DVA332 – Software Engineering 1: Basic Course
Date: 2023-01-09, 14:30–18:30

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) Software engineering is a discipline focusing only on documenting software systems. (1 point)

b) In the long run, it is usually cheaper to use software engineering methods and techniques for developing complex software systems. (1 point)

c) Agile processes are categorised as plan-driven processes. (1 point)

d) Functional requirements focus on qualities (e.g.,interoperability, usability) of the software systems. (1 point)

e) Operational and ethical requirements are functional requirements. (1 point)

f) A UML state machine diagram is an example of a behavioural model. (1 point)

g) Test-driven development is an example of an architectural pattern. (1 point)

h) High coupling generally improves updateability and maintainability. (1 point)

i) Unit testing should be performed using black-box testing techniques. (1 point)

j) To achieve a good test coverage, test cases should be designed randomly. (1 point)

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 point)

b) Describe two advantages of the incremental software process model over the waterfall one. (2 points)

c) You are planning a software project involving five developers. After gathering requirements and agreeing on an overall architecture where the system is divided in two relatively independent parts (A and B), the remaining work consists of the following activities:

ID	**Activity**	**Effort (person days)**
DI	Design the interfaces between the two parts	150
A	Implement part A	150
B	Implement part B	120
TA	Test part A	90
TB	Test part B	30
TS	Test the complete system	60

Your task is to make two versions of the plan for the remaining work:

- In the first version of the plan, all five developers should work together with one activity at a time (i.e., no parallel activities).
- In the second version of the plan, parts A and B should be developed in parallel, with 3 persons working on part A and 2 persons on part B.

For each of the two versions, draw a Gantt chart showing when the activities will happen in time. Activity durations and dependencies should be clearly visible in the chart. (5 points)

3. Requirements and design (10 points)

a) You have the following description of a Sick Leave Administration (SLA) system: *"All employees can register in the system when they are unable to work because they are sick. When they are no longer sick, they can register that they are able to work again. Team leaders can also view reports of the sick leave in their own team compared to that of the whole company. These reports are automatically generated by the system every night"*. Write 2 user requirements, 2 system requirements and 2 non functional requirements for this system. The requirements can either detail an already described functionality or extended the system description with new functionalities. (2 points)

b) Create a UML use case diagram for the text in 3. a). (3 points)

c) Start from the system description, the requirements and the UML use case diagram, and define (at least) 2 meaningful objects for the above system. Create a UML activity diagram describing a (non-trivial) behaviour involving those two objects. (5 points)

4. Verification and validation (10 points)

a) Consider the code below. Answer the following questions and motivate your answers. (4 points)

   i. What error could occur when strangeoperation is called? (2 points)

   ii. Is the fault that causes that error easier to find using black-box testing or code inspection? (2 points)

```
def strangeoperation(x, y):
  z = x/(y-388)
  return z
```

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 fum(int a, int b){
     if (a<15){
          a = 0
     }
     if (b>10){
          b = 10
     }
     return a-b
}
```