Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-08-29
Sal (2)	G33(2) G34(35)
Tid	14-18
Utb. kod	TDDC93
Modul	TEN1
Utb. kodnamn/benämning **Modulnamn/benämning**	Programutvecklingsmetodik, teori Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	9
Jour/Kursansvarig Ange vem som besöker salen	Kristian Sandahl
Telefon under skrivtiden	013-28 19 57
Besöker salen ca klockan	15
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson, 013-28 56 34 veronica.kindeland.gunnarsson@liu.se
Tillåtna hjälpmedel	2 handwritten A4 sheets. You may write on both pages A dictionary or English word-book. 2 handsrkivna A4 ark. Ni får skriva på båda sidor. Ett lexikon eller engelsk ordbook.
Övrigt	
Antal exemplar i påsen	

Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-08-29
Sal (2)	G33(2) G34(35)
Tid	14-18
Utb. kod	TDDC93
Modul	TEN1
Utb. kodnamn/benämning **Modulnamn/benämning**	Programutvecklingsmetodik, teori Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	9
Jour/Kursansvarig Ange vem som besöker salen	Kristian Sandahl
Telefon under skrivtiden	013-28 19 57
Besöker salen ca klockan	15
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson, 013-28 56 34 veronica.kindeland.gunnarsson@liu.se
Tillåtna hjälpmedel	2 handwritten A4 sheets. You may write on both pages A dictionary or English word-book. 2 handsrkivna A4 ark. Ni får skriva på båda sidor. Ett lexikon eller engelsk ordbook.
Övrigt	
Antal exemplar i påsen	

Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-08-29
Sal (1)	G33(6)
Tid	14-18
Utb. kod	TDDC88
Modul	TEN1
Utb. kodnamn/benämning **Modulnamn/benämning**	Programutvecklingsmetodik Tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	9
Jour/Kursansvarig Ange vem som besöker salen	Kristian Sandahl
Telefon under skrivtiden	013-28 19 57
Besöker salen ca klockan	15
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson, 013-28 56 34 veronica.kindeland.gunnarsson@liu.se
Tillåtna hjälpmedel	2 handwritten A4 sheets. You may write on both pages A dictionary or English word-book. 2 handsrkivna A4 ark. Ni får skriva på båda sidor. Ett lexikon eller engelsk ordbook.
Övrigt	
Antal exemplar i påsen	

Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-08-29
Sal (1)	G33(4)
Tid	14-18
Utb. kod	725G64
Modul	TEN1
Utb. kodnamn/benämning **Modulnamn/benämning**	Programutvecklingsmetodik Tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	9
Jour/Kursansvarig Ange vem som besöker salen	Kristian Sandahl
Telefon under skrivtiden	013-28 19 57
Besöker salen ca klockan	15
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson, 013-28 56 34 veronica.kindeland.gunnarsson@liu.se
Tillåtna hjälpmedel	2 handwritten A4 sheets. You may write on both pages A dictionary or English word-book. 2 handsrkivna A4 ark. Ni får skriva på båda sidor. Ett lexikon eller engelsk ordbook.
Övrigt	
Antal exemplar i påsen	

Linköpings Universitet
IDA
Kristian Sandahl

2019-08-29

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, 725G64

Note: When I visit the exam, I will take a slow walk among all students, so you don't need to sit with your hand raised. Just call for my attention when I pass your desk.

Instructions to students, please read carefully

- **Explicitly forbidden aids:** Textbooks, machine-written pages, photocopied pages, pages of different format than A4, electronic equipment.
- Try to solve as many problems as possible.
- Motivate all solutions.
- Please, write and draw clearly.
- Write solutions for different areas (fundamental part) and different problems (advanced part) on separate sheets of paper.
- Label all papers with AID-number, date of examination, course code, examination code, and page number.
- You may write solutions in either Swedish or English.
- Please, note that the problems are not necessarily written in order of difficulty.
- TIP! Read all exercises in the beginning of the exam. This will give you the possibility to ask questions about all parts of the exam, since the examiner will visit you in the beginning of the exam time.

Grading

The exam consists of two parts: Fundamental and Advanced.

The Fundamental part has problems worth 10 credits per area. Areas are: Requirements, Planning & Processes, Design & Architecture, Testing & SCM, and Software Quality. Thus, the Fundamental part can give maximally 50 credits.

The Advanced part has problems worth 50 credits in total. Each problem typically requires a longer solution of several pages.

The maximum number of credits assigned to each problem is given within parentheses at the end of the last paragraph of the problem.

Pass condition: At least 4 credits per area in the Fundamental part **and** at least 50 credits in total. The total amount of credits also includes the bonus credits you might have got in lecture exercises autumn 2018. This gives you the mark 3. If you have at least 4 credits for 4 of the areas in the Fundamental part, then you can still pass if you have more than 60 credits in total.

