DAT230/DAT231/DIT276 Requirements Engineering

Exam

Monday, October 29th, 2018

Examiner
Eric Knauss +46 31 772 10 80
Contact person during exam
Eric Knauss +46 31 772 10 80

Allowed tools / material

- Pen/pencil
- Eraser
- English dictionary

General information

Numbers within parentheses show the maximal points awarded for each question. Maximal points can be given if:

- The answer is correct.
- The presentation of the answer is readable and clear.
- The answer is given in English.

Each question must be answered on a new sheet of paper. Hence, the answers to Question 1 and 2 must NOT be written on the same sheet of paper. You are NOT allowed to write answers to questions on the backside of a sheet of paper!

Grading

The grades on this exam are based on your total score on the questions. For Chalmers students:
0 – 39 points: Fail
40 – 55 points: 3
56 – 67 points: 4
68 – 80 points: 5

For GU students:
0 – 39 points: Fail
40 – 67 points: G (Pass)
68 – 80 points: VG (Pass with distinction)

Review

Will be announced on Ping Pong

© Eric Knauss, 2018
Institutionen för Data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA & GÖTEBORGS UNIVERSITET

Part 1 – Multiple Choice (30p)

Question 1: Multiple-choice problems (30p)

This part consists of multiple-choice problems. These problems consist of pairs of *propositions* and *reasons*. For each problem, you shall answer by using one of the following answers:

A Both the proposition and the reason are correct statements. In addition, the reason explains the proposition in a correct way, i.e. the reason explains why the proposition is correct.

B Both the proposition and the reason are correct statements, but the reason does not explain the proposition.

C The proposition is a true statement, but the reason is false.

D The proposition is false, but the reason is a true statement.

E Both the proposition and the reason are false.

Correctly answered problems give **1 point**, while incorrect or missing answers give 0 points, regardless if you are partially correct in your answer!

	Proposition	*Reason*	*Answer* $A\|B\|C\|D\|E$
1) **(1p)**	Requirements engineering is a systematic approach to reduce the likelihood to develop the wrong system, e.g. one that does not solve the problem of a user.	A requirement is a condition or capability needed by a user to solve a problem or achieve an objective.	A
2) **(1p)**	With the term "tacit requirements" we refer to requirements that should not be part of a specification.	Tacit knowledge is knowledge that a stakeholder takes for granted or for other reasons does not explicitly mention.	D
3) **(1p)**	Symmetry of ignorance is a challenge of requirements elicitation that makes it difficult to map a problem to a solution.	Symmetry of ignorance refers to the fact that requirements engineers should ignore solutions and focus on problems instead.	C
4) **(1p)**	When a customer is inexperienced in a domain and a supplier is experienced a domain, the main goal of requirements engineering should be to reduce cost.	Since the supplier is more experienced, elicitation of customer requirements does not provide value and the potential to satisfy the customer can be maximized if the supplier is just providing their solution directly.	E
5) **(1p)**	Identification of stakeholders is an important part of requirements elicitation.	Stakeholder needs must be analyzed and refined. Without that, they are considered to be "raw requirements", not necessarily being requirements according to definition.	B

6) **(1p)**	If a stakeholder appears to talk about too detailed requirements, we should ask "why" questions.	"Why" questions raise the level of abstraction and allow to discuss the underlying problem.	A
7) **(1p)**	If a stakeholder uses unexpected or ambiguous wording during an interview, this should be ignored during the interview.	Missing information from an interview can easily be provided during analysis.	E
8) **(1p)**	It is crucial to satisfy some issues from each stakeholder group when processing the raised issues from focus groups.	All stakeholder groups have to feel that they get something.	A
9) **(1p)**	Questionnaires are a good elicitation technique in the early phases of a project.	They are easy to construct and easy to interpret.	E (D)
10) **(1p)**	Verbose requirements should be avoided.	Verbose requirements are hard to read and often not thought through.	A
11) **(1p)**	Component names, database fields, and similar details in a requirement indicate that it is problem-based.	Considerations of solutions and SW design should be avoided when specifying requirements.	D
12) **(1p)**	Context diagrams are useful for defining the project scope.	Context diagrams show the system under consideration in relation to other external entities such as other systems, datastores, or stakeholders. By showing critical interfaces it can define the context for use cases and task descriptions.	A
13) **(1p)**	Feature requirements can lead to a false sense of security.	Most stakeholders think that they understand feature requirements, but often, they misinterpret them.	C
14) **(1p)**	Feature requirements are useful when managing requirements for a large system.	Feature requirements are relatively short and independent descriptions of product functions and related data. It is easy to store them in databases and they can have many administrative attributes.	A
15) **(1p)**	Task descriptions describe the interaction between a user and a system in detail.	The main scenario of a task descriptions usually alternates steps that a user executes with responses or actions that the system executes.	E
16) **(1p)**	"Enter guest name" is a good task description.	Good task descriptions are closed (they end with a reaching a goal, with a pleasant feeling) and describe one session (a small set of related tasks).	D
17) **(1p)**	All non-functional requirements are quality requirements.	A non-functional requirement is either an attribute or a constraint of a system.	D
18) **(1p)**	Quality requirements are an exception to the rule that all requirements should be verifiable.	Since quality requirements are cross-cutting, criteria about whether they are met is better related only to the functional requirements they affect.	E

