



Forside

IS-304: 2022

Tittel: Smidig Utvikling av ETL system for Egde

Emnekode:	IS-304
Emnenavn:	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder:	Jaziar Radianti
Oppdragsgiver:	Egde Consulting AS

Studenter:

Etternavn	Fornavn
Bero	Dzenet
Hodzalari	Adel
Testad	Oskar

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

Sammendrag

Bachelor gruppen RefreshedIT har benyttet seg av samarbeidet mellom Universitet i Agder og Egde for å skrive en bedrifts basert bacheloroppgave. Det hele begynte da Universitet i Agder la til rette for at de ulike samarbeidsbedriftene kunne presentere seg selv og sine ulike oppgaver for student gruppene under arrangementet RefreshIT. Bacheloroppgaven går ut på å bruke Azure og Databricks til å transformere og laste opp data med oversikt over konsulenter, hvor de er utplassert, hvor lenge og hvilke ferdigheter en konsulent har.

Gruppens primære arbeidsmetodikk var Scrum. Valget for å benytte Scrum som arbeidsmetodikken vår i dette prosjektet falt naturlig, da vi har hatt positiv erfaring med Scrum fra tidligere prosjekt og har funket bra i forhold til gruppens dynamikk. Gruppen forstod før prosjektet startet at det var stor sannsynlighet for endringer underveis, og da virket det logisk å gå for en agil arbeidsmetodikk som Scrum. Gruppen hadde fra starten av et ønske om å gi Egde et produkt som kunne implementeres i fremtiden, vi har da gjennom prosjektet forholdt oss til ulike kvalitetssikringstiltak. Tiltakene vi har forholdt oss til er jevnlig kommunikasjon med produkteier, brukt offisiell dokumentasjon for kode, regelmessig pushet opp til GitHub og utført regelmessig testing av funksjonaliteter som ble laget.

Gruppen har til gjengjeld hatt et godt læringsutbytte fra starten av prosjektet, hvor vi har fått et innblikk i hvordan det vil være å forholde seg til andre viktige personer i et prosjekt. Vi har også fått muligheten og erfaringen av å jobbe med ny, moderne og ettertraktet teknologi og har derfor lært mye nytt som vi kan ta med oss videre.

Prosjektets resultat er et "proof of concept" (POC) på at en potensiell sammen-fletting av Azure, Databricks og Datalake og intern informasjon om konsulenter, kan føre til en mer behagelig og effektiv fremgangsmåte av logging. Systemet tillater at salgs og HR avdelingen kan enkelt hente ut spesifikke konsulenter som dekker en kundes behov uten å måtte gjøre dette manuelt og med flere ledd.

Innhold

Sammendrag	3
Figurliste.....	6
Tabelliste	6
1. Introduksjon	7
1.1 Om Egde Consulting AS.....	7
1.2 Prosjekt beskrivelse	7
1.3 Gruppen	9
2. Teknologi	9
2.1 ETL vs. ELT	9
2.2 Data varehus	11
2.3 Hva er en Datalake og Datalake house	12
2.4 Hva er databricks	12
2.5 Logic Apps og API	12
3. Prosjektstyring.....	13
3.1 Arbeids metodikk.....	13
3.1.1 Scrum og roller.....	13
3.1.2 Sprint og product backlog.....	14
3.1.3 Tidsestimering og Koordinasjon.....	14
3.1.4 Prosjektstyrings Verktøy	16
3.1.5 Kommunikasjon	17
3.2 Systemkrav	18
3.2.1 Kartlegging	19
3.2.3 Prosjektstyrings krav	19
3.2.4 Teknologiske krav.....	20
3.2.5 Kvalitets krav	21
3.3 Kvalitetssikring	23
3.3.1 Definerings av kvalitet	23
3.3.2 Teknologiske valg for å sikre kvalitet (Versjons kontroll, etc.)	23
4. Prosjektgjennomføring	26
4.1 Uthenting av data	26
4.2 Transformering og utsending av data.....	29
5. Produktet.....	34
6. Utfordringer.....	34
6.1 Risiko håndtering	35

6.2 SARS-CoV-2 utfordringer.....	36
6.3 Andre utfordringer.....	37
7. Refleksjon	38
7.1 Prosjekt gjennomføring.....	38
7.1.1 Scrumban	38
7.1.2 Risikohåndtering	39
7.1.3 Estimering og prioritering	40
7.1.4 Kommunikasjon	40
7.2 Kvalitet av sluttproduktet	41
7.2.1 Refleksjon på egen læring.....	41
7.3 Utfordringer	41
7.3.1 Covid-19	42
7.3.2 Roller og Rollefordeling.....	43
7.4 Brukerhistorie	43
8. Konklusjon	44
Referanseliste	45
9. Vedlegg	48
9.1 Gruppekontrakt.....	48
9.2 Styregruppemøter.....	52
9.2.1 Styremøte 1.....	52
9.2.2 Styremøte 2.....	53
9.2.3 Styremøte 3.....	55
9.3 Uttalelse fra Egde.....	57
9.4 Backlog	58

Figurliste

Figur 1 ETL Prosessen (BMC, 2022)	10
Figur 2 ELT Prosessen (Qlik, n.d.)	11
Figur 3 Tidestimering	14
Figur 4 Tidsestimering for sprinten	15
Figur 5 Loggføring av timer i Excel	16
Figur 6 Oversikt over ulike tilstander	17
Figur 7 Oversikt over Discord kanaler	18
Figur 8 Brukerhistorie i Jira	24
Figur 9 Kvalitetsikring	25
Figur 10 Uthenting av data	27
Figur 11 Skjerm bilde av oppkobling av datalake	28
Figur 12 Mangel på rettigheter	28
Figur 13 Transformering av data	29
Figur 14 Fyller nullverdier	30
Figur 15 Allokeringstabell	30
Figur 16 Erfaring tabellen	31
Figur 17 Konsulenter tabellen	31
Figur 18 Fil Struktur i Datalake	32
Figur 19 Fil Struktur i Datalake	32
Figur 20 Risiko Vurdering	35
Figur 21 Risiko Håndtering	36

Tabelliste

Tabell 1 Systemkrav	21
---------------------------	----

1. Introduksjon

Denne rapporten er skrevet med utgangspunkt at Egde ønsker å håndtere ustrukturert data for å gi en mer verdifull og effektiv allokeringsprosess av sine konsulenter. Rapporten vil redegjøre for de mest sentrale avgjørelsene som blir tatt under prosjektet, rapporten vil også vise systemet som blir utviklet og beskriver aktuelle systemutviklings prosesser og fremgangs måter. Første delen av rapporten handler om valgene som er tatt i forbindelse med teknologi, rammeverk og metodikker. Etterfulgt av en beskrivelse av verktøy som blir brukt i forbindelse med prosjektstyring og avslutter med en gjennomgang av produktet og hvordan det ser ut og til slutt en refleksjon over løsningen vår.

Gruppen synes at den problemstillingen Egde presenterte virket spennende, utfordrende og ikke minst relevant i forbindelse med dagens teknologi.

1.1 Om Egde Consulting AS

Egde er Sørlandets ledende fagmiljø innen digitalisering og teknologi (Egde, 2021). Bedriften har nærmere 100 ansatte fordelt på kontorer i Kristiansand, Grimstad og Porsgrunn. Ved hjelp av Egde sin helhetlige og tverrfaglige tilnærming til ulike aspekter innen digitalisering, hjelper de private og offentlige virksomheter til å legge grunnlaget for strategiske beslutninger som skal sikre effektiv fornyelse (Egde, 2021).

Vi har forholdt oss til to veiledere fra Egde, hvor begge spesialiserer i relevante fagområder som vi trenger i forbindelse med prosjektet. I tillegg til dette har vi fått full tilgang til Teams plattformen som bedriften benytter seg av og kan sende meldinger til alle ansatte skulle det være behov for å få kontakt med en spesifikk person.

1.2 Prosjekt beskrivelse

Prosjektet handler om å hente, transformere og sende ut data, altså en ETL (Extract, Transform, Load) prosess. Vi skal benytte oss av ulike teknologier som Azure Databricks, Azure Datalake, Python og MYSQL. Dataen vi skal jobbe med er utdelt fra Egde og må hentes (extract) fra Sharepoint og inn i en Datalake. Dataen skal videre hentes fra Datalaken og inn i Databricks. Her vil vi bruke Python og MYSQL for å transformere (transform) dataen og til slutt sende den ut (load) (se videre forklaringen av ETL, Datalake og Databricks konsepter i kapittel 2).

Prosjektet vil ikke bli et ferdigstilt produkt, men vil bli et eksperiment som tester om Databricks, potensielt kan erstatte andre programvareprodukter som har både større kostnader og unødvendig funksjonalitet. I tillegg vil vi teste Databricks sin oppkoblingsmulighet til bedriftens internprodukter.

Vi har også utarbeidet scoopet over tiden. I tillegg til at gruppen skal gjennomføre en ETL prosess, skal vi prøve og utfylle ønsket til Salgsavdelingen og HR-avdelingen om å lage en oppkobling fra databasen gruppen lager til Xledger. Xledger er det systemet Egde bruker for fremvisning av faktura nummer, Brutto/Netto fortjeneste per kunde og oversikt over kontrakten til konsulentene og bedriften de er solgt hos. Salgsavdelingen i Egde ønsker at systemet gir mulighet for å få frem de viktige variablene for hver kunde Egde har avtale med.

HR-avdelingen har også vært med på å utarbeide scoopet på produktet. Det er slikt at HR-avdelingen også ønsker at vi inkluderer en oversikt over alle ansatte og hvor mange barn de har under X-antall år, hvilken fagligbakgrunn ansatte har. Koble opp kompetanse med sertifiseringer, kurser som per dags dato ligger på e-post hos HR-ansvarlig. Oversikt over kursene som er "Påbegynt" eller "fullført", varslinger på kompetanseheving ha et varsel når en konsulent er ute av et oppdrag i forbindelse med kursing. HR-ansvarlig ønsker også en tilkobling mellom HR-systemet og Xledger.

Ettersom at vi underveis i prosjektet identifiserte nye systemkrav, måtte vi legge til rette for endringer. Vår agile arbeidsmetodikk gjorde det mulig å enkelt håndtere slike endringer når disse dukket opp.

1.3 Gruppen

Våren 2022 skal gruppen RefreshedIT utføre og levere sin bachelor i IT og Informasjonssystemer ved Universitet i Agder. RefreshedIT er en gruppe som siden starten av studieprogrammet har jobbet sammen og består av 3 medlemmer som utfyller hverandres styrker og svakheter på en god måte og har lært hvordan best jobbe sammen. Når det gjelder gruppens arbeid rundt prosjektet vil det i begynnelsen foregå digitalt på grunn av Covid-19, dette inkluderer forelesninger og samlinger på Universitetet i Agder sin side, men også møter med bedriften studentene skal utføre bachelorprosjektet sitt hos. Dersom Covid-19 restriksjonene lettes vil det sannsynligvis bli tilrettelagt for mer fysisk arbeid, men dette er noe som vil bli vurdert kontinuerlig av gruppen og bedriften. Gruppen hadde fra starten lyst til å jobbe med et reelt prosjekt fra næringslivet hvor en eventuell løsning og arbeid som utføres blir tatt i bruk og gir en form for verdi til bedriften.

2. Teknologi

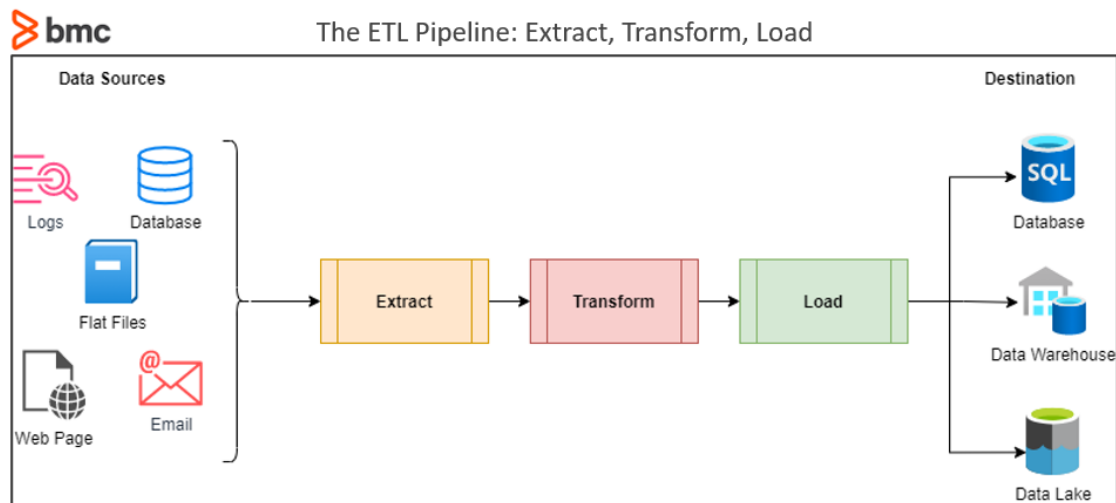
Videre i rapporten vil vi snakke om ulike systemer, verktøy og prosesser som vi har brukt gjennom prosjektet. I dette kapitlet går vi gjennom de ulike begrepene som blir brukt og gir en grunnleggende forklaring av konseptene. Her vil vi gå gjennom ETL, ELT, Databricks, Data varehouse, Logic Apps og API`er.

2.1 ETL vs. ELT

Både ETL (Extract, Transform og Load) og ELT (Extract, Load og Transform) er data integrasjons metoder som brukes for å overføre data fra en data kilde til en data lagrings plattform. Det er mange likheter, men også en del fundamentale ulikheter mellom de to metodene.

Extract, Transform, Load (ETL) metodologien henter rå data fra en eller flere datakilder til en sekundær server eller arbeidsområde. Her blir dataen transformert etter formål og spesifikasjoner og til slutt blir dataen fraktet videre til et endepunkt, som hovedsakelig er en form for lagrings plass som en database, data varehouse eller liknende (Bartley, K. 2021). ETL metoden blir ofte brukt når dataen som skal transformeres må tilpasses slutt destinasjonen, eller databasen (Se figur 1).

