

Praktikrapport 2020

Hos Efo ApS

Asger Hermind Sørensen
cph-as466@cphbusiness.dk

Praktikrapport for Datamatiker 5. Semester

Datamatiker
Cphbusiness Lyngby
Danmark
23. oktober 2020

Indhold

1. Fordord	2
2. Beskrivelse af praktikvirksomheden	3
3. Hvordan jeg har opfyldt mine læringsmål	4
3.1 viden	4
3.2 Læringsmål	4
2.3 Kompetencer	6
3. Udførte arbejdsopgaver	8
4. Refleksion	10
5. Bilag	11

1. Fordord

Denne rapport er skrevet ud fra mit praktikophold i Efo ApS hovedkontor i København. Jeg har arbejdet gennemsnitligt 35 timer om ugen både på Efo's kontorer samt hjemmefra.

Rapporten tager udgangspunkt fra praktikkens start den 17. august 2020 frem til den 9. oktober 2020. Praktikopholdet har været i gruppe sammen med Andreas Zoega Vestergaard Vikke og Martin Eli Frederiksen.

2. Beskrivelse af praktikvirksomheden

Efio er et DevOps konsulentfirma bestående af 2 afdelinger beliggende i København og Odense. Efio har i alt 8 DevOps ingeniører ansat som er meget nørdet inden for automatisering. Dette betyder selvfølgelig at der er et stort fokus på kontinuerlig integration og kontinuerlig levering. Efio er også samarbejdspartner med Amazon web service og praktiserer bl.a. serverless, containers og infrastructure as code.

3. Hvordan jeg har opfyldt mine læringsmål

Efio har to afdelinger, en i København og i Odense. De også endnu et praktikhold i Odense ud over os. Efio arbejder som konsulenter ude hos de virksomheder som de har som kunder, og er derved sjældent på egne kontorer. Hvad kunden har behov for kan variere meget, og derved også konsulentens arbejde. De fleste konsulenter har en fast kunde de er ude hos, og arbejder projektvis hos en kunde.

Efio har 4 kunde segmenter, de store virksomheder i lange ophold er det som efios konsulenter hovedsageligt foretager sig. Men nogle små virksomheder kan også have brug for hjælp til at få ordnet fejl som kunden ikke kan eller ved hvordan skal ordnes. Små virksomheder kan også købe klippekort til Efios konsulenter.

Efio laver et daily standup på slack, så alle stadig ved hvad hinanden arbejder på selvom de ikke ser hinanden eller arbejder sammen i en længere periode.

3.1 viden

Efio har to afdelinger, en i København og i Odense. De også endnu et praktikhold i Odense ud over os. Efio arbejder som konsulenter ude hos de virksomheder som de har som kunder, og er derved sjældent på egne kontorer. Hvad kunden har behov for kan variere meget, og derved også konsulentens arbejde. De fleste konsulenter har en fast kunde de er ude hos, og arbejder projektvis hos en kunde. Efio har 4 kunde segmenter[HUSK FODNOTE HER], de store virksomheder i lange ophold er det som efios konsulenter hovedsageligt foretager sig. Men nogle små virksomheder kan også have brug for hjælp til at få ordnet fejl som kunden ikke kan eller ved hvordan skal ordnes. Små virksomheder kan også købe klippekort til Efios konsulenter. Efio laver et daily standup på slack, så alle stadig ved hvad hinanden arbejder på selvom de ikke ser hinanden eller arbejder sammen i en længere periode.

3.2 Læringsmål

Joachim, direktør for Efio og vores tutor, har været inde over hele udviklingsprocessen, og har været med til at tænke kritisk og logistisk i en anden sans end skolen har. Joachim har gået meget op i at vi følger scrum helt rigtigt, og at vi inddrager dens elementer. Joachim startede sprintsne med at redegøre for hvad der skulle laves, og gjorde redefor hvordan det skulle udføres og kobles sammen. Det har derved stået os til opgave at prøve at udføre det, og hvis der opstod nogle problemer, snakkede vi internt eller snakkede med Joachim til standup møder vi normalt har om eftermiddagen. Jeg har stødt ind i flere problemer som jeg har skulle formidle til Joachim, hvor vi sammen prøver at finde

en løsning til det problem, som han synes passer bedst.

