

Written exam for Software Engineering Theory

Course codes TDDC88, TDDC93, TXDT02

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 through 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. Remember that the examiner has over 200 students, so plan for solving problems in a flexible order if you have to wait for clarification.

Grading

The exam consists of two parts: Fundamental and Advanced.

The Fundamental part has problems worth 10 credits per area. Areas are: Requirements, Design & Architecture, Testing, Planning & Processes, and Quality Factors. 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 includes also the bonus credits you might have got in lecture exercises autumn 2010. This gives you the mark 3 in the Swedish system and a C in ECTS. 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 in Swedish system	Translation to ECTS
0-49	UK	Fx
50-66	3	C
67-83	4	B
84-	5	A

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 give 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 statement A and B were true, and statement and statement 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 requirement “The system shall take an arbitrary number of integers as input, and as output calculate their average.” is a functional requirement.
- B. The requirement “All source code shall be written in LISP” is a functional requirements, since LISP is a functional language.
- C. The requirements “The system shall be available 99% of the time” and “The users shall be able to learn all functions of the system in 14 days” are two non-functional requirements.
- D. The requirements “The system shall accept the length of two sides of a triangle as input and calculate the length of the third side of the triangle as output.” is a functional requirement that can be said to be unambiguous, testable and feasible.

1 b) Draw one use-case diagram of a web-based bookstore, for instance Amazon, with two different actors and two different use-cases. Both actors shall be humans and have an association to at least one use-case. Each individual use case shall have a text that consists of at least 4 sentences and includes initiation and termination of the use-case. (4)

1 c) Suppose you are building a requirements database and have to determine the different fields that shall be filled in by the analyst for each requirement. It is obvious that you want a requirement identity number and a text describing the requirement. (So don't include them in the answer.) Suggest 4 different fields with 1-2 sentences of motivation for the need of each field. For instance:

Field 1: Requirement ID – This is needed to manage requirements, for instance, in prioritizing.(4)

Area 2: Design and Architecture

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

- A. A 2-tier client-server architecture with a fat client has the drawback that there is much network traffic needed compared to other client-server alternatives implementing the same application.
- B. A major drawback with architectural styles is that they are almost impossible to combine.
- C. The high reusability of filters in a pipe-and-filter architecture comes with the risk of performance penalties, since data might need parsing or checking for each filter.
- D. An advantage of a layered architecture is that incremental and iterative development methods can be supported.

2 b) Draw a UML class diagram modelling the following facts about cars:

- A car has exactly 4 wheels, otherwise it is not a car (we assume a car shall be in drivable condition)
- Each wheel has a tyre.
- The car owner must have at least one set of 4 summer tyres and one set of 4 winter tyres.
- Enthusiasts can have even more sets of tyres.
- Winter tyres can have studs.
- If the car owner have continental (stud-free) winter tyres, he or she might have a set of 4 snow-chains (4)

2 c) Explain the following terms in the context of software design: Coupling, Cohesion, Implementation view and Deployment view. (4)

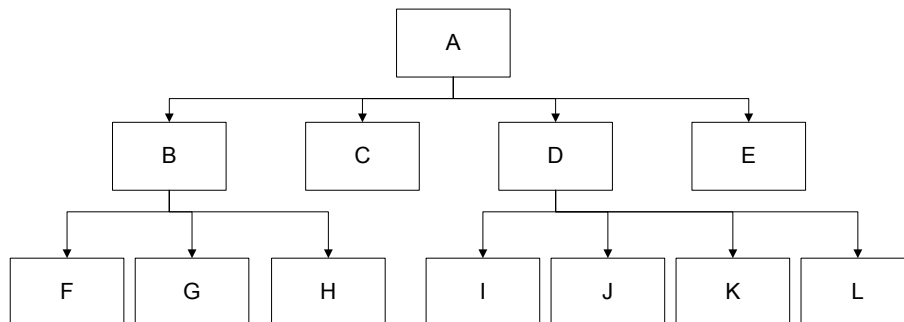
Area 3: Testing

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

- A. In control-flow testing, path coverage is stronger than branch coverage, which means that path coverage require at least as many test cases as branch testing, often much more.
- B. A program with a single if-then-else statement requires at least 3 test cases for branch coverage.
- C. If you make a data-flow analysis for a variable in a program you should pay special attention to the pattern when a variable is used, and then used again. This use-use pattern signals a serious problem of potentially inconsistent behaviour.
- D. A definition-use path of a variable x is a path through the program flow where the initial node contains a definition of x and the final node contains a usage of x.