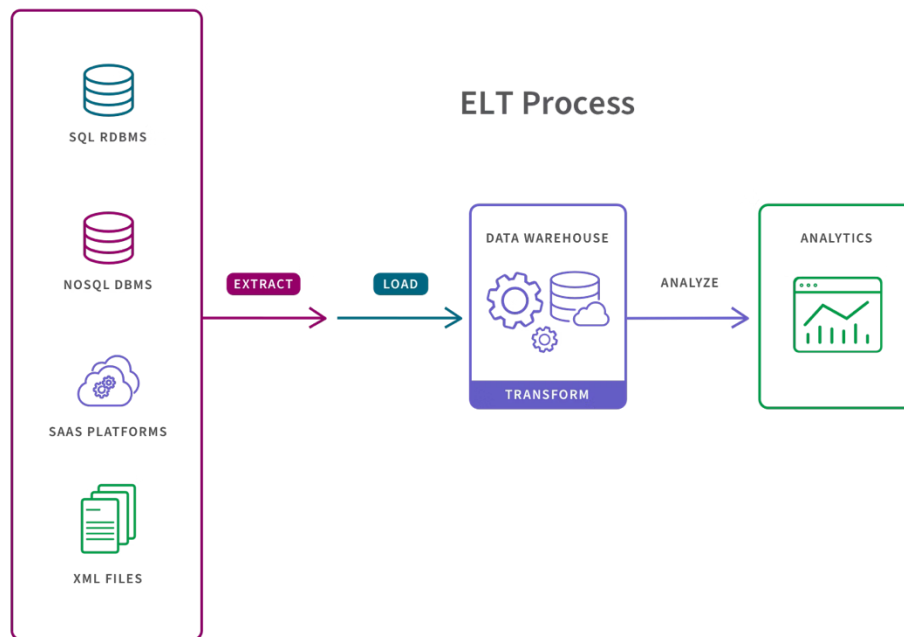
ETL metoden er også foretrukket når en bedrift eller aktør har en “On-premise” løsning med databasen som har en fast, uskalerbar og avgrenset tilknytting til minne og prosesseringskraft på server siden (Bartley, K. 2021)



Figur 1 ETL Prosessen (BMC, 2022)

Den mest åpenbare fordelen med et ETL system er at ETL har vært brukt i 20 år, og har en stor mengde hjelpemidler og dokumentasjon skulle det oppstå problemer ved implementering eller generell bruk. I tillegg er det den beste metoden for å flytte store mengder data til en destinasjon med spesifikke regler og strukturer. ETL har også ulemper, en av dem er at det ikke er den mest ideelle løsningen dersom man ønsker å oppnå sanntids eller “on-demand” data tilgang, hvor rask respons er nødvendig. I tillegg så tar det flere måneder å implementere en fullstendig ETL struktur innad en bedrift (InetSoft. n.d).

I motsetning til ETL, hvor transformerings laget må skje før opplastingslaget, eller “load” fasen (se figur 2). Er ikke det et krav for ELT (Extract, Load, Transform) metoden. ELT laster inn rådata direkte til den spesifiserte slutt destinasjonen fremfor å først flytte dataen til et mellomledd for transformering (Bartley, K. 2021). Transformerings skjer i databasen eller data varehuset, som tillater for flere ulike formateringer som kan bli hentet ut etter behov. ELT er en relativt ny løsning og krever at databasen hvor dataen blir lagret er skalerbar og sky basert for best resultat. Populære løsninger som fungerer bra med ELT-metodikken er Cloud Datavarehus som Snowflake, Amazon Redshift og Microsoft Azure (Bartley, K. 2021).



Figur 2 ELT Prosessen (Qlik, n.d.)

Den mest merkværdige fordel ved bruk av ELT fremfor ETL er at ELT er generelt et raskere system når det gjelder enkle oppgaver, og gjerne oppgaver som må gjentas flere ganger. I tillegg er ELT systemet billigere i lengden da du kan spesifisere hvor mye prosesserings kraft du trenger til enhver tid, og med det kan du alltid skalere opp eller ned alt etter behov (Data Intergration News, 2020).

Det er også et par ulemper ved bruk av et ELT-system. Den største ulempen er hvor mye system ressurser som kreves for å kunne brukes effektivt. Grunnen til det, er at for å lagre dataen som kommer fra ELT, så kreves det flere ressurser fra de aktuelle systemene som servere og andre datahåndterings lag (Laue, M. R, 2022).

2.2 Data varehus

Et data varehus er en type data lagring og data håndterings system som er designet til å forenkle og gjøre støtte mulig for business intelligence (BI) aktiviteter og ulike former for analyse (Oracle, n.d.). I et data varehus blir store mengder data sentralisert og brukt for å videre analysere viktige aspekter. Dette gjør de fleste organisasjoner og tar avgjørende beslutninger basert på denne analysen.

Disse ulike beslutningene kan ofte være avgjørende for en bedrift i forbindelse med å holde følge med andre bedrifter i et konkurransedyktig marked.

2.3 Hva er en Datalake og Datalake house

En datalake er en sentralisert plass hvor man kan lagre alt av strukturert og ustrukturert data av ulike mengder. Datalaken vil tillate analytiske handlinger, visualiseringer av store mengder data, sanntids analyseringer og maskinlæring for å deretter kunne ta bedre og avgjørende valg (LeClerc, B., & Cale, J, 2020).

Datalake-house er et datalagrings konsept som kombinerer sentrale elementer fra de to andre nevnte løsningene for lagring av data. Datalake house kombinerer de beste elementene fra Data Varehus og Datalake. Måten det blir gjort på er at et Datalake house tar for seg den generelle strukturen og oppbygningen av et data varehus. Samtidig som den tar for seg sortering, maskinlæring og liknende (Snowflake, n.d.).

2.4 Hva er databricks

Databricks er et sky-basert datateknisk verktøy som blir brukt av bedrifter til å prosessere, transformere og utforske store mengder data (Data Intergration News, 2020). Hovedsakelig blir Databricks brukt til å gå gjennom store mengder med data og bruke den dataen til ulike formål. Verktøyet gir tilgang til Spark SQL og aktive koblinger til visualiserings verktøy som Power BI, Qlikview og Tableau. Databricks kan også brukes til å bygge prediktive modeller ved hjelp av SparkML.

Når det gjelder utvikling i Databricks, som er hoved oppgaven vår dette bachelorprosjektet, får vi tilgang til kodespråkene R, Scala, Python og SQL via Apache Spark. Disse forskjellige programmerings språkene er det vi skal forholde oss til når vi lager systemet vårt.

2.5 Logic Apps og API

I prosjektets start fase ble vi informert om at automatisering av overføringen og flettingen av data fra Sharepoint og CV-partner var et krav. Etter flere henvendelser til veiledere med ulike løsninger kom vi frem til at Microsoft sin Azure Logic Apps ville passet utmerket til prosjektet vårt.

Azure Logic Apps er en sky-basert plattform som tillater brukeren å lage og kjøre automatisert arbeidsflyt og integrere ulike apper, informasjon, tjenester og systemer som er støttet (Microsoft, n.d).

Vi fikk også vite at CV Partner datakilden skulle hentes ut ved hjelp av et API, som vi får tilgang til av Egde. API står for “Application Programming Interface” og gir muligheten for en eller flere applikasjoner å snakke med hverandre (Mulesoft, n.d.).

Planene for bruk av Logic Apps i vårt system var å automatisere henting fra Sharepoint og overføring til Egde sin interne datalake. I tillegg til dette skulle det brukes et API som skal hente data fra CV Partner og overføre dette til datalaken, sammen med data fra Sharepoint. På grunn av manglende rettigheter og avansert oppsett ble vi enige med vår veileder at vi hopper over dette steget, slik at Egde selv kan vurdere løsningen vår på dette problemet.

3. Prosjektstyring

3.1 Arbeids metodikk

I dette kapitlet vil vi forklare vår arbeidsmetodikk. Her vil vi legge til rette for hvordan vi tilnærmer oss den agile SCRUM metoden, hvordan tidsestimering og prioritering er blitt gjort, hvilke prosjektstyringsverktøy som har blitt brukt og hvordan vi har kommunisert, både internt i gruppen, men også eksternt med Egde.

3.1.1 Scrum og roller

Vi har valgt Scrum til å være vår hoved metode for fordeling av arbeid og oppdatering av status på de gitte oppgavene. Ved hjelp av verktøyet “Jira” har vi en hybrid tilknytning til Scrum og kanban metodikk. Med dette menes at vi har de tradisjonelle standup møtene, samt holder med jevne mellomrom scrum retrospektiv møter sammen med vår veileder hvor vi gjennomgår ting vi ikke synes gikk bra under sprinten, hva vi skal forbedre og hva som gikk bra.

Scrum og kanban er veldig ulike agile rammeverk som spesialiserer i forskjellige aspekter av prosjektledelse. Scrum fokuserer på status rettet og grundig planlagt arbeid, hvor teamet oppdateres jevnlig med hva som jobbes med i regelmessige Sprint møter og liknende (Hamilton, T, 2022).

Kanban metodikken er mer rettet mot visualisering av arbeid og måler gruppens effektivitet med de oppgavene som er satt opp (Hamilton, T, 2022). Det er derfor ikke behov for møter i forhold til oppgaver når man bruker Kanban metodikk fordi alt av oppgavene blir tildelt på et brett som alle på laget har tilgang til.

Siden vi kun er 3 medlemmer i gruppen var det enkelt å fordele rollene, vi har en prosjektleder, en dev ops ansvarlig og en ansvarlig for dokumentasjon av arbeidsprosesser.

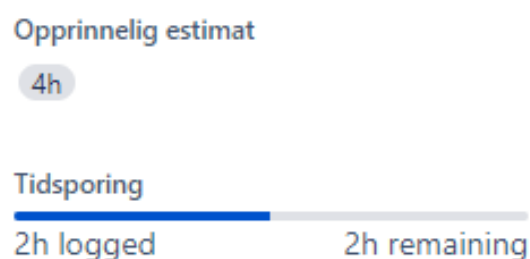
3.1.2 Sprint og product backlog

Ettersom at gruppen jobbet med et stort prosjekt så krevde det og finne ut av en god metodikk på strukturering av oppgaver. Derfor valgte vi å ta i bruk av Sprint og Product backlog. Vi brukte kun Jira som et dev ops program for å strukturere oppgavene. Som du ser i vedlegg 9.4, så viser bildet hvordan gruppen har valgt å strukturere oppgavene. Når en sprint er gjennomført så vil alle oppgavene som ikke er “fullført” bli automatisk sendt videre til neste sprint.

3.1.3 Tidsestimering og Koordinasjon

Gruppen anser tidsestimering som en viktig rolle for prosjektgjennomføring. Ved å tid estimere oppgavene vi gjennomfører under dette prosjektet, vil gruppen få et bedre innsyn på frister, ressurser og tidsbruken på oppgavene (“Estimating Time Effectively,” 2020).

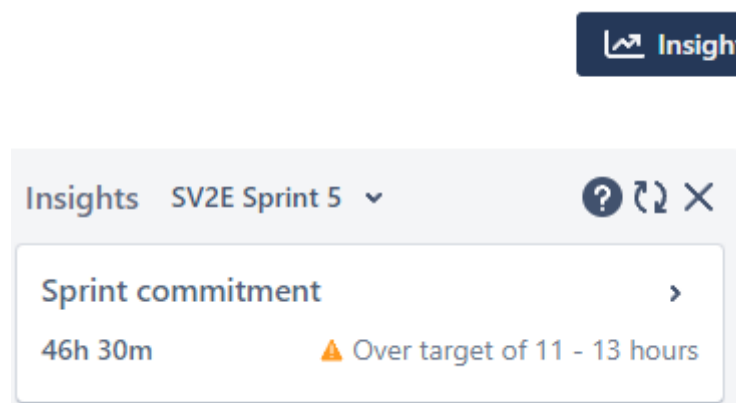
Verktøyet Jira gir også mulighet for å tids estimere alle oppgavene som blir opprettet under dette prosjektet. I Jira kan vi legge inn «Forventet tid» det vil ta for å gjennomføre oppgaven, men også «Den reelle» tiden det tok å gjennomføre oppgaven. Dette gir gruppen innsyn på hvor lang tid oppgaven egentlig tok i motsetning til det vi «trodde». Dette hjelper gruppen med å lære av egne feil og vil forbedre tids estimeringen av oppgaver.



Figur 3 Tidsestimering

Figur 3 viser hvordan tids estimeringen i Jira ser ut. Først når man lager en oppgave og skal legge inn tidsestimering så legger man inn det gruppen «forventer» at oppgaven skal ta. Etterhvert som oppgaven jobbes med, legger man inn timene brukt, timene brukt blir representert med «blå» linje og timene som er «igjen» er representert med «grå» linje. Også har vi det som heter «opprinnelig estimat» som er det estimatet gruppen opprinnelig la inn når oppgaven ble opprettet.

Det er også slikt at gruppen estimerer timer på alle sprintene som gjennomføres. Dette er for å sikre forventet fremgang for hver sprint. Gruppen har regnet ut at ukentlig så skal vi dedikere mellom 20-25 timer(t) til prosjektet. Ettersom at sprintene våre går over to-ukers intervaller så vil det bety at gruppen skal dedikere mellom 40 – 50t for hver sprint. Ved å bruke Jira sin «Insights» funksjon så får man frem timene som ble brukt for den dedikerte sprinten. Som man kan se i figur 4 så har gruppen brukt ca. 46t i femte Sprint.



Figur 4 Tidsestimering for sprinten

Det er også slikt at gruppen bruker Excel for å loggføre timer og dette er for å kvalitetssikre at timene som vises i Jira er «riktig». Ettersom at Jira er et nytt verktøy for gruppen så gjorde vi det slikt at vi både loggfører timer i Jira og i Excel. Da kan gruppen se eventuelle feil ved tids estimering. Figur 5 viser hvordan loggføringen blir gjort i Excel arket.

