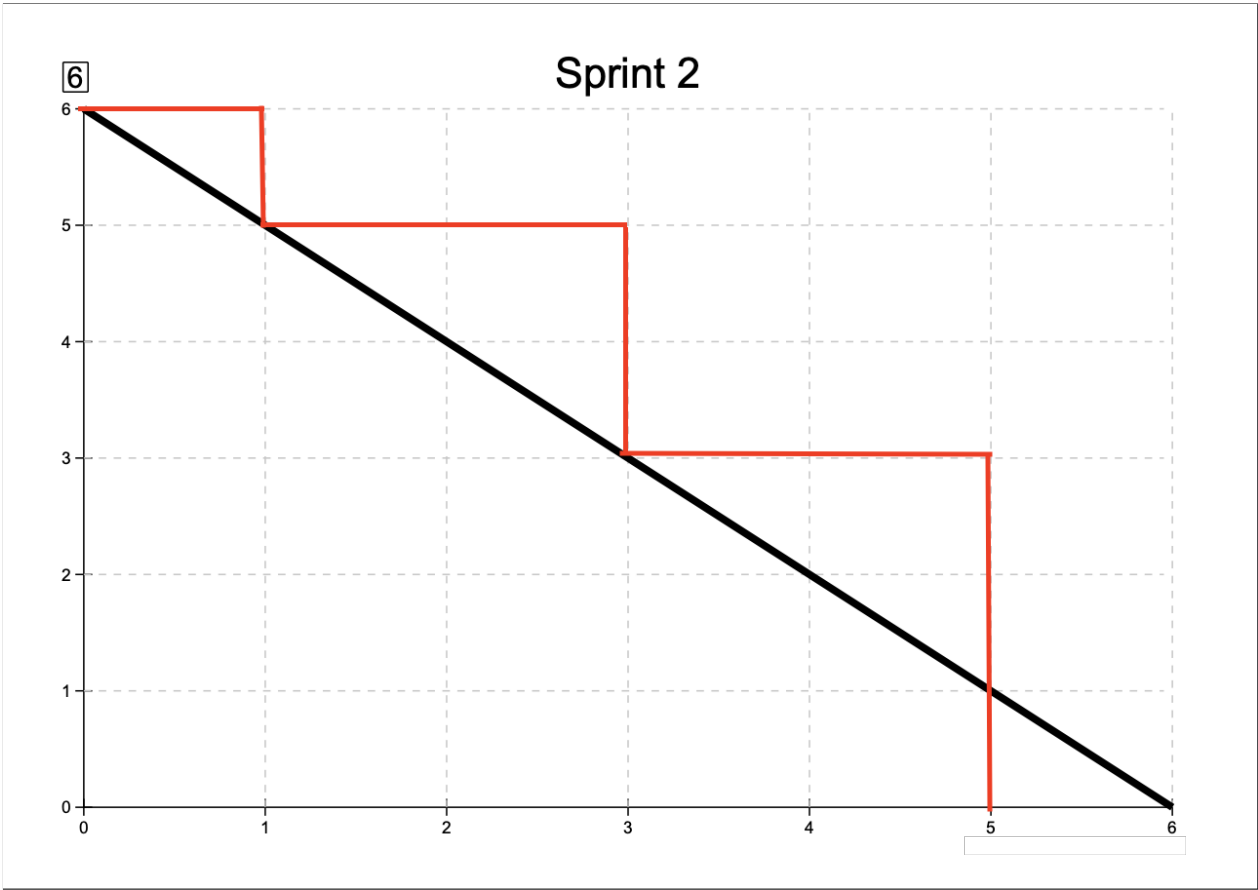
Burndown chart viser en oversikt over når brukerhistoriene ble gjort under sprinten. Y-aksen er størrelses-estimatet på sprinten og x-aksen er antall dager i sprinten. Grafen starter øverst på y-aksen siden ingen brukerhistorier er fullført når sprinten begynner. Når en brukerhistorie er ferdig så vises dette ved at grafen reduseres med estimatet til brukerhistorien. Lengre platåer i grafen indikerer at det har oppstått et problem som må adresseres.

