Grading instructions to exam 2022-10-26

There are many different solutions possible. This is working material. Misspellings and grammar errors do appear.

In almost reverse order

9. *Scenario:* You are the head of a software development company which normally uses agile methods where you pick compatible parts of eXtreme Programming, SCRUM, and Kanban depending on the project you get. Now you actually won a fixed-price contract and decided to use the classical waterfall model for that project.

Task: Select four different concepts (e.g. practices, artifacts, meetings etc.) from the agile methods or method frameworks mentioned above that you think can fit in the new project. For each concept write:

a. A description of how it works.(3-4 sentences)

b. Expected benefits in your new project. (2-3 sentences)

Also write how you will implement risk management in the classical waterfall model. (about 10 sentences). (10)

SCRUM meeting is a short (15 minutes) daily stand-up meeting where each team member reports:

- What has been done since the last meeting?
- What will be done until next meeting?
- Do I see any problems?

This will work well during the implementation and testing phases and gives the design team rapid, first-hand information about the project. Opportunities for helping others can be identified also in the classical waterfall project.

Product owner is a role who has the holistic view of the product with input from several direct and indirect stakeholders and market information. Both internal and external stakeholders' perception of the product are attended. The Product owner makes the analysis and defines the product to be delivered.

This role is needed to write the requirements specification and to assist others, e.g. architect, designers and testers when they define the goals of their activities. This works also in the classical waterfall project. It's not necessary to prioritize the requirements in a fixed-price project.

Pair programming means that two persons sit in front of the screen and program together. The code is discussed during the cooperation; sometimes the programmers leave the screen to investigate certain things a bit deeper. It is encouraged to shift the person using the keyboard regularly.

This way of working applies to the implementation phase, and it has been shown that the code becomes well elaborated with fewer bugs and good quality. By changing the person at the keyboard there is also a knowledge transfer between the programmers. This can be extended by changing the members of the pair also in a classical waterfall project.

Limit work in progress means that each team has a maximum number of tasks at hand. From experience a limit is set for instance of how many task that can be scheduled, ongoing, and documented for the team.

This practice applies to many phases of the classical waterfall model where work is sub-divided into tasks. The expected benefit is that things will be finished fast since there will be less waste of work with context switching as compared to having many parallel activities.

Risk management in the classical waterfall model can be performed for the entire project in the beginning You start with brainstorming to identify potential risks, then you analyze them in terms of probability and impact and calculate the risk magnitude indicator by multiplying them. The topmost 3-10 risks are planned for by formulating what we can to avoid, transfer, mitigate, or making a plan B. Since this is a fixed-price project with the classical waterfall model we must define things that keeps the price and the agreed requirements constant. We shall avoid going back to a previous phase. For instance, we can reduce our margin by making more expensive solutions. The risks are finally monitored regularly to check whether the information about the risks have changed or need to be changed. Since the project is very constrained, we need to plan the calendar time to make sure that each phase is done thoroughly.

Grading:

2p per agile concept: 1p for a description and 1p for a sensible motivation of how it helps the project.

2p for a sensible description of risk management. It is important that it is noted that the classical waterfall model puts constraints on what you can do.

8. *Scenario:* Somewhat simplified you can say that with a Swedish driver's license category B you are allowed to drive cars with a maximal weight of 3500 kg.

You are allowed to hook on a trailer with a maximal weight of 750 kg.

If the trailer weighs more than 750 kg you can still drive the car with the trailer if the sum of the weight of the trailer and the weight of the car is maximally 3500 kg.

Your friend has developed an app where you can calculate if you are allowed to drive the car with or without a trailer given their weights with a driver's license category B. Your job is to test the app.

Task: Identify input and output variables and valid and invalid equivalence classes. With valid we mean that you are allowed to drive with a category B driver's license. Invalid classes means that you are not allowed to drive with the category B driver's license. We assume that it is impossible to enter negative numbers.