19) **(1p)**	The quality grid is useful to describe trade-offs between different quality requirements.	The quality grid provides a matrix of the relevant quality requirements. This allows to define trade-offs between two quality requirements in the cells.	E
20) **(1p)**	When relying on "Open Target" when specifying quality requirements, the customer indicates that the degree of fulfillment of this quality requirement is not important to them.	It is often difficult to decide the value needed for some quality attribute. "Open Target" gives the supplier the responsibility to investigate a suitable target value.	D
21) **(1p)**	"Combinatorial creativity" combines both exploratory and transformational creativity.	A combinatorial creativity technique is generally a two-phase approach, where first, ideas are generated based on an exploratory technique, before these ideas are then discussed related to a transformed solution space.	E
22) **(1p)**	Requirements validation relates to the question "Are we building the right system".	Requirements validation is defined as the process of checking whether documented requirements match the stakeholders' needs.	A
23) **(1p)**	Since it is important that all requirements are verifiable, it is implied that all requirements must be atomic.	If requirements were not clearly identifiable and would not allow to reason about them individually, it would be impossible to state whether they are met.	A
24) **(1p)**	Consistency checks are done to compare different parts of a requirements specification, looking for inconsistencies.	CRUD checks are done to assess whether each entity in the data model is created, read, updated, and deleted, by an event, or a user task.	B
25) **(1p)**	A prototype is only useful for verification, but cannot be used for validation.	A prototype only allows to answer whether we are building "the system correctly", but not whether we are "building the right system".	E
26) **(1p)**	Relying on face-to-face communication instead of written specifications can lead to incomplete and wrong understanding of requirements.	Agile projects are not always able to achieve high-quality interaction with the customer organization.	A (B)
27) **(1p)**	Agile methods promise to mitigate communication gaps in requirements engineering.	Agile RE promotes regular interaction with customer and between teams.	A
28) **(1p)**	In large-scale agile development, most requirements analysis takes place on program level.	In large-scale agile, requirements analysis is spread out over portfolio, program, and team level activities.	B (D)
29) **(1p)**	It is important to elaborate and document each user story in detail.	A user story is the main mechanism to specify requirements in agile methods.	D (E)
30) **(1p)**	Requirements engineering relates to the trade-off between common product goals and autonomy of agile teams.	The fine balance between decisions on portfolio, program and team level must be maintained through strong communication of requirements-related knowledge.	A

Part 2 – Applying RE Knowledge (10p)

Question 2. Usability requirements and risk (10p)
Ms. Taylor has been in contact with Requirements Engineered Ltd., a company specialized in Requirements Engineering and bespoke development, to realize one of her project ideas. The company has worked for several months to come up with a specification. Now, Ms. Taylor would like to move to the next step and sign a contract for development of the resulting specification. She has asked you, as an expert in Requirements Engineering, to give her a second opinion on the following requirements.

R1	At most 1 of 5 novices shall encounter critical problems during tasks Q and R. At most 5 medium problems shall be encountered in total.
R2	Novice users shall perform tasks Q and R in 15 minutes. Experienced users shall perform tasks Q, R, and S in 2 minutes.
R3	Recording breakfast shall be possible with 5 keystrokes per guest. No mouse.
R4	80% of users shall find system easy to learn. 60% shall recommend system to others.
R5	Show 5 users 10 common error messages, e.g. Amount too large. Ask for the cause. 80% of the answers shall be correct.

Which issues and risks do you see in these requirements? Name one issue/risk per requirement, argue whether it applies to supplier or customer, and motivate why you think it is problematic/risky.

Solution: Generally, the risk for the customer is that they get what is specified, but not what they need. The risk for the supplier is that a requirement is hard to satisfy or only at excessive cost.
R1 does mainly pose risks for the supplier. How many novices are needed to test the delivery? How to classify critical or medium problems?
R2 does only pose risks for the supplier, same reasons as above.
R3 does mainly create risks for the customer; this is easy to test, but a software can still be lacking usability.
R4 has a lot of risk for both, a bit more to the supplier: How to test it without deploying it first? Expensive rework might be the result.
R5 big risk to customer, since under specified. Which users? Which error messages? Is 80% sufficient for usability? Again, a software might be bad, even though it fulfills this requirement.

Remark: Since the task is asking for issues, and problems in writing the requirement can also create a risk of misunderstanding, we were lenient with respect to alternative solutions.

Part 3 – Essays (40p)

For each topic below, write an essay that includes all of the listed concepts within the given maximum number of pages. The essays are graded based on

(a) how well the topic is described using the concepts in the list under each topic (thus judging the general flow and argumentation), and
(b) how well the concepts in the list are described and/or exemplified in the essay (thus, points are given for each concept that is mentioned and explained in sufficient depth).