Dato	Oppgave	Tid brukt (timer)	Navn		Dato	Møter/Forelesning	Tid brukt (timer)	Navn
21.03.2022	Jobbe med Assignment 4	5	Alle		24.03.2022	Møte med Microsoft	1	Dzenet og Oskar
22.03.2022	Jobbe med data Lakehouse	4	Alle		25.03.2022	Styringsgruppemøte	1	Alle
23.03.2022	Skrivedag	5	Alle		30.03.2022	Presentasjon	1	Alle
25.03.2022	Jobbe med lakehouse og delta lake	2	Alle					
28.03.2022	Presentasjoner	3	Alle					
29.03.2022	Commands, Presentasjon	4	Alle					
30.03.2022	Skrivedag etter Presentasjon	4	Alle					
31.03.2022	Skrivedag	5	Alle					
01.04.2022	Skrivedag	5	Alle					
	Total:	37				Total:	3	
Samlet Total								
Seksjon: Tid:								
Møter/Forelesning				3				
Oppgaver				37				
Total:				40				

Figur 5 Loggføring av timer i Excel

3.1.4 Prosjektstyrings Verktøy

Som nevnt innledende så finnes det mye forskjellige prosjektstyrings verktøy som man kan bruke. Gruppen har prøvd diverse verktøy som Trello, Azure DevOps, Jira, men etter en diskusjon med Egde så valgte vi å fortsette med Jira. Dette er fordi Egde bruker Jira og det var lettere for gruppen å få en god innføring i dette verktøyet. Verktøyet har en Scrum tilnærming. Rammeverket kombinerer viktige aspekter fra to populære agile metodologier, Kanban og Scrum (ProductPlan, 2021). I Jira har gruppen en helhetlig oversikt over hvilke oppgaver vi har, hvilke oppgaver vi jobber på og hvilke oppgaver som er ferdig. Jira brukes for å strukturere Sprintene, samt også for å tids estimere alle oppgavene. Mange av oppgavene blir formulert som en brukerhistorie og dette er for å simplifisere oppgaven for medlemmene i gruppen. Hver oppgave består av det Jira kaller "Child-issues" som er del-oppgavene.

Alle oppgavene som blir lagt inn kan bli satt i fire forskjellige “states”. Dette vises i figur 6 der de ulike tilstandene er: “Til utføring”, “under arbeid”, “Testing” og “utført oppgave”. Når en oppgave er under “til utføring” så er den enda ikke startet på, men målet er at den skal være fullført innen sprinten er ferdig. “Under arbeid” så Jobbes oppgaven med, men den er enda ikke fullført. “testing/review” delen er den delen der gruppen ser hva som har blitt gjort, og om oppgaven har møtt kravene som har blitt satt. Når alt er i orden så blir den flyttet til “utført oppgave”

Jira er et verktøy som inneholder mye gode funksjoner for prosjektstyring, noe som er grunnen til at vi har valgt å kun fokusere på Jira.



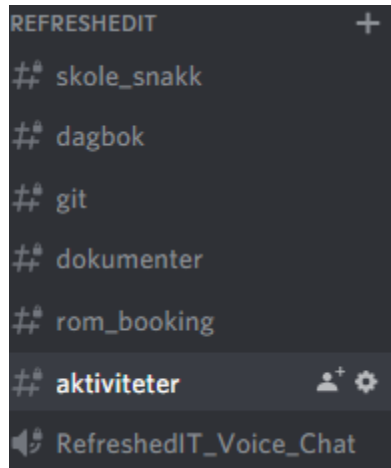
Figur 6 Oversikt over ulike tilstander

3.1.5 Kommunikasjon

Ettersom at kommunikasjon er et viktig element for god fremgang i prosjektet så har gruppen valgt å bruke god tid på planlegging av hvordan gruppen skal kommunisere med hverandre, og hvordan gruppen skal kommunisere med bedriften.

Den interne kommunikasjonen blant gruppemedlemmene foregår hovedsakelig på Discord som er en kommunikasjonsplattform. Discord er noe som ble mer fokusert på under SARS-CoV-2 Pandemien. Discord ble brukt hyppigere til faglige formål, samt har gruppen blitt bare bedre til å beherske disse applikasjonene som er grunnlaget til hvorfor vi har valgt å bruke Discord.

Gruppen har også satt opp «Kanaler» i discord som brukes til faglig formål, som for eksempel «Skole_snakk» kanalen som er gruppens hovedkanal, vi har også en «GIT»-kanal der man får regelmessige meldinger av en «robot» når det skjer endringer i koden vår og når det blir pushet opp.



Figur 7 Oversikt over Discord kanaler

Den eksterne kommunikasjonen mellom gruppen og veilederne skjer hovedsakelig på Teams. Inne på Teams så har vi opprettet en egen gruppe med våre to veiledere og en gruppe med salgslederen i salgsvdelingen. Når det skal gjennomføres Sprint review møter, eller generelt andre møter så gjennomføres de på Teams ettersom at man kan «video-chatte» med andre på plattformen.

3.2 Systemkrav

Systemkravene i dette prosjektet er utarbeidet i samarbeid med Salgsavdelingen og HR avdelingen, som hver for seg har behov for informasjon om sine konsulenter. Informasjonen som avdelingene ønsker, blir notert og beskrevet i tabell 1. I tabellen er de ulike kravene markert med en spesifikk prioritering som tilsier hvor viktig den er for Egde. Alle kriterier markert med “Must Have” er kriterier som Egde ønsker å ha på plass. Kriterier merket med “Should Have” og “Could Have” (MoSCoW) er kriterier som vi burde ha, eller kan ha, men som ikke er avgjørende. Disse kriteriene er inspirert av den populære MoSCoW prioriterings teknikken (ProductPlan, 2021).

3.2.1 Kartlegging

I samtaler med respektive ansatte i Egde, har vi gått gjennom ønsker og krav den gitte avdelingen har, hvilke aspekter av prosjektet som er mest relevant og kartlagt andre behov den avdelingen trenger for å kunne optimalisere prosessene innad bedriften. I tillegg så har vi ut gjennom prosjektet fått muligheten for jevn oppfølging av avdelingene slik at vi alltid var på riktig spor i henhold til krav og ønsker.

Ved starten av dette semesteret så skulle gruppen kun hovedsakelig samarbeide med veilederne. Ved et slikt samarbeid skulle studentene utføre en tradisjonell ETL prosess ved bruk av moderne teknologi som Databricks og Microsoft Azure. Underveis prosjektet så fikk gruppen nye samarbeids partnere. Dette var salgslederen og HR-ansvarlig i Egde.

Grunnen til at gruppen fikk nye samarbeidspartnere var fordi både salgsavdelingen og HR-avdelingen ønsket at gruppen skulle implementere funksjoner og verdier i ETL-prosessen. For eksempel så påpekte salgsleder at vi skulle se på hvordan vi kan koble opp det vi jobber med til regnskaps systemet i bedriften. Dette innebar å hente ut verdifulle tall og visualisere dem. Et slikt samarbeid er god læring for oss, der vi får prøvd å samarbeide med flere partnere i Egde også får gruppen et kjennskap til viktige personer i bedriften noe vi ser på som verdifullt. Vi blir også utfordret av Egde, noe som er veldig positivt ettersom at til gjengjeld får gruppen god og verdifull kompetanse.

3.2.3 Prosjektstyrings krav

Prosjektstyring er et av de viktigste aspektene av et prosjekt. Prosjektet i seg selv er avhengig av gode prosjektstyrings grunnlag for at det skal lykkes. Gruppen har fått en innføring i de ulike systemene som blir brukt internt av Egde, dette er noe gruppen har tatt med seg videre. I tillegg til en innføring av systemene som blir brukt, har gruppen også fått en rekke tips i forbindelse med oppgavefordeling og prosjekt strukturering.

For strukturering og utdeling av oppgaver, samt tidsestimering og kvalitetssikring har gruppen valgt å bruke Jira. Før oppgavene blir ferdigstilt har gruppen en testing seksjon hvor oppgaven og utfallet gjennomgås og vurderes.

3.2.4 Teknologiske krav

Egde er en voksende bedrift og i konstant utvikling og endring. Gruppen fikk i begynnelsen av prosjektet en innføring i smidig utvikling hvor vi fikk tydelig inntrykk at Egde verdsetter skalerbare løsninger. I forbindelse med skalerbare løsninger ønsker Egde at systemet skal kunne skalere ved at nye datakilder dukker opp, nye behov og dynamisk løsninger på å holde data oppdatert.

Hovedsakelig skal vi forholde oss til Sharepoint, Azure Datalake, Azure Databricks og bruke Logic App til transport av data fra de ulike lagene. Datakildene er foreløpig Allokeringssark og CV Portalen, men det er lagt vekt på at det skal være mulig å kunne flette inn flere datakilder i etterkant. Derfor er det viktig at vi som gruppe passer på å holde henting av kilder så dynamisk som mulig.

Egde ønsker i slutten av bachelor prosjektet et system hvor de kan hente ut informasjon om hvilke konsulenter som er ledige, hvilke konsulenter som er allokert og hvor mange prosent de er allokert. I tillegg ønsker de å vite om en konsulent har blitt tilbudt til en bedrift eller ikke. I tabell 1 følger kravene vi har kartlagt som nødvendige fra Egde sin side.

Systemkrav	Prioritering	Beskrivelse
Systemet må være enkelt å bruke. Det er ikke behov for komplisering uten behov	Must Have	I og med at det er flere ulike mennesker som skal bruke systemet, hovedsakelig fra Salg og HR avdelinger. Er det viktig at systemet er lett å benytte seg av og har et intuitivt design som Egde er kjent med fra før.
Det skal være mulig å hente ut data som viser informasjon om en konsulent er solgt eller ikke.	Must Have	Salgs avdelinger trenger å få en oversikt over konsulenter som er solgt til bedrifter. Dette er for å ikke "Selge" en konsulent flere ganger og skape problemer.
Det skal være mulig å hente ut informasjon om hvor mye	Must Have	I samtale med salgs avdelingen fikk gruppen inntrykk av at salgs avdelinger er avhengig av å vite når en konsulent er ledig

en konsulent er solgt til en bedrift		for nytt prosjekt så tidlig som overhodet mulig.
Det skal være mulig å sortere etter spesifikke utviklere/prosjektledere basert på erfaring	Should Have	Per dags dato har HR og Salgsavdeling en personlig tilknytting til de ansatte. Det vil si at de gjerne utenat vet hva vedkommende, spesialiserer i og hvor lenge vedkommende har vært i gjeldende fagområde. Dette er ikke gunstig i lengden.
Det skal være mulig å sortere etter spesifikke fagområder	Should Have	Egde har mange ulike fagområdet i verktøykassen sin, til tross for at per dags dato er disse fagområdene personlig gjenkjent av ledelsen, vil det være til fordel for Egde å få en måte å sortere/hente inn spesifikke fagområdet basert på kundenes behov.
Det skal være mulig å se når en konsulent skal velge ut en fadder	Could Have	HR-Avdelingen ønsker en oversikt over når en konsulent skal velge en fadder, eller en veileder av slag, etter et visst intervall etter ansettelse

Tabell 1 Systemkrav

Disse systemkravene har gruppen fått gjennom jevnlig kommunikasjon med produkteier. Vi har også fått mulighet til å tolke og vurdere løsninger selv basert på informasjonene vi har fått. Men det har vært hjelpsomt for gruppen å få tildelt kontaktpersoner, og å snakke med vedrørende prosjektet på behovs basis.

3.2.5 Kvalitets krav

Under prosjektet har vi mange oppgaver å forholde oss til. Vi får oppgaver både fra UiA og fra Egde. Dette vil i sin egen essens være utfordrende, men det er også en utfordring i seg selv å holde oversikt over alle de ulike oppgavene vi får tildelt. Vi har fått en rekke ønsker fra Egde om hva de har behov for i systemet sitt, det er da vår jobb å sørge for at dette blir levert.

På bakgrunn av samtaler med ansatte i Egde har vi dannet følgende krav:

- Skalerbart
 - Egde er enda i utvikling og vokser stadig. I en samtale med produkt eier fikk vi vite at vi burde ta hensyn til fremtidsrettede valg. Mer spesifikt ble vi opplyst om at det er stor sannsynlighet at det dukker opp nye data kilder i fremtiden, og at det hadde vært ønskelig om disse data kildene enkelt kan tilføyes det eksisterende systemet.
- Dynamisk
 - Hard kode er ikke en god, langsiktig løsning for dette prosjektet, spesielt ikke hvis skalerbarhet er et kvalitetskrav. Derfor må vi finne en løsning for at de ulike kildene blir flettet inn dynamisk i koden.
- Konsistent
 - Egde ønsker et system som skal være enkelt å bruke, dette kan vi oppnå med å være konsistente med valg når vi lager og designer systemet. Valgene vi tar må gå gjennom hele systemet, og må forholde seg i samme flyt. Dette er uklart på hvordan vi skal klare, men med åpen dialog med produkt eier vil ikke det være et stort problem.
- Vedlikehold
 - Dette systemet skal vare. Derfor er det viktig at vi lager systemet med vedlikehold i tankene. Vi må passe på at det ikke er store endringer som må gjøres etter kort tid. Men det er uansett forventet at noen endringer vil måtte forekomme ettersom tiden går. Vedlikehold punktet henger godt sammen med Skalerbarhets punktet. Det skal være enkelt å legge til og fjerne aspekter av systemet etter behov.
- Automatisert
 - Systemet skal forholdsvis være automatisert til en viss grad. Informasjonen om for eksempel, utviklere skal hentes inn regelmessig. Dersom det oppstår en endring i en fil fra Sharepoint, skal dette videreføres inn i ETL systemet.

3.3 Kvalitetssikring

Kvalitetssikring er å oppfylle de kravene som har blitt satt til kvalitet (Leif Halbo, 2020). Siden prosjektet vårt omhandler en ETL prosess, der vi jobber med sensitiv data, er det derfor avgjørende at vi er strengere og flinkere på å sikre kvalitet.

3.3.1 Definerer av kvalitet

For å kunne sikre kvalitet er det viktig å finne ut hva kvalitet er i vårt prosjekt, og hvordan man skal oppfylle dette. Men hva er egentlig kvalitet i dette prosjektet? Det vi ønsker å levere til Egde er et godt produkt som henter data, transformerer den, og laster den opp til en Delta Lakehouse. Hvis denne prosessen har noen avvik eller feil, vil ikke systemet fungere ettersom alle ledd henger sammen. Det er derfor avgjørende for oss å ta i bruk de riktige verktøyene og bruke ressursene på en effektiv, men også en sikker måte.

Det er ikke kun i selve systemet at kvalitet oppstår, det oppstår også i kommunikasjonen vår med Egde. Vi er nødt til å ha en tett dialog med Egde, der vi oppdaterer de på ting som er gjort og forsikrer oss om at det vi jobber på oppfyller deres ønsker. Dette gjennomføres ved å ha Sprint retrospektiv i slutten av hver sprint, samt å være aktive gjennom en Teams-chat dersom mindre spørsmål eller problemer dukker opp.

