## o4-SOFT-A5: Requirements and Specifications National Examination (3 hours duration)

#### **Notes:**

- 1. If a doubt exists as to the interpretation of any question, the candidate is urged to submit, with the answer sheets, a clear statement of any assumptions made.
- 2. This is an open-book exam with one book of the candidate's choice.
- 3. No calculator is permitted (nor is one needed).
- 4. Most questions are open-ended and require an answer in essay format. The candidate should write clearly and organize her or his answer with care.
- 5. This exam has 11 pages including this page and space for answers. Write your answers in the space provided. Do not be verbose.

**Part 1 (30 points):** For the following questions, a statement regarding software requirements engineering is made and a reason is provided for the statement. You will have to determine which of the following situations will apply with a brief rationale (explanation):

- A. Both the statement and the reason are true, and the reason is a valid explanation for the statement.
- B. Both the statement and the reason are true, but the reason does not explain the statement.
- C. The statement is true but the reason is false.
- D. The statement is false but the reason is valid in some situations.
- E. Both the statement and the reason are invalid (false).

This part has 10 questions, each of which is worth 3 points.

1. Statement: It is often better to wait and prioritize requirements at the end of the requirements gathering process. Reason: A subset of the requirements becomes clearer as the requirements gathering process proceeds.

2. Statement: Prioritization using an ordinal scale requires more work compared to prioritization using a ratio scale. Reason: When a ratio scale is used, mutual priority fractions also need to be elicited.

3. Statement: Good requirements engineering achieves the same high level of completeness for different types of requirements. Reason: Both trivial and non-trivial requirements should be specified with the same accuracy to make best use of spent effort.

4. Statement: If a prototype to a large degree fulfills the requirements then the verification of the implementation can be avoided. Reason: If the integrator limits its responsibility to prototypes of a small fraction of the subsystems then the customer's risk of verification can be reduced significantly.

5. Statement: If precedence and coupling is taken into account in release planning it is likely that the solution space becomes smaller. Reason: The number of possible release plans that fulfill the constraints will in general be greater if more constraints are introduced.

6. Statement: To replace large parts of the written communication between developers and customers with face-to-face communication can pose risks. Reason: If intensive communication cannot be achieved, the risk that the requirements engineering process leads to incorrect requirements may increase significantly.

7. Statement: Test-driven development can increase traceability. Reason: Test cases can capture requirements but do not link to design and code.

8. Statement: Screens and prototypes are more suitable when describing requirements at domain level compared to task descriptions. Reason: Task descriptions lack information about how actors achieve their goals.

9. Statement: Quality requirements are often more difficult to specify than data requirements. Reason: Quality requirements often affect more system components than data requirements and the specification of quality levels is often made in relation to sliding scales.

10. Statement: Focus groups can help make sure that each stakeholder gets something he/she wants from the product. Reason: The results of a focus group meeting include lists with each stakeholder's top wishes.

Part 2 (25 points): Requirements Elicitation. Describe the process of requirements elicitation using no more than two pages. Your essay should cover issues such as barriers to elicitation, things to elicit, the suitability of elicitation techniques, analysis of stakeholders, interviews, focus groups, and prototypes.



### May 2015 - 04-SOFT-A5

Part 3: Non-functional Requirements (25 points). Answer, in essay format, the following questions regarding non-functional requirements.

(i) What are non-functional requirements? (5 points)

(ii) How do non-functional requirements map to a use-case model? (5 points)

(iii) Suppose you are involved in developing software requirements for a life-critical avionics system. Describe three non-functional requirements (or software quality attributes) that this system should possess. You should describe each of the three requirements briefly. (15 points)

# Part 4: Excellent Requirements and Requirement Collections (20 points).

- 1. Excellent requirements are:
  - (i) Complete
  - (ii) Correct
  - (iii) Feasible
  - (iv) Necessary
  - (v) Prioritized
  - (vi) Unambiguous
  - (vii) Verifiable

Briefly explain each of the seven characteristics listed above (7 x 2 points = 14 points).

### May 2015 - **04-SOFT-A5**

- 2. Excellent requirements collections are:
  - Consistent
  - (i) (ii) Modifiable
  - (iii) Traceable

Briefly explain each of the three characteristics listed above (3 x 2 points = 6 points).