

Dokumentasjon 2 – Scrum Sprint 1: Uke 1

Teamnavn: Los Angeles

Navn	Epost	Studieretning
Abdallah Jacob Adam Abbo	abab020@student.kristiania.no	Cybersikkerhet
Benedikte Dybsjord	bedy002@student.kristiania.no	Frontend & Mobilutvikling
Riffat Hashim	riha019@student.kristiania.no	Frontend & Mobilutvikling
Raja Daud Ibrahim Nawaz	rana003@student.kristiania.no	Cybersikkerhet
Magan Ahmed Yusuf	mayu001@student.kristiania.no	Cybersikkerhet
Mahdi Ali	maal049@student.kristiania.no	Cybersikkerhet
Azaan Khan	azkh001@student.kristiania.no	Cybersikkerhet

Prosjekt: Dashboard for sikkerhetsstyring (risiko, tiltak, SoA, policyer)

Sprinttype: Scrum Sprint 1: Uke 1 – PRO203

Dato: Høst 2025

Tabellen under viser Product Backlog slik den så ut ved slutten av Sprint 1, inkludert EPICs, PBI-er, user stories, akseptansekriterier, story points og prioritert rekkefølge.

EPIC	PBI / Feature	Kort beskrivelse (User Story)	Akseptansekriterier (kort)	Story Points	Prioritet (1 = høyest)
EPIC: Prosjektforståelse & Backlog-etablering	Forstå Echomedic-case og problemdefinisjon	Som team vil vi forstå caset slik at vi bygger riktig løsning	Case forstått, problemdefinisjon skrevet, krav identifisert	2	1
	Opprette & strukturere Product Backlog	Som PO vil jeg ha en strukturert backlog med epics, PBI-er og SP	Epics er definert, PBI-er opprettet, prioritert, estimert	3	9
EPIC: Risikovurdering & risikoregister	Risiko-workshop – identifisere risikoer	Som sikkerhetsansvarlig vil jeg identifisere risikoer	Minst 10 risikoer dokumentert	5	2
	Lage risikomatrise	Som leder vil jeg se risikoer i matrise	5 x 5 matrise, fargekoder	3	4
	Opprette risikoregister	Som leder vil jeg ha komplett risikoregister	Register med alle kolonner ferdig	8	3
	Sikkerhetstesting / kvalitetssikring	Som team ønsker vi å bekrefte kvalitet	Testscenarier, funn dokumentert	5	8
EPIC: Statement of Applicability (SoA)	Velge relevante ISO 27001-kontroller	Som konsulent vil jeg velge riktige kontroller	Kontroller valgt og begrunnet	3	5
	Utkast til Statement of Applicability	Som leder vil jeg ha SoA-dokument	SoA skrevet med status/relevans	5	6
EPIC: Policyer (tilgang, logging, hendelser)	Tilgangsstyringspolicy	Som organisasjon vil vi styre tilganger riktig	Roller, passordkravm, policy utkast	3	7

	Logging- og hendelseshåndteringspolicy	Som CTO vil jeg ha logging og hendelseshåndtering	Hva logges, lagring, varselrutiner	3	10
EPIC: Prototype / UI (portal)	Opprette frontend-prosjekt	Som utvikler vil jeg sette opp prosjektet	React + Vite satt opp og kjører	2	11
	Login UI – første prototype	Som bruker vil jeg logge inn	UI, validering, ally, ingen backend	5	12
	Risikoliste UI – mock data	Som leder vil jeg se risikooversikt	Tabell m/mockdata, badges, API-klar	5	13
EPIC: Backend & DevOps	Opprette GitHub-repo	Som utvikler vil jeg ha repo og struktur	Repo, branches, README	2	14
EPIC: Dokumentasjon & eksamensleveranser	Dokumentasjon 2 – Sprint 1	Som student vil jeg levere dokumentasjon	Sprintlogg, backlog, review, retro	8	15

Product Backlog – prosess og resultat

Arbeidet med å etablere product backloggen startet med en felles gjennomgang av Echomedic-caset og problemstillingen. Basert på caset identifiserte vi hovedområdene i prosjektet og grupperte disse som EPICs. Vi endte opp med syv EPICs som representerer de sentrale arbeidstrømmene:

- Prosjektforståelse & backlog-etablering
- Risikovurdering & risikoregister
- Statement of Applicability (SoA)
- Policyer (tilgang, logging, hendelser)
- Prototype / UI (portal)
- Backend & DevOps

- Dokumentasjon & eksamensleveranser

Hver EPIC ble deretter brutt ned i Product Backlog Items (PBI-er), formulert som user stories med tilhørende akseptansekriterier. PBI-ene dekker både sikkerhetsfaglige leveranser (risikoanalyse, SoA, policyer), tekniske leveranser (UI, repo, DevOps) og dokumentasjonsarbeid. Totalt opprettet vi fjorten PBI-er.

