

Exam Question 2020

IN5140

Questions asked

- Will BPMN models be in the exams?
 - Yes, most possible
- How much details do we need to know? Every icon and symbol there is or just the most important ones?
 - Only the most important ones. Lanes, actors, events, activities

How to answer on the exam?

The student must be able to show that they **understand** the curriculum and are able to **apply** it to different cases and **discuss** the results

In general:

- Answers are mostly correct and concise, no irrelevant text
- Answers are well structured
- The student reasoned about the topics and gives rationale
- Content is related to the material in the lectures and slides

Task 1 (15%)

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.

Task 1 (15%)

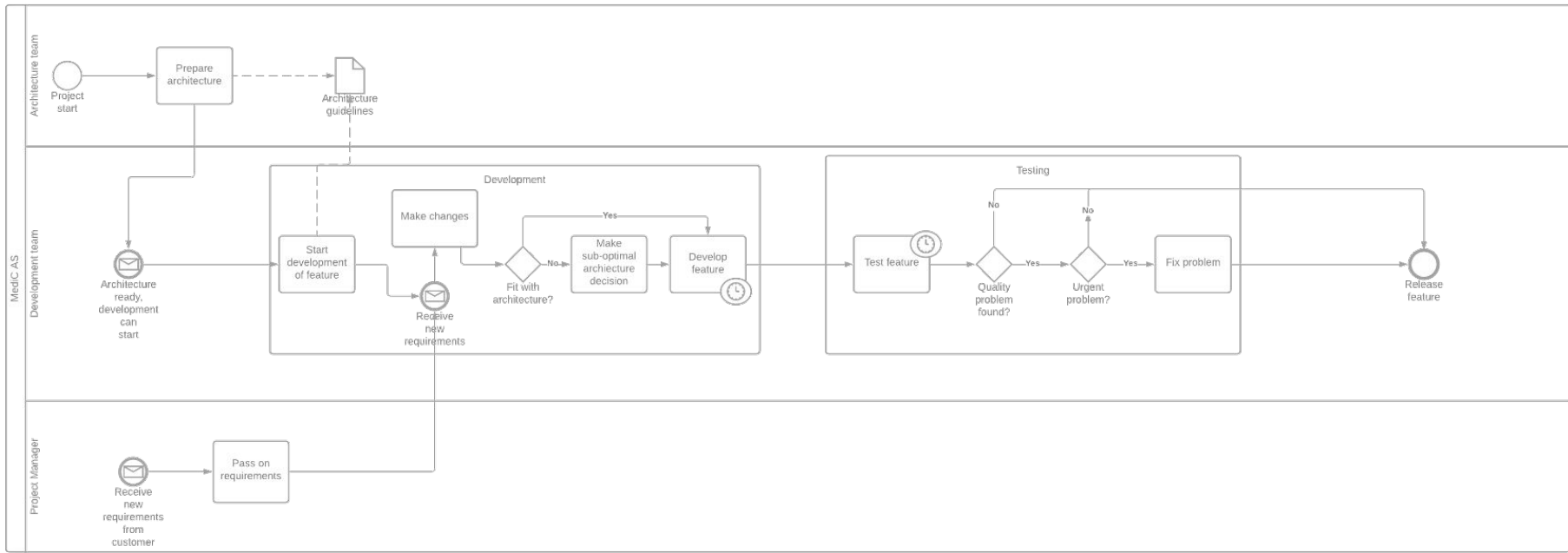
- **Resistance to change:** is often a challenge when implementing change. People usually like the old and familiar and are resistant to new and uncertain changes.
- **It takes time to implement changes:** An already high workload while implementing significant change in the whole company may be a challenge. Do people have time to change the way they work and as the whole company is subject to this change? Will it be done in big swoop or incrementally? Both approaches lead to their own set of challenges.
- **Losing trust and motivation:** It is easy to be invisible in a large crowd. When scaling up, organizations must continue to build around motivated people and give them trust, but it is easy to think that the more people, the higher the need for control.
- **Coordination:** The more people working towards a common goal, the more coordination is needed. It is difficult to get an overview of what everyone is working on, often resulting in spending resources on something another person is already doing.
- **Misalignment:** Misalignment between employees while implementing huge change. Some may be more enthusiastic than others which can create or widen rifts in opinion and willingness to change.
- **Lack of coaching/training:** in agile methods can lead to different interpretations of agile methods and how to implement them, resulting in asynchronous expectations and difficulties in communication.
- **Lack of management:** Lack of management participation in the administration and planning of company-wide change can result in differences in implementation and approach while also making the bottom layers of the workforce less willing to change.
- Other:
 - Lack of investments - physical spaces not adequate
 - Risk of falling back to the old practices of working
 - Inconsistent processes and practices across teams (they will have their routines and it will be hard to standardize the practices all over the teams)
 - Lack of skills with agile methodologies
 - Lack of customer availability