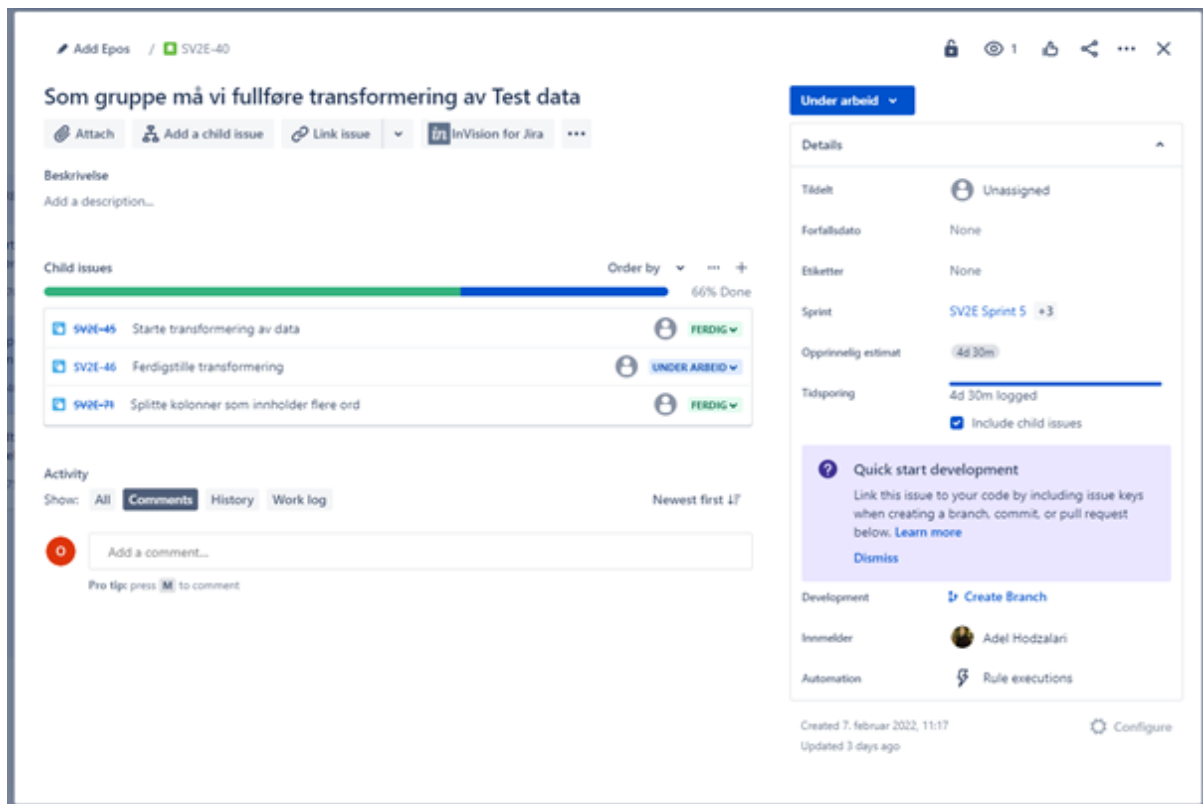
Vi ønsker også å sikre kvalitet i vår egen læring gjennom dette prosjektet. Vi dokumenterer derfor alt vi gjør, enten i form av rapport-skriving, dagbok, eller dokumentasjon av kode. God dokumentasjon, spesielt dokumentasjon rundt ETL prosessen, vil ikke bare hjelpe oss med læring, men den vil være til stor hjelp for Egde når de skal ta i bruk dette prosjektet.

3.3.2 Teknologiske valg for å sikre kvalitet (Versjons kontroll, etc.)

Som nevnt tidligere så vil bruken av verktøy og ressurser hjelpe oss med kvalitetssikring. Det finnes flere ulike verktøy som kan hjelpe til, men det viktige er å velge riktig. I vårt tilfelle har vi valgt å bruke Jira og Scrum for dokumentasjon og prosjektstyring, og GitHub, Azure og Databricks for versjonskontroll.

«Jira er et oppgavehåndteringssystem som hjelper deg med å holde styr på oppgaver i et prosjekt eller i et team. I utgangspunktet er programmet laget for å støtte programvareutvikling, men har også blitt tatt i bruk i flere andre disipliner innen alle typer bransjer.» (Jira - hva er det egentlig? | Artikkel | Computas, 2020). Vi bruker dette verktøyet til å loggføre timer, oppgavefordeling, prioritering av oppgaver og ikke minst kvalitetssikring.

Gjøremålene blir delt opp i ulike brukerhistorier som er tilpasset gjøremålet, og brukerhistoriene inneholder mindre oppgaver om gjøremålet.

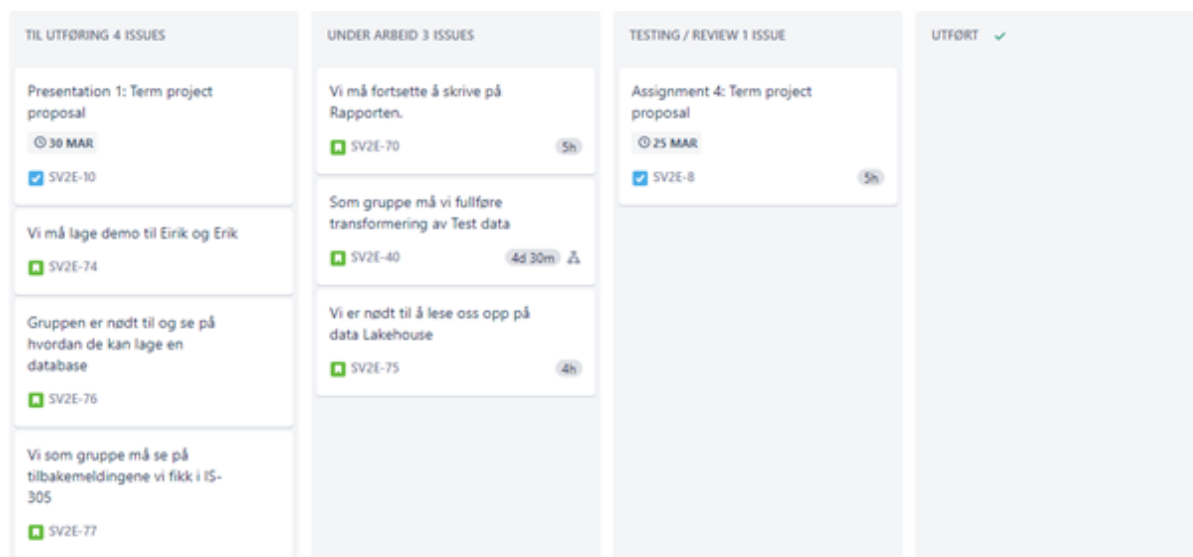


Figur 8 Brukerhistorie i Jira

Brukerhistoriene blir fordelt på Jira (Se figur 8 og 9) og kan befinne seg i fire typer “boards”, eller tilstander. Disse fire er:

- Oppgaver som skal utføres
- Oppgaver under arbeid
- Oppgaver i test/review
- Oppgaver som er utført

Ved å bruke 4 tilstander sikrer vi oss at hver brukerhistorie og de tilhørende oppgavene som er fullført blir testet og oppfyller det vi og Egde ser på som kvalitet.



Figur 9 Kvalitetssikring

For versjonskontroll og å sikre kvalitet i koden vår, bruker vi Github. Githuben er direkte knyttet opp mot databricks, dette betyr at hvis en endring blir gjort på et “sheet” i databricks så kan denne lagres og automatisk bli pushet opp i githuben vår. Måten vi har strukturert “repositorien” er å dele opp i ulike “branches” der hver “branch” representerer en funksjon, som for eksempel data “cleaning”. Inne i disse “branchene” har vi delt opp i 2 mapper, “shared” og “users”. Her vil “users” mappen fungere som vår “dev” mappe, altså inneholde det vi jobber på per dags dato. Når det vi jobber på er fullført vil dette bli flyttet til mappen vår i “shared”.

Grunnen til at kvalitetssikring er så viktig for oss er fordi dataen vi jobber med er sensitiv. Vi jobber med personlig data for ansatte hos Egde som ikke kan bli lastet ned eller vist på noen som helst måte. Det er derfor disse tiltakene for å sikre dataen, men også kvalitet er nødvendig.

4. Prosjektgjennomføring

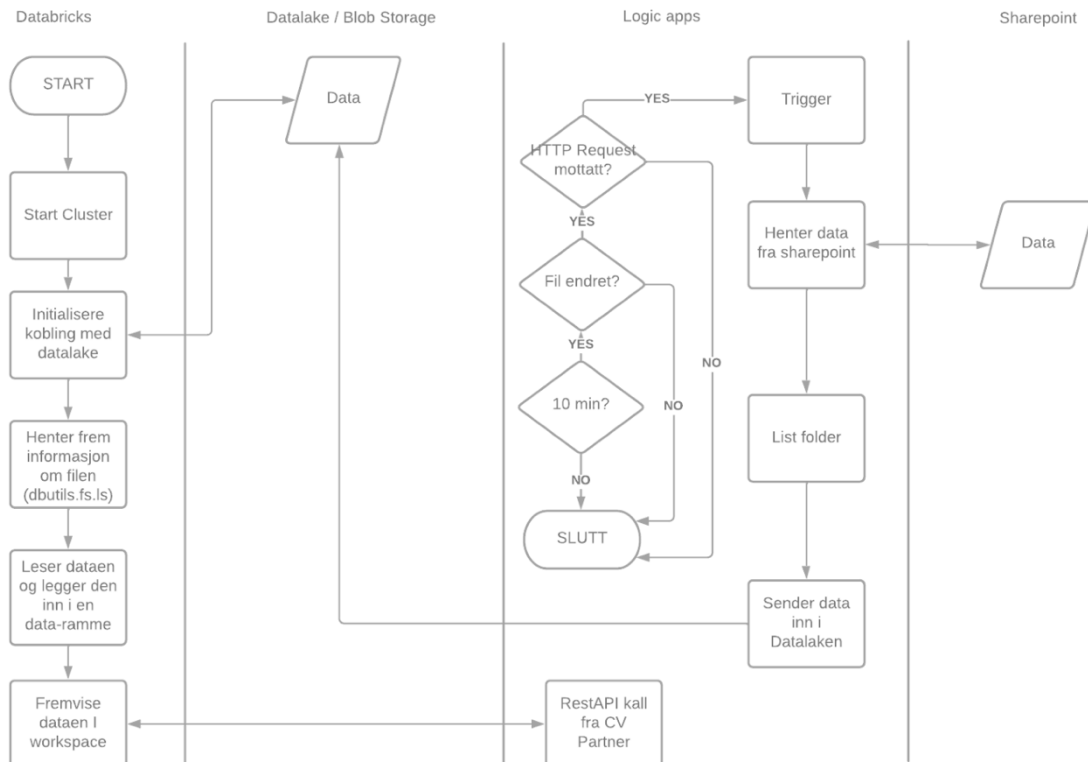
En ETL Prosess består gjerne av tre hoved deler, det er, Extract, Transfrom og Load. Ved gjennomføring av en slik prosess er det viktig at man har riktig verktøy og den riktige kunnskapen. Det finnes flere verktøy man kan bruke ved gjennomføring av en slik prosess, derfor skal man alltid prøve og definere hvilken teknologi som skal brukes og eventuelt hvilken plattform teknologien skal utføres i.

Før det siste semesteret starten så var vi i dialog med veileder der vi ble introdusert til de forskjellige teknologiene vi skulle ta i bruk av. Det gruppen gjorde før semester start var å lese seg opp på de forskjellige teknologiene presentert. Dette ga oss et innblikk på hva gruppen skulle jobbe med og gjorde planleggingen enklere.

Ved semester start så begynte gruppen på fulltid og jobbe med prosjektet. Da var det slikt av vi brukte Databricks som en plattform for å utføre «Transfrom», Sharepoint og datalake var gruppens «Extract» point og til slutt så var en relasjons database eller Data lakehouse gruppens «Load» point.

4.1 Uthenting av data

Hvordan uthenting av data fungerer teoretisk kan variere fra prosjekt til prosjekt. I vårt prosjekt var den opprinnelige planen å hente ut allokerings dataen direkte fra SharePoint, for så å videreføre dette til den interne datalaken. Deretter skulle denne dataen blitt hentet inn i Databricks, som i vårt tilfelle skal fungere som et mellomledd for håndtering av data. Vi startet derfor å se på mulighetene ved å hente ut dataen fra SharePoint og inn i datalake. Vi forstod tidlig at “logic apps” var riktig metode å bruke, og begynte derfor å undersøke hva som måtte til for å kunne gjennomføre dette. Videre så undersøkte vi også hva som måtte til for å overføre dataen fra datalaken vår til Databricks. I tillegg til allokeringsdata så hadde vi data fra CV partner også, denne dataen skulle bli levert direkte inn i datalaken vår i CSV-format. Utfra disse faktorene dannet vi en overordnet oversikt som vises på figur 10.

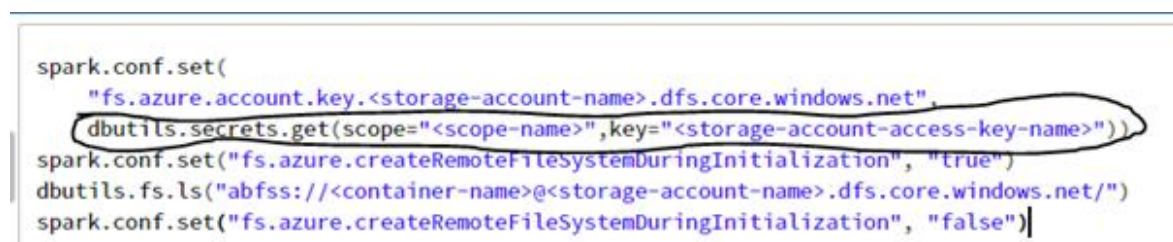


Figur 10 Uthenting av data

Gruppen forsøkte å gjennomføre planen vist i figur 10 og startet med å se på “Logic apps”. Her møtte vi på et problem som gjorde at vi ikke fikk tilgang til “Logic apps” funksjonen. Dette skyldes at i vårt samarbeid med Egde, så har vi fått tilgang på gjestebbrukere i Azure. Dette er brukere som har begrensninger på hvilke funksjoner og oppgaver som kan utføres. En av begrensningene er at disse brukerne ikke har Azure Active Directory (Azure AD), noe som Azure bruker for å verifisere brukere og gir tilganger på ulike funksjoner innad i systemet. Vi var derfor nødt til å se på løsninger som ikke krevde Azure AD, etter mer research fant vi en løsning som tok i bruk Microsoft Flow, og som i motsetning til «logic apps» brukte Acces Key i datalake som verifisering. Her møtte vi også på problemet med rettigheter og hadde ikke tilgang til å bruke nøkkelen. Vi videreformidlet problemene til våre kontaktpersoner i Egde som ikke fikk løst problemet, og vi ble dermed bedt om å sette dette på vent og dataen ble manuelt plassert i datalaken.

