Grading instructions to exam 2017-10-19

There are many different solutions possible. Misspellings and grammar errors do appear

8. *Scenario:* Oh, no! After the election in 2018 Sweden got a prime minister who likes to post drastic and provocative messages on Facebook. This creates lot of unnecessary conflicts that aren’t even appreciated within the prime minister’s own party.

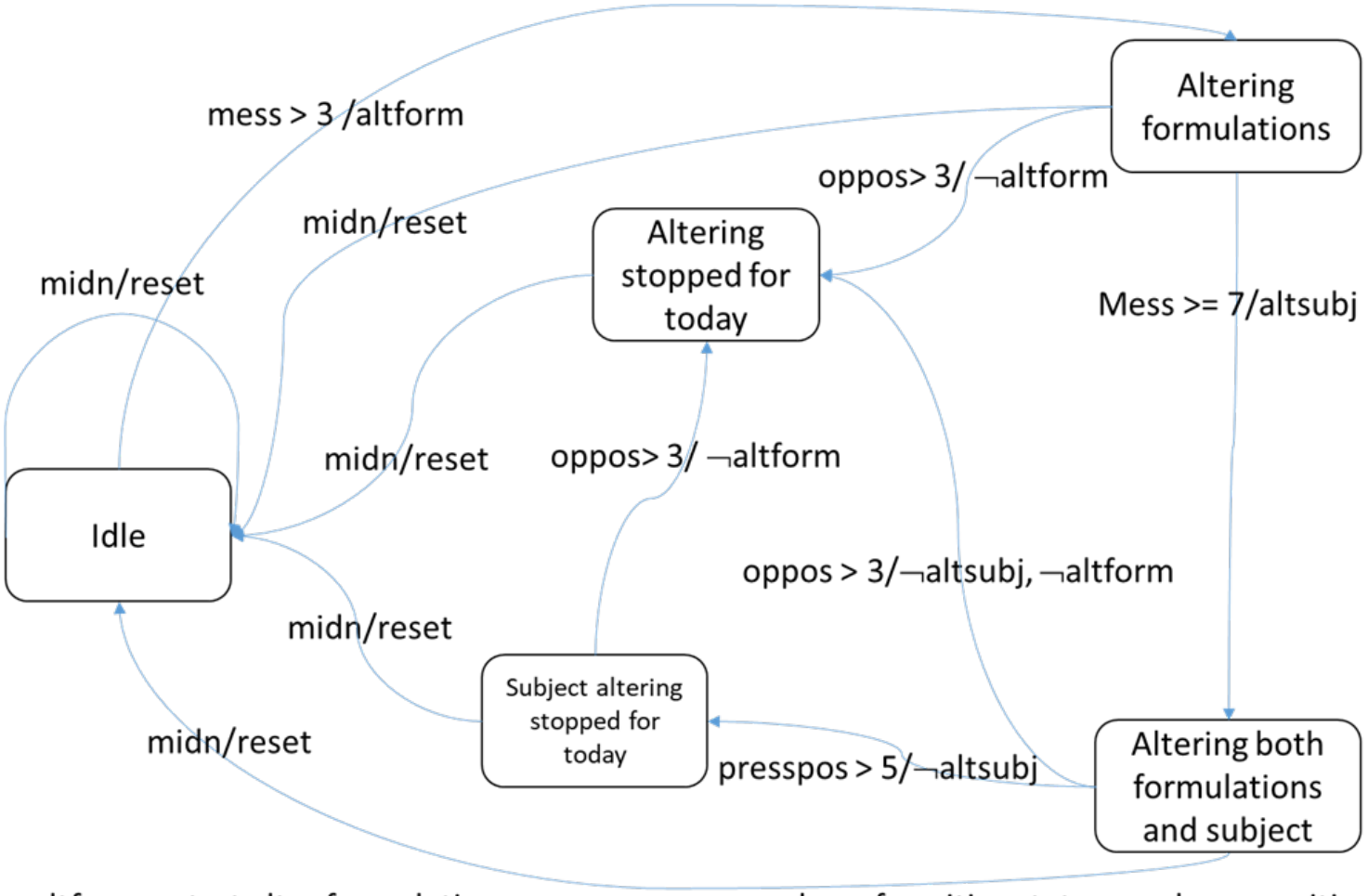
Luckily you have hacked the prime minister’s account and you have planted a softbot that alters the formulations or even choice of subject of the messages so that they are aligned with a more normal debate climate. But, the softbot has to be careful of not overusing its capacity, because then its existence might be revealed. The softbot has a partner that watches the debate in the press.