Higher marks are given based on fulfilled *pass condition* **and** higher amounts of credits according to the following table:

Total credits	Mark
0-49	U (no pass)
50-66	3
67-83	4
84-	5

Multiple choice questions

In multiple-choice questions, we will ask you to write down the letters A, B, C, or D for the one or two statements that you think are true. Note that you should not write down the statements that you think are false. There are exactly two true statements per question, so answering with three or four alternatives with gives 0 credits.

For each statement that you select that is correct (i.e., that the statement is in fact true) you get one credit. For each statement that you select that is incorrect (i.e., that the statement is in fact false, but you believed it was true) you get minus one credit. Each multiple-choice question can give maximum 2 credits and minimum 0 credits, i.e., you cannot get negative credits for one multiple choice question.

Example 1: Assume that you have written down statements A and C. If now statements A and B were true, and statements C and D were false, you would get +1 credit for writing down A, but -1 credit for writing down C. Hence, the total credits for the multiple-choice question is 0.

Example 2: Assume that you have written down statement B. If now statement A and B were true, and statement and statement C and D were false, you would get +1 credit for the multiple-choice question.

Example 3: Assume you correctly wrote both statement A and B. If now statements A and B were true, and statements C and D were false, you would get +1 credit for writing down A, and +1 for writing down B. Hence, the total credits for the multiple-choice question is 2.

Good Luck!

Kristian

Problems

Part 1: Fundamental

Area 1: Requirements

1 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. The goal of *requirements elicitation* is that the requirements engineer gets an understanding of the true needs of the customer.
B. We say that a requirement is *unambiguous* if it can only be interpreted in one way by different people.
C. The *UML stereotype* <<extend>> links a *use-case* to the *classes* participating in realizing the use-case.
D. The IEEE Standard 830 for Software Requirements Specification mandates the use of only *natural language* in requirements descriptions.

1 b) Draw a UML *Use-Case Diagram* of a system for a social network system, such as Facebook or LinkedIn. There shall be two different *use-cases* and two different *actors* in the diagram. Use-case descriptions for the use-cases shall be written in complete sentences. (4)

1 c) Write down two *functional* and two *non-functional* requirements of a system storing, searching and playing streamed videos, such as YouTube Netflix, or SVTPlay. (4)

Area 2: Planning and Processes

2 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. The risk "We will develop the current customer's requirements in a non-optimal order" is both a *project specific* and a *direct* risk.
B. The *risk magnitude indicator* is calculated by multiplying the probability and the impact of a risk.
C. By defining a *contingency plan* we will be able to lower the probability of a risk occurring.
D. It is good to manage many *risks* since many things might go wrong. About 30-100 risks for a 10-person team during half a year.

2b) Describe three advantages and one drawback of *iterative software development.* (4)

2c) Describe the difference between a *phase* and a *task* in a *Gantt chart*. Also, describe the difference between a *milestone* and a *tollgate*. (4)

Area 3: Design and Architecture

3 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. The purpose of *design patterns* is to describe new and creative solutions to programming problems.
B. The *Façade* design pattern is useful when you want to provide a simple interface to a complex set of subsystems.
C. The *Observer* design pattern is useful when you want to create only a single instance of a class.
D. The *Strategy* design pattern reduces the number of conditional statements.

3 b) Describe three advantages of a *layered architecture style*. Describe the concept of *layer bridging* in terms of what we gain and what we lose when using layer bridging. (4).

3 c) Draw a *UML Sequence diagram* of a deposit machine for PET-bottles. The machine consists of:

- An input device with a EAN-code reader
- A touch screen with information to the user and buttons for commands
- A receipt printer
- A control unit

Main flow description: The user enters bottles into the input device. The control unit calculates the amount due and display that on the touch screen as soon as a new bottle is accepted. This goes on until the user presses the "finished" button on the touch screen. The control unit changes screen to information about the amount and two buttons: one to donate the amount to charity and one to get cash by bringing a receipt to a shop. If donate is selected a heart-warming "thank you" message is shown on the screen. If cash is selected, the screen shows a neutral good-bye message and the control unit orders the printer to print a receipt.

Use at least one *fragment* for your solution. (4).

https://commons.wikimedia.org/wiki/File:Aldi-Pfandr%C3%BCckgabeautomat.JPG

Area 4: Testing and SCM

4 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. When performing *Equivalence class testing,* you try to create test cases covering as many invalid equivalence classes as possible in each test case.
B. To make *Boundary value test cases* you need access to the source code of the system.
C. It is possible to write a program that requires the same number of test cases for both *branch coverage* and *full path coverage.*
D. *Exhaustive testing* means that you test all possible input values to a program.

