IT UNIVERSITY OF COPENHAGEN

BDSA 2014

Assignment 35.1

Anders Wind Steffensen - awis@itu.dk
Christopher Blundell - mail@itu.dk
Pierre Mandas - mail@itu.dk

September 2, 2014

0.1 Q1-2

A programming language is a notation for representing algorithms and data structures. List two advantages and two disadvantages of using a programming language as the sole notation throughout the development process.

Advantages:

> By only using a programming language, you can quickly produce a prototype than if you had to dedicate time to documentation and planning. This jumpstarts the iteration process which allows the developers to test, find bugs and improve where needed. Another advantage is that when working on small projects, simply producing the code is cost- and timeefficient.

Disadvantages:

> When not documenting, extendability and maintainability becomes much more difficult - especially if there isn't a sole developer as the code might not be easily readable.
>
> Another disadvantage is that you easily can lose the overview of your project, by for example not using UML-diagrams.
>
> By only writing code and not communicating with the customer, the program can easily become unsatisactory for the customer (no validation).

0.2 Q1-4

What is meant by "knowledge acquisition is not sequential"? Provide a concrete example of knowledge acquisition that illustrates this.

By referering to the book, knowlegde acquisition is not sequential means that you cannot just pour knowledge into ones head. You might have to reconsider the knowledge learned and all experiences might not catch on as well as others.

For example, when developing software and delivering a prototype to the costumer, one might experience that what was first understood by the technical specification was not what the customer actually wanted.

Therefore the knowledge gained was not correct and you might have to start over and therefore it is not sequential.

0.3 Q1-6

Specify which of these statements are functional requirements and which are nonfunctional requirements:

- "The TicketDistributor must enable a traveler to buy weekly passes."
  Functional.
- "The TicketDistributor must be written in Java."
  Nonfunctional.
- "The TicketDistributor must be easy to use."
  Nonfuctional.
- "The TicketDistributor must always be available."
  Functional.
- "The TicketDistributor must provide a phone number to call when it fails."
  Functional.

0.4 Q1-8

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.[1] A major design issue is how to provide access to the accounts when the customer cannot establish an online connection.[2] One proposal is that accounts are made available on the mobile computer, even if the server is not up.[3] In this case, the accounts show the amounts from the last connected session.[4]

(1) *Assume you are developing an online system for managing bank accounts for mobile customers*

> This sentence is in the application domain, because it does not go in into details with how the system needs to work, but explains system requirements.

(2) *A major design issue is how to provide access to the accounts when the customer cannot establish an online connection.*

> This is stillin the application domian because it explains problems that the users encounter even without the system.

(3) *One proposal is that accounts are made available on the mobile computer, even if the server is not up.*

> This sentence is in the solution domain. It goes into detail as with servers and how the systems can solve a specific problem.

(4) *In this case, the accounts show the amounts from the last connected session.*

> This sentence is clearly in the solution domain as it explains how a solution in the system can be created.