The specification of the softbot is:

1) Never change the first message of the day.
2) After three messages have been posted in a single day, start altering the formulations.
3) If seven or more messages have been posted during a single day, start altering both the formulations and subjects.
4) If both the formulations and subjects are altered and the press writes more than five positive articles about the prime minister, stop altering subjects.
5) If formulations are altered and more than three positive statements about the prime minister from the opposition are cited in the press, stop all alterations.
6) At 24:00 o'clock every night, the system is reset and no alterations are made.

Task: Draw a UML state diagram showing the behavior of the softbot. (10)

Assumptions:
we reset also number of articles in press and oppositon at midnight (not specified)



altform = start alter formulations
altsubj = start alter subjects
midn = current time = 24:00
reset = mess, oppos, presspos := 0

oppos = number of positive statemens by opposition
presspos = number of positve articles in press
mess = number of messages posted by PM

-1 credit per missing requirement
-1 credit per missing state
-1 credit per semantic ULM fault
-1 credit per syntactic UML faults, but only one instance. If for example, the event and actions are interchanged on several transitions. It’s still only -1 credit

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

Task: Identify input and output variables of the softbot. Identify five equivalence classes and write a test case for each of the equivalence classes. (10)

Input variables (refer to the abbreviations in the state diagram)

Mess
Presspos
Oppos
Current time
Current state

Output variables:
Altform
Altsubj
Next state

State abbreviations:
I = Idle
AF = Altering formulations
AFS = Altering both formulations and subjec
SAS = Subject altering stopped for today
AS = Altering stopped for today

Equivalence classes (per state and variable)
EC1: Current time < 24:00
EC2: Current time = 24.00 (it is also possible to regard time as a system mode)
EC3: Current state = I & 0 <= Mess <= 3
EC4: Mess < 0 (invalid)
EC5: Current state = AF & 3 < Mess < 7
EC6: Current state = AF & 7 < =Mess
EC7: Current state = AFS & 0 <= Presspos <= 5
EC8: Presspos < 0 (invalid)
EC9: Current state = AFS & 5 < Presspos
EC10a: Current state = AF & 0 <= Oppos <= 3
EC10b: Current state = AFS & 0 <= Oppos <= 3
EC10c: Current state = SAS & 0 <= Oppos <= 3
EC11: Oppos < 0 (invalid)
EC12a: Current state = AF & 3 < Oppos
EC12b: Current state = AFS & 3 < Oppos
EC12c: Current state = SAS & 3 < Oppos

Test table

Id	Current state	Mess	Presspos	Oppos	Time	Altform	Altsubj	Next state	EC
1	I	2	0	0	10:00	no	no	I	EC3
2	AF	4	0	0	11:00	yes	no	AF	EC5
3	AF	8	0	0	12:00	yes	yes	AFS	EC6
4	AFS	8	6	0	13:00	yes	no	SAS	EC9
5	AFS	8	0	4	14:00	no	no	AS	EC12b

Other solutions exist.

EC3: Current state = I & 0 <= Mess <= 3
EC5: Current state = AF & 3 < Mess < 7
EC6: Current state = AF & 7 < =Mess
EC9: Current state = AFS & 5 < Presspos
EC12b: Current state = AFS & 3 < Oppos

1 credit per Equivalence class
1 credit per correct test case

This was an easy solution. For some test cases you need to know the input state and the output state as well.

