

## Task 1

Describe the main challenges that the company will be facing when trying to scale up their processes. Try to mention as many challenges as you can without going too much into details.

Fra powerpointen i IN5140 gitt 14.Oct.2020 side 14<sup>1</sup> ser at det er mange utfordringer vi kan møte når man scaler opp en prosess, men hovedproblemene jeg tror selskapet vil møte på er:

- Kommunikasjon mellom gruppene(teams)
- Selv organisering modellen av gruppene
- Samme teknisk nivå på gruppene
- Forskjellig forståelse av hva agile er
- Motstand mot forandring
- Valg av framework
- Ledelse av flere team og kommunikasjon mellom dem

Kommunikasjon mellom gruppene var kanskje ikke et problem når det var kun 2 grupper å ha forhold til, men når det blir 15 så har vi en økning i antall grupper på 750%. Denne økte mengden team gjør at det er mer informasjon å forholde seg til når flere lag jobber på et project. Fra agile 11.prinsippet om selv-organisere grupper så kan dette lage mer problemer med 15 grupper enn med 2 siden det er lettere å holde styr på 2 selvorganiserende grupper en 15 for ledelsen.

Å holde likt kvalitetsnivå for teknisk kunnskap på alle gruppene så det ikke er noen forskjell på resultat produktet mellom gruppe 1 og gruppe 2.

Variasjoner i hva gruppene tenker på som agile kan føre til forskjeller i prosess eller følging av prosess i forskjellige grupper.

Motstand til forandring i gamle ansatte og ledelse struktur samt forandringer i posisjon og ansvar.

Det var ikke gitt noe framework for bedriften i oppgaveteksten så valg av riktig framework som passer prosessen og som er scalable for bedriften er viktig.

Med økt mengde grupper og mest sannsynlig prosjekter delt over flere grupper er ledelse å kommunikasjon mellom dem essensielt.

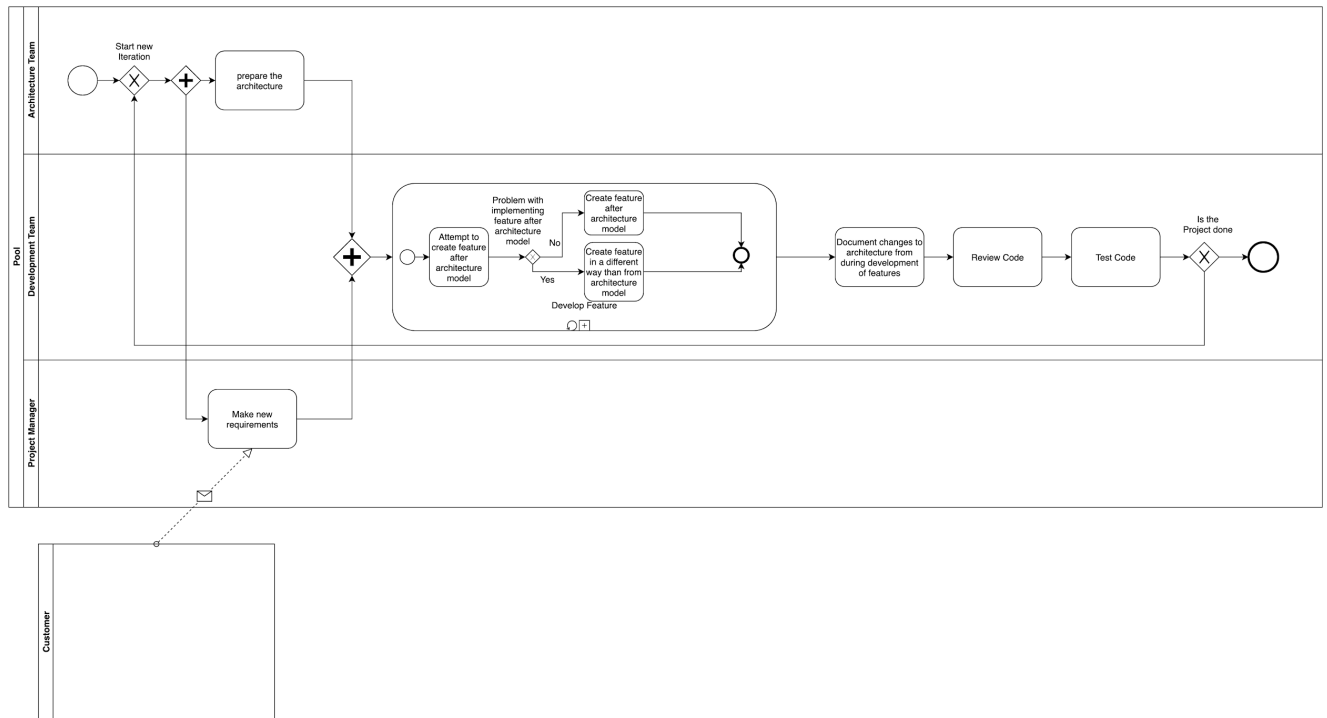
---

1

[https://www.uio.no/studier/emner/matnat/ifi/IN5140/h20/lectures/in5140\\_2020\\_10.14.1\\_largescale\\_02handouts.pdf](https://www.uio.no/studier/emner/matnat/ifi/IN5140/h20/lectures/in5140_2020_10.14.1_largescale_02handouts.pdf)

## Task 2

Model the process described above using BPMN. Make your own assumptions and add comments to your BPMN diagram.