Til tross for å ha møtt på et problem ved overføringen fikk vi bekreftet at metodene vi fremviste fungerte, men på grunn av manglende rettigheter så var det ikke mulig å bruke de.

Videre i prosjektet jobbet vi med å hente ut data fra datalake og sende dette inn i Databricks, hvor dataen som ble hentet skulle bli transformert og videreført. Måten datalake og Databricks blir koblet sammen slik at filer enkelt kan hentes ut, er ved bruk av «mounts» og «secret-scope». Du lager først et «secret-scope» som er en kolleksjon av nøkkelverdier definert av et navn. Nøkkelverdiene som finnes i dette scopet er som regel «Acces Keys» og har en unik verdi, dette kan være nøkler til flere datalakes (mssaperla, 2022). I figur 11 ser du koden vi kjørte for oppkoblingen. For at en slik oppkobling skal fungere er det krav om å ha et eksisterende scope, med den nødvendige nøkkelverdien.



```
spark.conf.set(
    "fs.azure.account.key.<storage-account-name>.dfs.core.windows.net",
    dbutils.secrets.get(scope="<scope-name>", key="<storage-account-access-key-name>"))
spark.conf.set("fs.azure.createRemoteFilesystemDuringInitialization", "true")
dbutils.fs.ls("abfss://<container-name>@<storage-account-name>.dfs.core.windows.net/")
spark.conf.set("fs.azure.createRemoteFilesystemDuringInitialization", "false")
```

Figur 11 Skjerm bilde av oppkobling av datalake

De to stegene av uthenting som blir beskrevet var de mest utfordrende stegene, men de var viktig å få på plass tidlig under prosjektet, det neste var å transformere dataen. Ettersom at alle oppkoblingene var på plass og vi fikk hentet test filene fra vår test container i CSV-format, ble vi tilsendt første versjon av Allokeringssarket til Egde. Dette arket kom i xlsx-format noe som skapte småproblemer for oss. Det å hente xls- og xlsx- filer viste seg å være et problem ettersom at databricks ikke støttet dette. Vi var nødt til å laste ned et eksternt bibliotek “crealytics/spark-excel” for å kunne hente ut og lese slike filer. Dette var noe vi ikke hadde tilgang til å gjøre så kontaktpersonene våre var nødt til å fikse dette. Da biblioteket var på plass, støttet vi på enda et problem. Problemet nå var at vi ikke hadde tilgang til å utføre denne konfigurasjonen på grunn av manglende rettigheter vist i figur 12. Vi fant aldri ut av hva problemet var, men tror det var en oppkoblingsfeil i den nye containeren i datalaken. Vi fikk istedenfor Allokeringssark-filene sendt som CSV-filer fremfor xlsx og begynte nå å transformere.

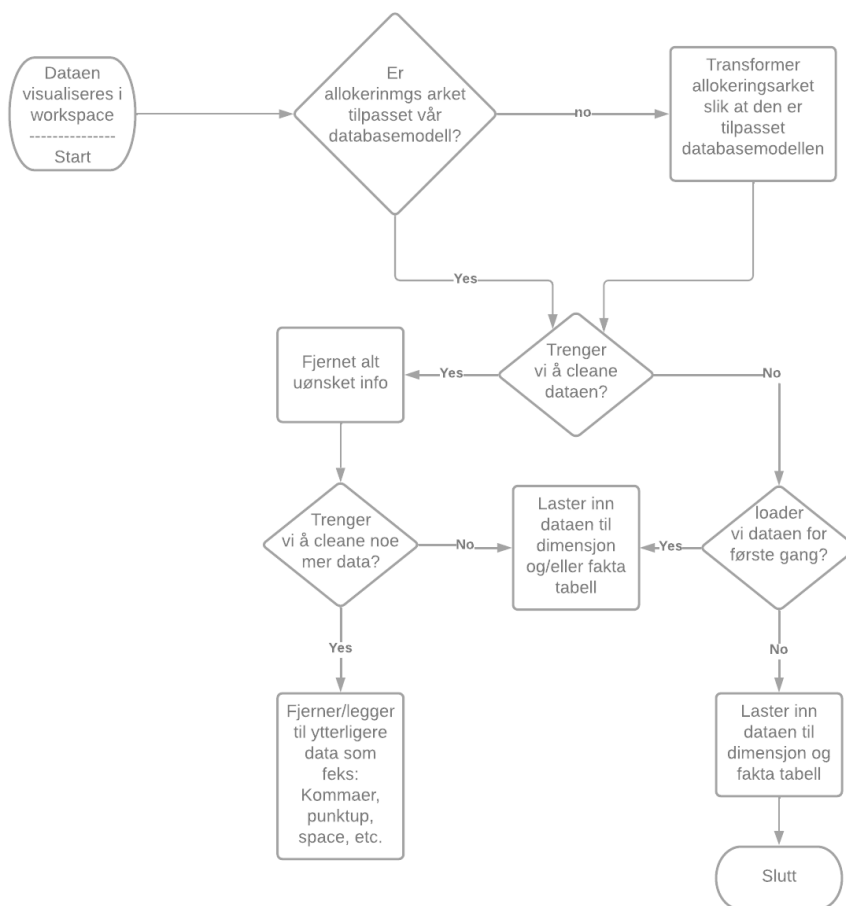


```
KeyProviderException: Failure to initialize configuration
Caused by: InvalidConfigurationException: Invalid configuration value detected for fs.azure.account.key
```

Figur 12 Mangel på rettigheter

4.2 Transformering og utsending av data

Da vi startet med transformeringen, tok vi utgangspunkt i de systemkravene vi hadde skaffet fra Egde. Vi ønsket å transformere Allokeringsskemaet slik det ble mer forståelig og inneholdt kun den nødvendige dataen. Vi startet med å lage 4 ulike tabeller der to var fakta tabeller og to var dimensjonstabeller og et flowchart vist i figur 13.



Figur 13 Transformering av data

Disse tabellene skulle visualisere hva slags data som skulle tas i bruk og hvor dataen skulle lagres. I starten fikk vi beskjed fra Egde om at dataen skulle lagres i en database, så vi jobbet dermed mot dette. Under transformeringen brukte gruppen hovedsakelig Python og Structured Query Language (SQL). Da vi fikk Allokeringsskemaet var dataen allerede strukturert, så det vi startet med å gjøre var å «rengjøre» kolonnene for å gjøre de uniforme.

Det neste steget var å splitte kolonnen «NAVN» opp i «FORNAVN» og «ETTERNAVN». I denne prosessen oppsto det et lite problem der den ene kolonnen «LONG_DESCRIPTION» ble sammenflettet med «NAVN». Løsningen på dette var å slette «LONG_DESCRIPTION». Videre i prosessen byttet vi ut nullverdier med logiske verdier, som vist på figur 14. Grunnen til dette var for å gjøre det enklere for Egde å visualisere og jobbe med dataen senere.

```
#Bytter de forskjellige NULL verdiene til en logisk verdi
df13 = df12.fillna({'NATIONALITY':'Ikke satt',
                    'UPN':'Ingen Epost',
                    'EXTERNAL_USER_ID':'Ikke satt',
                    'EXTERNAL_UNIQUE_ID':'Ikke satt',
                    'UPDATED':'Ikke oppdatert',
                    'UPDATED_BY_OWNER':'Ikke oppdatert',
                    'CUSTOM_TAG__IMPORTSTATUS':'Ikke OK',
                    'CUSTOM_TAG__FAGOMRÅDE':'Ingen fagområde'})

#Viser frem datasetet
display(df13)
```

Figur 14 Fyller nullverdier

Det neste steget i prosessen var å velge ut de kolonnene med data som vi mener er nødvendig å ha med og sende dataen til sine respektive tabeller. Disse tabellene blir laget i Databricks og vil kun fungere som en visualisering av databasen. Dette er den generelle prosessen vi gjennomgår både med dataen vi får fra Allokeringssarket og med dataen vi får fra CV Partner. I vårt tilfelle blir data fra allokeringssarket fordelt og sendt til de to tabellene «allokeringer» og «muligheter». Figur 15 viser en oversikt over Allokeringstabellen, altså hvor konsulenter er allokert, hvor mye de er allokert og start og slutt dato.

	ETTERNAVN ▲	FORNAVN ▲	TILLEGGSROLLE ▲	KONSULENT_KATEGORI ▲	FAGGRUPPE ▲	FAGOMRÅDE_PRIMÆR ▲
1	Gjeruldsen	Eirik	Ingen Rolle	Ingen Kategori	Ingen Faggruppe	Ikke Definert
2	Gjeruldsen	Eirik	Ingen Rolle	Ingen Kategori	Ingen Faggruppe	Ikke Definert
3	Gjeruldsen	Eirik	Ingen Rolle	Ingen Kategori	Ingen Faggruppe	Ikke Definert
4	Gjeruldsen	Eirik	Ingen Rolle	Ingen Kategori	Ingen Faggruppe	Ikke Definert
5	Gjeruldsen	Eirik	Ingen Rolle	Ingen Kategori	Ingen Faggruppe	Ikke Definert
6	Gjeruldsen	Eirik	Ingen Rolle	Ingen Kategori	Ingen Faggruppe	Ikke Definert
7	Gjeruldsen	Eirik	Ingen Rolle	Ingen Kategori	Ingen Faggruppe	Ikke Definert

Figur 15 Allokeringstabell

Tabellen «muligheter» inneholder sensitiv data på hvilke salgsmuligheter Egde har, sannsynlighetsregning og prioritet og kan dermed ikke vises.

Dataen vi får fra CV Partner blir fordelt på de to tabellene «erfaringer» og «konsulenter». Disse to tabellene inneholder personlig data knyttet opp til konsulentene vist i figur 16 og figur 17.

	ETTERNAVN ▲	FORNAVN ▲	EMAIL ▲	DEPARTMENT ▲	SKILL_NAME_NO_ ▲	YEAR_EXPERIENCE ▲
1	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	Python	2
2	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	Utvikling i Python	1
3	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	Smidig utvikling med SCRUM	1
4	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	C#.NET	0
5	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	SSIS	0
6	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	XGBoost	2
7	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	Keras	3
8	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	Tensorflow	2
9	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	Jupyter	2
10	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	Object detection	2

Figur 16 Erfaring tabellen

	ETTERNAVN ▲	FORNAVN ▲	EMAIL ▲	DEPARTMENT ▲	UPDATED ▲	UPDATED_BY_OWNER ▲	MONTH_FROM ▲	YEAR_FROM ▲	MONTH_TO ▲
1	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	2020-08-03 10:56:50 UTC	Ikke oppdatert	6	2019	8
2	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	2020-08-03 10:56:19 UTC	Ikke oppdatert	9	2019	4
3	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	2020-08-12 06:39:18 UTC	Ikke oppdatert	8	2020	NaN
4	Gjeruldsen	Eirik	eirik.gjeruldsen@egdeconsulting.no	Grimstad	2022-01-27 14:26:09 UTC	2022-01-27 14:26:10 UTC	1	2019	5

Figur 17 Konsulenter tabellen

Da vi var ferdige med selve transformeringer gjensto kun den siste delen av oppgaven, utsendingen av den transformerte dataen. Som beskrevet ovenfor trodde vi at dataen skulle sendes inn til en database og en SQL-server, men underveis i prosessen med å finne en løsning til SQL server ble vi informert om å undersøke konseptet Delta Lake. Dette ble da vår nye vinkel og løsning på å sende ut data. Vi satte oss ned sammen med våre veiledere i Egde og lagde en ny Datalake med de rettighetene vi trengte. Da denne var satt opp jobbet vi med å koble den nye datalaken til databricks på samme måte som forklart tidligere. Etter det jobbet vi med å sende de ferdige tabellene inn i den nye datalaken i Delta format. Dette visste seg å være litt krevende og vi brukte mye tid på å forstå selve konseptet med Delta Lake og hvordan filene kunne bli sendt i Delta format. Filene ble sendt og figur 18 og figur 19 viser en oversikt over den nye datalaken.

