DVA332 – Software Engineering 1: Basic Course
Date: 2022-11-02, 8:30–12:30

Responsible teachers: 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 – 32.5 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 Swedish or English.

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

Good luck!

1. Miscellaneous software engineering knowledge (8 points)

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

a) It is important that requirements are testable.

b) User stories should be as detailed as possible.

c) High coupling generally makes a system more difficult to maintain.

d) Renaming variables is a type of refactoring.

e) Architectural decisions often have a big impact on software quality.

f) An advantage of the layered architectural style is that a layer can be changed without affecting lower layers.

g) Controlled experiment is a suitable research method for investigating complex questions or tasks that take a long time.

h) For research to produce scientific results, all validity threats must be removed.

2. Software quality and development processes (8 points)

a) Explain the difference between external and internal quality properties and give one example from each category. Explain why high internal quality is beneficial (what software development activities are affected negatively by low internal quality, and why). (4 points)

b) Explain why late requirement changes are more problematic in plan-based development than in agile development. Also, give an example of a type of software system where plan-based development is more suitable than agile, and explain why. (4 points)

3. Scrum (8 points)

a) Describe the responsibilities of the *Scrum master* and the *Product owner*. (4 points)

b) The following features were collected at the start of a project:

Size	Name	Importance
10	Add case	100
5	Adjust colors	10
10	Assign supervisor	80
5	Daily update view	40
8	Export to pdf	30
2	Generate report	10
20	Import from email	10
10	Manage staff	50
2	Overview dashboard	100
5	View case progress	60
20	View statistics	50

One week into the first sprint, the client presents the following new feature and suggests that you start working on it right away, and just extend the sprint with one more week:

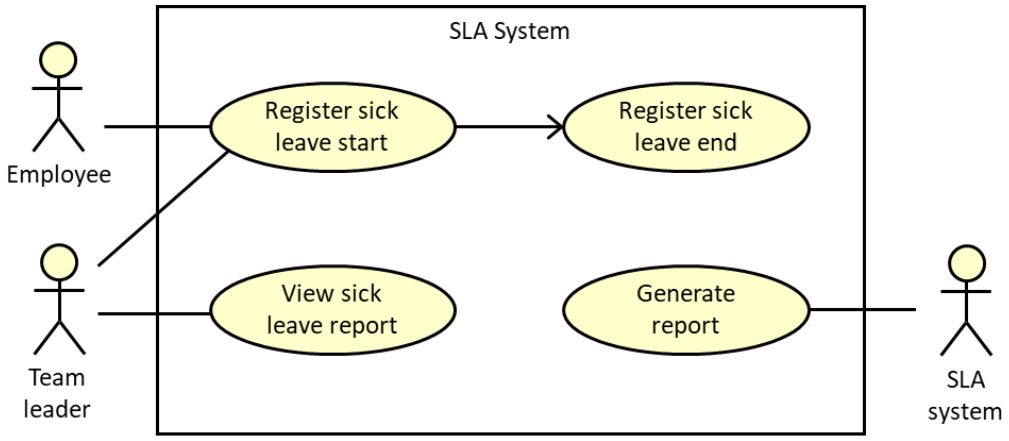
Size	Name	Importance
10	Facebook integration	150

First, assuming that the velocity of the team is 30, what should the sprint backlog look like after the first sprint planning? Next, describe how the situation with the new feature should be handled in Scrum. (4 points)

4. UML (8 points)

a) Given the following description of a Sick Leave Administration (SLA) system, explain why the use-case diagram below is incorrect (there are multiple reasons), and describe how it could be changed to avoid those problems (4 points):

"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."



b) Model the following behaviour in a UML state machine diagram. (4 points)

*"The lift has three positions (Low, Mid and High) and you can use the **Up** and **Down** buttons to change position. The **Door** button can be used to open and close the lift door when it is in the Low or High position. It is not possible to change position when the lift door is open.*

5. Software verification (8 points)

a) Explain the difference between *reviews*, *static analysis* and *testing*. Give one example of a software quality problem that can be found by reviews but not by testing. (4 points)

b) For the function `scalesmallest`, defined below, give minimal sets of input combinations that result in *i)* complete statement coverage and *ii)* complete path coverage. (4 points)

```
int scalesmallest(int s, int x, int y){
   if (s<0) {
      s = -s
   }
   if (x<y) {
      return s*x
   }
   else {
      return s*y
   }
}
```