4 b) Describe the following steps of the highest levels of testing: *Function test, Performance test, Acceptance test, and Installation test*. Focus on **what** is tested, not **how** testing is done. (4)

4c) Describe the work-flow of a *version handling* system, such as Git or SVN, where two developers, Adam and Eve, are changing the same piece of code without knowing of each other's actions. They started from the same version and Eve stored her changes to the central repository before Adam. (4)

Area 5: Software Quality

5 a) Which of the following statements are true? Answer with the statement letter only, no motivation is needed. (2)

A. In the *staged* representation of CMMI a *Maturity level* is associated with a set of *Process areas.*
B. To determine if a *Process area* in CMMI is satisfied, you check the fulfilment of *Specific* and *Generic* goals, associated with the process area.
C. The majority of the *Process areas* of CMMI are associated with level *5: Optimizing.*
D. The *Process areas* of CMMI are all independent of each other.

5 b) Describe two *metrics* that can be measured on source code and that can be used to predict *reliability* of the system under development. Remember to motivate your answer. (4)

5 c) In general guidelines for *Total Quality Management* you sometimes hear the word "non-personal software" or "egoless programming". Explain what this is and what benefits this can offer to a software company. (4)

Part 2: Advanced

6. Make a template of headlines of a *project plan*. For each headline, add a small instruction within brackets of what information that goes under that headline.

Example:

1. Project description
1.1 Background to the project

< Major functions. Identify orderer and main users. Expected benefits from the perspective of orderer. Earlier solutions used. >

1.2 Constraints

<...>

(end example)

You shall assume that requirements specification and quality management are documented in separate documents. At least 10 headlines are needed for full credits. 1.1 in the example does not count; 1.2 in the example does if you write instructions. (10)

7. *Scenario:* You are a newly hired R&D consultant for a small software development company that sells embedded software for the retail market. Your main products are apps that are placed in smart phones (for scanning inventories) and point-of-sales (POS) systems placed in tablets (the main system that the shop assistants are using when helping a customer). Currently, the company is using a development process that is similar to the classic waterfall model. The company do not have any tool support in their development except standard text editors, compilers, and a file server. There are10 employees in the company:

- 1 CEO
- 2 Sales persons
- 1 Marketing manager
- 1 Project manager
- 2 testers
- 3 developers

Task: Write an e-mail to the CEO of the company, where you argue for that the company should change their way of working. Assume that the CEO is mostly familiar with traditional project management, and that you suggest them to use a variant of *Scrum*, together with some useful agile practices. Make sure that you explain each concept carefully so that the CEO understands why this is useful/not useful and how it might be implemented in the organization. Below, you will find a list of concepts and topics for which you are supposed to argue either for or against:

- Sprint
- Time-boxing
- GANTT-chart
- Waterfall model
- Self-organization
- Product Backlog
- Continuous Integration
- Automated Testing
- Scrum Roles
- Software Review

Please note that your email should be an arguing and convincing e-mail, not just a list of statements. (20)

8. *Scenario:* A Car Cruise System (CCS) monitors all cars on special highways. A special feature is to build and automatically drive queues of cars. This means that the driver releases steering and speed control to the CCS, which monitors and drives the cars in a queue with small distances between the cars in a fast, but safe and even speed. When driven in the queue the attention of the driver is checked through biological sensors. If the driver is close to fall asleep, the car is moved out of the queue and parked at the nearest parking lot. All cars in a queue communicate via an ad hoc 5G network, and sends GPS positions to the CCS. There are lot of sensors and actuators to detect and fix potential problems found in the close environment of the car and the queue. A car enters a queue by telling the CCS that its driver is interested in queue driving and the CCS starts inviting cars in the neighborhood. A car can also be invited if it is close to a queue. The driver can press a button to leave the queue, a procedure done in different steps, such as increasing the distance around the car and moving it to another lane before handing off control to the driver.

Task: Draw a *UML class diagram* of the main concepts related above. Some concepts are given as classes, such as the Car, Queue, Driver, and CCS. Other concepts might be modeled as *classes* or *attributes*. Don't forget the *multiplicity* and names on *associations*. The solutions are judged for their clarity to display the relations between the most important concepts using a correct UML syntax. Methods are less important and are only added if they increase the clarity of the solution. Use about 8-10 concepts from the text above. (10)

9. *Scenario:* The same as in problem 8 above.

Task: Draw a *UML State Diagram* for the class Car. The *state diagram* shall consist of at least 5 *states*. Don't forget *triggering events* and *actions* on the transitions. The solutions are judged for clarity, consistency and correct use of *UML State diagrams*. (10)