Create a test table when you perform boundary value testing as it has been taught in the course. You should be able to answer by using the table editor in Wiseflow. (15)

Input variables: Trailer weight (TW, kg), Car weight (CW, kg)

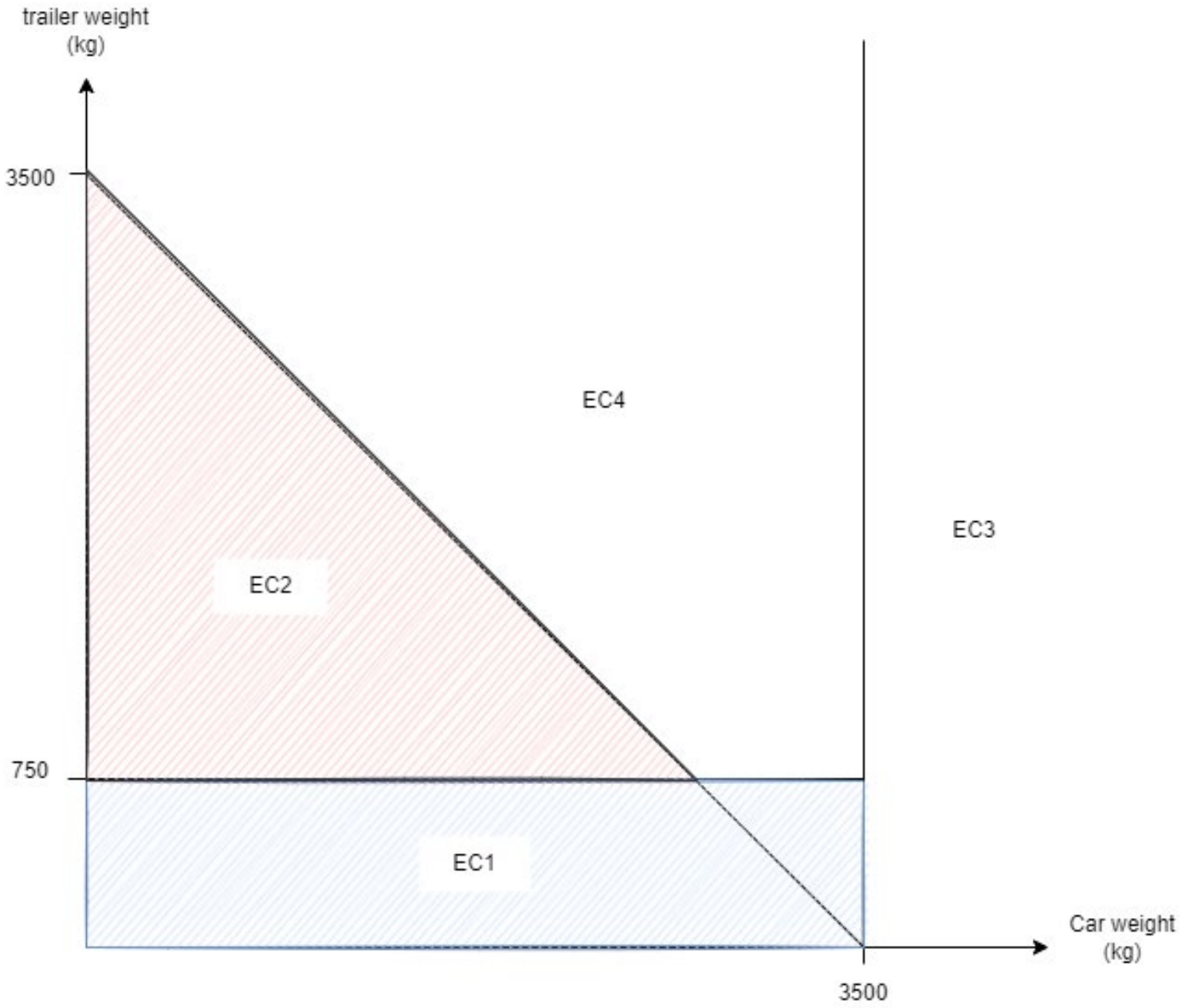
Output variable: Permitted drive with B license (PB, Boolean)

EC1: 0 <= TW <= 750 and 0 <= CW <= 3500 (valid)

EC2: 750 < TW and 0 <= TW + CW <= 3500 (valid)

EC3: 3500 < CW (invalid)

EC4: 0 <= CW <= 3500 and 750 < TW and 3500 < TW + CW (invalid)


trailer weight
(kg)
3500
750
EC4
EC3
EC2
EC1
Car weight
(kg)
3500

ID	Input: TW	Input: CW	Output PB	Comment
TC1	749	3000	T	Testing the border of allowed TW
TC2	750	3000	T	D:o
TC3	751	3000	F	D:o
TC4	600	3499	T	Testing the border of allowed CW
TC5	600	3500	T	D:o
TC6	600	3501	F	D:o
TC7	749	2749	T	Testing the border of allowed TW + CW
TC8	750	2750	T	D:o
TC9	751	2751	F	D:o

Grading:

For full credits, correct input/output variables, correct equivalence classes, correct test table

We accept solutions without the comment column in the test table.

