Exam problems 2

10/12-14

Group F

Alexander Badju, Fredrik Helander, Jonathan Klingberg, Jonathan Knorn, David Lundberg & Niklas Sjöberg

Problem 1: Specifying technical interfaces [Lau:5]
Proposition: To understand message descriptions and protocols it is important to know the semantics of the communication. To achieve this one can use task descriptions or essential use cases to describe what each party does at each end of the communication.
Reason: The protocol is often the most complex part of a technical interface. When specifying a technical interface it is important to start prototyping the communication early.
Correct answer: ***B*** (Both the proposition and the reason are true but the reason *does not* explain the proposition)
Motivation: The proposition is a true statement as essential use cases and task descriptions can be used to describe the domain level. The reason is also a true statement, however, it does not answer the proposition.
Reference: Lau:5 p. 214-216
Learning objective: 1.1.1 1.1.4 1.2.3
Main responsible: Fredrik Helander

Problem 2: Usability tests vs Heuristic evaluations [Lau:6]
Proposition: A heuristic evaluation is better than usability tests at detecting problems with software usability.
Reason: It is difficult to create usability tests that cover all possible problem scenarios, which an expert is more likely to detect.
Correct answer: ***E*** (Both the proposition and the reason are false)
Motivation: Unlike code reviews carried out by programming experts, usability experts often don't understand the user as well as a programming expert understands a computer. The usability expert therefore might "detect" problems that are unrealistic for the real users, and simultaneously miss problems that the real users would face often. Usability tests are less likely to have this problem as they are performed on real users (if done correctly). The reason is partly true, as it might be difficult to create completely covering usability tests, yet there is nothing that suggests an expert would fare better at it.

Reference: Lau:6, pp. 254-255
Learning objective: 1.1.1, 1.1.4
Main responsible: David Lundberg

Problem 3: Specify response time in multi-user systems [Lau:6]
Proposition: When specifying the end-user response time for an internet based system it is good practice to specify a maximum time limit.
Reason: The customer who buys the server system wants to ensure that the end users experience a low response time.
Correct answer: ***D*** (The proposition is false, but the reason is a true statement)
Motivation: Max time limit should not be specified for systems where other parties such as Internet are involved. Since the server developers aren't responsible for internet response time it's better to specify response time limits for the server system assuming that internet has zero delay and then specify another requirement for the internet provider separately. The buyer definitely wants the users to experience low response times but that's not an answer to the proposition.
Reference: Lau: p. 244-246
Learning objective: 1.1.1, 1.1.4
Main responsible: Jonathan Klingberg

Problem 4: Issues with ambiguity [Lau:9]
Proposition: When writing requirements it is good praxis to explain the purpose of each requirement in domain related terms, to protect against ambiguity.
Reason: Ambiguity problems arise when developers think they understand a requirement but in reality the interpretation is not cohesive with the customers view of the situation.
Correct answer: ***B*** (Both the proposition and the reason are true but the reason *does not* explain the proposition)
Motivation: The proposition is true, it is good to explain requirements in domain related terms to safeguard against ambiguity. The reason is true as well, but it doesn't explain the proposition since the reason just states the fact that developers can meet ambiguity issues when working with requirements.
Reference: Chapter 9, Checking and Validation. *9.1 Quality criteria for a specification-Unambiguous* p.376
Learning objective: 1.1.1, 1.1.4
Main responsible: Alexander Badju

Problem 5: Numerical Assignment [PRIO]
Proposition: The numerical assignment technique shows which requirements are more important than others, and also how much more important they are.
Reason: The numerical assignment technique first groups the requirements into different priority groups, and then rank them against each other within each group.
Correct answer: ***E*** (both the proposition and the reason are false)
Motivation: The proposition is false since the numerical assignment technique prioritizes the requirements in an ordinal scale. The reason is also false, as requirements in the numerical assignment technique are only divided into priority groups, and not ranked against each other within the group. The reason is actually a combination of numerical assignment and ranking, which is used in the Planning game in Extreme programming.
Reference: PRIO chapter 4.4
Learning objective: 1.1.3, 1.1.4
Main responsible: Niklas Sjöberg

Problem 6: Highly dependent requirements [INTDEP]
Proposition: To reduce the number of pairwise comparisons when examining dependencies, it can be a good idea to identify the "most dependent" requirements and then only analyze the pairwise assessments from these.
Reason: Since highly dependent requirements often affect many other requirements this is a good method to speed up the process of identifying dependencies.
Correct answer: ***A*** (Both the proposition and the reason are correct statements,
AND the reason explains the proposition in a correct way.)
Motivation: In the study, between 67% and 79% of the interdependencies were covered by identifying the 20% "most dependent" requirements and performing the assessment only with these. The conclusion was that this could be a good method to find the majority of the dependencies.
Reference: INTDEP chapter 3.4
Learning objective: 1.1.4, 1.1.7
Main responsible: Jonathan Knorn