E.1 Sprint 1



Figur 5: Burndown chart for sprint 1

E.2 Sprint 2



Figur 6: Burndown chart for sprint 2

E.3 Sprint 3



Figur 7: Burndown chart for sprint 3

F Testplan

F.1 Testplan for brukerhistoriene i sprint 1

ID	Story	Korrekt gjennomførelse av ID	Oppskrift til bruker
1	As an author, I want to create content, so that I can add it to the site.	Logge seg inn som author på siden, så man kan lage og publisere artikkel.	Logg deg inn som author og legg ut en artikkel
2	As an author, I want to submit my content, so that editors can read and publish it.	Logge seg inn som author på siden, så man kan legge artikkelen sin til "godkjenning" av en editor	Logg deg inn som author og lagre en artikkel til "godkjenning"
3	As an unregistered visitor, I want to view content, so that I can consume content I am interested in.	Gå på forsiden og se på artiklene	Gå på forsiden og se på artiklene du er interessert i
4	As an unregistered visitor, I want to browse content, so I can choose what to consume	Få et overblikk over ulike artikler på hovedsiden, og ha mulighet til å klikke inn på de.	Gå på forsiden, og velg en artikkel du vil konsumere

Figur 8: Testplan til sprint 1, ID 1, 2, 3 og 4.

F.2 Testplan for brukerhistoriene i sprint 2

ID	Story	Korrekt gjennomførelse av ID	Oppskrift til bruker
28	As an executive editor, I want to create categories, so that the content can be organized.	Executive editor kan legge til kategorier i tekstfeltet og legge dem til med en knapp. Executive editor skal se at kategorien er lagt til.	Logge inn med executive editor og legge til en kategori. Lag en artikkel og se at du får tildelt en kategori til denne.
29	As an executive editor, I want to assign unpublished content to copy editors so that they can proofread it	"Publish" en artikkel, se at den blir lagt til i unpublished og at kun executive editor og admin har tilgang til å tildele artikler til copy-editors.	Publishe en artikkel, se at den blir lagt til i unpublished og at kun executive editor og admin har tilgang. Logge inn med executive editor og legge til copy editors. Logge inn som editor og se at du har blitt assigned en post.
30	As an executive editor, I want to unpublish content, so that it can be taken off the site.	Bruker logger seg inn som executive editor, navigerer til en artikkel, trykker så på "delete". Skal ikke være mulig å gjøre dersom man er innlogget som f.eks. en author	Logg deg inn som executive editor og fjern en artikkel fra hovedsiden.