For å få et felles estimatgrunnlag gjennomførte vi en planning-poker-økt i teamet. Vi brukte Fibonacci-skalaen (1, 2, 3, 5, 8, 13) og diskuterte kompleksitet, usikkerhet og arbeidsmengde før vi landet på story points for hvert PBI. Dette ga et felles eierskap til estimatene.

Backloggen ble deretter prioritert ved å ta et utgangspunkt i hva som er mest kritisk for prosjektets fremdrift og for Echomedic som kunde. Vi valgte å prioritere:

- Forståelse av caset og prosjektmål
- Risikoanalyse og grunnleggende risikostruktur (matrise og register)
- SoA og grunnleggende policyer
- UI-prototype for login og risikoliste
- Tekniske rammer (GitHub, frontend-prosjekt)
- Dokumentasjon og eksamensleveranser

I Scrumwise organiserte vi backloggen slik:

- Alle EPICs ble opprettet og fungerer som overordnede grupper
- Alle PBI-er ble lagt inn som backlog-items og knyttet til riktig EPIC.
- Story points ble registrert per PBI.
- Prioritet visualiseres gjennom rekkefølgen i backlogg-listene og sprint-tilordning.

Dette resulterte i en komplett, strukturert og transparent product backlog som oppfylles akseptansekriteriene:

- Alle EPICs er definert
- Alle Features/PBI-er er opprettet
- Backlog er prioritert
- Story points er estimert

Backloggen fungerer nå som styringsverktøy for videre sprintplanning.

Sprint 1 – Sprintmål

Sprintmålet for Sprint 1 var å etablere et solid fundament for prosjektet gjennom:

- forståelse av Echomedic-caset
- en strukturert product-backlog
- første versjon av risikoarbeidet (workshop, matrise og risikoregister)
- første utkast til sentrale sikkerhetsdokumenter (SoA, tilgangspolicy, logging-/hendelsespolicy)
- en første UI-prototype for login og risikoliste med mock-data
- etablering av GitHub-repo og frontend-prosjekt
- dokumentasjon av sprinten (Dokumentasjon 2)

Målet var å gjøre prosjektet klart for videre utvikling i Sprint 2, ved å kombinere både cybersikkerhets-leveranser og tekniske leveranser.

Sprint 1 – Planlegging (referat)

Tid: Mandag 1. desember, formiddag

Deltakere: Hele temaet Los Angeles (7 studenter)

Formål: Definere sprintmål, velge hvilke PBI-er som skulle inn i Sprint 1, og avklare ansvar.

Agenda og hovedpunkter:

1. Gjennomgang av product backlog
 - Vi gikk gjennom alle EPICs og PBI-er og diskuterte avhengigheter.
 - Vi identifiserte hvilke PBI-er som var nødvendige for å kunne vise et første helhetlig resultat til Echomedic (risiko + SoA + policy + UI).
2. Fastsettelse av sprintmål
 - Vi ble enige om sprintmålet beskrevet over: «etablere fundamentet for risiko- og sikkerhetsrammeverket og første UI-prototype».
3. Valg av PBI-er til Sprint 1

Følgende PBI-er ble lagt inn i Sprint 1 (ref. Scrumwise-skjermbildene):

- Førstå Echomedic-case og problemdefinisjon
- Opprette & strukturere Product Backlog
- Opprette GitHub-repositorium og struktur
- Opprette frontend-prosjekt (React + Vite)
- Login UI – første prototype
- Risikoliste UI – mock data
- Velge relevante ISO 27001-kontroller for Echomedic
- Utkast til Statement of Applicability (SoA)
- Tilgangsstyringspolicy – første utkast
- Logging- og hendelseshåndteringspolicy
- Lage risikomatrise (sannsynlighet x konsekvens)
- Risiko-workshop – identifisering av risikoer
- Opprette første versjon av risikoregister
- Dokumentasjon 2 – Sprint 1 leveranse

4. Estimering og kapasitet

- Vi la sammen story points for sprinten og så at vi la inn en relativt høy mengde arbeid til første sprint.
- Vi hadde ikke empirisk velocity fra tidligere, og undervurderte noe hvor mye tid risiko- og dokumentasjonsarbeid ville ta.
- Dette ser vi igjen på slutten av sprinten ved at tre PBI-er ikke ble ferdig (risikomatrise, risikoregister, risiko-workshop).

