Item 1

1 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score. (2)

A	*End-to-end traceability* necessitates to link requirements with source code and test cases.
B	Functional requirements are *user requirements* while non-functional requirements are *system requirements*.
C	In a high-quality *requirements specification document*, requirements should be numbered and prioritized.
D	In a UML Use Case diagram, the *system boundary* should include all the relevant actors and use cases.

1 b) *Scenario*:
The platform diving scoring system (PDSS) at the Paris 2024 Olympics is responsible for calculating the score and ranking of athletes competing in platform diving. Platform diving has races for individual and synchronized events separately for men and women divers. Each race consists of a qualifier and a final. Regardless of event, each male diver needs to make 6 dives while each female diver needs to make 5 dives.

In individual race, each dive is scored individually by seven jury members, on a scale of 1-10, including half-point margins. The highest two scores and lowest two scores are eliminated. The remaining three scores are added together to achieve the execution score, which is multiplied by the dive's degree of difficulty to determine the total score of the dive. In synchronized diving events, pairs of divers compete, but the exact scoring is not detailed here further.

After each round of dives, the system needs to calculate and display the current ranking (highest score is better). In the qualifier, the best eight divers qualify to the final where they compete for the gold, silver and bronze medals (for 1st, 2nd and 3rd place).

Task: Create a *UML Use Case diagram* for the PDSS with key actors and use cases. *Give appropriate names for these actors and use cases*, but *no detailed descriptions are needed*. (Logging in and logging out are basic functions, not to be considered as individual use-cases.)
Append a UML Use Case diagram and write in the text which appendix that contains the answer to this question. (4)

1 c) *Scenario*: Same as in Problem 1 b) above

Task: Write *two user stories* and *one non-functional requirement* for the PDSS system. You should follow the best practices for writing user stories and natural language requirements. Name the *high-level software quality factor* related to the non-functional requirement in accordance with the classification of the ISO/IEC 25010 standard. (4)

Item 2

2 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score. (2)

A	Achieving both *high cohesion* and *high coupling between modules* (at the same time) are beneficial for complex software systems.
B	The *Façade design pattern* helps reduce dependencies between subsystems.
C	Typically, there are more *concrete states* of an object than *abstract states*.
D	*Design models* and *implementation models* provide approximately the same level of details.

2 b) Consider that a modern *mobile application* is being developed for the judges of the Platform Diving Scoring System (PDSS) (described in Section 1).
Decide whether you would apply each of the following architectural styles for the architecture of this web-based PDSS system. Give a short justification for each of your decision, i.e. why the specific architectural style is applicable in the context of TMS, or why it is not meaningful. (4)

Model-View-Controller, *Pipes and Filter* (Two separate questions!)

B I U á

0 / 10000 Word Limit

2 c) Describe two key characteristics of a (functional) *software component*? How can one address *fault-tolerance requirements when allocating software components* to the hardware platform? (4)

B I U á

0 / 10000 Word Limit

Item 3

3 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score. (2)

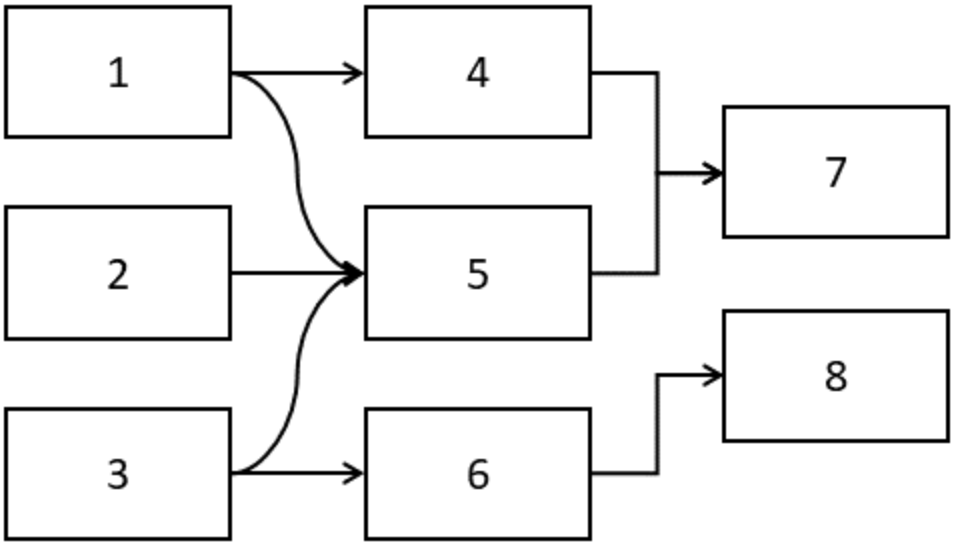
A *Top-down testing* is a good strategy for the integration testing of complex user interfaces.