Please make an effort to write readable. Essays that are difficult to read or difficult to understand will render a deduction of received points. An unreadable essay will receive zero points! Each essay should start on a new sheet of paper.

Question 3: Topic: Completeness of Elicitation **(10p)**
Max 1 page!
List of concepts: 1 example of an early-phase elicitation technique, 1 example of a mid-phase elicitation technique, 1 example of a late-phase elicitation technique, tacit knowledge, effect of asking why

[2 points for each concept: [for elicitation techniques; 1 point for the technique and 1 point for why it is important for the specific elicitation phase]

- **Early elicitation technique**; any of the following: *Interviews*, *Group interviews/sessions*, *Observations*
- **Mid Elicitation technique**; any of the following: *Task demo, Questionnairs (survey), Brainstroming.*
- **Late Elicitation technique;** any of the following: *use cases/scanarios, modelling/data flow diagrams, prototyping*
- **Tacit knowledge:** *Unkonwn knowns. Can only be obtained through interaction and social networks.*
- **Effect of asking why***: identify goals or problems necessitating a requirement from a customer*

Question 4: Topic: Understanding Quality Requirements **(10p)**
Max 1 page!
List of concepts: Quality Grid, Planguage, Open Metric, Constraint, Attribute

[2 points for each concept: 1 point for what the concept is about and 1 point for an example given]

- **Quality grid**; maps quality factors to criticality. Helps suppliers and customers identiy critical quality requirements to focus on and at what stage of the system deployment i.e. operation (e.g., usability, integrity), revision (e.g., maintainability) or transition (e.g., portability, reusability).
- **Planguage**; allows for measurable, testable quality requoirments t be written. *Uses a set of keywords to guide the quantifcation of the quality criteira, for instance, for a quality requirment on hotel forecast speed, define* ***scale*** *(e.g. data volume or time length),* ***must*** *(minimum accepted criteria),* ***wish*** *(customer's ideal criteria), and* ***past*** *(current status relating to the quality criteria e.g, curently takes one hour)*
- **Open metric;** Deferred decision for a specific measurement criteria or indicator for a quality requirement. E.g., letting a supplier specify the accuracy of a forecast.

- **Constraint:** *constrains solution space beyond what is necessary for meeting specific functional, performance or quality requirements; could be physical, cultural, environmental etc. E.g., system data collection must meet GDPR guidelines, or organization requires that only Microsoft products are used for implementation, or even budget contraints [Student must give at least one concrete example of a constraint]*
- **Attribute***: specific performance or quality requirment. E.g., speed, space bounds, or usability requirements. For instance, ordering an item shall require no more than 5 clicks, or, page load shall be within 2s. [Student must give at least one concrete example of an attribute]*

Question 5: Checking and Validation **(10p)**
Max 1 page!
List of concepts: Contents check, Structure check, Consistency checks, Reviews, Tests
[2 points for each concept: 1 point for what the concept is about and 1 point for an example given]

- **Contents check**: inspection to check that the specification contains relevant sections such as project background, business goals, quality requirements, system boundaries etc.
- **Structure check**: checks the quality of the specification itself, e.g., does each requirement have an ID, are requirements verifiable, does each requirement have a purpose, etc.
- **Consistency checks:** checks consistency between different abstraction levels of specified requirements e.g., do data requirements exist for tasks or virtual windows involving data input. CRUD matrix is useful for this.
- **Reviews**: involves reviewing of all parts of the spec by customers and developers, goal-means analysis to ensure that critical issues are covered and that requirements are justified, and risk assessment by each party (developer and customer) to improve high risk areas.
- **Tests**: Could be simulation and walk throughs (e.g., walk through task descriptions), or prototype tests to check that requomets are meaningful, or pilot tests that may involve installation and operation to assess the cost and benefit.

Question 6: Quality Requirements in Agile Requirements Engineering (**10p**)
Max 1 page!
List of concepts: Portfolio level, program level, team level, agile estimation, architecture
[2 points for each concept]

- **Portfolio level**: high level requirements engineering that considers investment themes, e.g., a product portfolio consists of one or more similar products. Architecture is defined at this level and decisions made at this level feed into road mapping and release planning.
- **program level**: Includes product management and release management in terms of features provided by products. Also involves requirements analysis, e.g., assessment of quality requirements, which are often cross-cutting several functional requirements. Road mapping the product releases and prioritization of requirements. Often interferes with team autonomy.
- **team level**: Focus is on implementation of specific qualities, features, and components delivered by a team. Has three major roles involved in RE, these include product owner, agile coach, and developers/testers. Artifacts managed include user stories and

tasks. User stories provide measurable and testable requirements from a user's perspective; mostly used for functional requirements.

- **agile estimation**: source of requirements updates and measurement of effort. Techniques used include planning poker. Sometimes challenging due to changing requirements
- **Architecture**: High level design decision influenced by quality requirements. Defined at portfolio level.