Name	
☐	📁 [..]
☐	📁 Allokeringer
☐	📁 Erfaringer
☐	📁 Konsulenter
☐	📁 Muligheter

Figur 18 Fil Struktur i Datalake

Name	
☐	📁 [..]
☐	📁 _delta_log
☐	📄 part-00000-906bfa1-6f6f-48a0-b6f4-717b7fb335bd-c000.snappy.parquet
☐	📄 part-00001-58b19690-1c56-486a-bc84-225dfde9af6f-c000.snappy.parquet
☐	📄 part-00002-c5aec03a-53a1-4220-b266-bafabc5f90dd-c000.snappy.parquet
☐	📄 part-00003-f1a9852a-c31e-46e6-ae3c-6fd17dfee8e6-c000.snappy.parquet

Figur 19 Fil Struktur i Datalake

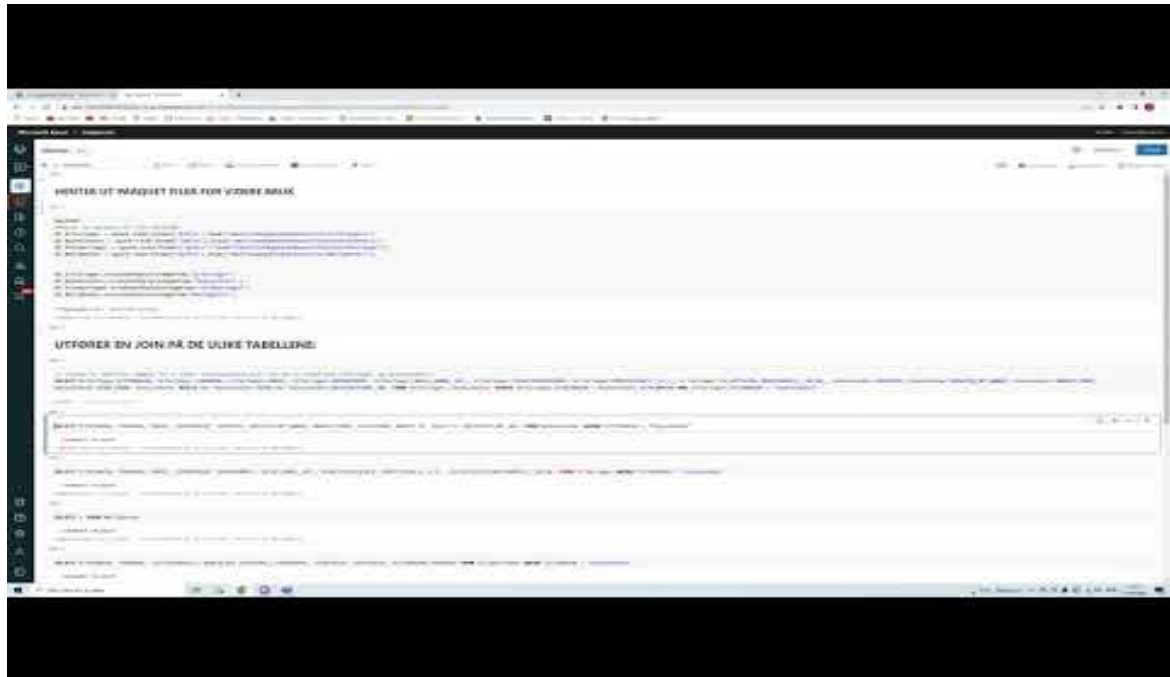
Det neste Egde ønsket å gjøre var å kunne visualisere den dataen vi har jobbet med. De ønsket å kunne enkelt gå inn og se og bruke dataen, så vi startet å lese oss opp på Power BI. Vi fant en løsning som ville fungere, men igjen så manglet vi tilgang, fordi det var krav på Azure AD verifisering. Vi viste løsningen og dokumentasjonen for å gjennomføre en slik oppgave til Egde, så dette er noe de kan ta i bruk senere. Det vi valgte å gjøre istedenfor var å hente disse “parquet” filene tilbake i Databricks og kjøre SQL queries for å fremvise dataen. Ved å bruke queries kan man sortere etter det man ønsker, enten det er navn, allokerings prosent, egenskaper om konsulenten osv. Under vises eksempler på queries som kjøres:

- SELECT** Erfaringer.CV_PARTNER_USER_ID,
 Erfaringer.ETTERNAVN,
 Erfaringer.FORNAVN,
 Erfaringer.EMAIL,
 Erfaringer.DEPARTMENT,
 Erfaringer.SKILL_NAME__NO_,
 Erfaringer.YEAR_EXPERIENCE,
 Erfaringer.PROFICIENCY__0_5_,
 Erfaringer.IS_OFFICIAL_MASTERDATA__IN_NO_,
 Konsulenter.UPDATED,
 Konsulenter.UPDATED_BY_OWNER,
 Konsulenter.MONTH_FROM,
 Konsulenter.YEAR_FROM,
 Konsulenter.MONTH_TO,
 Konsulenter.YEAR_TO,
 Konsulenter.EMPLOYER__NO_,
 Konsulenter.DESCRPTION__NO_
 FROM Erfaringer, Konsulenter **WHERE**
 Erfaringer.CV_PARTNER_USER_ID = Konsulenter.CV_PARTNER_USER_ID **AND**
 Erfaringer.ETTERNAVN = "Gjeruldsen"
- SELECT** * **FROM** Konsulenter **WHERE** ETTERNAVN = "Gjeruldsen"
- SELECT** * **FROM** Erfaringer **WHERE** ETTERNAVN = "Gjeruldsen"
- SELECT** * **FROM** Muligheter
- SELECT** * **FROM** Allokeringer **WHERE** ETTERNAVN = "Gjeruldsen"

5. Produktet

For å vise produktet på en god måte, har vi valgt å lage en kort demo-video hvor vi viser og forklarer hva de ulike funksjonene gjør som vi har laget. Det er verdt å nevne at vi bare viser de ferdige tabellene da vi ikke har mulighet til å vise transformeringen da den seksjonen av koden inneholder sensitiv data i form av nøkler og personal info.

Link til Demo her: [RefreshedIT - Bachelor Demo](#)



6. Utfordringer

Utfordringer er uforutsigbare, og gruppen kan ikke vite alle utfordringer som de kommer til å støtte på. Dette betyr at gruppen må ha gode rammer for håndtering av fremtidige utfordringer. Under dette prosjektet bruker vi Weekly Scrum der gruppen møtes "ukentlig". Under et slikt møte tar vi opp temaer som, hva skal vi gjøre, hva har vi gjort og hva kan vi gjøre bedre. Det dedikeres også ti minutter av Weekly scrum møtet der vi ser på utfordringer gruppen har møtt på. Dette er en fremgangsmåte på hvordan uforutsigbare utfordringer vil bli anerkjent og løst.

6.1 Risiko håndtering

Under prosjektet så har gruppen opplevd mange diverse utfordringer. Derfor er det viktig at gruppen har en oversikt over mulige risikoer, og en plan for å håndtere disse.

Ved risiko håndtering så har gruppen bestemt seg for å ta i bruk av risiko matrisen som er presentert av vår professor. Matrisen er i Excel med satte formler & regler der den fargelegger & regner ut risikoer.

En matrise vil hjelpe gruppen i å visualisere “risikoene” som forekommer under prosjektet. I figur 20 så er det slikt at det er en utregning i Excel arket som regner ut den totale “risikoen” i prosjektet. Dette gjør programmet ved og gange “konsekvens” med “sannsynlighet”. Om den totale poengsummen overskrider “10” poeng så blir risikoen ansett som en “kritisk” risiko. Det er også slikt at gruppen har et “tiltak for at ikke det skal skje” & “tiltak på redusering av skade” som betyr at gruppen kan diskutere sammen og legge inn forskjellige tiltak som vil hjelpe med og behandle risikoen som har oppstått. Vi vurderer nye risikoer hver retrospektiv og behandler gamle risikoer samtidig.

Risiko Vurdering			
	1 til 5	1 til 5	
Risiko	Konsekvens	Sannsynlighet	RISIKO
1 Programmet er for vanskelig og bruke	2	2	4
2 Vi rekker ikke og gjennomføre prosjektet pga mangel på gruppe-medlemmer	5	2	10
3 Gruppen begynner og miste motivasjon	5	2	10
4 Sikkerhets brudd I datalaken	5	1	5
5 Sykdom balnt gruppe medlemmene	2	5	10
6 over implementere eller under implementere	3	1	3
			0

Figur 20 Risiko Vurdering

	Tiltak for at det ikke skjer	Tiltak, redusere skade
1	Jobbe tett I tett med kundene	Dokumentere til alltid sånn at programmet blir uoversiktlig
2	ikke miste gruppemedlemmer	God kommunikasjon blant gruppe medlemmene
3	Lytte på hverandre, og gjennomføre tilatk som hever motivasjonen	Endre på fremgangs måte, gjøre kanskje noen gøye aktiviteter når det jobbes med skole
4	Privatisere access keys, alltid skjule dem/holde dem privat	Cold backup av datalaken, evt extracte dataen fra gammel data lake og sende det over til ny
5	Ta godt vare på seg selv	Ha oversikt over hva personen som er syk skal jobbe med, evt gjøre oppgaven som mangler for personen om det er tidskritisk
6	Alltid holde god kommunikasjon med produkt eier, og ved uklarhet så alltid ta en prat med veileder/produkt eier for å havne tilbake på spor.	Kommunikasjon med produkt eier/veileder

Figur 21 Risiko Håndtering

En slik modell har vært svært behjelpelig for gruppen ettersom at vi alltid har hatt en metode på hvordan vi skal håndtere diverse risikoer. Som nevnt tidligere så har gruppen opplevd mange utfordringer, som for eksempel utfordringer med SARS-CoV-2, Reduksjon i gruppemedlemmer, osv. Da har en slik prosess vært veldig verdifullt for gruppen ved risikohåndtering.

6.2 SARS-CoV-2 utfordringer

Dette semesteret var utfordrende. Den globale pandemien påvirket gruppen både psykisk, men også fysisk. Pandemien har påvirket gruppens overordnet arbeidsmetodikk, arbeidsmoral og engasjement i forelesninger. En utfordring som påvirket gruppen på et høyt nivå, var at vi ikke fikk møtt bedriften vår på starten av semesteret. Ettersom at Egde ikke ønsket og ta noen sjanser, var bedriften stengt ned i starten av 2022.

Gruppen visste ikke hvem som var hvem, og vi fikk ikke fysisk møtt veilederne våre før senere i semesteret. En slik situasjon påvirket gruppens motivasjon, og det å jobbe med prosjektet virket bare som en vanlig skoleoppgave, ikke noe mer ordinert. Dagene var fylt med discord møter og teams møter, der vi kun holdt kontakt med bedriften digitalt.

En annen utfordring med Covid-19 var sykdom blant medlemmer. Det var medlemmer i gruppen som ble smittet med SARS-COV-2(Covid-19) noe som gjorde at arbeidsflyten ble sterkt redusert. I Kristiansand så var det store tilfeller med Covid-19 tidlig på året noe som gjorde at gruppen fikk frykt for og møtes pga. smitte.

6.3 Andre utfordringer

Kanskje den største utfordringen dukket opp etter en måned inn i prosjektet. Vi fikk da vite at to av gruppemedlemmene droppet ut av studiet, på grunn av personlige årsaker. Dette medførte til at gruppen opplevde et større arbeidspress. Selv om oppgaven ble skalert til kun 3 gruppemedlemmer, følte vi fremdeles at arbeidsmengden og innsatsen vi måtte legge ned økte.

Gruppen møtte også på utfordringer under Retrospektiv møtene. Under et Retrospektiv så skal gruppen gjerne være samlet der vi går gjennom hva som ble gjort, hva som ikke ble gjort og hva som burde gjøres under den fullførte Sprinten. Vi går også gjennom hva som var bra, hva vi burde fortsette med og hva vi burde droppe til neste sprint. Det som var utfordrende, var at ved et retrospektiv møte skal man helst være fysisk til stedet med gruppen der hvor man gjennomfører møtet ved bruk av lek eller annen kreativ metodikk. For eksempel så er det sånn at Egde gjennomførte sine Retrospektive ved bruk av VR-briller, lek, tavle, post-it lapper, osv. Dette gjøres for å skape engasjement og stemning blant teamet. Vi kjente veldig tidlig at å gjennomføre et Retrospektiv gjennom «Teams-møte» ikke var like motiverende og «gøy» som og ha det fysisk på Egde lokalene.

7. Refleksjon

I dette kapittelet så skal gruppen presentere refleksjonene som har blitt gjennomført gjennom prosjektet. Det skal også reflekteres over prosjekt gjennomføringen, kvalitet av produkt og utfordringene vi har møtt på gjennom semesteret.

7.1 Prosjekt gjennomføring

I disse seksjonene så skal gruppen presentere våre refleksjoner rundt, prosjekt gjennomføring, den agile metoden kombinert med kanban metodikk, risikohåndtering, estimering, prioritering og kommunikasjon

7.1.1 Scrumban

Under dette semesteret så har gruppen gjennomført en annen vri på arbeidsmetodikken. Ettersom at gruppen har vært vant med å gjennomføre den tradisjonelle Scrum metoden så har vi i år prøvd på en annen fremgangsmåte der vi har kombinert Scrum med Kanban også kalt "Scrumban" (Abraham, 2021).

Gruppen jobbet hardt for å følge de mest sentrale Scrum prinsippene. Scrum består av mange forskjellige prinsipper og det er ikke alltid i alle prosjekter at man har behov for å ta i bruk av alle prinsippene som Scrum tilbyr. Gruppen tok aktiv i bruk av Weekly Scrum, Scrum roller, den agile fremgangsmåten, Retrospektiv, Sprint planning og Product backlogs

De aspektene vi ikke tok i bruk var Daily scrum og Sprint review. Manglene skyldes gjerne at gruppen ikke hadde behov for Daily scrum møter ettersom at vi gjennomførte Weekly Scrum. Når vi holdt Scrum møtene med litt lengre mellom rom var det alltid litt mer å prate om i weekly møtene som vi hadde satt opp gjennom semesteret.

Det gruppen sitter med på slutten av dette prosjektet er at en slik fremgangsmåte er noe som funker veldig bra for oss. Dette er fordi man får fortsatt de "Scrum" fordelene i tillegg til den visualiserings delen som Kanban metodikken tilføyer. Det var heller ikke behov for store endringer ettersom at gjennomføringen av Scrum var det samme, det eneste som var tilføyd av "nytt" var verktøyet Jira som var den delen som integrerte kanban i metodikken vår.

Gruppen håper på at neste gang det skal jobbes med et tilnærmet prosjekt at får vi prøvd ut Scrumban blant litt flere kollegaer og litt mer fysisk til stedet.

7.1.2 Risikohåndtering

Årets semester var litt mer unikt i forhold til tidligere år. Grunnen til dette er håndteringen av risikoer. Fra tidligere semester og prosjekter så hadde ikke gruppen fokusert like mye på risikohåndtering som vi gjorde i år. Grunnen til at vi ikke hadde fokusert på det tidligere var på grunn av manglende behov. Men i år så var det ekstra utfordrende ettersom at gruppen var rammet av COVID-19 og strenge COVID-regler. Under den utfordrende tiden så møtte gruppen på forskjellige risikoer noe som var grunnlaget for det ekstra fokuset på risikoer i år.

Gruppen brukte Risiko matrisen som professoren vår foreslo å bruke, matrisen kan man se på figur 20 og 21. Denne matrisen har vært veldig behjelpelig, ettersom at man får visualisert risikoene som kan oppstå i prosjektet, samt visualiserer vi med farger og poeng sum noe som gir gruppen et mer helhetlig bilde over risikoene som er i matrisen.