B *Continuous integration* is especially challenging for small teams.

C In testing, an *oracle* helps decide what test inputs need to be investigated

D Handling *stateful services* is particularly challenging for continuous deployment.

3 b) Consider the following dependency graph for between software components identified by numbers 1-8. For example, an arrow from Component 1 to Component 4 indicates that Component 1 depends on (or requires service from) Component 4.



An integration testing strategy consists of a sequence of steps where each steps clearly identify the components that are integrated and tested together in that step. For example:
{1}, {1,2}, {1,2,3} means that first Component 1 is tested in isolation, then Component 1 and Component 2 are tested together, then Components 1,2 and 3 are tested together.

Task: Mark all the strategies below which comply with a *bottom-up integration testing* approach. There are exactly two correct answers, wrong answers give negative score.

A	{7,8}, {4,6,7,8}, {1,3,4,6,7,8}, {1,2,3,4,5,6,7,8}
B	{7,8}, {4,5,6,7,8}, {1,2,3,4,5,6,7,8}
C	{8}, {6,8}, {3,6,8}, {3,6,7,8}, {3,4,5,6,7,8}, {1,2,3,4,5,6,7,8}
D	{7}, {5,7}, {2,5,7}, {2,4,5,7}, {1,2,4,5,7}, {1,2,4,5,7,8}, {1,2,4,5,6,7,8}, {1,2,3,4,5,6,7,8}

3 c) Describe the key ideas and workflow of *test-driven development.* (3)

B *I* U á

0 / 10000 Word Limit

3 d) Describe the *workflow of using feature branches* in Git which is an open-source distributed version control system. Mention *one advantage* and *one disadvantage* of this workflow. (3)

Item 4

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

A	Activities on the *critical path* must be done on time; otherwise, the entire project will be delayed.
B	A useful property of a *Gantt chart* is that you can mark the current date in the diagram and get a good picture of the progress so far.
C	The *available time* for a task can be calculated as the difference between *float time* and *slack time*.
D	A *burn-down chart* as used in SCRUM keeps track of the financial expenditure of the project.

4 b) The *classical waterfall life-cycle model* consists of several distinct *phases*, such as *requirements*, *architecture*, *design*, etc. Describe the principles of the *classical waterfall life-cycle model,* that is, how it works. Focus on the *principles*, not the phases. Moreover, write down two advantages of the *classical waterfall model*. (4)

B I U á

0 / 10000 Word Limit

4 c) Describe the four different meetings of SCRUM: *daily SCRUM, sprint planning meeting, sprint review meeting,* and *sprint retrospective*. (4)

B I U á

0 / 10000 Word Limit

Item 5

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

A	The Acronym PDCA as used in the Shewhart Cycle means Program, Deploy, Compile, and Analyze.
B	In a mature organization all quality factors are equally important and must be treated in a similar way.
C	In ISO 9000-3 it is important not to press suppliers too hard to get the lowest prices. They need to keep some room for innovation to the benefit of both suppliers and buyers.
D	In a *usage-based* view of product quality, there can be different opinions about quality, which means that you need statistics do make an objective statement of the quality.

5 b) Suppose that you work in a company at *CMMI level* 1 that has high requirements on *usability* of the products. Select the two most important *process areas* of *CMMI level* 2 or 3, that you would like to focus on first to meet those high requirements. Briefly describe the process areas and motivate how they can contribute to high *usability*. (4)

B I U á

0 / 10000 Word Limit

5 c) Compare the two software review methods *inspection* and *code peer-review*. The code peer review is organized in such a way that at least one other team member must approve a change made by someone in the team. Draw a table with the review methods in the columns and comparison criteria as the rows. The comparison criteria are: resources needed, time duration, type of artefacts reviewed, and type of *defects* that can be detected. (4)

B I U á

0 / 10000 Word Limit

Item 6

6. *Scenario*: (extension of scenario in Section 1)
The platform diving scoring system (PDSS) at the Paris 2024 Olympics is responsible for calculating the score and ranking of athletes competing in platform diving. Platform diving has races for individual and synchronized events separately for men and women divers. Each race consists of a qualifier and a final. Regardless of event, each male diver needs to make 6 dives while each female diver needs to make 5 dives.

In an individual race, each dive is scored individually by seven jury members, on a scale of 1-10, including half-point margins. The highest two scores and lowest two scores are eliminated. The remaining three scores are added together to achieve the execution score, which is multiplied by the dive's degree of difficulty to determine the total score of the dive.