5. Ansvarsfordeling

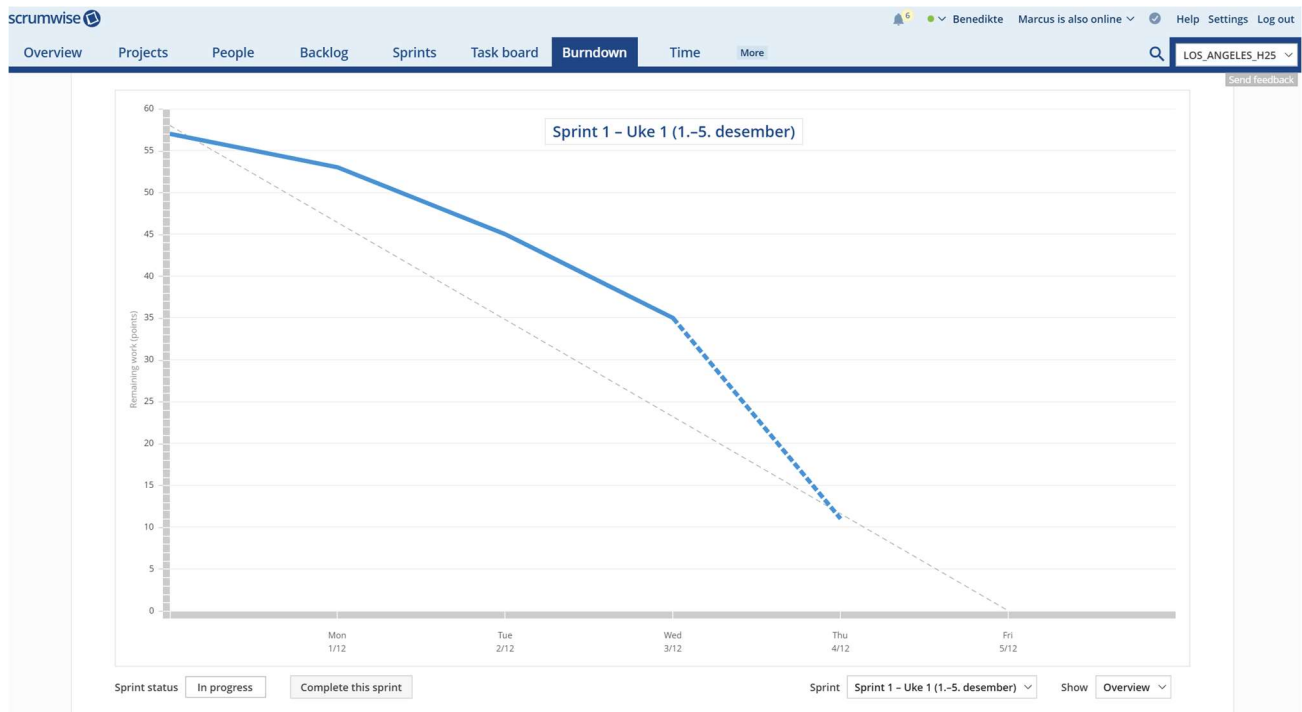
- Frontend-studentene tok hovedansvar for:
 - o GitHub-repo, frontend-prosjekt, Login UI, Risikoliste UI og Dokumentasjon 2.
- Cyber-studentene tok hovedansvar for:
 - o Risiko-workshop, risikomatrise, risikoregister, SoA, ISO-kontroller og policyer.
- Vi presiserte at alle likevel kan bidra på tvers, spesielt på dokumentasjon.

6. Arbeidsform og verktøy

- Scrumwise brukes til backlog, sprint, burndown og timeføring.
- GitHub brukes til versjonskontroll.
- Discord/Messenger for løpende kommunikasjon.
- Vi avtalte korte daglige standups (10-15 min) for status og hindringer.