En slik fremgangs måte har gitt gruppen innblikk i hvor viktig det er og ta risikoer seriøst ettersom at de kan være ansvarlig for og ødelegge flyten og motivasjon under et prosjekt. Samt er det også viktig å snakke med hverandre om mulige risikoer og om de som allerede har oppstått. Ved å diskutere risikoer så er gruppen bedre egnet til å håndtere disse. Dersom den samme risikoen skjer igjen, og gruppen er klare over det, er det lettere å løse problemet ettersom at man har erfaring i fra tidligere risikoer.

Gruppen er fornøyd med risiko matrisen som har blitt tatt i bruk og konkluderer med at denne matrisen eventuell en lignende matrise skal brukes i fremtidige prosjekter.

7.1.3 Estimering og prioritering

I starten så strevde vi med å forstå hvordan Jira fungerte. Dette medførte til at timer som ble ført inn i Jira i starten av prosjektet, ikke er helt nøyaktig, men siden vi også har estimert i Excel så kan disse timene dobbeltsjekkes slik. Underveis i prosjektet så har det til tider blitt glemt å føre timer, og vi var nødt til å gå tilbake og føre inn manuelt. De gangene det har skjedd har vi heller ikke hatt nøyaktig tidsestimering, men ved å bruke dagbøkene som blir skrevet etter hvert møte får vi en omtrent oversikt over timer brukt.

For prioritering av oppgaver har Jira fungert veldig bra, men igjen så strevde vi med å forstå oss på verktøyet i starten. Oppgaver ble prioritert feil eller ikke i det hele tatt, men dette påvirket ikke selve arbeidet vårt. Selv om vi ikke hadde arbeidsoppgavene og prioriteringen av disse dokumentert, hadde vi full oversikt. Dette oppnådde vi på grunn av kontinuerlige samtaler med veilederne våre både i digitale møter, men også over chat. Da vi hadde forstått Jira og hvordan man prioriterer oppgaver gikk dette automatisk. Oppgaver som dukket opp underveis i prosjektet ble diskutert i gruppen og lagt til i backloggen vår.

7.1.4 Kommunikasjon

Kommunikasjon har vært veldig kritisk for oss i dette prosjektet, og spesielt den eksterne kommunikasjonen. Vi som gruppe har ikke hatt problemer med den interne kommunikasjon innad i gruppen, dette skyldes at vi kjenner hverandre personlig og har kontaktinformasjonen til gruppemedlemmene på flere plattformer. Da kritiske situasjoner har oppstått har vi alltid fått kontakt med hverandre. Den eksterne kommunikasjonen derimot har vært god, men det har vært tilfeller der vi har sittet fast på en oppgave og ikke fått fremgang på grunn av manglende kommunikasjon. I starten av prosjektet var ikke dette et problem, da fikk vi som regel svar innen en time, men utover i prosjektet ble veilederne våre mer opptatte av egne arbeidsoppgaver som medførte til senere svar. Måten vi løste dette på var å samle opp spørsmål og problemer som dukket opp underveis og fremviste disse når veilederne hadde ledig tid.

7.2 Kvalitet av sluttproduktet

Ettersom at vi har jobbet med ukjente teknologier og ukjente fremgangsmåter har det vært ekstra vanskelig å bedømme kvalitet selv. Vi har til tider vært usikre på om det vi har jobbet på, faktisk er riktig fremgangsmåte. For å kunne forsikre oss om at kvaliteten på produktet svarer til Egde sine forventinger har det derfor vært viktig å ha tett dialog. I starten av mai hadde vi en presentasjon for den ene faggruppen hos Egde, der vi presenterte prosjektet vårt og forklarte hvordan vi tilnærmet oss de ulike teknologiene. Noen av disse teknologiene er fortsatt litt ukjente for Egde, spesielt Delta Lake teknologi, derfor er de meget fornøyde med den informasjonen vi leverer. Generelt er Egde fornøyde med arbeidsinnsatsen og sluttproduktet. De har selv sagt at dette er prosjekt som de ønsker å bygge videre på, og det blir kanskje et nytt bachelor-prosjekt til neste år. Uttalelsen til Egde bekrefter at vi har oppfylt deres krav og at kvaliteten på produktet er av ønsket kvalitet (Se vedlegg 9.3).

7.2.1 Refleksjon på egen læring

Underveis i dette prosjektet har vi lært veldig mye. Ettersom at teknologiene og fremgangsmåtene vi har brukt i prosjektet var helt nye for oss, har vi brukt mye tid på å forstå og skaffe en god nok kunnskap til å benytte oss av disse. Vi har blitt introdusert til Azure skytjenester, noe som er veldig ettertraktet kunnskap, vi har lært nye kodespråk (Python og Scala) og vi har lært nye lagringsmetoder, i form av Delta Lake. I tillegg til dette så har vi fått erfart virkelighetsnært arbeidsliv, der vi har operert som konsulenter for “kunden” vår Egde. Slik praktisk kunnskap er sjeldent, og det er svært studenter som får oppleve en slik oppgave. Siden alle gruppemedlemmer skal jobbe etter endt studie har slik læring vært til stor hjelp.

7.3 Utfordringer

Gruppen har møtt på en del utfordringer gjennom prosjektet som ikke bare var knyttet opp mot det praktiske med oppgaven, men også logistikken rundt møtene, rettigheter i prosjekt miljø, motivasjon og Covid-19. Under dette kapittelet vil det bli drøftet rundt de ulike og mest merkbare utfordringene vi hadde som gruppe, og hvordan vi løste dem. I tillegg vil vi nevne hva som kunne blitt gjort på en annen måte.

7.3.1 Covid-19

Covid-19 var en utfordring gjennom prosjektet som gikk utover gruppens arbeidsflyt og motivasjon. I store deler av prosjektet var det vanskelig å finne en plass hvor alle i gruppen følte seg komfortable nok til å jobbe uten å måtte bekymre seg for Covid-19. Dette medførte til at gruppen ut gjennom nesten hele prosjektet deltok i gruppe workshops og møter hjemmefra. I tillegg til dette, var det også en utfordring i seg selv å ha møter over Teams, da dette ikke alltid ga ønsket resultat og var i etterkant ofte vanskelig å få tak i gjeldende person for en forklaring. Helhetlig tok avgjørende valg lengre tid på grunn av at vi måtte kontakte veileder og/eller aktuell person via epost eller melding, som ofte tok noen timer, til dager.

I retrospekt kunne gruppen tatt mer initiativ til å sette opp fysiske møter hos Egde og/eller skole. Hvis vi hadde hatt flere møter fysisk med hverandre hadde det vært vanskeligere å miste fokus på oppgavene og vi hadde vært mer effektive under arbeidet. Til tross for at vi potensielt kunne vært mer effektive i arbeids møtene våre, er gruppen enig i at vi ikke tror det nødvendigvis hadde utgjort en stor forskjell på sluttproduktet og er dermed bare en ting vi kunne gjort annerledes.

Det ble også laget tiltak for at vi skulle få en grei flyt og oversikt på ting som måtte gjøres til tross for at alt var digitalt. Et av tiltakene som fungerte bra for gruppen var å alltid ha et fast møte på mandager hvor vi gikk gjennom planen og gjøremålene for uken. I disse digitale Weekly-Scrum møtene diskuterte vi rundt hva som gikk bra forrige uke, hva vi kunne gjort bedre og la planene for hva vi burde fokusere på den kommende uken. Ikke nok med det var det faste møter annen-hver uke med vår veileder fra UiA hvor vi fikk input på hvordan vi gjorde det generelt i oppgaven og hvor vi burde fokusere.

7.3.2 Roller og Rollefordeling

Før prosjektet begynte fikk gruppen beskjed av Egde at vi skulle velge en prosjekt-leder og Dev-Ops ansvarlig. Ansvarsområdet for prosjekt-lederen var å være kontakt personen som UiA kunne forholde seg til innad i gruppen, samt at Egde også hadde en person som kunne snakke for gruppen og ta viktige møter. I tillegg til disse to rollene ønsket vi å ha en rolle for dokumentasjon av møter og intervjuer. Fordelingen av disse rollene var ikke et stort problem ettersom at vi alle kjenner hverandre godt. I dette prosjektet opererte Adel Hodzalari som prosjektleder, Dzenet Bero som ansvarlig for dokumentasjon og Oskar Testad som Dev-Ops.

7.4 Brukerhistorie

Etter en grundig vurdering av alle gruppemedlemmene så valgte vi å ikke ta i bruk av bruker historier under dette prosjektet. Grunnlaget bak gruppens valg er på grunnen av hvordan systemet er laget. Ettersom at systemet vårt ikke er et interaktivt system med et visuell design så hadde ikke gruppen behov for fordelene en brukerhistorie gir. Ofte med bruker historier så er hensikten og få et «perspektiv» av sluttbrukeren på hvordan en funksjon skal utformes ("User Stories and the Alternatives," 2020).

Hadde gruppen hatt mer tid i dette prosjektet så hadde det blitt inkludert Brukerhistorier. Dette er fordi gruppen hadde startet med å jobbe på front-end designet på systemet, samt så hadde vi hatt et system som hadde vært rettet til en større kunde-gruppe. Som utgjør et større behov for Brukerhistorier.

Vi har brukt et «konsept» av en Brukerhistorie der vi har ført inn oppgaver i Jira på en unik måte. I Figur 6 så kan man se at oppgaven er skrevet i et tredje perspektiv, men dette er kun for selve oppgaven og ikke systemet.

8. Konklusjon

Oppgaven vi fikk utdelt fra Egde var å gjennomføre en ETL prosess, der data fra to ulike datakilder skulle transformeres og lagres som en tabell i en Datalake. Teknologiene vi skulle benytte oss av var teknologier innenfor skytjenesten Microsoft Azure. Siden teknologiene vi brukte var nokså ukjente for Egde hadde vi frie tøyler til å utforske og finne den fremgangsmåten vi trodde egnet seg best for Egde. Underveis i prosessen har vi møtt på flere utfordringer, men bruk av gode hjelpemidler, rammeverk, god kommunikasjon og samarbeid har vi sammen med Egde håndtert disse og gjennomført hele prosessen.

Sluttproduktet er en POC av vår ETL prosess, der vi har dokumentert alle ledd i prosessen. Vi henter data fra en Datalake, videre blir dataen sendt inn i vårt arbeidsmiljø som er Databricks, her blir dataen transformert til ønsket nivå og til slutt sendes dataen ut som Delta format i en ny Datalake. Vi er fornøyde med det resultatet vi oppnådde, men prosessen kan alltid utbedres. Vi rakk ikke å transformere all dataen til ønsket nivå, ettersom at vi ikke hadde nok tid. Vi ønsket også å kunne benytte oss av Power BI slik at dataen kunne blitt visualisert på en mer brukervennlig måte, men dette manglet vi rettigheter til. Vi oppfordrer Egde til å jobbe videre med prosjektet, og implementere et nytt system med dette prosjektet som utgangspunkt.

Referanseliste

Abraham, A. (2021, January 29). Scrumban — The unwanted Child. Retrieved from Agile

Insider website: <https://medium.com/agileinsider/scrumban-the-unwanted-child-272b18482c0b#:~:text=Scrumban%20is%20an%20agile%20management>

Aven, T. (2016). Risk assessment and risk management: Review of recent advances on their foundation. *European Journal of Operational Research*, 253(1), 1–13. Science Direct.

<https://doi.org/10.1016/j.ejor.2015.12.023>

ecfan. (2018, June 29). Automate tasks for enterprise integration - Azure Logic Apps.

Retrieved from Microsoft.com website: <https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview>

Estimating Time Effectively. (n.d.). Retrieved May 13, 2022, from RWA Insight website:

<https://insight.rwabusiness.com/blog/posts/2020/march/estimating-time-effectively/#:~:text=Once%20you%20can%20estimate%20the>

ETL Pros and Cons | InetSoft Technology. (n.d.). Retrieved from www.inetsoft.com website:

https://www.inetsoft.com/business/solutions/etl_pros_and_cons/

ETL vs ELT: Key Differences, Side-by-Side Comparisons, & Use Cases. (2020, January 27).

Retrieved from Rivery website: <https://rivery.io/blog/etl-vs-elt/>

ETL vs ELT: What's the Difference? | Zuar. (2022, January 3). Retrieved May 13, 2022, from

Zuar | Blog website: <https://www.zuar.com/blog/etl-vs-elt-whats-the-difference/>

ETL vs. ELT and the Benefits of Data Transformation in the Cloud. (2020a, January 9).

Retrieved May 13, 2022, from Best Data Integration Vendors, News & Reviews for

Big Data, Applications, ETL and Hadoop website: <https://solutionsreview.com/data-integration/etl-vs-elt-and-the-benefits-of-data-transformation-in-the-cloud/>

ETL vs. ELT and the Benefits of Data Transformation in the Cloud. (2020b, January 9).

Retrieved from Best Data Integration Vendors, News & Reviews for Big Data, Applications, ETL and Hadoop website: <https://solutionsreview.com/data-integration/etl-vs-elt-and-the-benefits-of-data-transformation-in-the-cloud/>

Halbo, L. (2020, January 6). kvalitetssikring. Retrieved from Store norske leksikon website: <https://snl.no/kvalitetssikring>

Jira - hva er det egentlig? (2020, September 1). Retrieved May 13, 2022, from Computas website: <https://computas.com/artikkel/jira-hva-er-det-egentlig/>

mssaperla. (n.d.). Secret scopes - Azure Databricks - Workspace. Retrieved from docs.microsoft.com website: <https://docs.microsoft.com/en-us/azure/databricks/security/secrets/secret-scopes>

Om oss. (n.d.). Retrieved May 13, 2022, from Egde Consulting website: <https://egde.no/om-oss/>

Oracle. (n.d.). What is a Data Warehouse? Retrieved from www.oracle.com website: <https://www.oracle.com/database/what-is-a-data-warehouse/>

ProductPlan. (2018). What is MoSCoW Prioritization? | Overview of the MoSCoW Method. Retrieved from Productplan.com website: <https://www.productplan.com/glossary/moscow-prioritization/>

Scrum Vs. Kanban: Know the Difference. (n.d.). Retrieved from www.guru99.com website: <https://www.guru99.com/scrum-vs-kanban.html>

User Stories And The Alternatives. (2020, August 27). Retrieved May 13, 2022, from Scrum

Inc website: <https://www.scruminc.com/user-stories-and-the-alternatives/>

What is a Data Lakehouse? (n.d.). Retrieved from Snowflake website:

<https://www.snowflake.com/guides/what-data-lakehouse>

What is ELT? (Extract, Load, Transform) The complete guide. (n.d.). Retrieved May 13, 2022,

from Qlik website: <https://www.qlik.com/us/elt>

What is Scrumban? | Definition, Overview, and Examples. (n.d.). Retrieved from

www.productplan.com website: <https://www.productplan.com/glossary/scrumban/>

Wickramasinghe, S. (n.d.). What's ETL? Extract, Transform & Load Explained. Retrieved from

BMC Blogs website: <https://www.bmc.com/blogs/what-is-etl-extract-transform-load-etl-explained/>

9. Vedlegg

9.1 Gruppekontrakt

Målsetning

Gruppen ønsker å oppnå en A, men B er akseptabelt.