Tekniske og Analytiske Arbejdsmetoder

Vi har brugt rigtig meget scrum til udviklingen af det produkt vi sidder med. Jeg kender til, eller har hørt om, alt vi har arbejdet med. Der blev taget meget fokus i definition of ready og definition of done, som jeg kun kort er blevet informeret om på skolen. Definition of ready består af INVEST:

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

For at en user story er klar til at kunne blive sat i sprint backlog, skal den opfylde INVEST. Det kræver at userstoryen:

- Er uafhængig af andre user storys.
- Kan ændres og diskuteres, det er ikke en kontrakt for en feature.
- Bringer en form for værdi, enten til systemet, brugerne eller aktionere.
- Kan rimeligt estimeres.
- Ikke er for stor til ikke at kunne passe i et sprint.
- Kan testes, og kræve nok information til at der kan laves test driven development eller behaviour driven development

Hvis en user story kan opfylde alle de krav kan den derved tages med i sprintet. Da vi kørte dette projekt, skulle alle være enige om hvert punkt i INVEST var opnået før definition of ready var opfyldt. Hvis ikke diskuterede vi for og imod punktet, og lavede ændringer hvis nødvendigt. Når en user story er klar til at blive gennemgået og skal ligges over som færdig skal den først kunne opnå definition of done: Definition of Done:

1. automatiseret godkendelse af acceptens kriterier
2. gennemgået af holdmakker(e)
3. fusioneret med master

4. Deployere til produktion

Første punkt bliver kørt igennem Cucumber's Behave, som kører accept test i python som en del af Behaviour Driven Development (BDD). Der testes lokalt før man pusher op, men alle behave test køres automatisk oppe på CI-pipelinen når man pusher op på en feature branch. Der kan ikke merges ind til master uden at man laver et pull request. Dette pull request kræver at en anden fra ens hold gennemgår ens kode, og tjekke at det man har lavet, er i orden og at koden opnår en vis kvalitet. Derved får man klaret punkt to og tre ved pull requestet, når en holdmakker godkender ens pull request kan man merge ind til master. Til sidst opretter man en ny udgivelse på GitHub, det er CD pipeline sat op til at lytte efter. Når en ny udgivelse er lavet, sender GitHub et request til pipelinen om at der er lavet en ny release, som CD-pipelinen lytter på, så går den i gang med at bearbejde. Når pipelinen er kørt igennem og alt fungerer, er der blevet deployeret til produktion.

Struktur

Vi har brugt Jira til at opstille struktur omkring den individuelle arbejdsgang. Jira er en platform der bruges til agile udviklingsmetoder. Jira har hele vores sprint backlog, hvor hver user story er brudt ned i subtasks. Alle user storys kommer fra sprint planning meeting med PO, hvor vi sammen har siddet og bedømt hvor mange user storys vi kan tage på et sprint, med henhold til den viden vi besidder på daværende tidspunkt. For at kunne bedømme hvor stor en user story er har vi brugt poker planning sammen med risk evaluation, for at bedømme om den størrelse der er givet til user story har en lille eller stor chance for at ramme forkert. Risk evaluation bliver baseret på hvor meget vi kender til de teknologier som user storyen indebærer. Hver dag laver vi daily standup, og snakker med hinanden om hvad vi arbejder på, og om vi har nogle problemer vi eventuelt har brug for noget hjælp til. Derved har vi altid et overblik over hvor hele holdet er henne. Hvis der ikke er nogen der har brug for hjælp, har vi arbejdet individuelt videre på den user story man er på, eller tilknytter sig en ny user story eller subtask.

2.3 Kompetencer

Tilegnelse af ny viden

De første tre uger gik næsten kun ud på at tilegne sig viden omkring serverless computing, AWS og deres tjenester. Meget af det har været svært at forstå, da det er noget helt andet end det vi har arbejdet med på skolen. Selve brugen af tjenesterne har ikke været utroligt svært, men forståelsen af hvordan de forskellige tjenester fungerer har været det jeg har haft meget svært med. Joachim har gennemgået mange af de ting som jeg har fundet svært, og virkelig gået i dybde med de forskellige systemer og tjenester som vi har taget i brug så jeg kan få en dybere forståelse.

Faglig og tværfaglig

Jeg har hovedsageligt gjort brug af systemudvikling med scrum. Herved har vi også haft Peter inde over, som er virkelig god til at se problematikker for brugeroplevelsen. Her har vi snakket sammen om hvordan vi kunne forbedre designet af vores produkt, for at gøre det mere brugervenligt.

3. Udførte arbejdsopgaver

sprint 0

Før der kunne begyndes på projektet, var jeg nødt til at lærer omkring AWS og de tjenester som de tilbyder. Dette indeholder bl.a. at få et certifikat fra Amazon om gennemførelse som AWS Technical Professional[HUSK FODNOTE HER]. Herudover brugte jeg lang tid på at få online undervisning omkring AWS og deres tjenester samt hvordan man bruger dem igennem a cloud guru.

sprint 1

I projektets første sprint startede jeg med opbyggelse af en NoSQL database, DynamoDB. Jeg har efterhånden skulle opbygge en del databaser med relationer, men det er ikke muligt med en NoSQL database, og logikken er helt anderledes. Det er ikke noget vi har lært på skolen, men har heller ikke været anset nødvendigt. Resten af gruppen har opbygget Continuous Integration (CI) og Continuous Deployment (CD) pipelines. Det har vi kun fået en dags undervisning i, derved var det svært at forstå hvordan det fungerede til at starte med, og siden jeg ikke selv har bygget den er min viden om den stadig begrænset.

sprint 2

Idet der blev arbejdet på meget nye tjenester, blev andet sprint sat af til Spikes. Spikesne omfatter brugen af lambda funktioner, som er begivenhedsdrevet serverløst computing platform, så man kan køre kode uden at skulle håndtere servere.