-1 no id
-1 for wrong relations per variable
-2 for not have all input variables in the table
-1 missing variables
-2 missing output

7. SCRUM is a popular framework for organizing development teams in software engineering. Select five concepts that are prescribed by SCRUM. For each of the concepts, write down:

a) A description of the concept.
b) A motivation of why the concept is agile (follows the agile manifesto).
c) A motivation of the benefits of using the concept from the team perspective.
d) A motivation of the benefits of using the concept from the customer perspective.

(20)

Sprint:

a) A sprint is a time-boxed iteration where the team is working on product backlog items. Normally 1-4 weeks.

b) It's agile in the sense that the work is divided into several iterations where the work is evaluated after each iteration.

c) The benefit for the team is that they will have a clear focus, hopefully undisturbed, for a couple of weeks.

d) The benefit for the customer is the possibility to influence the product backlog frequently.

SCRUM-meeting:

a) A SCRUM meeting is a short daily, stand-up meeting where the team-members answer three questions: What have I done since last meeting? What will I do until next meeting? Are there any problems?

b) It's agile since it promotes direct communication over written reports.

c) The benefit for the team is that everyone is updated with first-hand information frequently. It's possible to deal with problems quickly.

d) The benefits for the customer is indirect; things that can be problematic get resolved fast.

The team:

a) The team is itself a role in SCRUM. The team has the cross-functional knowledge needed and is self-organized.
b) It's agile in the sense that collaboration between individuals are prioritized.
c) The benefit for the team is that it will have the competence close at hand and can avoid searching for experts.
d) Since a well-working team will develop good software fast, the customer can quickly deploy the result of the software. Potentially, the price can be lowered, but that depends on the competitiveness of the market.

Product backlog:

a) The product backlog is a list of items that need to be performed in the project. The items are sorted in order of priority.

b) It's agile if the backlog contains user stories that are fairly independent and can be re-poetized often. It can also replace detailed requirements and detailed release plans. *It can, however, be misused if the items are large and complex.*

c) The benefit for the team is that they are trusted to make estimates, and can be sure to always focus on the most valuable work.

e) The benefit for the customer is the possibility to influence the product backlog through the product owner.

Sprint retrospective

a) A sprint retrospective is a meeting where the team evaluates and possibly improves its way of working after each sprint.

b) It's agile in the sense that also the way of working gets updated in small, frequent iterations, just as the product.

c) The benefit for the team is that this allows tuning ways of working frequently, which leads to efficiency *and hopefully job satisfaction.*

d) Since a well-working team will develop good software fast, the customer can quickly deploy the result of the software. Potentially, the price can be lowered, but that depends on the competitiveness of the market.

1 credit per sensible item a)-d) if the concept is prescribed in SCRUM. Max 4 credits per concept.

6. *Scenario:* Your company of 10 skilled software engineers has been assigned to develop a safety critical software in a medical equipment. The calendar time for the development is not too short, but there is no room for unnecessary delays. After the start in January 2018 you are supposed to deliver in June 2018. Hence, you have started making preparations by:

- Sending one of your developers to a course in formal methods for requirements engineering.
- Placing an order for a logic inference tool to help you proving the consistency and correctness of your requirements.
- Outsourcing the coding to a company in India that is evaluated on the CMMI level 5.
- Contracting a medical expert of 100 hours to be used in different types of reviews and testing.

Task: Identify two project-specific risks with high risk magnitude. Create four different plans per risk: avoidance, transfer, mitigation, and a contingency plan. The risks shall use information from the scenario.
(10)

Examples:

Risk: The use of formal methods might prolong the development time.
Plan:

- Avoidance: Stick to known methods e.g. inspections and simulation.
- Transfer: Hire an expert consultant to work with formalization and verification.
- Mitigation: Start a study circle in using the tool with known requirements.
- Contingency plan: Send more people to the course so they can help each other.