We accept solutions without the explaining diagram as long as the ECs are OK

When it comes to testing TW + CW <= 3500 the students don't need to test the intersection with
TW = 750. Any values on the line TW + CW = 3500 will do.

There are several ways to express the ECs depending on how you assume the program is implemented.

Deductions:

-1 p per missed I/O variable

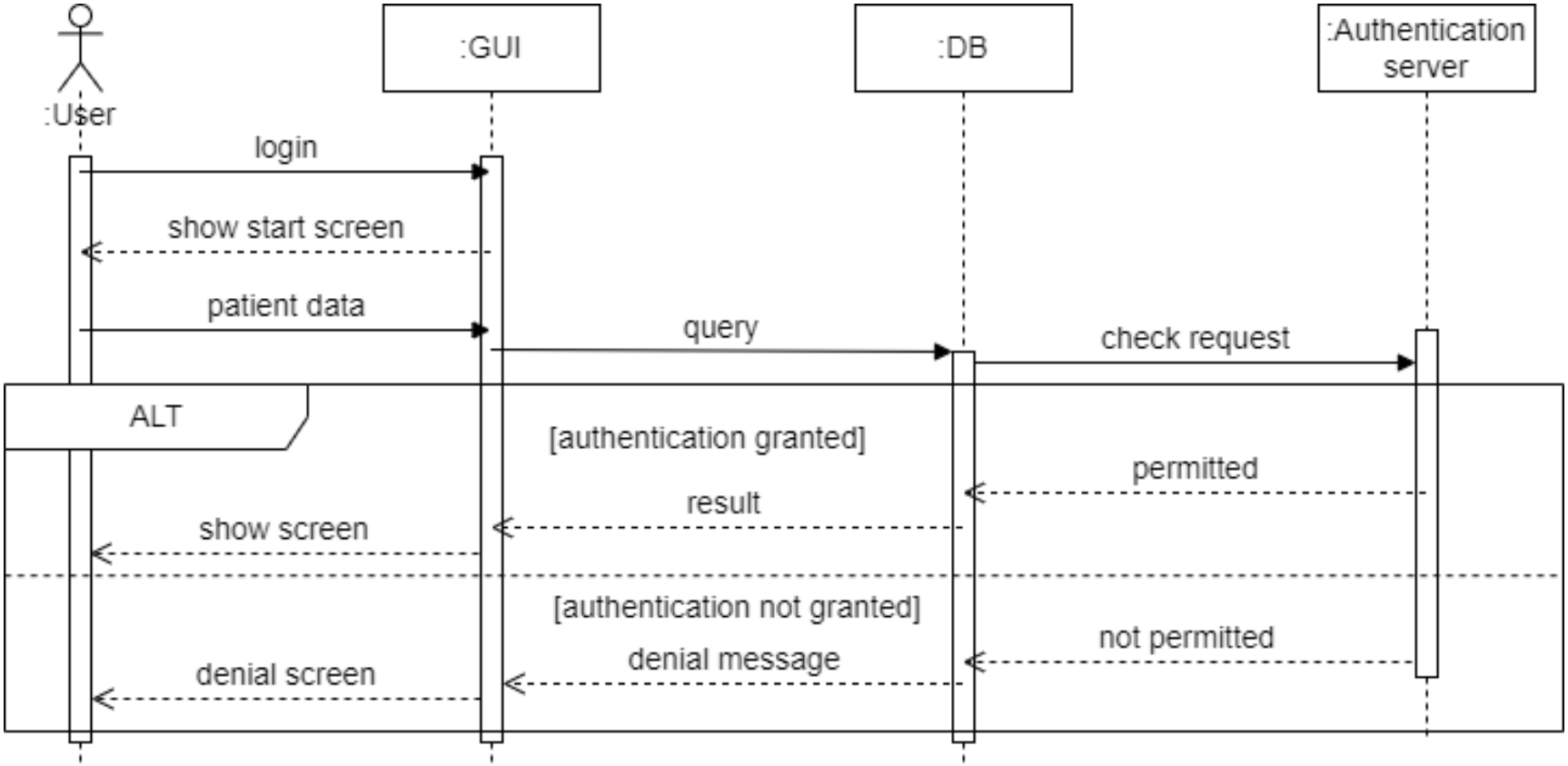
- 1 p per wrong or missing EC

- 1 p per missing or wrong test case

-1 p per missing column apart from Comment in the test table

7. *Scenario:* You are about to design an application of a patient record database for a healthcare district and have formulated the following use-case:
Name: Retrieve data about a specific patient
Actors: User, GUI, database, authentication server
Description: The user logs in to the system through a GUI. The user fills in the personal number of the patient and selects the type of information that is of interest on the starting screen. The user presses the “send” button. The GUI formulates a query to the database. The database checks if the user is eligible to retrieve the wanted information for the patient with an authentication server. If authentication is granted, the result of the query is sent from the database to the GUI. The GUI presents the results in a results screen.

