DVA332 – Software Engineering 1: Basic Course
Date: 2023-03-03, 8:30–12:30

Responsible teachers: Alessio Bucaioni 021-101437
Jan Carlson 021-151722

Help allowed: Language dictionary

Max points: 40
Approved: Minimum 20 points

Grade 5: 33 – 40 p	Grade A: 36 – 40 p
Grade 4: 27 – 32.5 p	Grade B: 33 – 35.5 p
Grade 3: 20 – 26.5 p	Grade C: 27 – 31.5 p
	Grade D: 23 – 26.5 p
	Grade E: 20 – 22.5 p

Write on one side of the sheet only. Start each assignment on a new sheet.

Answer in English.

Assumptions must be made when there is not enough information provided to solve an assignment and all assumptions must be specified and explained in order to achieve full points.

Good luck!

1. Miscellaneous software engineering knowledge (10 points)

Are the following statements true or false? Motivate your answer.

a) Following good software engineering practices is more important for development of complex systems than for simple systems.

b) The waterfall model is an example of incremental development.

c) In agile development, the requirements should change as often as possible.

d) “Regular users should not have access to the administration functionality” is an example of a non-functional requirement.

e) "The system should never crash" is a verifiable requirement.

f) A UML class diagram is an example of a behavioural model.

g) Pipes-and-filters is an example of an architectural pattern.

h) Low coupling generally improves updateability and maintainability.

i) Unit testing should only be performed using black-box testing techniques.

j) To achieve a good test coverage, test cases should be designed randomly.

2. Software processes and planning (10 points)

a) For each of the fundamental software engineering activities, describe the differences between the waterfall model and agile processes. (3 points)

b) Describe sprint planning (when and how is it done and what is created)? (3 points)

c) You are planning a software project involving 10 developers. The plan is to first develop a common framework and then divide the team in two groups to develop the two parts (A and B) separately. Finally, the whole team will integrate the two parts into the framework and test the whole system:

ID	**Activity**	**Effort (person days)**
DF	Design the common framework	50
IF	Implement the common framework	140
IA	Implement part A	80
IB	Implement part B	100
IT	Integrate and test the whole system	80

First, decide how to best split the team in two groups developing parts A and B, to minimize the overall project time, and state the resulting duration for each activity.

Next, draw a Gantt chart showing when the activities will happen in time. Activity durations and dependencies should be clearly visible in the chart. (4 points)

3. UML (10 points)

a) Draw a conceptual UML class diagram capturing the following concepts and relations of an e-sport tournament: *A tournament consists of a number of matches of two types: Solo matches between 8 players, and team matches where two 4-player teams compete against each other. A player can participate in both solo and team matches, but a player can be part of at most one team.* (3 points)

b) Model the following description of a marketplace price negotiation in a UML activity diagram with three swim lanes: *The buyer makes an offer, and the seller can either accept it, reject it, or make a new offer. In case of a new offer, the seller has the same three options. This is repeated as long as a new offer is made. When an offer is accepted, the marketplace registers the final price and then notifies both the seller and the buyer that an agreement was reached. A rejected offer just means that the negotiation ends.* (4 points)

c) Draw a UML use-case diagram capturing the following functionality of a customer helpdesk system: *All customers can browse the FAQ archive and then add their own question if they don't find it there. Gold level customers also have the possibility to chat live with one of our employees. Regular customers can upgrade to gold level by pressing the upgrade button.* (3 points)

4. Verification and validation (10 points)

a) Explain how input partitioning and boundary values can be used to select good black-box test cases in general, and then give concrete examples for this function. (4 points)

```
/* returns the number of days in that month that year, */
/* or 0 for incorrect inputs. */

int daysInMonth(int year, int month)
```

b) For the following function, give three minimal sets of input combinations that results in complete statement, branch and path coverage, respectively. (6 points)

```
int fun(int a, int b, int c){
     if (a<b){
          a=b;
     }
     if (c<10){
          return a*c;
     }
     else {
          return a*10;
     }
}
```