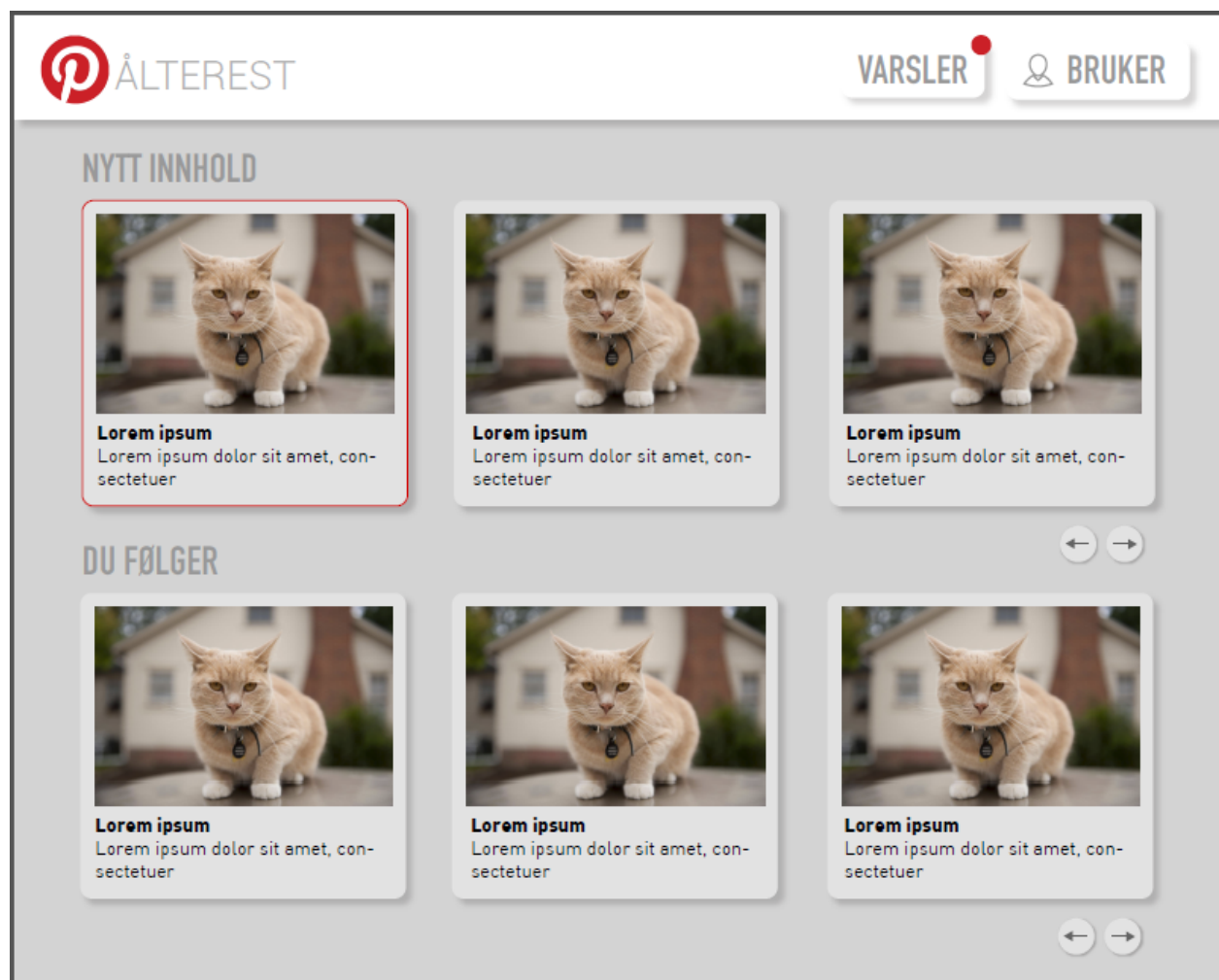
Figur 9: Testplan til sprint 2, ID 28, 29 og 30.

F.3 Testplan for brukerhistoriene i sprint 3

ID	Story	Korrekt gjennomførelse av ID	Oppskrift til bruker
7	As an admin, I want to assign and manage the roles of registered users, so that I can control what they can do on the site (what rights they have)	Dersom man logger seg inn som noe annet enn admin så skal man ikke kunne endre rettighetene til en bruker. Dersom man har logget seg inn som Admin vil man kunne endre rettighetene til en bruker	Logge deg inn og endre rettighetene til en bruker
8	As an admin, I want to do everything the other roles are able to do.	Logge seg inn som admin på siden, og ha mulighet til å browse artikler som en vanlig bruker har. Den skal også ha muligheten til å: (1) lage nye artikler, (2) Gi feedback til authors, som en copy editor, (3) Å "assigne" work til seg selv og drive med content editing. (4) Subscribe til categories og authors. (5) Kunne endre rettighetene til bruker	Logg inn som admin, og prøv å gjør handlinger som andre type brukere har mulighet til å gjøre
20	As a registered user, I want to update my profile, so that I can keep my information up to date.	Gjelder kun for bruker som er innlogget som registered user. Kunne oppdatere informasjon om profilen sin	Gå inn på "Min Side" eller lignende og oppdater ønsket informasjon
5	As a copy editor, I want to give feedback on the authors content by commenting on the pages, so that I can review content.	Kunne kommentere på authors sitt innlegg, og i tillegg kunne finne tilbake til kommenterte innlegg	Trykk på kommenter-knappen og skriv noe. Finn fram til tidligere kommenterte innlegg
9	As a registered user, I want to be able to do everything the unregistered visitors can do.	Bruker logger inn på siden og skal kunne aksessere det samme innholdet som en uregistrert bruker kan	Logg inn som registered user. Forsøk så å navigér rundt på siden, og sjekk at du har lesetilgang til de ulike artiklene
10	As a copy editor, I want to see a list of all content that needs proofreading and assign the work to myself, so that all copy editors know what work is in progress.	Man skal ikke ha mulighet til å assigne work til en selv dersom man ikke er innlogget som copy editor (en executive editor kan assigne til copy editors). Ha mulighet til å "assigne" work til seg selv. "Work"-et skal da ha en indikator festet til seg som indikerer at arbeidet er tatt. Andre brukere skal da ikke ha mulighet til å assigne "work"-et til seg selv.	Logg inn som copy editor. Få en oversikt over all content som skal redigeres. "Assign" work til deg selv. Sjekk da at arbeidet kan da ikke "assignes" av andre brukere.