Task 2 (20%)

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

Task 2 (20%)

MediC experiences some challenges when scaling up. Architects are part of a special **architecture team** and **prepare the whole architecture upfront** without adjusting the architecture later. During development, **developers find issues** that were not considered by the architects, and they need to implement features in a different way from what is described in the initial architecture models. The architects become upset about developers not following the architecture, and the developers do not trust the architects' guidelines. When the teams need to take care of the new information found during implementation, they **take sub-optimal architecture decisions**, because they do not involve the architects. **The project manager** is in dialogue with the customer and **passes new requirements to the development teams continuously**. The development teams **implement features**, and the **code is reviewed** and **tested** just before release. However, it is difficult and time consuming to fix all the quality problems that are found, **only the very urgent ones are fixed**. The **release is delivered anyway**. During the next iterations, there is no budget to fix previously found quality problems, so they stay in the system. The project manager expects the teams to proceed to deliver new features fast, but now the teams seem to get stuck fixing bugs and changing all parts of the software for any small new changes that are requested.



Task 3 (20%)

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:

Task 3 (20%)

- *Technical debt is a design or implementation construct that is expedient in the short term, but makes future change more costly*
- *Social debt is the presence of sub-optimality in the development community, which causes a negative effect*
- *Process debt is the occurrence of sub-optimal process design*

How are they related?

Task 3 (20%)

Social debt

- *developers does not trust the architectural guidelines*
- *architects become upset when developers does not follow architecture guidelines*

leading to...

Technical debt

- *developers making sub-optimal architecture decisions*

Task 3 (20%)

Debt ID	Short description	Type of debt	Consequences	Other debts that are affected
Debt 1	No communication between developers and architects	Social	Developers end up making sub-optimal architecture decisions	Technical
Debt 2	Waterfall approach to architecture	Process	Architecture does not respond to new information found in implementation	Technical, social
Debt 3	The code is reviewed and tested just before release	Process	Leads to bugs not being fixed due to time limitations	Technical
Debt 4	Project manager expect developers to deliver fast, but developers gets stuck fixing bugs	Social	High stress for developers leading to poorer code	Technical
Debt 5	Sub-optimal architecture	Technical	The system is difficult to maintain and change	Social

Task 4 (30%)

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.

The purpose of Lean is to optimize the process in all parts of an organization to achieve a common goal. It is known for eliminating waste and can overall increase productivity and morale. Based on this, it could be a fitting approach for MediC AS.

Avoiding waste:

Find a balance between upfront and emergent design. Investing too much time into architecture and design of software products upfront can be wasteful.

A useful Lean technique is “**Just-in-Time**” which essentially means “Do not produce it before somebody demands it”. The technique is useful for reducing waste, since only what the customer actually wants is produced. I propose this technique since the probability of an architectural decision to be wrong decreases over time. This is because more information is available and more experience is gathered. The architecture should be made just in time, if not; delay it. This could help MediC AS reduce their social debt involving developers and architects. The architecture should then be more flexible to accommodate new requirements and issues found by developers.

Supporting change:

- Do shorter release cycles – you will have faster feedback. Therefore, you will have time to fix things based on the feedback and not build on top of them until it is too late and the feedback is not that helpful.
- Continuous Integration – will help the team to scale up as described above while monitoring the changes.
- Develop unit and integration tests – developers will be sure that their new features are not breaking anything covered by those tests in the system. This way they will have better quality of the code.

Flow:

Improve the inter-team communication – introduce Slack as a tool that will help the communication between teams as well as between members of the same team. This way everything will be more transparent and honest. This will also help the Social Debt that has been accumulated in MediC.