Risk: There are communication problems between your company and the vendor in India.
Plan:

- Avoidance: Let your own company also do the coding.
- Transfer: Outsource to a company in your own cultural neighborhood.
- Mitigation: Arrange visits on both sites for people involved.
- Contingency plan: Let your own company also do the coding and pay overtime.

Buy a tool with code generation ability

1 credit per sensible, project specific risk
1 credit per sensible item in the plans
We accept several risks but they only get 2 credits, so just phrasing 10 risks will only give 2 credits.

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

A. The duty of the inspectors in an inspection process is to resolve as many defects as possible found in the inspected artifact.
B. The inspection leader (moderator) can act as a recorder in an inspection process.
C. The author does not report known or recently found bugs, they are handled separately.
D. The reader keeps the pace of the inspection meeting to ensure that people are not stuck in details.

B, D

5 b) *Scenario:* You have a fairly large development company and you have a good reputation when it comes to understand the customers' true needs and you deliver fast. However, you need to spend too much time in repairing defects of the products, since your reliability of the products vary a lot.

Task: Describe a CMMI process area that you think would help you to deliver less faulty software. Summarize the process area's purpose and introductory notes in 5-6 sentences, and motivate how this could help you. (4)

Example:

Verification.

Verification means to check that different work products meet the requirements. Typical activities are testing and reviews. These activities are performed iteratively during the entire development. For each activity you set up a verification environment and specify procedures and criteria. Verification does not only apply to code, but also other artefacts, such as architecture. The choice of which products that shall be verified, and selection of suitable verification methods is essential.

This will help me to get a systematic and repeatable process for testing and reviews. It is especially important to select well-defined criteria for verification to avoid variance in quality.

2 credits for a good description that carries more information that just the purpose and general statements

1 credit purpose only

2 credits for a good motivation

5 c) Describe two metrics that can help you to measure the usability of a product developed in an iterative fashion. It’s important that the metrics can be applied on prototypes of different fidelity and maturity through the project. (4)

Description: Number of good and bad features recalled by users.
How to obtain data: Set up a test scenario. Let test users run the scenario. Collect number of good and bad features in a questionnaire afterwards. This can be run also with paper prototypes.
How to calculate the metric: Take the average of number of good and no. bad features. Two values.
Relevant quality factor: Relevance – many good and few bad features indicates a good match with the users' mind-set.

Description: Frequency of help and documentation use
How to obtain data: Set up a test scenario. Let test users run the scenario. Let the facilitator measure how many times the users search for help or documentation during the scenario. This can be done also with paper prototypes, even if documentation has to be oral.
How to calculate the metric: Take the average number of times the help or documentation was used.
Relevant quality factor: Efficiency – if the user has to consult the help or documentation relatively often, it slows down the processing time per task.

2 credits for a well-motivated metric as above. Can I measure this?
If it's obvious that the metric can be collected with prototypes, no motivation is needed.
It's more problematic if the student measures time without commenting that this is only a relative measure.
-1 for no motivation of usability

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

A. The cost of using a bottom-up integration testing strategy is the need to develop drivers.
B. A benefit of using a top-down integration testing strategy is that defects of lower-levels off-the-shelf components can be found early.
C. The cost of using a big-bang integration testing strategy is the need to implement both stubs and drivers.
D. A benefit of using a sandwich integration testing strategy is that it works well for large systems with many levels in the decomposition tree.

A, D

4 b) Describe the four types of acceptance testing:
benchmark test, alpha test, beta test, and parallel testing.
(4)

Benchmark, run a set of standard test cases. Mostly used to compare different solutions.

Alpha test, test by running the system under conditions as similar to regular use as possible. The environment is typically a development environment, or another controlled environment.