Figur 10: Testplan til sprint 3, ID 7, 8, 20, 5, 9 og 10.

G Designutkast



Figur 11: Designutkast

H Epost

H.1 Epostutveksling med produkteier

Etter reestimering av brukerhistoriene som skulle implementeres i sprint 3 fant vi ut at vi hadde mulighet til å ta med flere brukerhistorier i sprinten. Men vi måtte i så fall endre prioriteringene av brukerhistoriene, på grunn av at den neste brukerhistorien med høyest prioritering hadde for høyt størrelsesestimat. Vi sendte derfor en epost til produkteier hvor vi forespurte om vi kunne ta med to ekstra brukerhistorier som hadde lavere prioritet enn den neste vi egentlig kunne tatt med. Produkteier sa seg enig i vår vurdering.

I Brukermanual

I.1 Rammeverk

I.2 Django

Rammeverket vi har brukt for produktet kalles Django. Hva er Django? Django er et rammeverk programmert i Python. Django hjelper webutviklere med en kjappere og smidigere måte å utvikle nettsider på. Når en utvikler en nettside er det mye en må tenke på: Innlogging, Utlogging, registrering, sikkerhet med mer. Dette sørger Django automatisk for. Rammeverket har innebygd databasesikkerhet, passordvalidering, skjemavalidering og mye, mye mer.

I.2.1 Hvorfor valgte vi Django?

Vi valgte å bruke Django som rammeverk da vi hadde hørt at det var et verktøy som var lett å bruke, samt hadde støtte for databaser. I tillegg bruker Django programmeringsspråket Python (og HTML, CSS og JavaScript), som var kjent for samtlige i gruppen.

Da ingen i gruppen hadde tidligere erfaring med databaser og sikkerhet, så vi Django som er godt valg for rammeverk.

I.3 Kode

I.4 Programmeringsspråk

Som nevnt har vi brukt rammeverket Django, som hovedsaklig bruker programmeringsspråket Python. I tillegg bruker Django **HTML**, **CSS** og **JavaScript**. (De vanligste webspråkene).

I.4.1 Versjoner

Versjonene til programmeringsspråkene vi har brukt er som følger:

- **Python 3.7**
- **HTML 5**
- **CSS 3**
- **JavaScript 1.8.5**

I.5 Sammenheng mellom språkene

Siden vi bruker Django, er det mye forskjellig kode. Hovedsaklig brukes HTML, CSS og JavaScript til front end (det som synes på siden + interaktivitet), mens Python brukes til back end (behandling av data, databaser osv...) Når en bruker går inn på en av sidene på nettsiden, sendes en GET request til serveren (Django). Denne GET requesten behandles av Python, som deretter sier ifra til serveren hvilken HTML som skal vises og hva i den som skal vises. På samme måte behandler Python POST requests (når brukeren sender et skjema, som f.eks innlogging). Databaser bruker som regel språket SQL, men i Django skrives også databasekommandoer i Python, som Django deretter automatisk gjør om til SQL.