3 b) Assume you have a system that accepts five to nine inputs which are integers between 100 and 1000. The interval is inclusive, that is the values 100 and 1000 are also valid inputs. Which inputs of test cases do you select if you want to accomplish boundary value testing? Motivate your choice. (4)

3 c) The following figure illustrates the component hierarchy in a software system.



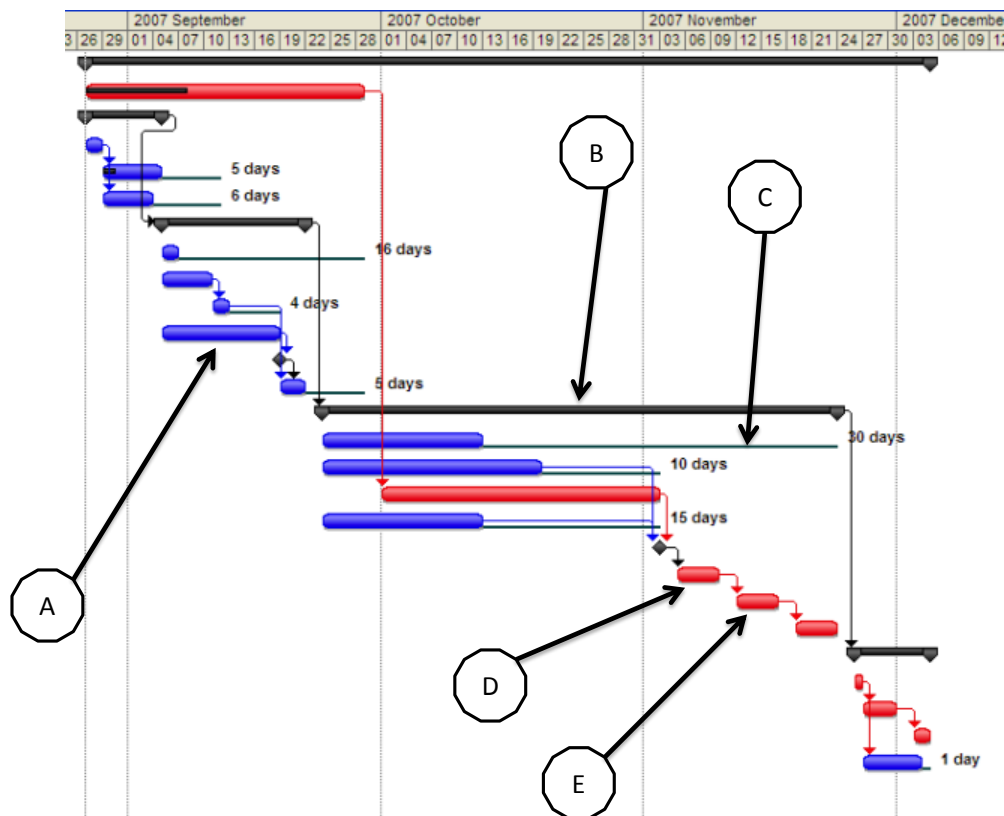
Describe the sequence of sessions of integrations and tests of the components using a bottom-up approach and a top-down approach. You may assume that all individual components have passed individual unit tests before your integration test starts. (4)

Area 4: Planning and Processes

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

- A. A drawback of the Waterfall model is that information flows so fast that it is practically impossible to plan milestones and tollgates.
- B. SCRUM provides a set of practices for ways of organizing development work, but does not give much guidance in which software techniques to use.
- C. The “planning game” in the context of eXtreme Programming refers to the fact that you often can deliver good software unexpectedly fast.
- D. Iterative and incremental life cycles reduce the risk of failing the project goal since it is possible to gather more knowledge of the customers’ true needs by demonstrating prototypes and getting feed-back before the project is completed.

4 b) Consider the following excerpt of a Gantt chart.



Name and explain the elements of the Gantt chart marked as A, B and C. Name and explain the meaning of the little arrow between elements D and E. (4)

4 c) You are a good project leader and have identified the following risks:

1. Communication problems between the teams in Linköping and Stockholm.
2. We will not be able to manage as many simultaneous users as required.
3. Our team will not be able to make a good-looking layout of the User manual.
4. Our configuration manager will probably take a half-year off for parental leave during the coming 1.5 years.