Beta test, test by running the system under conditions as similar to regular use as possible. The environment is at one or several customers' sites.

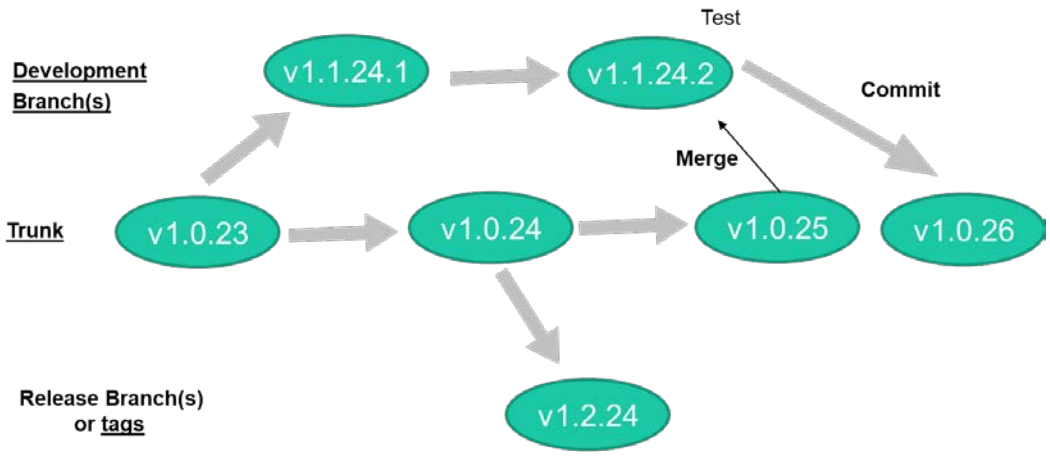
Parallel testing, test a new system by running the old system in parallel for a period of time and compare results and/or performance.

1 credit per correct description

4 c) Explain with an example the following parts of a history tree in version handling: trunk, development branch, merge, and release branch. (4)

Trunk: the mainline where all work are integrated
Development branch: A special branch for development of certain functions or correcting bugs in parallel to the trunk.
Merge: contributions from two (or more) branches are combined, conflicts might show
Release branch: a branch from the trunk that packages a release, preceded by baseline testing. Alternative formulation: the release is stable

History Tree



1 credit per correct explanation

We can accept a diagram only if numbering is correct.

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

A. An implementation (code) view of a system can be illustrated with a UML package diagram.
B. An execution view of a system shows which source code artifacts that shall be compiled together.
C. A deployment view of a system shows on which hardware different components shall execute.
D. An implementation (code) view of a system can be used to analyze the need of bandwidth and processing power for a client-server architecture.

A, C

3 b) Give two examples of things you can do when you create the architecture that can improve the maintainability of the system. Clearly motivate your answer (Hint: a clear motivation is typically 3-5 sentences.) (4)

Examples:

Low coupling. If there is a limited way components depend on each other the understandability of the entire architecture increases. It is thus relatively easy to localize the component that need to be changed. Moreover, the time it takes to follow up on whether a change in a component implies a need of change in another component decreases. The risk of having merge conflicts decreases if we are not changing many parts of the system.

