Lecture "Software Engineering SoSe 2023

Free University of Berlin, Institute of Computer Science, Software Engineering Group Lutz Prechelt, Linus Ververs, Oskar Besler, Tom-Hendrik Lübke, Nina Matthias

Exercise sheet 5 **Use cases** for 2023-05-22

Task 5-1Æ: Use cases

Learning Goal: Understand the concept of use cases; know criteria and guidelines for creating good use cases.

a) What is a *use case* in the context of software development (independent of UML) and what is its purpose?

b) Research *principles* for writing good use cases. Use at least two different sources (cite them, as always).

Note: The following keywords may be helpful in your research: *Writing (good/effective) use cases, Alistair Cockburn*.

1. First of all, take the "product view": What actually makes a good use case? Formulate at least three criteria that are specific to use cases (criteria such as "understandable" or "complete" do not count).
2. Now switch to the "process view": How do you write such a good use case? Look for guidelines on the writing process and explain at least the core message of at least three of these guidelines in your own words.

c) What is the difference between a *scenario* and a *use case*?

Task 5-2Æ: UML use case diagrams

Learning Goal: Be able to recognize when and for what UML use case diagrams are appropriate.

a) What is a use case diagram as opposed to a use case? What are the advantages and disadvantages of using UML use case diagrams? Name at least two of each.

b) What does the *Actor* model element represent in a use case diagram?

c) What other model elements does UML know for use case diagrams besides the actor and the use case itself? Name and explain three.

d) What types of relationship can two use cases have to each other in a UML use case diagram? Name and compare at least two.

Task 5-3: Develop use case and UML diagram

Learning objective: To be able to formulate use cases in a structured way and to relate them in UML diagrams.

a) Based on one of your requirements (Exercise **4-2**), formulate a use case for your software. Do not choose a trivial use case, i.e. its success scenario should contain more than three steps and it should have at least one extension.

First read the source: http://www.bredemeyer.com/pdf_files/use_case.pdf
Create your use case according to the template presented in the source.[1].

[1] Remember that a use case should be written in such a way that it can be understood by all project participants. Therefore, please refrain from using the constructs suggested by Coleman, which are based on programming languages, such as IF-THEN-ELSE, REPEAT-UNTIL and IN PARALLEL. These inflate the complexity of use cases far too much, just when the main purpose of use cases should be precisely the technical reduction of complexity.

b) Create a UML use case diagram that shows the relationships of at least three use cases of your software to be developed. (You do not need to formulate these use cases).

Add your formulated use case and use case diagram to your wiki page in addition to the normal electronic submission.