I.5.1 Eksempel på sammenheng (Registrering)

Her vil du se et eksempel på hvordan Django håndterer POST og GET requests, samt en sammenheng mellom programmeringsspråkene brukt.

Dette er et eksempel på hvordan registreringen på siden er implementert. Det er 6 filer som spiller en rolle her, hvor fire av de er nødvendige Django-filer.

Filen som behandler GET og POST request kalles **views.py**:

Views.py

```
def register(request):
    if request.method == 'POST':
        form = RegForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            form.save()
            profil = Profile(user=form.save())
            profil.save()
            args = {'form': form, 'username': username}
            return redirect('/home', args)

    else:
        form = RegForm()

    args = {'form': form}
    return render(request, 'home/register.html', args)
```

Vi ser her at hvis requesten er POST skal skjemaet på siden valideres og lagres. Det skal opprettes en profil for den ny-registrerte brukeren og deretter skal brukeren omdirigeres til hjemmesiden. Hvis requesten er GET, skal skjemaet bare være tomt og `/home/register.html` skal vises. Vi ser også at denne funksjonen bruker et form, som er definert i **Forms.py**:

Forms.py

```
class RegForm(UserCreationForm):

    def __init__(self, *args, **kwargs):
        super(UserCreationForm, self).__init__(*args, **kwargs)

        for fieldname in ['username', 'password1', 'password2']:
            self.fields[fieldname].help_text = None

    class Meta:
        model = User
        fields = (
            'username',
            'first_name',
            'last_name',
            'email',
            'password1',
            'password2',)

    def save(self, commit=True):
        user = super(RegForm, self).save(commit=False)
        user.first_name = self.cleaned_data['first_name']
        user.last_name = self.cleaned_data['last_name']
        user.email = self.cleaned_data['email']

        if commit:
            user.save()

        return user
```

Skjemaet bruker Django's innebygde registreringsskjema. **Fields** under **class Meta** beskriver hvilke felter av skjemaet som skal vises og **save()** lagrer informasjonen i en ny databasemodell. Modeller er oppgitt i **Models.py**:

Models.py

```
class Profile(models.Model):
    user = models.ForeignKey(User, related_name='user', on_delete=models.CASCADE)
    alder = models.CharField(max_length=3, blank=True)
    interesser = models.CharField(max_length=500000, null=True, blank=True)
    tlf = models.CharField(max_length=12, blank=True, null=True)

    def __str__(self):
        return self.user.username

    class Meta:
        verbose_name_plural = "Profiler"
```

Dette er en databasemodell for en brukerprofil. Den er skrevet i Python, men for at databasen skal skjønne hva det betyr, gjør Django denne koden om til SQL.

Den siste Django-filen kalles **Urls.py**. Denne sier hva som skal skje når brukeren skriver noe i adressefeltet:

Urls.py

```
url(r'^register/', views.register, name='register')
```

Denne linjen sier at hvis brukeren skriver inn 'localhost:8000/home/register' i adressefeltet, skal **register** viewet fra **views.py** kjøres. Deretter sier **register** viewet ifra at det er **register.html** som skal vises:

Register.html

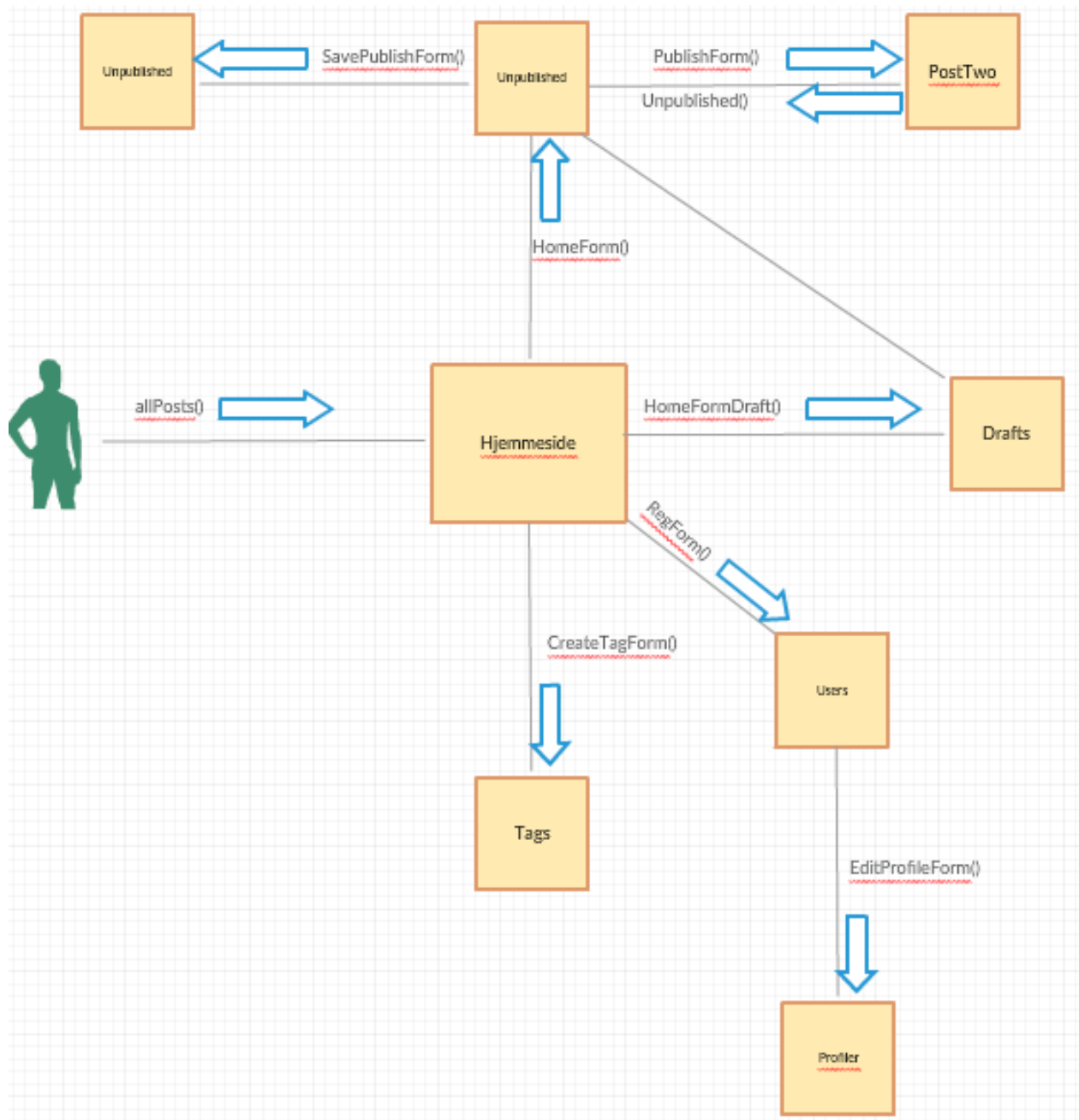
```
{% extends 'base.html' %}

{% block content %}
    <div class="page-container">
        <form class="container" id="article" method="post" enctype="multipart/form-data">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit" class="submitBtn">Register</button>
        </form>
    </div>
{% endblock %}
```

Her ser vi igjen et eksempel på Django rammeverket. alle linjene med {% %} eller {{ }} er 'Django kode'. Øverste linjen sier at HTML-filen skal arve fra en annen HTML fil. {{ **form.as_p** }} sier at skjemaet som kommer fra **Views.py** skal vises som er skjema. {% **csrf_token** %} er et eksempel på hvordan Django håndterer sikkerhet. Uten denne vil ikke formet vises, da den mangler verifikasjon i henhold til Djangos regler.

Den siste filen er helt vanlig **CSS**.