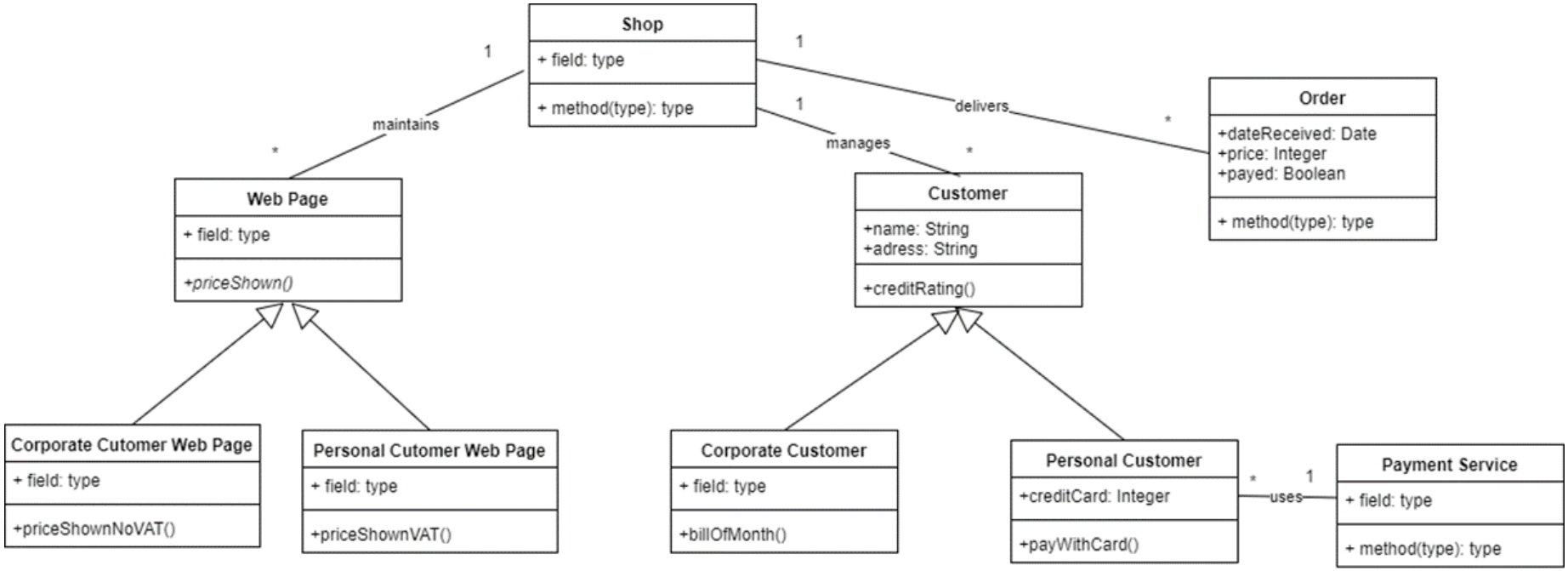
High cohesion. If all sub-parts of the components are strongly related, we have probably done a good analysis and separation of the most important functions. This will increase the understandability of the purpose and construction of the components once they are identified. It also contributes to make changes local, which eases navigation in the code as well as regression testing. If a change occurs it is likely to affect only one or a few components

There are many variants such as: Use a layered architecture, use well-known interfaces, use design patterns, use abstractions, and design an architecture with good traceability to code.

2 credits per well motivated thing to do
-1 credit per short motivation

3 c) *Scenario:* A shop has two kinds of customers: corporate customers and personal customers. The customers place orders for the shop and the shop stores their name, address, and credit rating. At the shop's web-pages, a corporate customer is shown prices without VAT, whereas a personal customer is shown the price including the VAT. Another difference is that a corporate customer gets monthly bills whereas a personal customer pays with credit card using a third-party paying service.

Task: Draw a UML class diagram describing the situation above. Use at least one generalization association. Don't forget to put name and multiplicity information on the other associations. (4)



Straightforward solution:
It is of course also allowed to write an implementation model, where management software and databases are visible. We also buy other solutions, such as using explicit attributes.

