

Arbeidskrav assessment and feedback guidelines – PG3402 Microservices

The purpose of this document is to provide guidelines for how the arbeidskrav is assessed, as well as some guidance for peer review between students and student groups.

The goal of the arbeidskrav is to provide a foundation for the project, detailing the main topic and discussing the most important and interesting elements.

Overall, the submission should be clear, concise, and focused on those important and interesting elements.

Overview – description of the project – about half a page

Does the submission clearly describe the domain and the topic that it aims to cover. The overview is restricted to half a page. The idea is that elevator pitches are short descriptions of the context and the key element of novelty that the project brings.

Think of it as an answer to the following questions:

“What does the project do that is new, interesting, better suited to a particular environment or context?”

In short: “Why should the listener be interested in more information about this?”

The focus in terms of assessment and feedback is on:

Clarity – the overview should clearly explain the domain and what is new.

Brevity – the overview should be brief. An elevator pitch is usually only a couple of minutes long. The goal is to get interest for the more detailed presentation and to create a framework to guide the requirements process and the product backlog.

Focus - due to the need for the submission to be clear and short, it is important that the overview focuses on the most important and interesting components that the project adds.

When giving feedback on this aspect, consider what would make the description clearer, and assess how interesting and novel the proposed addition is.

Priority user stories – 2-3 user stories that are central to your system

The user stories describe the functioning of a system, from the perspective of the user. The user stories focus on the goals the system is meant to achieve, and illustrate how the system will deliver value to the customer.

The user stories can focus on those aspects identified in the overview as the essential topic, the areas where the system brings novelty.

These user stories are also selected to best show that added value. What is essential for the system to demonstrate that the idea presented in the overview is feasible, and can be accomplished in the available time, with the available resources and skills.

When giving feedback on the selected user stories, focus on which of the stories are the most interesting and relevant, and assess if they capture the ideas presented in the overview.

Requirements – a few requirements that flow from the user stories above

The requirements showcase the same user stories as presented in the previous section, but from the perspective of the system itself.

How is the system expected to address those user stories?

What kinds of data sources and data processing is needed in the system, in order to accomplish the stated goals?

In terms of feedback: focus on how the data is processed and obtained, and how the system can be designed to encourage a multi-service approach. In particular, focus on how the key user stories, showing more added value for the user, are going to be implemented.

Architecture overview – a small diagram of how your system will be structured

The idea is to set up the arbeidskrav for beginning the project itself. The architecture should clearly identify needed services (at least 3, since both synchronous and asynchronous communication approaches are required).

The architecture should also focus on the key functionality and key user stories and requirements. The idea is to show how the system can be designed in order to deliver the functionality.

Initial planning and backlog

The project planning should include:

MVP: minimum viable product. What is the least amount of functionality that illustrates the key functionality discussed in the overview?

Think of this as the equivalent of an E: what is the minimum required to show that the user stories are, at the very least, achievable or meaningful? These should show some implementation of the required functionality, and should show some added value: data should be processed somehow, analyzed, rather than just delivered.

For example, a basic backend with a few services that still show the functionality.

Likely scenario: What are the items of functionality that would best showcase the functionality in the overview? Additional features are added, some kind of frontend that allows the assessor to check your user stories. The user stories should be planned for, and some means of showing that they are met should be included.

Optimistic scenario: What additional requirements would showcase more advanced skills, allow new functionality to be investigated and delivered?

This is a “best case scenario” where, if there is time, more advanced features can be implemented. I would include here anything that has to do with external data sources (since the final project should be a showcase that runs even without additional data sources).