Task: Model the use case as a UML sequence diagram. The diagram shall take the case when authentication is not granted for the combination of user, patient, and type of information. Also, give one advantage and one disadvantage if we replace use-cases with a set of sequence diagrams instead. (10)



Advantage: More clear and detailed descriptions

Disadvantage: Hard to understand for stakeholders without knowledge in UML and programming

Grading:

8 p for a correct diagram. We accept that return arrows are solid.

-1 p per syntactic and semantic mistake. If the same mistake appears several times, it is still only -1 p. For instance, if there are no role rectangles, it is still -1 p.

-1 p per missing role

-2 p for wrongly used role

-1 p if there is no fragment

1p per advantage and disadvantage, max 2 p

6. Describe three different metrics that can be used to assess the usability of an interactive software product, for example, a mobile/computer game. The metrics shall be obtained in different ways:

- One metric shall be possible to obtain with a test panel running a final release of the software and then answering a questionnaire.
- One metric shall be possible to obtain by making observations of a test panel running a final release of the software.
- One metric shall be possible to obtain without involvement of user representatives.

For each of the metrics answer the following questions:

a. What do you measure? (Called “Description” in the metrics slides.)
b. What procedure do you need to preform to get the data?
c. What resources do you need to collect the data?
d. How do you calculate the numerical values(s)?
e. How does the metric relate to usability? Use arguments such as: “A high number of indicates high usability since …” (15)

Panel questionnaire of full release

Description: Number of good and bad features recalled by users

Procedure: Set up a test scenario. Let test users run the scenario. Collect number of good and bad features in a questionnaire afterwards.

Resources: Test equipment, a panel of test users, a questionnaire

How to calculate the numerical value: Take the average of number of good and no. bad features. Two values.

Indication of usability: Appropriateness recognizability (Relevance); many good and few bad features indicates a good match with the users' mind-set.

Observation of users of full release

Description: Percentage of time spent in help-functions

Procedure: Arrange that a panel of end-users are given representative scenarios and have an observer recording total time, and the time spent with help-functions.

Resources: Test equipment, a panel of test users, observer

How to calculate the numerical value: Divide the time spent with help-functions with the total time for each scenario and end-user. Translate to percentage.

Indication of usability: A relatively high percentage might indicate that the system has a long learning curve or low relevance for the purpose. In both cases usability will be relatively low.

Without the panel

Description: Maximum response time

How to obtain the metric: Identify a set of representative operations. Set up test-cases for them through the GUI. Preferably the test cases are automated. Instrument the code to measure the execution time of the operations. Run a large number of test sessions.

Resources: Automated GUI testing tool, test cases, programmer

How to calculate the numerical value: Calculate the average execution time of the tests.

Indication of usability: A relatively low response time indicates that the system is fast and avoids waiting time and interruptions that can be stressful for the end-user

5 a) Which of the following statements are true? Two statements are true. Wrong answers give minus credits (2)

- ☑ In object-oriented software, a high maximum depth in the inheritance tree indicates that the software is harder to understand.
- ☑ A high number of reused code lines can be an indicator of high reliability with the argument that the reused parts have been tested and run for a longer time than newly produced code
- ☐ If we draw a flow-graph of a program, that only contains binary decisions, then the cyclomatic complexity, V(G), of the program grows quadratically with the number of decisions. (e.g. if you double the number of binary decisions, V(G) becomes four times higher).
- ☐ The reliability can be approximated by MTBF/(1-MTBF), where MTBF denotes Mean Time Between Failures.

5 b) *Scenario:* You are the manager of a company of 15 people and you have passed the state of being a start-up. The software helps the customers well and you have a good reputation for well tested software. Sometimes you are a bit late in delivering a new release, but it is tolerable. However, your staff complains about the large amount of overtime. They believe that this is because you need to spend much time in adapting the components you chose; nothing seems to work right off the shelf.