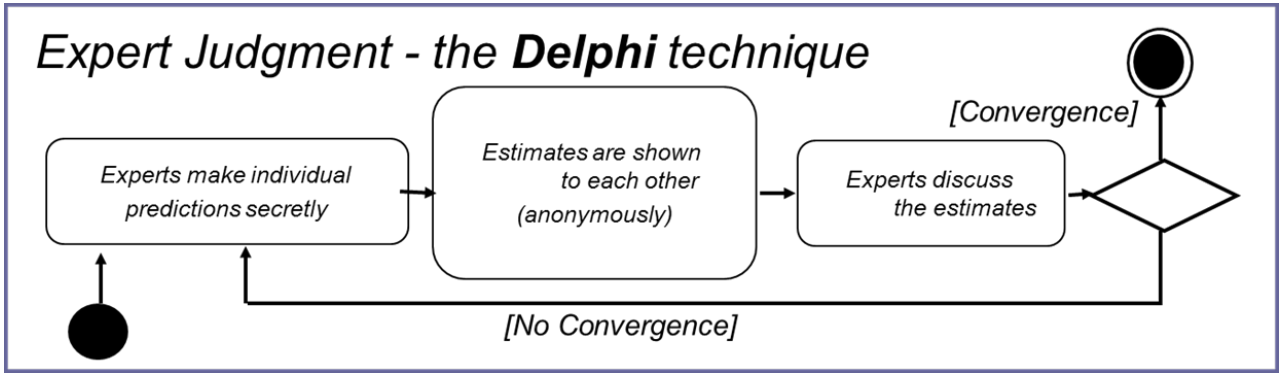
If they fulfill all requirements above and no UML mistakes they get 4 credits.
-1 credit per failed requirement
-1 credit per type of syntactic of semantic mistake, but only one minus per misunderstanding. For instance, assume that they have turned two generalization associations upside down, then we withdraw 1 credit, for the wrong understanding. But we don’t withdraw 2 credits even though the problem was manifested twice.

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

A. The buffer time is the difference between internal and external deadlines.
B. A toll-gate is described by activities that has to be finished before a certain date.
C. The critical path in a Gantt-chart contains activities performed by the most expensive resources.
D. A consequence of time-boxing can be that you need to postpone the implementation of low-priority functions.

A, D

2b) Describe the Delphi method for effort estimation. Describe a potential limitation of the method. (4)



A potential limitation can be:

- The team of experts might lack all competence needed;
- It takes calendar time to book the busy experts for the job; or
- After a while experts might learn to make conflict-avoiding estimates.

1 credit per described activity in the activity diagram.

1 credit for a sensible limitation.

2c) Describe two advantages and two disadvantages of moving from the classical waterfall model to an iterative model. (4)

Advantages:

- Misunderstandings and inconsistency are made clear early (e.g. between requirement, design, and implementation)
- Encourage to use feedback -> elicit the real requirements
- Forced to focus on the most critical issues
- Continuous testing offers project assessment
- Workload is spread out over time (especially test)
- The team can get "lesson learned" and continuously improve the process
- Stakeholders get concrete evidence of progress

Disadvantages:

- Problem with current business contracts, especially fixed-price contracts.
- With short iterations it can be hard to map customer requirements to iterations.
- Overhead added
- Requirements selection problem
- Stressful learning period

2 credits per sensible advantage, max 2 credits
2 credits per sensible disadvantage, max 2 credits

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

A. The RAM model is focused on handling requirements of memory management of parallel execution.
B. In the analysis phase of requirements engineering you use interviews and observation techniques to determine the true needs of the customer.
C. Software reviews can be used to validate a requirements specification.
D. Traceability of a requirement can be a link to the elements of the design that contribute in the realization of the requirement.

