

Lauesen Chapter 5,7

Proposition:

Embedded trace information gives backwards traceability from code to requirements.

Reason:

Giving bits of codes and design artefacts a requirements ID makes the requirements fulfillment traceable.

Answer:

A - Both the proposition and the reason are correct statements, and the reason explains the proposition in a correct way.

Motivation:

As you have the building parts to a system each refer to which requirement they fix you can then use that data to validate that all the requirements are taken care of and from where, since you can list all the features the implementations deals with you can there check that every requirement is included in the list.

Reference:

Lauesen: Chapter 7, page 314.

Learning Objectives:

1.1.3, 1.2.4

Main Responsible:

Henrik Persson

Lauesen Chapter 6

Proposition:

The QUPER model aims to make a fast estimation of the dependencies between different quality indicators for a set of quality requirements.

Reason:

By choosing good quality indicators the possibility of making better decisions for release planning increases. How many quality indicators that should be managed depends partly on a balance between the effort to obtain the information to identify a quality indicator and the benefit of this indicator.

Answer: D - The proposition is an incorrect statement, but the reason is correct.

Motivation:

Proposition is false: The QUPER model could be used to avoid making complete predictions of otherwise inherently difficult relations between cost, benefit and quality when considering a set of quality requirements. In short, the model aims to give a broader overview and understanding of the quality requirements and making better release planning decisions. This model involves analysis of the concurrents and estimations about how much an increase in quality would cost/ which benefit this quality actually would bring. To manage the dependencies between different quality indicators the breakpoints could be compared, but it is hard to try to estimate all existing dependencies. Often experts are needed for the crucial cases.

Reason is true: By choosing the quality indicators carefully you can find special cases that can improve the possibility of making reasonable predictions, and find a good balance between competing quality attributes. The number of quality indicators needed depends both on the effort obtaining information about the indicators and the benefit this indicator brings. Moreover, the amount depends on which the most-strategic use cases, domain or product it considers.

Reference:

QUPER 87, 89-91

Learning Objectives:

1.1.3, 1.1.4, 1.2.1, 1.3.5

Main Responsible:

Hanna Andréason

Lauesen Chapter 9

Proposition:

Performing a CRUD check is an effective way of finding inconsistencies and missing operations in a specification.

Reason:

If it is not specified how the entity classes in a data model are going to be created and deleted, and how their values are going to be read and updated, the specification can not be considered complete.

Answer:

A - Both the proposition and the reason are correct statements, and the reason explains the proposition in a correct way.

Motivation:

During a CRUD check it is checked whether every entity class in the data model can be created and deleted, and also whether their values can be read and updated. These are very basic functionalities of the entities of a specification and should be specified. This can for example be accomplished by making a CRUD matrix by looking at the events, functions, and user tasks of a specification.

Reference:

Lauesen: Chapter 9, pages 386-388.

Learning Objectives:

1.1.1, 1.3.3

Main Responsible:

Tim Borglund

MDRE, PRIO, PR

Proposition:

The 100-dollar prioritization technique should only be used once per a set of requirements.

Reason:

By using it several times a stakeholder might prioritize their favorite to a much higher priority based on previous results.

Answer:

A - Both the proposition and the reason are correct statements, and the reason explains the proposition in a correct way.

Motivation:

Based on previous result one stakeholder could put all their money on one requirement if their favorite requirement didn't make it to the top priority before and this might influence the result heavily. At the same time they might not give any money to requirements that will get much money anyway.

Reference:

PRIO 4.4.2 page 32

Learning Objectives:

1.1.1, 1.1.3, 1.2.1, 1.2.5

Main Responsible:

Erica Karlsson

INTDEP, AGRE

Proposition:

The cost of implementing a change request, in Agile Requirements Engineering, is lower than in traditional software development.

Reason:

Two requirements can have more than one type of relationship between them in an interdependency.

Answer:

B - The proposition and reason are correct but does not explain one another.

Motivation:

In agile requirements engineering the requirements are constantly validated with the customer and with that the risk of a major change in the requirements are minimized. So the cost is decreased compared to traditional software development due to the lower amounts of big changes. Two interdependent requirements can have different relationship. For instance one requirement needs the other requirement to function and the other requirement increases the value of the first requirement for the customer. Depending on dependencies between requirements the cost of addressing a change is affected but it doesn't explain the difference in agile and traditional methods.

Reference:

INTDEP 2.1 page 105 , AGRE page 116

Learning Objectives:

1.1.1, 1.1.3, 1.1.5, 1.1.7, 1.3.5

Main Responsible:

Elin Hanstorp