Task: Select and describe a CMMI process area that you believe will help your company to improve. A description of a process area contains major ideas of the process area and the specific goals. For full credit, the purpose statement alone is not enough. Also, motivate why the process can help you. (4)

Technical Solution (TS) describes a set of processes for selecting, designing, and implementing the software. This applies to all abstraction levels of the system. Selection is to develop alternative solutions to the requirements and evaluate them with different criteria to select the best. Different types of prototypes can be used to find additional information. After the selection a more detailed design is developed to provide all information for the implementation work to start taking quality factors into account. Finally, implementation is made according to the design. Sufficient documentation for future maintenance is provided.

TS will help us to be more conscious about how we select the solutions. Looking for reuse opportunities is good, but it will not help us reduce the extra work if we don’t have all information needed before the implementation.

Grading:

2 p for a sensible process area description that shows that the student has not only been re-writing the

1 p if only the process area and the purpose is mentioned

2 p for a good motivation, with connections to the scenario

5 c) Compare the review methods Inspection and Walk-through in terms of:

- Goal
- Participants Process
- Data collected

Hint: Make a table with the methods as columns and comparison criteria as rows. Write 1-2 sentences in each cell. Preferably, you can use the table tool in Wiseflow. (4)

Criteria	Inspection	Walk-through
Goal	Detect defects	Detect anomalies, improve the product.
Participants	Moderator, recorder, reader, author, inspectors	Author (walk-through leader), recorder, team member
Process	Formal four-step process with meeting	Preparation, Informal meeting
Data collected	Per anomaly: Classification, Category, Ranking General data: product Id, date, team, inspection time, volume of material to be inspected, decision	Anomalies, recommendations, and actions

Grading: Half a point for each sensible cell.
Round up to nearest integer.

4 a) Which of the following statements are true? Two statements are true. Wrong answers give minus credits (2)

- ☑ The classical waterfall model can be good for fixed-price projects
- ☐ A problem with the classical waterfall model is that it is hard to understand.
- ☑ A prerequisite to use iterative development is that the requirements can be divided in smaller parts
- ☐ A problem with the iterative model is that work cannot be done in parallel.

4 b) Assume that you have a contract with the customer stating that a list of features must be present in the coming release. Give examples of things you can do if:

- Two people leave the project.
- Your manager wants you to deliver the product a month earlier.
- Your strategic product manager decides that you need to deploy on more platforms than was originally agreed.
- You realize that your time estimations for the features were too optimistic

We don't assume that all the above problems occur simultaneously, just account for them one by one. For each of the changes, write a short motivation, 1-2 sentences, of how it will help. (4)

- Two people leave the project.

We can renegotiate the deadline and deliver later. This way we can deliver the features with less resources.

- Your manager wants you to deliver the product a month earlier

We can ask the manager to lower the quality, for instance to work with a more limited set of web-browsers. This way the features are delivered but with less flexibility for the customer.

- Your strategic product owner decides that you need to deploy on more platforms than was originally agreed.

If we already have the competence within the team, we can renegotiate the deadline. Thus the features are there but later than planned.

- You realize that your time estimations for the features were too optimistic.

This is a resource problem. You can negotiate to add more people in the project, then you get all the features, but to a higher cost. This only works if the problem is discovered early, if we try to add more people late in the project there is a risk for huge delays, since it takes a while until the new persons are productive.

Grading:

1 p for each sensible change with a motivation

Some students want to use risk management also to mitigate the risk of the problems that can occurs. This is OK as long as there is a motivation for how the mitigation strategy can work. The number of features is fixed, this cannot be compromised.

4 c) Shortly describe the following concepts in planning with a GANTT-chart: phase, slack time (also called float time), critical path, and mile-stone.
A short description is typically 1-2 sentences. (4)

Phase - a set of activities in the chart that belong together, representing a main part/phase in the project.

Slack-time - the buffer available to complete a task before it will cause delays for future dependent tasks. As long as delay of the task is shorter than slack time, no delay on the entire project will occur

Critical path - a sequence of tasks that affects the time it takes to complete the whole project, meaning that if delays occur for a task on the critical path, it will cause overall delays in the project as a whole.

Milestone - internal sub-goal for the project. A date (deadline) at which a certain (sub) goal shall be achieved. Green "upside down" triangle in Gantt chart.