I.6 UML

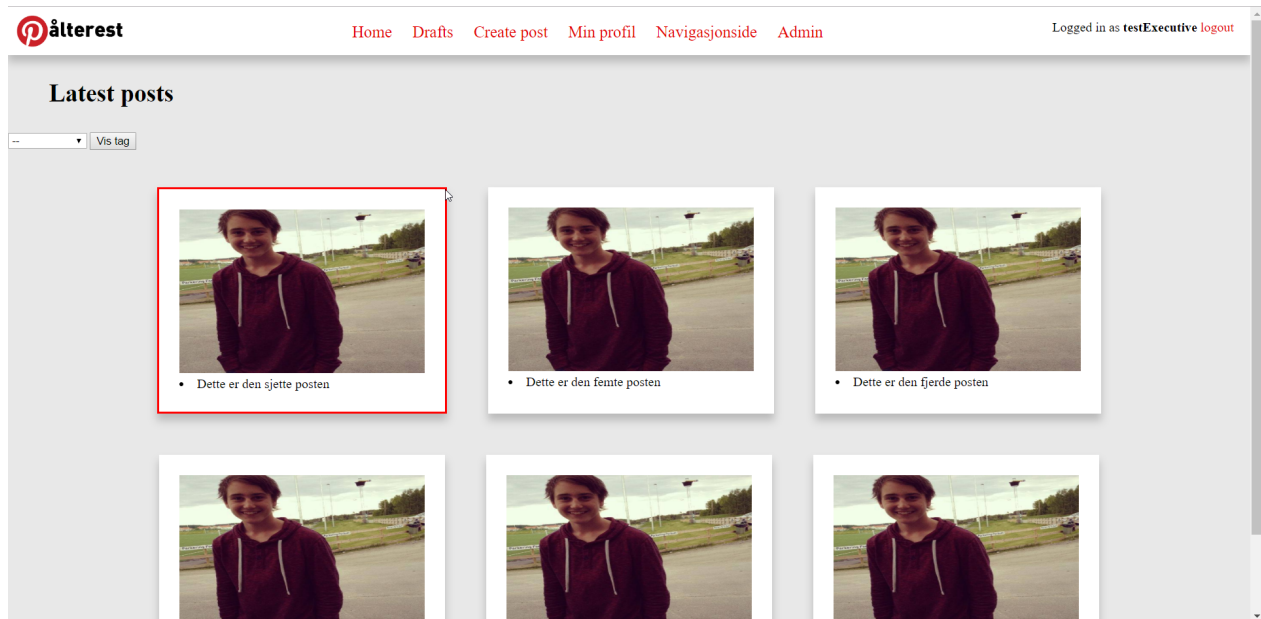


Figur 12: UML diagram av nettsiden

Figur 12 viser et UML diagram av nettsiden. Den viser hvilke modeller objektene lagres i i databasen og hvilke funksjoner og skjemaer som kjøres når brukeren utfører ulike operasjoner på nettsiden

I.7 Design

Designet for nettsiden var inspirert av Pinterest. Derfor har nettsiden et simpelt design med rød og hvit som primære farger. Vi var enige fra starten av at vi ønsket å designe nettsiden med brukervennlighet som øverste mål. Derfor har vi prøvd å få nettsiden så oversiktlig som mulig. Designet på nettsiden er hovedsaklig skrevet i CSS.



Figur 13: Forside av nettsiden

I.8 Teknisk

I.8.1 Python

Det første en må gjøre for å ta i bruk produktet er å installere Python 3.7. Python kan lastes ned fra Pythons hjemmested

I.8.2 Git

Deretter må git installeres. Dette kan gjøres på flere måter:

- For Windows, Mac og Linux
 - Denne nettsiden
- For Mac
 - Åpne terminalen og skriv inn følgende:

```
$ sudo dnf install git-all
```

- For linux
 - Åpne terminalen og skriv inn følgende:

```
$ sudo apt install git-all
```

I.8.3 Kloner prosjektet fra GitLab

Åpne terminalen og naviger til stedet du vil at mappen med prosjektet skal lastes ned. Dette gjøres med **cd** kommandoen. Feks:

```
$ cd Users/Downloads
```

for Mac og Linux

Eller for Windows:

```
$ cd c:\Users\User\Downloads
```

Videre må en skrive inn kommandoen:

```
$ git clone https://gitlab.stud.idi.ntnu.no/it1901-2018/01.git
```

I.8.4 Kjøre serveren

For å kjøre serveren må du igjen kjøre **cd** kommandoen fra forrige punkt, bare denne gangen, naviger deg inn i mappen der **manage.py** filen fra det nedlastede prosjektet ligger. Deretter:

```
$ python3 manage.py runserver
```

og du er igang!