En af de lambdafunktioner skal kunne skrive til en Simple Queue Service (SQS), som holder beskeder i kø som derefter kan køres af andre lambdafunktioner. Der skulle dertil også en lambda-funktion til at kunne tage en besked fra SQS-køen og gøre noget ved den besked.

En anden lambdafunktion skulle kunne skrive fra en Simple Notification Service (SNS), som kan skrive beskeder, e-mails, SMS mm. ud til alle som abonnere på et emne som den SNS omhandler. Man kan bruge en lambdafunktion til at skrive beskeder ud fra SNS, som derved bliver sendt ud til alle der abonnere, som kan være en anden lambdafunktion som gør noget ved den besked.

sprint 3

Her havde jeg fokus på Slack Connection Service (SCS), som er at snakke med Slacks API igennem den app vi havde lavet. Det bestod af at kunne sende en knap til en kanal i Slack, som en bruger kan trykke på. Her brugte jeg meget postman for at finde ud af hvordan Slacks API fungerede,

som vi har brugt meget i skolen.

Når brugeren trykker på knappen, skal der poppe et Modal op. Modalet blev designet sammen med PO under sprint planning. Modalet skulle udfyldes med de kunder som brugeren kunne have været ud hos, som var blevet forudsagt og ligger i databasen. Inden modalet bliver sendt til brugeren skal modalets template blive udfyldt med det data der ligger i databasen. Modalets template er bygget i JSON, og kunne derved nemt manipuleres med andet data. Jeg kender til brug af API og manipulation af JSON fra skolen af, og var derved ikke en større udfordring.

sprint 4

PO ville gerne have ændret måden data blev gemt på i databasen når en bruger valgte noget i modalen. Han ville gerne kunne se hvis en kunde var blevet valgt fra hvis den allerede var valgt, og derved tilføje en soft-delete på kunden i databasen. Der skulle også registreres tid ud fra hver eneste kunde valgt, også hvis kunden blev valgt fra efter skulle tidsregistrering af den blive i databasen.

4. Refleksion

Virksomheden har fået en slack applikation som de kan bruge til at indregistrere timer de har været hos kunder. Dette er noget Joachim gerne vil have bygget i længere tid og har prøvet at få bygget med andre praktikgrupper også.

Selv har jeg lært utroligt meget nyt, serverless computing og opsættelse og brug af AWS. Det har givet utroligt mange udfordringer, da det er noget helt andet end det vi har lært på skolebænken.

Det som jeg tror kunne have været rart at have med inden jeg tog i praktik var en mere fastsættelse for brugen af scrum mens man er i skole. Ved faktisk at tage brug af definition of done og definition of ready. Samt gøre mere ud af at blive lært hvordan man ordentligt opbygger user stories med estimeringer. Mens vi sad i skolen, blev nogle af de her koncepter nævnt men aldrig taget i brug, eller krævet at vi tog dem i brug, og det ville have været rart så man havde gjort det til en vane at det er sådan man udvikler.

Mens vi arbejdede på projektet, var master branchen låst og der blev kun arbejdet på branches. Når man så var færdig lavede man pull requests. Jeg har ikke lært om pull requests og hvordan det fungerer, men jeg synes det er utrolig smart at man får kigget hinandens arbejde igennem, og på den måde for formindsket fejl. Ved at vi også hver i ser har arbejdet på sin egen branch har der været så få merge konflikter, og dem der har været har været så banale. Jeg ser meget positivt på at arbejde på den måde, og kommer selv til at arbejde på den måde fremadrettet når jeg skal arbejde i grupper.

Arbejdet i erhvervslivet har ikke haft den største indflydelse på mig, har arbejdet meget i praktikken som jeg har arbejdet i skolen. Forstået som at man har aftalte mødetidspunkter og aftalte tidspunkter man er færdig for dagen. Den gruppe jeg har været med i den periode jeg har været på skolen, har haft en del struktur og fælles ledelse. Jeg har kun set positivt på at komme ud i erhvervet, og glad for at se at scrum bliver taget noget mere seriøst end jeg selv har taget det førhen, det er også det der har været med til at vise mig hvor stor værdi scrum kan have for udviklingen af et system.

5. Bilag