1 p for each sensible description

3 a) Which of the following statements are true? Two statements are true. Wrong answers give minus credits. (2)

- ☑ In the software engineering terminology an error is a human mistake leading to a fault in the program, which if executed can cause a failure.
- ☑ Usability testing very often involves using a panel of test users performing representative tasks of the system.
- ☐ If you want to achieve branch coverage testing of a program you can be satisfied with a smaller number of test cases than you would need if you require statement coverage testing.
- ☐ A drawback of Equivalence class testing is that only valid input is tested.

3 b) Describe the process of Test-Driven Development (TDD). What is the alternative to TDD? What benefits are proponents of TDD expecting to accomplish? (4)

Test-driven development mans that you iterate over the following cycle:

1. Write a test that fails
2. Write code that make the tests work
3. Refactor to avoid redundancy and bad design

This means that you write test before coding. The alternative is to finish the code and then write the test cases. Proponents of TDD claims that this will help you to:

- Get faster development
- Avoids unnecessary work
- Spend time thinking about what the software will do
- The tests can play the role of detailed documentation.

Grading:

2p for a reasonably correct process description

1p for an alternative

1p for at least one good benefit

3 c) What is the difference between the centralized workflow and the feature branch workflow in continuous integration? Write down one advantage of the centralized workflow, also write down one advantage and one disadvantage of the feature branch workflow. (4)

The major difference is that with feature branch workflow it is possible to make a branch where development can be done in parallel to main in several steps over longer time instead of merging with main for each step of development.

Advantage of centralized workflow: Simple to set up

Advantage of feature branch workflow: Better control of the other of inclusion of features to main.

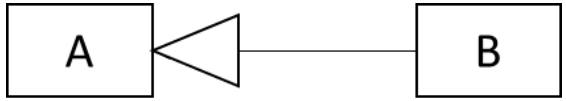
Disadvantage of feature branch workflow: Branches that live too long in parallel to main can be difficult to merge.

1 p per sensible answer

2 a) Which of the following statements are true? Two statements are true. Wrong answers give minus credits. (2)

- ☑ The composition association implies “no sharing”. Instances of B can only be owned by a single instance of A in the diagram:

- ☐ Generalization means that A inherits all properties and operations of B in the diagram:



- ☐ The Observer design pattern can be used when we need to use different variants of the same algorithm in a class.
- ☑ The Façade design patterns can be used to provide a simple interface to a complex subsystem.

2 b) Describe two ways you can make an architecture for a system that is easy to maintain. Don't forget to motivate your answers. 2-3 sentences per solution might suffice. (4)

A layered architecture means that the system is partitioned in abstraction layers. Components within a layer communicate as wanted, communication between layers follows a predefined protocol. It is only possible to communicate with the layers immediately above ore below the layer.

This eases maintenance since changes can be kept local to a layer and ripple changes outside the layer are handled through the interface.

Any architecture where we can accomplish low coupling and high cohesion. Low coupling means that it is easy (realistic) to follow up on potential ripple effects of changes made in one component. High cohesion often eases understanding of the component.

Grading:
2 p per pair description of the architecture and a motivation. We put more emphasis on the motivations, so even a quite simple description in well-known terms gives if the motivation is good.

2 c) Let's put you in the teacher's role. You have asked the students to "Draw a small example of a UML State diagram for a book in a library." You received two solutions, A and B. Identify the incorrect solution. Give the incorrect solution constructive feedback of about 4 sentences. (4)