Sprint 1 – Gjennomføring, burndown og tidsbruk

Burndown-graf



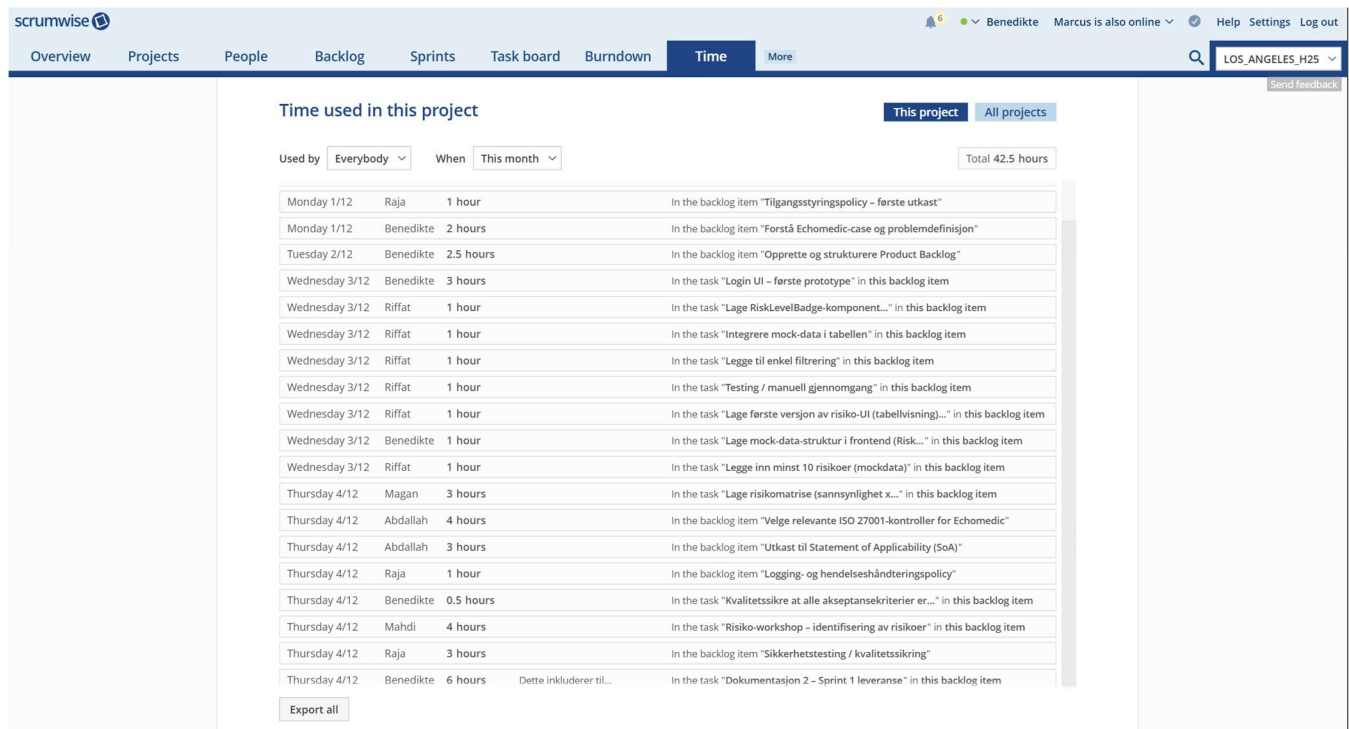
Figur 1: Burndown-graf

Kommentar til burndown-grafen (Figur 1):

- Sprinten startet med rundt ca. 58 story points (visuelt estimert fra grafen).
- Grafen viser en jevn reduksjon mandag-onsdag, og et kraftigere fall torsdag når flere PBI-er ble ferdigstilt («Done»).

- Mot slutten av sprinten ligger vi noe foran ideal-linjen, men med restarbeid på tre PBI-er innenfor risiko-EPICen.
- Mønsteret illustrerer at mye arbeid ble gjort samlet midt i sprinten, og at risikoarbeid og registreringsoppgaver ble mer tidskrevende enn planlagt.

Tidsbruk



Figur 2: Tidsbruk

Oppsummering av tidsbruk (Figur 2):

- Totalt logget arbeid i sprinten: 42,5 timer i Scrumwise.
- Tidsloggingen viser at:
 - o Frontend-studentene hadde størst timebruk på Login UI, Risikoliste UI og Dokumentasjon 2.
 - o Cyber-studentene hadde størst timebruk på ISO-kontroller, SoA-utkast, policyer, risikomatrise og risikoworkshop.
- De fleste timene ble logget onsdag og torsdag, noe som tyder på at vi kunne fordelt arbeidet mer jevnt utover sprinten.