Mine forventninger til BPMNen er at ved start starter prosjektleder å snakke med kunden for å lage nye krav, i oppgaveteksten så det ikke noe om håndtering og lagring av informasjon i prosessen så det er utelatt fra modellen. Selv om teksten sier at release blir levert hver gang har jeg valgt å avslutte prosessen kun når hele prosjektet er ferdig. I testing av kode gjennomføres også bug fix, hvor kun de viktigste blir gjort i forhold til hvor mye tid de har.

## Task 3

Explain to the owner of MediC that there are several kinds of debt, and why it's important to understand what debts exist and how they are related. Create a backlog of minimum 5 debt items according to the description above and use a table similar to the following:

Det er flere typer dept i software utvikling, men vi ser på 3 typer social dept, process dept og technical dept. Technical debt er snarveier tatt under software utvikling som gjør at framtidige endringer til softwaret vil kreve mer eller være umulig. Å skrive code med mange avhengigheter vil spare tid i det korte perspektiv, men vil føre til problemer for selskapet på lang sikt. Process Debt er uoptimal prosess design, avvik fra optimal formell prosess eller mangler i infrastruktur som kan ha fordeler i kort sikt. Men problemer kan skape både kort sikt sløsing av ressurser men også konsekvenser som på lang sikt har stort negativt innvirkning på prosess interessenter. Social Debt er uoptimaliteter i sosial samarbeid for bedriften, prosjekter og/eller grupper(teams) som kan forårsake negative effekter. Dept kan ha fordeler på kort sikt, men på langt sikt så lønner det seg ikke, så for en bedrift som holder på å utvide prosessen sin er det

Iurt å ta hensyn til dept og håndtere det på en god måte eller så kan det føre til problemer med konkurranse egenskapene til bedriften.

Dept ID	Short description	Type of dept	Consequences	Other debts that are affected
Debt 1	Kommunikasjon mellom architect teamet og development teamet	Sosial Dept	Development team må endre arkitekturen og uoptimale arkitektur forandringer blir tatt siden ingen arkitekter er tilstede for å hjelpe	Dept 2 Dept 3
Debt 2	Arkitektur blir ikke justert av arkitekt teamet når development teamet finner problemer de ikke tok hensyn til	Technical Dept	Avgjørelsene for forandring av arkitektur blir gjort av development team ou uoptimale løsninger blir valgt.	Dept 1
Debt 3	Mistillit mellom arkitekt teamet og utviklingsteamet	Sosial Dept	Kommunikasjon mellom teamene blir dårligere og samarbeid blir vanskeligere	Dept 1
Debt 4	Prosjektleder legger til nye krav under utvikling som fører til mindre tid for fiksing av problemer	Process Dept	Flere features blir laget, men kun de viktigste feilene blir rettet så mange av featurene blir uoptimale og dårlige.	Dept 5
Debt 5	Tidligere feil hindrer framtidige endringer og ny funksjonalitet	Technical Dept	Leverings tid blir tregere og endringer blir vanskelig å fikse etter som kompleksitet økte(deler ble	Dept 4

			avhengige av hverandre) mens det var påviste problemer i koden.	
--	--	--	---	--

#### Task 4

Describe an approach to improve the current process. Describe how the approach would solve at least two of the challenges described above. One suggestion is to use techniques from Lean. Use the Goal-Question-Metrics to define a strategy to measure that the process has been improved. In particular, characterize the measurements you suggest to collect.

Det er en del elementer i prosessen som kan forbedres men mitt forslag er å bytte den nåværende prosessen med en tilpasset kanban. Fokuset med denne nye prosessen er å forsikre kvalitet på featurene som blir laget som vil føre til minke antall bugs, senke technical dept og ut ifra det senke lead time. Ved tilpassing av den nye kanban prosessen trenger vi å forbedre kommunikasjon mellom arkitektur teamet og development teamet. Dette kan vi gjøre ved å la arkitekt teamet ha deltakere på møte om endringer i arkitekturen fra feil funnet i utvikling fra development team som kun development team var på før. Vi vil også at i stedet for at arkitektur teamet lager all arkitekturen å gjør ingen endringer etter at de har sendt det til development de i stedet starter en prosess parallell til development team hvor hvis development team finner feil i antagelsene i arkitekturen eller ting som ikke er tatt hånd om lages en feilrapport som blir sendt til arkitektur teamet som gjør at de kan utarbeide et forslag til endringer i arkitekturen i forkant av møte holdt av utviklingsteamet hvor endringene i arkitektur blir bestemt (som de nå får delta på) som gjør at optimale løsninger blir valgt. Dette fører til økt arbeidsmengde for arkitektur teamet, men vil mest sannsynlig minske lead time for prosjektene. Dette vi da også forbedre kommunikasjon mellom teamene. For å passe på at featurene har god kvalitet vil vi bruke et kanban board med en limit på antall features som kan jobbes på. Vi vil dermed prioritere features som er mest viktig. Dette kan føre til at i en syklus lager vi mindre feature en den nåværende prosessen, men det vil føre til at framtidige features vil bli lettere å implementere og ta kortere tid som vil føre til at prosjektet i sin helhet vil ta mindre tid.

For Goal-Question-Metrics:

Jeg har foreslått 2 hovedmål ved endringene mine minske technical dept fra feature utvikling og forbedre kommunikasjon mellom teamene for å øke effektivitet og redusere lead time.

Så mine mål er:

Goal: lower lead time for full project

Purpose: Improve lead time

Issue: problems in code and bad communication between teams

Object (process): the development process

Viewpoint: from the project manager's viewpoint

Question: What is the general lead time for current projects based on number of features

Metric/variable: Time used until the project was done per feature in the project.

Measured as: lead time/feature count

Question: How many features are generally completed in an iteration of development

Metric/variable: number of features made with new process divided by number of iterations.

Measured as: features/iteration count