Diagram A

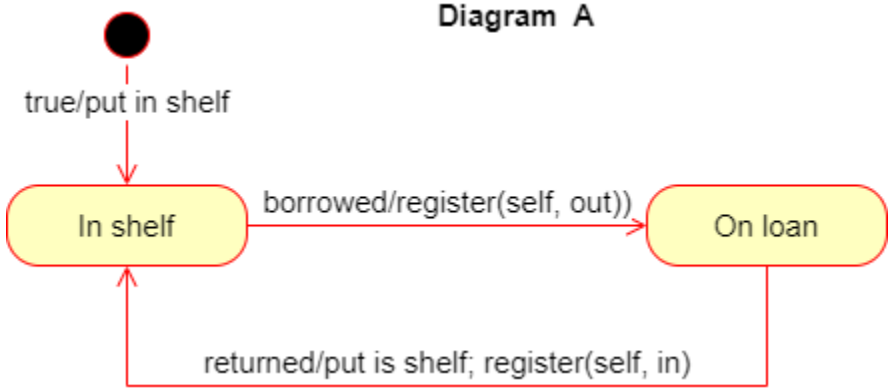


Diagram B

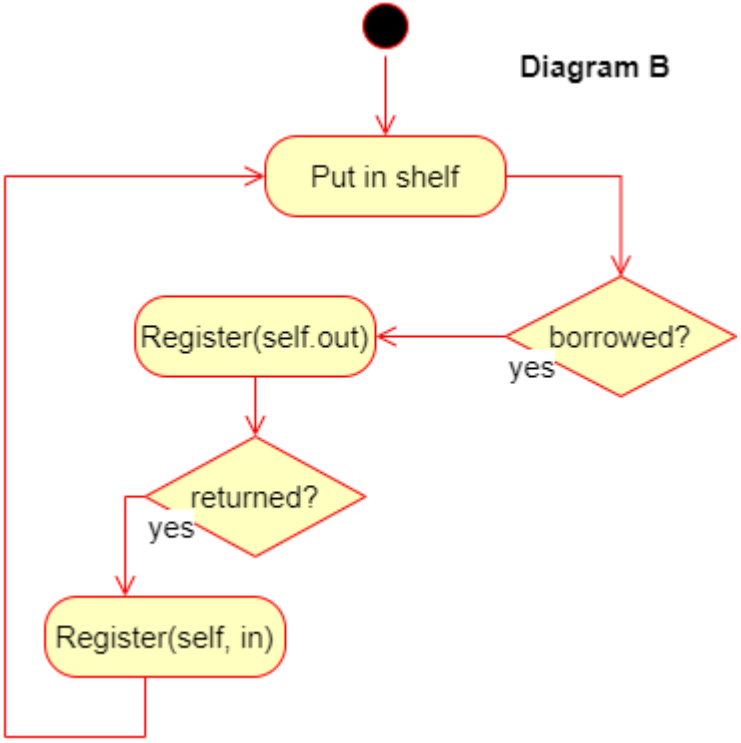


Diagram B shows as problematic solution.
Feedback:

This is more like an activity diagram where the states describe activities, not the state of the class Book.

There are no real transitions where an event triggers a behavioral change.

The diamonds try to mimic the events, and the following activity becomes the action.

The diamonds have no alternatives for the case when the condition is not met.

Grading:

1 p for identifying B as problematic

1 p per reasonable piece of feed-back that will help the student to understand what's wrong.

1 a) Which of the following statements are true? Two statements are true. Wrong answers give minus credits. (2)

- ❑ We say that two requirements are testable if, and only if, they can always be satisfied simultaneously
- ✓ ❑ Prototyping can be used both for requirements elicitation and requirements validation.
- ❑ An Entity-Relationship diagram is useful when we want to describe the dynamic behaviour of an embedded control system
- ✓ ❑ The IEEE Standard 830 for Software Requirements Specification encourages that you to adapt the disposition of headlines to your particular application.

1 b) *Scenario:* You are developing a home page for a consumer product test magazine that provides readers with results of professionally performed tests with recommendations. Anyone can see headlines, read summaries of the tests, and buy an article containing the full test with a credit card. The purchased article can be accessed for seven days. Subscribers have full access to all test articles during the subscription period and can also make comments under the articles sharing their own experience of the product with other subscribers. The editor publishes the tests and writes the summaries and explaining information about the test. The editor can remove subscribers' comments and archive outdated tests.

Task: Your task is to draw a UML use-case diagram of the site comprising at least two different use-cases and two different actors. Don't forget the use-case texts (also known as descriptions)

Use-case: Buy a test

Actor: Reader

The reader opens the home page.

The reader selects which tests that will be bought.

The reader will be prompted for personal details.

The reader clicks “Continue” and is prompted for credit card details.

The reader clicks “Pay” and is shown a confirmation page from the bank.

The reader acknowledges the payment with mobile ID and goes back to the payment page.

After the order is processed, a link with the articles is sent to the reader’s mail.