- Vi ser også at enkelte oppgaver (særlig risiko- og SoA-relaterte oppgaver) krevde flere timer enn opprinnelig forventet ut fra story points.

Sprint 1 – Resultat (Sprint Review)

Tid: Torsdag 4. desember (kveldstid)

Deltakere: Hele teamet

Formål: Vise hva som ble levert i sprinten, og vurdere sprintmålet opp mot faktisk resultat.

Hva ble demonstrert

1. UI / tekniske leveranser
 - Login UI – første prototype
 - Vi demonstrerte en fungerende innloggingside bygget i React + Vite, med enkel validering og fokus på brukervennlighet.
 - Risikoliste UI – mock data
 - Vi viste en tabellvisning av risikoer med mock data, risk level-badges og struktur som senere kan kobles mot backend/API.
 - GitHub-repo og frontend-prosjekt
 - Repo, branching-strategi og prosjektstruktur ble presentert. React-prosjektet bygger og kjører uten feil.
2. Sikkerhetsfaglige leveranser
 - Velge relevante ISO 270001-kontroller for Echomedic
 - Minimum ti kontroller er valgt, begrunnet og klare til å brukes i SoA.
 - Utkast til Statement of Applicability (SoA)
 - Første versjon av SoA-tabell er laget med kontroll-ID, relevans, status og begrunnelse.
 - Tilgangsstyringspolicy – første utkast
 - Kort policy (1-2 sider) med roller, ansvar, autentiserings- og passordkrav i tråd med Normen og ISO 27001.

- Logging- og hendelseshåndteringspolicy
 - Policyutkast som beskriver hva som skal logges, krav til sikring av logger og rutine for håndtering av sikkerhetshendelser.
- 3. Prosjektforståelse og backlog
 - Forstå Echomedic-case og problemdefinisjon
 - Problemdefinisjon er formulert og diskutert i teamet.
 - Opprette & strukturere Product Backlog
 - Vi viste en komplett product backlog i Scrumwise med EPICs, PBI-er, estimer og prioritet.
- 4. Dokumentasjon
 - Dokumentasjon 2 – Sprint 1 leveranse
 - Utkast til denne rapporten ble gjennomgått, med plan for ferdigstilling og innlevering.

Items som ble «Done»

I Scrumwise er følgende PBI-er markert som fullført («Done») ved slutten av sprinten:

- Forstå Echomedic-case og problemdefinisjon
- Opprette & strukturere Product Backlog
- Opprette GitHub-repositorium og struktur
- Opprette frontend-prosjekt (React + Vite)
- Login UI – første prototype
- Risikoliste UI – mock data
- Velge relevante ISO 27001-kontroller for Echomedic
- Utkast til Statement of Applicability (SoA)
- Tilgangsstyringspolicy – første utkast
- Logging- og hendelseshåndteringspolicy

(Dokumentasjon 2 settes også til Done når denne rapporten leveres).

Items som ikke ble ferdigstilt

Tre PBI-er står igjen som In progress:

- Lage risikomatrise (sannsynlighet x konsekvens)
- Risiko-workshop – identifisering av risikoer
- Opprette første versjon av risikoregister

Disse oppgavene hadde høy kompleksitet og var med tidskrevende enn antatt, spesielt fordi de krever både sikkerhetsfaglig vurdering og strukturert dokumentasjon. Teamet besluttet i sprint review at alt restarbeid flyttes inn i Sprint 2, med mulig oppdeling i mindre tasks.

Vurdering opp mot sprintmålet

Sprintmålet ble delvis oppnådd:

- Vi oppnådde målet om fundament for UI, SoA, policyer, backlog og teknisk struktur.
- Vi kom godt i gang med risikoarbeidet, men fikk ikke ferdigstilt risikomatrise, risikoregister og full risiko-workshop innenfor sprintens tidsrom. Dette blir hovedfokus tidlig i Sprint 2.

Sprint 1 – Retrospective (referat)

Tid: torsdag 4. desember, etter sprint review (kveldstid)

Deltakere: Hele teamet

Format: «Start – Stop – Continue» + konkrete tiltak for Sprint 2

Hva fungerte bra (Continue)

- Tverrfaglig samarbeid: Frontend- og cyberstudentene jobbet tett sammen, spesielt rundt SoA og policyer der teknisk forståelse og sikkerhetskrav måtte møtes.
- Bruk av Scrumwise: Backlog, sprintboard, burndown og timeføring ga god transparens for hele teamet.

