

Assignment 1

Tore Kjelds (tokj@itu.dk)

September 14, 2022

[Link to Github](#)

C#

```
int GreaterCount<T, U>(IEnumerable<T> items, T x)
    where T : IComparable<T>;

int GreaterCount<T, U>(IEnumerable<T> items, T x)
    where T : U
    where U : IComparable<U>;
```

In the first example, T has to implement the IComparable interface. In the second example, T has to implement U, which implements IComparable. The first method, there is no certainty that the method can compare T and U, since only T has the interface implemented. The second method, there is a certainty that the method can compare T and U, since they both implement the IComparable interface.

Software Engineering

Exercise 1

Exercise 2

CoronaPas would be a Interactive transaction-based application. I base this on the communication aspect between the server which verifies that the user has had vaccine. An argument could be made that this also falls in the category of systems of systems, since coronapas interacts with alot of different software systems.

Git would also be an Interactive transaction-based application. I base this on the transactions that occur when pulling and pushing.

Exercise 3

CoronaPas would be considered a *Generic product*, since it is offered on the open market to any one who wishes to download. An argument could be made that there are elements of *bespoke* software, since it was commissioned by another company.

Git Would be considered a *Generic product*, since it is offered to any one, who wishes to use.

Exercise 4

Git

I would argue that git focusses equally on the quality attributes. Git needs to be dependable, as if the service would be down, the companies using it would be losing money since the people working would'nt be able to push their commits to the pipeline. Efficiency is also an important attribute for git, since the amount of individual transaction occurring are very large, it is important to handle them efficiently. Lastly, maintainability is also important for git, since the software needs to evolve with its users needs. **CoronaPas**

I would argue that the Corona pas application leans more heavily towards security, since the application handles sensitive information about the user, and that specific legislations has strict requirements when it comes to handling sensitive user data. It would also skew towards dependability since alot of our institutions and places require people to show coronapas, meaning unexpected downtime would hinder alot of normal day to day life. Of course Efficiency is also important, since most of the people in Denmark will be using the application, there will be a lot of transactions being made. **Insulin pump control**

The insulin pump, would focus more heavily on the quality attribute dependability, since the result of having an inadequate dose or too frequent or infrequent dosages could lead to dangerous medical outcomes.

Exercise 5

1. Gitlet was written to show how Git works, not as a fully fledged software product. Meaning that it does not require the additional work of having a meaningful architecture since it is a learning tool.
2. You could look through the source code and look for file dependencies and look at just the overall file structure. Through the dependencies you could get an overview of the hierarchy.
3. Firstly, git and gitlet is written in two different programming languages, c and javascript. Gitlets design is alot simpler, having it all contained in one file, while git has lots of different files and dependencies.
4. As gitlet was meant to be a simpler version of git, it focusses alot more on the acceptability product characteristic. Since it was designed to be more

understandable and usable for a larger demographic. Git on the other hand focusses more on maintainability by having a set architecture.

Exercise 6

Softwareproblemer skadede mere end 100 patienter på amerikansk hospital

1. The problem was, that if a doctor filled out a formular wrong the formular would go into an unkown queue, instead of giving an error message. Its a tricky situation because either this was intentional an design decision and the problem was caused by improper training when accesing the queues, or the problem was caused by improper error handling.
2. There are two solutions depending on the which one of the problems that we are dealing with. The first potential problem could be solved with better training, if it is true that the unkown queue is a feature and not a bug. Otherwise better error handling, to make sure that the formular is not able to sent out, unless it was filled out correctly.
3. Having the impetus focussed on the people using the system, instead of focussing on developing a system that is not so prone to error, would in my opinion be a better choice. So developing a system with better error handling seems the better choice.
- 4.

Kodefejl i Sundhedsplatformen: Fem patienter har fået forkert dosis medicin

1. The problem was caused by code changes, which created a bug in an integrated system.
2. The problem could have been caught by having better automated testing, and running those tests every single time a new code change happened.
3. The description of the solution in the article corresponds pretty well with the one propossed above.
- 4.