Pairs of divers compete in synchronized diving events, which are scored by a total of 11 judges. Three judges focus on scoring the execution of each individual diver of the pair (a total of six judges). The remaining five judges score the pair's synchronization. Of the 11 total judges' scores, five scores are used: the median execution score for Diver 1, the median execution score for Diver 2, and the middle three synchronization scores. The five scores are then added together, multiplied by 0.6 (to more closely align with scores from the individual events), then multiplied by the dive's degree of difficulty to determine the total score of the dive.

After each round of dives, the system needs to calculate and display the current ranking (highest score is better). In the qualifier, the best eight divers qualify to the final where they compete for the gold, silver and bronze medals (for 1st, 2nd and 3rd place).

Task: Draw a *domain model in the form of a UML Class Diagram* for the PDSS system showing the *domain classes* and their *relationships* as well as potential *generalizations*. Specify *multiplicities* for your *associations and compositions*, and give them *meaningful names*. Provide a total of *6 attributes* mentioned in the description above *together with their type*. Give a *brief textual justification* of your *key design decisions* (one sentence per key design decision).

Append the UML Class diagram and write in the text which appendix that contains the answer to this question. (10)

Item 7

7. *Scenario*: (Same as in Section 1)

The platform diving scoring system (PDSS) at the Paris 2024 Olympics is responsible for calculating the score and ranking of athletes competing in platform diving. Platform diving has races for individual and synchronized events separately for men and women divers. Each race consists of a qualifier and a final. Regardless of event, each male diver needs to make 6 dives while each female diver needs to make 5 dives.

In individual race, each dive is scored individually by seven jury members, on a scale of 1-10, including half-point margins. The highest two scores and lowest two scores are eliminated. The remaining three scores are added together to achieve the execution score, which is multiplied by the dive's degree of difficulty to determine the total score of the dive. In synchronized diving events, pairs of divers compete, but the exact scoring is not detailed here further.

After each round of dives, the system needs to calculate and display the current ranking (highest score is better). In the qualifier, the best eight divers qualify to the final where they compete for the gold, silver and bronze medals (for 1st, 2nd and 3rd place).

Task: Describe the high-level business process (workflow) of the PDSS Ticket Management System by a *UML Activity diagram* (which may include activities performed by different actors). It is sufficient to cover only an individual event. Append your *UML Activity* diagram and write in the text which appendix that contains the answer to this question. (10)

Item 8

8. *Scenario*: A software service automatically calculates the baggage surcharge fees for flight tickets with checked in baggage according to the following rules. An Economy Class ticket includes one piece of checked-in baggage up to a weight of 23kg. A Business Class ticket includes two pieces of checked-in baggage. An Economy Light Class ticket includes no checked-in baggage, so a passenger needs to pay for each checked-in luggage. The cost of each extra baggage (over what is not included in the ticket) is 600 SEK.

No checked-in baggage may exceed 35kg in weight. If the weight of any checked-in baggage exceeds the maximum allowed weight in the given ticket class, then a surcharge of 100 SEK is paid for each extra kilogram over the limit of 23kg, *for each checked-in baggage* (i.e. not only for those which are actually over the limit).

Task: Identify input and output variables and valid and invalid equivalence classes. A valid class means that a passenger is allowed to check-in and bring the respective baggage to aircraft – potentially after paying a baggage surcharge fee. We assume that it is impossible to enter negative numbers.

Create a test table while performing weak equivalence class partitioning as it has been taught in the course. Include a test case also for each invalid equivalence class.

Item 9

9. "Productivity is a measure of performance that compares the output of a product with the input, or resources, required to produce it. The input may be labor, equipment, or money." (https://www.investopedia.com/terms/p/productivity.asp)

Describe *five different metrics or measurements* that can be used to measure *productivity of a software development organization*. The metrics or measurements can be on any level: individual, team, or system level. Use the format as was introduced on the slides in the Software Metrics lecture, that is, (i) description, (ii) how to obtain data, (iii) how to calculate the metric, (iv) relevant quality factor (in this case: productivity). (10)

B I U á T^2 T_2

0 / 10000 Word Limit

Item 10

10. Describe each of the four dependent project parameters. In addition, give *an example in the software engineering context of the Platform Diving Scoring System* (detailed in Section 6) where *two parameters are locked*, and describe what you can do with the remaining two parameters if a problematic situation appears. (10)

B I U á T^2 T_2

0 / 10000 Word Limit

Item 11

Don't write anything. The administrator will fill in your registered bonus credits for autumn 2023 here..

0 / 10000 Word Limit