I.8.5 Avslutte serveren

For å avslutte serveren går du inn i terminalen hvor serveren kjører og klikker **CTRL+C**

I.9 Ikke-teknisk

I.9.1 Komme deg inn på nettsiden

Start serveren ved instruksjoner under I.8.4. Åpne deretter din favotittnettleser og gå inn på localhost:8000

I.9.2 Registrer deg

1. Hvis du allerede er logget inn med en av testbrukerene, klikk logg ut
2. Klikk logg inn øverst til høyre
3. Klikk **Er du ikke registrert? Registrer her"**
4. Skriv inn et passende brukernavn og passord og klikk **registrer** ¹

I.9.3 Oppdatere profil

1. Logg deg inn med ønsket bruker
2. Klikk **"min profil"**
3. Klikk **edit"**
4. Fyll inn ønsket innhold
5. Klikk **submit"**

¹Er passordet for svakt eller brukernavnet opptatt vil du blir sendt til registreringssiden på nytt. Hvis registreringen er vellykket, vil du bli sendt til hovedsiden

I.9.4 Se en post

1. Start serveren og gå inn på nettsiden (I.8.4, I.9.1)
2. Du vil se en mengde poster, klikk deg inn på en av de

I.9.5 Legge ut en post

1. Klikk logg inn øverst til høyre
2. Logg inn med en av brukerne gitt i I.10.1²
3. Klikk på "**Create post**"
4. Lag en post og klikk submit³
5. Logg ut og logg inn med en adminbruker⁴
6. Klikk på navigasjonsside og deretter "**Review posts**"
7. Du vil nå se en liste over alle poster som ligger til godkjenning. Klikk deg inn på den du lagde i 4
8. Klikk submit

I.9.6 Lagre en post

1. Gjenta steg 1 - 3 i I.9.5
2. Istedenfor å klikke **submit**, klikker du **save**

I.9.7 Lage tag

1. Logg deg inn med en bruker med executive editor rettigheter eller høyere (I.10.1)
2. Klikk **navigasjonsside**"
3. Klikk "**create tag**"
4. Skriv inn ønsket tag og klikk **save tag**"

²Merk: Brukeren må være author eller høyere. Nylig registrerte brukere vil ikke være authors

³Du blir sendt til hjemmesiden, men posten er lagt til godkjenning av admin/executive editor

⁴Bruk "adminsom brukernavn og "adminsom passord"

I.9.8 Fjern en post fra hjemmesiden

1. Logg deg inn med en bruker med executive editor rettigheter eller høyere (I.10.1)
2. Gå inn på en post (I.9.4)
3. Klikk **delete** ⁵

I.9.9 Endre rollen til en bruker

1. For dette må en være innlogget som Admin. (I.10.1)
2. Klikk admin
3. Naviger deg inn på "**users**"
4. Klikk på brukeren du vil endre rollen til
5. Bla deg nedover til du finner **groups**"
6. Legg til og fjern grupper her
7. Bla nederst på siden og klikk **save**"

I.10 Nyttige tips til nettsiden

I.10.1 Brukere

Her er en liste over brukere som ligger i databasen, hvis en har lyst til å teste med flere forskjellige roller:

Gruppe	Brukernavn	Passord
Author	testAuthor	qwertyuiopå
Editor	testEditor	qwertyuiopå
Executive editor	testExecutive	qwertyuiopå
Admin	admin	admin

⁵Sletter ikke posten, bare fjerner den fra hjemmesiden. Den ender opp i "unpublished"