There are 4 major strategies to plan for handling these risks. For each of the risks above: name a strategy and give an example of how the strategy can be implemented. (4)

Area 5: Quality factors

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

- A. The structure of the staged CMMI model is built up by maturity levels that in turn are defined by a set of company values that must be fulfilled for each level.
- B. Test driven development, TDD, is necessary to achieve CMMI level 3.
- C. Configuration management, CM, is necessary to achieve CMMI level 2.
- D. Measurement and Analysis, MA, is introduced on CMMI level 2, and thus precedes Risk Management, RKSM, that is on CMMI level 3.

5 b) Your manager wants the following questions answered each month:

1. How fast can a beginner user become productive with our present release?
2. How reliable is our system?
3. How large is the system?
4. How much overtime did you spend on error-handling last month?

For each of these tasks, define a metric that can help you answering the question, at least give you partial information to the question. Use the form introduced in the lecture, example:

Description:	Mean Time Between Failures (MTBF)
How to obtain data:	In your bug tracking system, measure the time span it takes to repair a bug. Measure the mean time each failure appears.
How to calculate the metric:	$MTBF = \text{average time span to repair a bug} + \text{average time of the appearance of a failure}$
Relevant quality factor:	Availability, if the MTBF is high we seldom take the software out of production. (4)

5 c) Name and describe four roles of a software inspection. (4)

Part 2: Advanced

6. You are developing and running a world-class portal for private persons buying boats. The users can specify the wanted attributes of the boat, their skills and plans for their usage of the boat. Your system can assist them in the specification process and even suggest good attributes given their skills and plans. You also have an optional warning mechanism if the system thinks that the attributes do not match the skills and plans of the owner.

Your system has great access to various dealers and a good auctioning system that can help your users to find and purchase a good object. If they decide to buy abroad your system can guide them through the necessary paper work, and order their travel to the pick-up location. Alternatively, they can choose a delivery service and pick up the boat at your office in Gothenburg harbor. (With a nice view!)

The payment is flexible, your system have access to major credit cards and bank transfer systems. Your system can also help the users with financing, but then you need more information about their private economy. When taking a loan, it is not possible to use electronic signature, you have to exchange paper copies with the user to grant the loan. Needless to say, the security must be very high.

But before cashing in you must model your system:

- a)** Draw a high level architecture with boxes and arrows of the main components and subsystems you need. Describe how security requirements are met in the system. (5)
- b)** Draw a class diagram centered on the class User. The diagram shall contain at least 5 attributes and 5 relations to other classes. The other classes need only to be mentioned by name. (5)
- c)** Draw a state diagram of the class Boat. Use at least 5 states, other than the start and stop. Don't forget to specify what is happening on the transitions. (5)
- d)** Draw a sequence diagram with at least 5 roles or objects for a situation where a customer pays for the boat and services with a part of the cost from a bank transfer, and part of the cost with a loan arranged by your system. (5)
- e)** Write down 5 functional test cases for testing a part of the system. Describe with a sentence the intended part you are testing. (5)

7. You are a leader of a small firm of 20 employees in Linköping, 20 in Stockholm and 10 in Singapore. You and your co-workers have a good record in developing training systems for defense industry. This means that you are used to GUI programming, Ada, Assembler and Secure solutions. Since your customers require extensive certification procedures and full control of the development you work with the Waterfall model, with some very small through-away prototyping. You happen to be between two large contracts and have taken the task to implement your friend's boat buying portal (described in problem 6) within a year. This is another type of software so you have to reconsider your way of working.

a) Which process model should you use? You are thinking of three alternatives: i) stay with the Waterfall, ii) Use a phased-iterative method, such as RUP or OpenUP, or iii) go for an Agile solution, such as eXtreme Programming.

Make a table of the three alternatives as columns and write advantages and drawbacks for these models in your company. You shall have at least 10 items in the table. An item can be either a drawback or an advantage. The bottom row is the number of advantages – number of drawbacks. Which column got the highest value? Write a sentence motivating if this was your expected choice of process. (10)

b) You visited a guest lecture from industry at the university and got the recommendation to work with test-driven development. Describe shortly the idea of test-driven development and list expected advantages of that practice. (5)

c) Ada and assembler are not used in web-based systems. List at least three different languages or techniques you need special training for in your coming project. Add an one-sentence motivation for each item on your list. (3)

d) You use Source Safe for version handling, but are thinking of changing to Subversion or Git. Select one of them, and give two arguments for why they would suit you. (2)

e) Write down four non-functional requirements for the system that emphasize the quality factors security and usability. Each quality factor shall be addressed in at least one requirement. (4)

f) Specify a functional requirement or feature that contributes to either security or usability. Motivate why. (1)