Software Engineering and Project Management - Written Exam

2023-03-09. **Duration:** 14:00 - 17:00

Start by reading all the questions, to see if anything is unclear. One of the teachers will visit the exam around 14:30 to clarify questions.

The exam has a total of 40 points and you need 21 points to pass the exam (extra points will be added afterwards, once the exam is already graded). The grade is on the scale U/G.

Answers must be written in English. Dictionaries may be used.

Start each numbered question on a new page. Please hand in the pages in the correct order.

For each question, I give the maximum points of the answer.

The format and extension of your answer should adapt to the type of questions:

- **Specific questions:** a concept, a list… the *optimal* answer is usually around a single paragraph (This does not include drawing).
- **Short questions:** explanations, reviews, comparisons… the *optimal* answer is usually a few (2 or 3) paragraphs (This does not include drawing).
- **Questions for reasoning:** your arguments are presented in a concise, well-reasoned way,… and written with your own words. The *optimal* answer is usually less than 2 pages. (This does not include drawing).

A checklist of common mistakes that cost points:

- Answer all **5** questions. A bad answer never gives less points than no answer.
- Read the question again after you have written the answer. Verify that you have actually answered the question. Verify that you answered *all* parts. Verify that you have *not* hidden the answer between many other irrelevant comments about the topic. *Cursives are added to highlight the key elements of the questions.*
- In particular, don't forget to give an example if that is requested, and make it a concrete one.
- When a question asks you to compare two things A and B, make sure to highlight the contrasts: their differences. I do *not* want a full description of A and a full description of B, leaving it to me to find the differences. It is better to use tables than paragraphs of text to compare.

CONTINUES NEXT PAGE

Specific questions.

Question 1. [3 points].
Describe what are the key elements that distinguish plan-based software engineering processes from iterative software engineering processes : *Enumerate* at least two (2) *criteria*.

Question 2. [3 points].
Describe how the model-view-controller architectural pattern works.

Short questions.

Question 3. [6 points].
3.a (2/6 points) *Explain* what is the role of the *Forming* phase according to the Tuckman Team Model.

3.b (2/6 points) *Explain* what is the role of the *Storming* phase according to the Tuckman Team Model.

3.c (2/6 points) *Compare* the Tuckman Team Model and the GRIP model: *Indicate* at least two (2) differences and one (1) similarity between them. Three (3) in total.

Questions for reasoning.

Question 4. [14 points].
Your team is entrusted with the validation and verification of CoffeeVega, a mobile application to automatically inform their users when to sow their coffee seeds or gather the coffee fruits to have maximum flavor.

4.a (5/14 points) *Explain step by step* the inspection formal meeting activity, with special emphasis on *which* are the actors involved and *what* is their role.

4.b (4/14 points) *Propose* and *describe* a different type of inspection method than a formal meeting, which also guarantees that everything will be inspected. *Indicate* also two (2) *disadvantages* of the selected method.

4.c (5/14 points) In the context of the inspection activity described in 4b., *choose* one (1) inspection activity that your team will perform in order to *validate* the process described in 4b. The activity must be different from the formal meeting and the activity described in 4b. Then, *justify why* this particular inspection activity is the best one. Provide an *example* of a less suitable activity than the one you have chosen.

CONTINUES NEXT PAGE

Question 5. [14 points].

Your company has been tasked with the development of a new online social media service for managing small towns. The user wants the service to provide an interface where the citizens of the town can post pictures of things they want to be fixed, share these pictures in the platform, comment on each other's posts and, if they think the original issue should be fixed, upvote it. The city council can filter and review the most popular pictures, interact with the citizens publicly and reply to the original post (e.g., attaching a message and a picture of the issue fixed).

The final application is intended for small towns with circa 100 000 inhabitants and, assuming the citizens will be engaged, an estimate of 100 posts per day.

5.a (5/14 points) *Explain step by step* the process of requirements engineering, with special emphasis on *which* actors and/or stakeholders are responsible for each step of the process and *what* are their responsibilities.

5.b (3/14 points) *Formulate* two (2) non-functional requirements, each one of different type, for the software described at the beginning of this question. *Indicate* the type of each requirement.

5.c (6/14 points) *Formulate* one (1) complete functional requirement, with a corresponding diagram, related to the application described at the beginning of this question. Include *at least* one (1) alternate flow and one (1) exception flow.