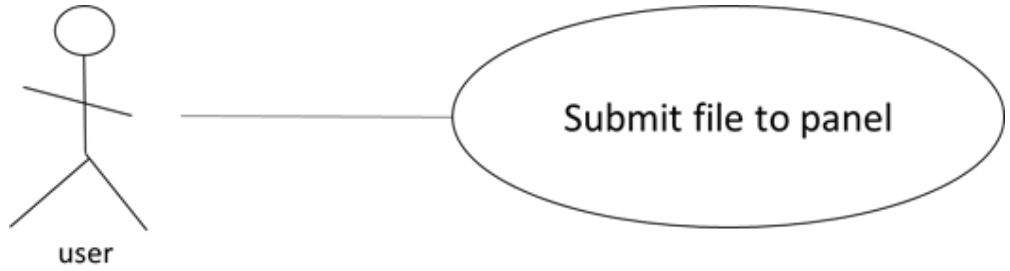
C, D

1 b) *Scenario:* In a social media system users can upload pictures, short movies, and sound files. There are features of basic editing of the media files as well as sharing and commenting functions. You have a detailed personal profile including your ambition level and dreams when it comes to media production. It’s possible to submit files to panels of well-renowned members that can give you constructive criticism on how to improve your media production.

Task 1: Write a use-case of the system described above and draw a UML use-case diagram. Write two user stories of the above system where at least one is related to the use-case. (4)

Use-case: Submit file to panel.
Actor: User
Text:
The user logs in to the system.
The user navigates to the album of the media files for submission.
The user selects the media files to submit.
The user selects the panel from a dropdown list.
The user clicks on the SEND button and gets a confirmation question.
The user answers yes.
The media files are submitted.
The user gets a receipt of submission.

Diagram:



User stories:

As a user I want to submit photos for judgement so that I can get feed-back to improve.
Priority: x, Estimate y

As a user I want to upload photos so I can inspire my connections.
Priority: z, Estimate w

Use-case text of 3-5 sentences, 1credit.
Use-case diagram with correct actor and a verb phrase as a title, 1 credit.
We accept if the student draws a system boundary around the use-case.
Two correct user stories, I credit each.
Students must know the difference between the terms use-case and user stories.

Task 2: Write a quality requirement and one design constraint of the system described above. For each of the two requirements, write a short motivation on which level of testing they can be tested. (Hint: a motivation is typically 1-2 sentences.) (4)

QR: The system must be available 160 hours per week.

This requirement is tested during acceptance testing, since the system must be delivered to the customer's hardware to give a correct availability measurement.

DC: The client shall be developed in JavaScript.

This requirement can be tested already at the implementation by reviewing the files checked in. Alternatively it can be double-checked at integration testing.

1 credit for a testable quality requirement.
1 credit for a design constraint.
1 credit for each sensible motivation.

Pass condition

To pass the exam (alternatives)

1. a) at least 4 credits in all areas in fundamentals **and**
   b) at least 50 credits in total
2. a) at least 4 credits in at least 4 areas **and**
   b) at least 60 credits in total

Part I: Fundamentals

- Requirements
- Planning and Processes
- Design and Architecture
- Testing and SCM
- Software Quality

10 credits per area. Max 50 credits.

Part II: Advanced

50 credits, distributed over 2-5 questions.

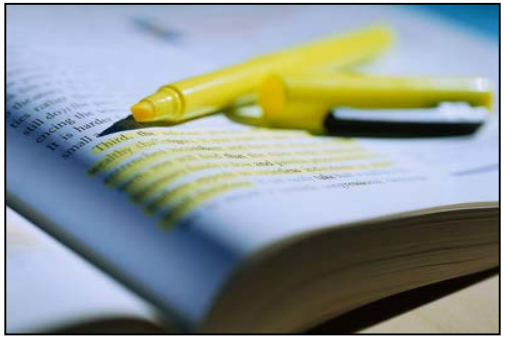
- argue, compare, and analyze different concepts and techniques.
- construct and/or design solutions to larger problem.
- explain more advanced and specific topics.

Grades if pass condition is met

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

Allowed aids

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



Explicitly forbidden aids

- Textbook
- Machine-written pages
- Photocopied pages
- Pages of other format than A4
- Electronic equipment

Hints

- Register for the exam
- Never guess on two alternatives of multiple-choice questions
- Use a pencil
- Use ergonomic aids
- Have the nerve to read through the exam first
- Use time-boxing and buffer time
- Do as the exam invigilators say

Thanks for listening!

GOOD LUCK!