Satisfying the customer:

- Instead of planning too much in the beginning, include the customer as much as possible in the process. So that you will get more information.
- Deliver more frequently and smaller increments - so that you get feedback and adjust the future decisions based on that. This way the architects will be adapting to a more balanced design and not do everything upfront. Because some requirements might not be needed anymore or are not understood correctly. The more information you get the better the decisions that each team will make will be.

Visualization:

Tracking the development progress – introduce Jira to keep track of the:
Visions/Objectives, Plans and milestones, Tasks, Metrics, Timing or technical problems

Task 4

The purpose of Lean is to optimize the process in all parts of an organization to achieve a common goal. It is known for eliminating waste and can overall increase productivity and morale. Based on this, it could be a fitting approach for MediC AS.

Avoiding waste:

Find a balance between upfront and emergent design. Investing too much time into architecture and design of software products upfront can be wasteful.

A useful Lean technique is “**Just-in-Time**” which essentially means “Do not produce it before somebody demands it”. The technique is useful for reducing waste, since only what the customer actually wants is produced. I propose this technique since the probability of an architectural decision to be wrong decreases over time. This is because more information is available and more experience is gathered. The architecture should be made just in time, if not; delay it. This could help MediC AS reduce their social debt involving developers and architects. The architecture should then be more flexible to accommodate new requirements and issues found by developers.

Task 4 (30%)

GQM is no longer a part of the course curriculum.

But - metrics/measures are still important!

Task 4 (30%)

Goal - Improve the communication between the teams

Metrics

- *No. of changes made by the developer teams without consulting architects*
 - *Data collected by project manager through daily reports*
- *Developer/architect satisfaction of collaboration*
 - *Data collected by project manager through surveys with likert scales (1-5)*

Task 4 (30%)

Goal - Reduce the amounts of defects

Metrics

- *Density of defects (Number of defects / Size of system)*
- *No. defects found by tests before/after process changes introduced*
 - *Project Manager responsible for retrieving data from tools*

Task 5 (15%)

Describe how using microservices and the cloud could help to improve the situation described previously by the owner, but also what they need to be aware of.

Assumption: It seems like the owner has a rough idea of what microservices and the cloud are. I will therefore skip a detailed explanation and the difference between types of communication.

A problem with our current process is that one big chunk of software is made all at once. This is difficult to test, and we are quickly accumulating technical debt because of it. Microservices enable higher abstraction and allow us to focus on creating smaller and more manageable services. This would also fit in with our new lean approach and techniques like Just-in-time, as well as reducing waste. Instead of having our architects designing the entire architecture upfront, they can now focus on designing a more manageable amount, and focus on doing it right. If the developers find issues that are not described in the initial architectural model, they only need to change the relevant architecture. This supports the lean principles “supporting change” and “reducing waste” which we should be working towards. Microservices can also be easier to test since the test environment is more isolated. In the worst case, we find out the microservice does not work, it won't necessarily affect the others. Furthermore, microservices can enable our teams to work more independently, since they can focus on one or few microservice each without dependencies wasting time. It also allows more frequent deliveries - if one service is ready to be deployed it can be without affecting the other services.

The downsides to microservices is that it can be difficult to scale up. The higher the number of microservices there are the more challenging it is to manage them. The software can turn into an architecture with high complexity that can be difficult to have an overview over. Microservices can also bring longer cycles as one has to test, build and release. There is also a possibility that downtimes within one of few services may affect the entire system if the dependencies are not loosely coupled enough.

Moving to the cloud could add more benefits to the decision. The cloud allows even more abstraction, which can make the system more manageable. The cloud also provides a high number of opportunities to choose your tech stack since the cloud does many things for us (without us having to care about it). One example of this is automation. A problem with microservices was scaling. Some of these problems are solved by the cloud. You only pay for what you need, and can therefore scale up and down as needed. This is more difficult when having everything stored locally. The cloud is also known for being more resilient and generally failsafe.

This sounds great from a technical perspective, however, as an owner that is managing economic resources I should tell you that the process can be costly as it requires much work. The customer is not likely to notice much difference either. Having someone with experience in the cloud is advisable, as the change can be challenging. You either train someone from your company first, get outside consulting service, or get the Cloud vendor consultants. You mentioned that the chief architect suggested this. If he/she is motivated to deploy the change and have some experience with this, I would absolutely consider it.

Questions?