- Planning poker: Estimeringsøkten bidro til gode diskusjoner om kompleksitet og avhengigheter.
- Leveranser av UI og policyer: Vi leverte konkrete, synlige resultater både på frontend og sikkerhetsdokumentasjon

Hva var utfordrende (Stop / forbedre)

- Overcommitting i første sprint: Vi tok inn for mange story points uten å kjenne på vår faktiske velocity. Dette førte til at risiko-PBI-ene ble påbegynt, men ikke ferdigstilt.
- For grove PBI-er innen risiko: Spesielt «Opprette første versjon av risikoregister» (13 SP) var stort og burde vært brutt ned i mindre, sprint-bare oppgaver.
- Sen oppstart på enkelte oppgaver: Noe av risikoarbeidet startet sent i sprinten, samtidig som policyer og SoA også skulle fullføres.
- Ujevn timeføring: Mange timer ble logget tett opp mot slutten av sprinten. Vi kunne hatt bedre jevn progresjon og mer konsekvent oppdatering av gjenstående arbeid.

Hva vi vil gjøre annerledes i Sprint 2 (Start)

1. Velge sprintomfang basert på faktisk kapasitet
 - «Først estimere, så velge en realistisk mengde PBI-er».
 - For Sprint 2 vil vi ta inn færre story points og lage en tydeligere prioritering.
2. Bryte ned store risiko-oppgaver
 - De gjenstående PBI-ene for risikovurdering og risikoregister deles opp i mindre tasks (f.eks. per risikokategori eller per dokument-del).
3. Strammere Definition of Ready og Definition of Done
 - Vi formaliserer hva som må være på plass før en oppgave tas inn i sprint (DoR), og hva som kreves for å sette «Done» (DoD), spesielt for dokumentasjons- og risikooppgaver.
4. Bedre jevn fordeling av arbeid og timeføring
 - Vi avtaler tydeligere del-frister inne i sprinten (mid-sprint checkpoints) og sørger for at timeføring gjøres daglig.

Refleksjon rundt estimering og en lærerkommentar

Emneansvarlig presiserte at vi ikke har erfaring med å kjøre en sprint, og at vi derfor bør:

«...starte med backlog og deretter sprint backlog. Ved estimering hvert item dere har (10-15 første f.eks) så ser dere hvor mye det er. Når dere så velger ut hvilke items dere tror dere rekker i løpet av uken så ser dere hvor mange storypoints dette blir.»

I Sprint 1 fulgte vi først del av rådet: vi etablerte en strukturert product backlog og estimerte PBI-ene med planning poker. Vi tok likevel inn for mange story points i sprinten, uten reell historikk på hvor mye teamet klarer å levere på én uke.

Konsekvensen var:

- Vi levere godt på UI, SoA, policyer og prosjektforståelse,
- men undervurderte tidsbruk på risikoanalyse og risikoregister, slik at tre risiko-relaterte PBI-er ble stående som «In progress».

Til Sprint 2 tar vi med oss følgende justeringer:

- Vi bruker faktisk resultat fra Sprint 1 til å anslå en mer realistisk team velocity.
- Vi prioriterer ferdigstilling av risiko-EPICen før vi legger inn nye store oppgaver.
- Vi vil visualisere total story points for sprinten og aktivt diskutere «cut-line» før vi starter sprinten.
- Vi aksepterer heller å levere litt mindre, men 100% ferdig, enn å ha mange påbegynte PBI-er.

Dette er i tråd med Scrum-tanken om empirisk prosesskontroll: vi lærer av første sprint og justerer kapasitet, planlegging og nedbrytning til neste.

Oppsummering

Sprint 1 ga oss:

- Et tydelig fundament:
 - o forståelse av Echomedic-caset
 - o strukturert product backlog
 - o GitHub-repo og frontend prosjekt

- første UI-prototyper for login og risikoliste
- førsteutkast til SoA og sentrale sikkerhetspolicyer
- Viktig læring rundt:
 - estimering og sprint-omfang
 - hvor tidskrevende risikoanalyse og dokumentasjon faktisk er
 - behovet for å bryte ned store oppgaver og sikre jevn progresjon

Til Sprint 2 går vi inn med et mer realistisk syn på kapasitet, tydelig prioritet på å fullføre risikoarbeidet, og mer modne Scrum-rutiner for planlegging, oppfølging og retrospektiv.

Skjerm bilde av «Backlog»-siden i Scrumwise:

