Assignment 1: Week 35

HJSK and MGAE

September 2017

Exercise 1

> In the following description, explain when the term account is used as an application domain concept and when as a solution domain concept:
>
> ”Assume you are developing an online system for managing bank accounts for mobile customers. A major design issue is how to provide access to the accounts when the customer cannot establish an online connection. One proposal is that accounts are made available on the mobile computer, even if the server is not up. In this case, the accounts show the amounts from the last connected session.”

The term Application domain specifies which area of a specific term that needs to be examined to solve a problem. In the text ”account” is used in two different ways. Account is used as an application domain from the start of *A major design* and ends at *An online connection* here the problem is discussed and no solution is provided. From the part starting with *One proposal* and ending with *Last connected session* account is being used in a solution domain, where a solution concerning the accounts is being discussed

Exercise 2

> A passenger aircraft is composed of several millions of individual parts and requires thousands of persons to assemble. A four-lane highway bridge is another example of complexity. The first version of Word for Windows, a word processor released by Microsoft in November 1989, required 55 person-years, resulted into 249,000 lines of source code, and was delivered 4 years late. Aircraft and highway bridges are usually delivered on time and below budget, whereas software is often not. Discuss what are, in your opinion, the differences between developing an aircraft, a bridge, and a word processor, which would cause this situation.

Whereas civil and aerospace engineering are older and well established fields, software is much more recent. Aircrafts and bridges have been made a thousand times before, but many software projects are unique which make their costs and deadlines difficult to estimate. A project can be nearly finished only to end up scrapped or delayed indefinitely due to the complexities of software.

Exercise 3

3.1

Explain the differences between software development and software engineering.

If we simplify the meaning of software development to programming off a checklist, software engineering is much more rationale-driven. To model a program, from the technical specifications to the architecture, problem-solving skills are required.

Software engineers need to understand the system they want to build before they can come up with an optimal solution, while software development just need programming. [Bruegge and Dutoit, 2010]

3.2

Describe the history behind the term 'software engineering'.

The term was coined in 1968. Software developers were not able to set concrete objectives and predict the resources necessary to carry them out. [Bruegge and Dutoit, 2010]

To improve the quality of software and customer satisfaction, some best practices were introduced which fall under software engineering.

3.3

Explain the expression 'there is no silver bullet' in relation to software systems.

The phrase is an analogy to werewolves and how a silver bullet can magically eliminate this monster. Brooks argues that the existence of a silver bullet in software development is unlikely to exist due to several complexities in this field. Firstly, the rate of which technology is advancing far outpaces the advancement of tools and practices in the hands of developers. The use of both high-level and OOP languages may have significantly increased software productivity, but fundamental elements of development continue to cause problems due to their nature.

A software entity is inherently complex because of the many states it can take, which makes it hard to understand and test in depth. The discipline is also considerably more abstract compared to other scientific fields due to the rules being made by people and not nature. Lastly, it has no physical structure which restricts the number of ways our mind can understand it. [Frederick P. Brooks, 1987]

Because of these reasons, there might not be a key or silver bullet to the complications of software development.

Github

Link to the technical delivery:

> https://github.itu.dk/hjsk/BDSA2017.Assignment01

References

Bernd Bruegge and Allen H. Dutoit. *Object-Oriented Software Engineering*. 3rd edition, 2010.

Jr. Frederick P. Brooks. *No Silver Bullet: Essence and Accidents of Software Engineering*, April 1987.