Use-case: Publish a test

Actor: Editor

The editor receives an article from an author

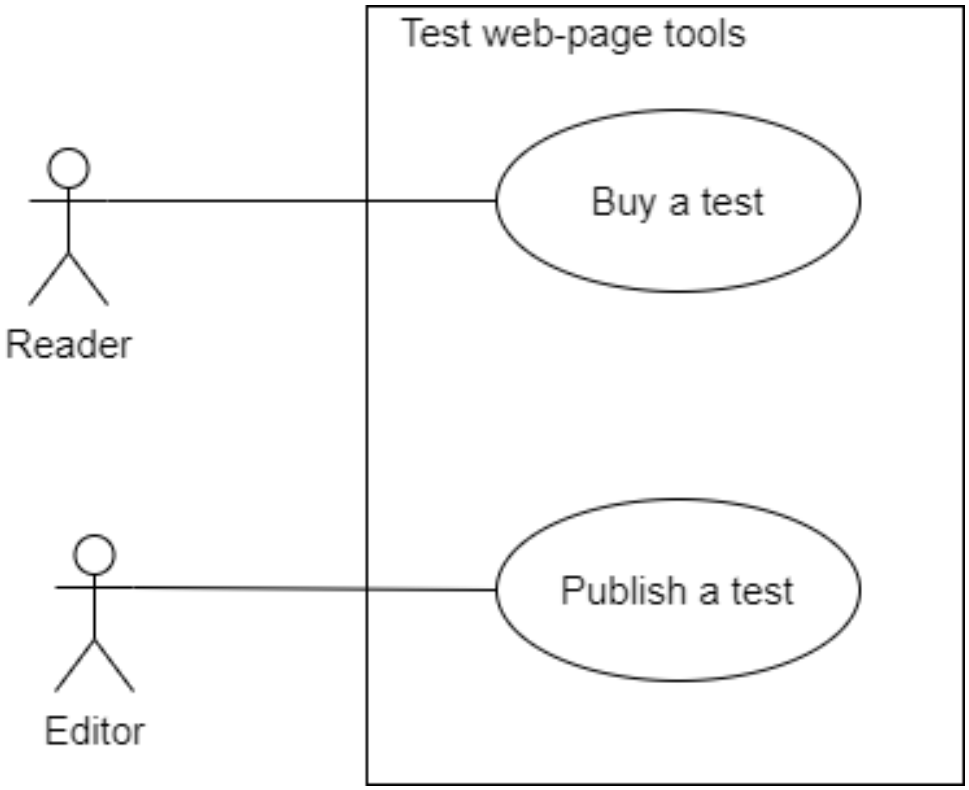
The editor reads and edits the article and writes a summary and explaining information

The editor logs into the publish page

The editor uploads the content

The editor presses the “Send” button

The editor checks the result in a web browser.


Test web-page tools
Buy a test
Reader
Publish a test
Editor

Grading:

2p per good use-case

at least 3 sentences per use-case, there shall be a clear start and stop

actors are roles, not user1 and user2

actor can be a sub-system

Two single use-case diagrams minus 2 p.

Only a correct diagram 1 p. Diagram missing minus 1 p

Use-case name: a verb phrase, if not minus 1 p

1 c) Define the following concepts in the context of requirements engineering: non-functional requirements, unambiguous requirement, user story, human bias. About 1-2 sentences per concept is probably enough.(4)

Non-functional requirements can be a quality requirement stating targets of quality factors or a design constraint that limits the solution space of the system.

An unambiguous requirement can only be interpreted in one way by the readers.

A user story describes the behavior of the system at low level from the perspective of a stakeholder. Priority and estimate are added.

Human bias is that we interpret information from our already presumed conception of the phenomenon we are studying.

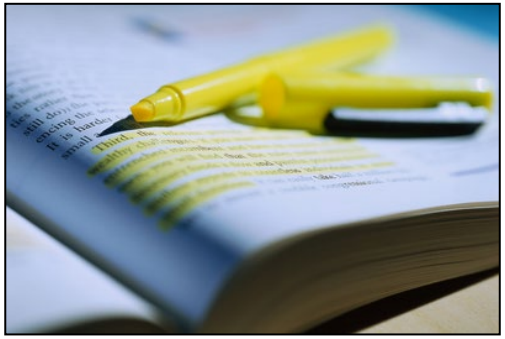
1p per sensible explanation. Well-chosen examples are OK

Marks

Total credits	Mark
0-49	U
50-66	3
67-83	4
84-	5

Allowed aids

- Two sheets of **handwritten** A4 papers (you can write on both sides)
- One volume of dictionary to or from English or an English wordbook.



Explicitly forbidden aids

- Textbook
- Machine-written pages
- Printout from drawing software
- Photocopied pages
- Pages of another format than A4
- Electronic equipment

Hints

- Register for the exam
- If necessary, reserve a guest computer
- Never guess on the first multiple-choice questions of each KA
- Bring a good pen for diagrams and sketches
- Use ergonomic aids
- Have the nerve to read through the exam first
- Use time-boxing and buffer time
- Do as the exam invigilators say
- If you have questions, wait until we pass your table.

Thanks for listening!

GOOD LUCK!