Resultat:

Resultatet av gruppearbeidet blir karakteren og eventuelt annen læring.

Medlemmer:

Dzenet Bero, Adel Hodzalari, Oskar Testad

Overordnet Regler:

Gruppen vil avtale møtetidspunkt etter behov hvor det fordeles arbeidsoppgaver og hvor det blir tatt en helhetlig vurdering på enkeltes bidrag til gruppens sluttresultat.

1. Regler:

1. Gruppen vil bruke Messenger og Discord som hovedkommunikasjons plattform og mobilnummer til nød.
2. Gruppen vil bruke Intellij, visual studio code, github, google docs og Miro som arbeidsverktøy.
 - a. Gruppen skal bruke et Versjonskontroll verktøy på prosjektet
3. Gruppen møtes etter behov.
4. Det forventes at medlemmene møter til avtalt gruppearbeid hvor oppmøte er avtalt.
5. Informer om forsinkelse så fort de oppstår via Messenger, Discord eller per telefon til leder eller viseleder.

6. Det forventes at gruppemedlemmene holder seg oppdatert på kommunikasjonsplattformen til gruppen. (Henviser til punkt 1.1)
7. Det forventes at gruppen aktivt bruker Miro og ser hva som må gjøres.
8. Når gruppearbeid er avtalt forventes det en 45 minutters arbeidstid og 15 minutters pause. (Kan justeres etter behov under møtene)
9. Alle medlemmer forventes til å være i stand til å gjøre lik innsatts i forbindelse med oppgaver.
10. Hvis jobben din hindrer deg i å gjøre en oppgave, forventes det at du legger inn innsatsen på fritiden.

2. Gruppen vil bruke et 4 strike system til alvorlige brudd og manglende innsatts:

1. Alvorlig brudd og manglende innsatts er herunder definert som:

- a. Store forsinkelser / Gjentatte forsinkelser (20-30min)
- b. Tydelig mangel på innsatts
- c. Grad av manglende innsatts vurderes av gruppen i plenum
- d. Dersom gruppen ikke kommer til enighet, får gruppeleder avgjørende stemme for endelig avgjørelse.

2. Strike systemet:

- a. Strike 1: Muntlig advarsel
- b. Strike 2: Straff i form av ekstra arbeid.
- c. Strike 3: Samtale med gruppe leder og en siste advarsel.
- d. Strike 4: Medlemmet blir utmeldt fra gruppen uten videre advarsel.

3. Gruppen vil også ha en felleskasse for mindre brudd:

1. Mindre brudd blir herunder definert som:

- a. Små forsinkelser på rundt 15min (50kr i kassa)

- b. Forstyrrende eller distraherende adferd under gruppe arbeid (50kr)
- c. Pranks som er voldelig eller går TYDELIG utover helse og sikkerhet. (100kr)
- d. Svikt i forbindelse med tildelte gruppe oppgaver (50kr)
- e. Andre brudd og konsekvenser vurderes fortløpende av gruppen

4. Konflikt:

Om det skulle oppstå konflikt rundt strikes og felleskasse, eller generell oppførsel i forhold til samarbeid vil det håndteres på følgende måte:

- 1. Konflikten vil bli holdt internt og medlemmene som opplever problemet kan ta det opp med vedkommende eller annet medlem for å diskutere en løsning.
- 2. Om konflikten ikke kan bli håndtert internt vil vi søke hjelp fra lærer.
- 3. Om konflikten ikke kan bli håndtert av (a, b) vil gruppen stemme over medlemskapet.
 - a. I disse tilfellene vil vedkomne ikke ha stemmerett - for å unngå uavgjort
 - b. Alle stemmer sammen, gruppeleder har avgjørende stemme i uavgjort situasjon

5. Roller:

- 1. Enkelte medlemmer har tildelt roller som har sine ansvar og oppgaver:
 - a. Prosjekt Leder: Adel Hodzalari
 - b. Dev-Ops: Oskar Testad

6. Normer:

Det er lov å spise i gruppemøter.

Det forventes at gruppemedlemmene hjelper hverandre.

Denne gruppekontrakten er opp for endring i løpet av semesterets gang

Signaturer:

Dzenet Bero

Adel Hodzalari

Oskar Testad

Dzenet Bero Adel Hodzalari Oskar Testad

9.2 Styregruppemøter

9.2.1 Styremøte 1

Styringsgruppemøte 1

Informasjon:

Sted: Digitalt møte på Teams

Dato: 18.02.2022

Tid: 10:00 - 11:00

Deltakere:

- Prosjektgruppen (RefreshedIT)
- Produkteier
- Veileder

Agenda:

- Første sprint review med Veileder til stedet.
- Status og fremgang
- Plan for neste sprint

Sprint Review

Fremgangsmåten for sprint review fikk vi anbefalt av Egde, og måten vi gjorde det på var å finne frem et Miro-bord, og laget en tabell som er delt opp i 3-deler. De tre delene består av 1. Hva skal vi slutte med? 2. Hva skal vi fortsette med? 3. Hva skal vi start med? I disse kategoriene legger vi til lapper, med ting vi mener passer.

Under reviewen kom det frem blant annet at det er litt forsinkelser i møter og at gruppen ikke er så effektive. Det kom også frem at gruppen skal fortsette med å fokusere på Databricks oppgaven og vi ble enige om at vi skulle begynne å komme mer presist til møtene.

Status og Fremgang

I status forklarte vi fremgangen vår så langt og at vi strevde med rettighetene, samt noen komplikasjoner med Databricks som vi ikke helt forstod. Vi fikk beskjed om at vi skulle få bedre veiledning i løpet av uken når de har funnet ut hva problemet er.

Plan for neste sprint:

Planen videre ble vi enige om er å få de tingene som ikke fungerer i databricks til å fungere som de skal. Samt lese videre på rapporten og sakte, men sikkert utarbeide den.

9.2.2 Styremøte 2

Styringsgruppemøte 2

Informasjon:

Sted: Digitalt møte på Teams

Dato: 18.03.2022

Tid: 08:30 - 09:00

Deltakere:

- Prosjektgruppen (RefreshedIT)
- Produkteier
- Veileder

Agenda:

- Sprint Review
- Status og fremgang
- Plan for neste sprint

Sprint Review

Vi benyttet oss av samme fremgangsmåte som sist, ved at vi brukte Miro som et verktøy til å effektivt visualisere hva vi synes går dårlig, hva vi synes går bra og hva vi skal begynne å gjøre. I denne omgang kom vi frem til følgende:

- Hva skal vi slutte med?
 - a. Bruke lang tid på å sette i gang på møtene
 - b. Bli flinkere på å holde fokuset i møtet.
- Hva skal vi fortsette med?
 - c. Jobbe med Databricks
 - d. Skrive videre på rapporten
- Hva skal begynne med?
 - e. Begynne å lese gjennom rapporten og rette
 - f. Spør oftere hvis det er noe vi sliter med for bedre framgang.

Vi fikk tips av veiledere å kanskje legge av tid i begynnelsen av møtene for “catch-up” i stedet for rett på arbeid.

Status og Fremgang

I denne status oppdateringen valgte vi å vise hva vi klart til nå, og hvordan vi tenker å angripe de nye problemstillingene som dukket opp under sprinten. Vi har enda problemer med tilganger som må løses, men ellers har vi klart å få god fremgang på databricks fronten.

Vi strever derimot litt med å få inn relevant data da det excel arket vi har fått er svært manglende.

Plan for neste sprint:

Planen videre blir å få Databricks oppe å kjøre fullstendig, samt at vi skal begynne å flette inn litt CV-Partner med Databricks. Dette blir kanskje mulig å få gjort via API, men ikke sikkert det funker på grunn av manglende rettigheter.

9.2.3 Styremøte 3

Styringsgruppemøte 3

Informasjon:

Sted: Digitalt møte på Teams

Dato: 29.04.2022

Tid: 08:30 - 09:00

Deltakere:

- Prosjektgruppen (RefreshedIT)
- Produkteier
- Veileder

Agenda:

- Første sprint review med Veileder til stedet.
- Status og fremgang
- Plan for neste sprint

Sprint Review

Vi benyttet oss av samme fremgangsmåte som sist, ved at vi brukte Miro som et verktøy til å effektivt visualisere hva vi synes går dårlig, hva vi synes går bra og hva vi skal begynne å gjøre. I denne omgang kom vi frem til følgende:

- Hva skal vi slutte med?
 - g. Slutte å hoppe frem og tilbake på om vi skal ha workshop i siste liten.
 - h. Slutte å forsove seg.
- Hva skal vi fortsette med?
 - i. Utarbeid rapport og/eller fullføre den.
 - j. Bygge videre på demo, eventuelt ferdigstille
 - k. Skrive videre på rapporten
- Hva skal begynne med?
 - l. Gjennomgang av rapport og kvalitetssikring
 - m. Legge en jevn plan for skrivningen
 - n. Levere rapporten for evaluering av veileder, eller aktuelle bekjente.

Vi fikk tips av veiledere å prioritere rapporten mer og mer jo nærmere innleveringen kommer. Veilederen vår fra Egde forteller også at vi har klart å gjøre god fremgang og at det ikke er mye igjen før vi er i mål med det de forventet å få dette bachelor prosjektet.

Status og Fremgang

Status oppdateringen valgte vi å trekke frem vår fremgang i form av en enkel demo, dette var for å vise hva vi tenkte å gjøre og hvordan vi tenker å gjøre videre. Her fikk vi gode innspill på hvor vi kan forbedre og hva vi burde fokusere på fremover. Vi har også valgt å se bort ifra oppkobling av API mot CV-partner grunnet manglende rettigheter og at det ville tatt for lang tid å utføre enn det er verdt. Vi har derimot fått ulike CV-er i JSON format som vi kan bruke, direkte levert i Datalaken vår.

Plan for neste sprint:

Planen for de neste sprintene blir å prøve å lage en grei demo, snakke med sluttbrukere og fremvise det vi har for alle aktuelle mennesker. I tillegg skal vi videre bygge på det vi har ut ifra responsen vi får på de ulike veilednings møtene. Ettersom det nærmer seg innlevering, tenker vi å legge Databricks litt på hylla, til vi er ferdige med rapportene. Og så fortsette de 2 siste ukene før den endelige presentasjonen.

9.3 Uttalelse fra Egde



Høsten 2021 arrangerte Universitetet i Agder RefreshIT på Radisson Blue Caledonien Hotel i Kristiansand. Her møtte flere lokale aktører opp for å presentere seg selv for studentene, for så å legge frem hvorfor studentene skulle samarbeide med de representative aktørene. Egde ble gjennom arrangementet en populær aktør som mange studentgrupper ønsket å samarbeide med. Gjennom en intervjurunde og intern prat falt Egdes valg på RefreshedIT bestående av Adel, Dzenet og Oskar.

Egde er en bedrift i vekst som stadig utforsker ny teknologi. Gjennom prosjektet har studentene bidratt til å undersøke potensial rundt Egdes interne data gjennom nye teknologisatsinger. Dette var utgangspunktet for oppgaven studentgruppen har arbeidet med gjennom denne våren.

Kravene til prosjektet er definitivt møtt og resultatet er av svært tilfredsstillende grad da datagrunnlaget som ble gitt både har blitt strukturert og utarbeidet på en måte som kartlegger en prosessflyt i teknologivalgene.

Veiledere hos Egde var aldri i tvil om at gruppens prosjektleder, Adel, hadde kontroll på arbeidsoppgaver og fordeling av disse mellom gruppens medlemmer.

Studentene har videre presentert resultatet for Egdes faggruppe "Data og Analyse" hvor de presenterte prosjektet, hvilke verktøy som var benyttet, samt teknologivalg tatt underveis.

RefreshedIT har på en god måte løst utfordringen og levert et resultat de kan være stolte av. Dette har studentene oppnådd gjennom godt samarbeid internt i gruppen, og kontinuerlig kommunikasjon med Egde når større utfordringer ble oppdaget underveis.

Egde er en profesjonell aktør som verdsetter kvalitet i arbeid og metodikk. Her har studentene levert som ønsket og Egde er stolte av å kunne være en del av denne bacheloroppgaven med studentene.

Vi takker studentene for deres innsats og ønsker dem lykke til videre!

12.05.2022

På vegne av Egde

Eirik Gjeruldsen og Erik Nergaard

9.4 Backlog

SV2E Sprint 7 18 Apr – 1 May (12 issues)			0m	2w 1d 30m	1w 1d 6h 30m	Complete sprint	...
SV2E-70	Vi må fortsette å skrive på Rapporten.	4d 6h	UNDER ARBEID				
SV2E-9	Assignment 5: Final portfolio	13 MAY 1d 1h	UNDER ARBEID				
SV2E-77	Vi som gruppe må se på tilbakemeldingene vi fikk i IS-305	5h	TESTING / REVIEW				
SV2E-74	Vi må lage demo til Eirik og Erik	3h	FERDIG				
SV2E-81	Vi må lage en måte å presentere fremdrift på (Funksjon?)	3h	FERDIG				
SV2E-11	Presentation 2: Final term project	27 APR 1h	FERDIG				
SV2E-40	Som gruppe må vi fullføre transformering av Test data ETL	4d 30m	FERDIG				
SV2E-00	Presentasjon for Egde	1d 7h	FERDIG				
SV2E-76	Gruppen er nødt til og se på hvordan de kan lage en database		FERDIG				
SV2E-79	Vi må gå gjennom kommentarer på rapport og utarbeide RAPPORTSKRIVING	7h	UNDER ARBEID				
SV2E-86	Vi må lage et første utkast av ny struktur til bachelor rapport	1d 1h	UNDER ARBEID				
SV2E-87	Vi må skrive rapporten for IS-305	2d 4h 30m	UNDER ARBEID				
+ Create issue							