Sektion 1

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

A	Since requirements elicitation happens in the early phase of software engineering, requirements are not used in *iterative software development methodologies*.
B	Three characteristics of a *good requirement* include feasibility, necessity and testability.
C	*Generalization* relation may exist between two actors or between two use cases.
D	*Prototyping* is typically part of the requirements specification document.

1 b) *Scenario*: The ticket management system (TMS) of UEFA is orchestrating the sales of tickets for football fans of the UEFA 2024 European Championship. Each match at UEFA 2024 is played on a specific date and in the stadium of a host city between the home team and the away team. The number of available tickets for a specific match depends on the capacity of the stadium in the host city. TMS allows fans to make bids for match tickets in multiple rounds, where a predefined number of tickets gets sold in each round for each match. In each round, a fan can make a bid for several matches, but at most 4 tickets can be requested for each match by the same person. Once the bidding deadline of a round has passed, the system randomly distributes the tickets available for sale in the given round among the fans who made a bid. TMS notifies if the ticket application of a person has been approved for a given match by sending an email. In such a case, the respective fan is required to buy all the requested tickets for that match by a given deadline using an external credit card payment service, otherwise, the tickets will be reallocated.

Task: Create a *UML Use Case diagram* for the TMS 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)

B I U á

0 / 10000 Word Limit

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

Task: Write *two functional requirements* (one on user-level, one on system level) and *one non-functional requirement* for the TMS system. You should follow the best practices for writing 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)

Sektion 2

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

A	The *platform description* contains the logical software modules that will be deployed on the physical architecture of the system.
B	*Block diagrams* capture the high-level dependencies between two UML Sequence Diagrams.
C	*Isolation of software components* is a key principle for safety-critical systems.
D	The *Observer design pattern* is characterized by four key classes, namely, the *Subject*, the *ConcreteSubject*, the *Observer*, and *ConcreteObserver*.

2 b) Consider that a modern, multi-tier web-based application is being developed for the Ticket Management System (as described in Section 1).

Scenario: The ticket management system (TMS) of UEFA is orchestrating the sales of tickets for football fans of the UEFA 2024 European Championship. Each match at UEFA 2024 is played on a specific date and in the stadium of a host city between the home team and the away team. The number of available tickets for a specific match depends on the capacity of the stadium in the host city. TMS allows fans to make bids for match tickets in multiple rounds, where a predefined number of tickets gets sold in each round for each match. In each round, a fan can make a bid for several matches, but at most 4 tickets can be requested for each match by the same person. Once the bidding deadline of a round has passed, the system randomly distributes the tickets available for sale in the given round among the fans who made a bid. TMS notifies if the ticket application of a person has been approved for a given match by sending an email. In such a case, the respective fan is required to buy all the requested tickets for that match by a given deadline using an external credit card payment service, otherwise, the tickets will be reallocated.

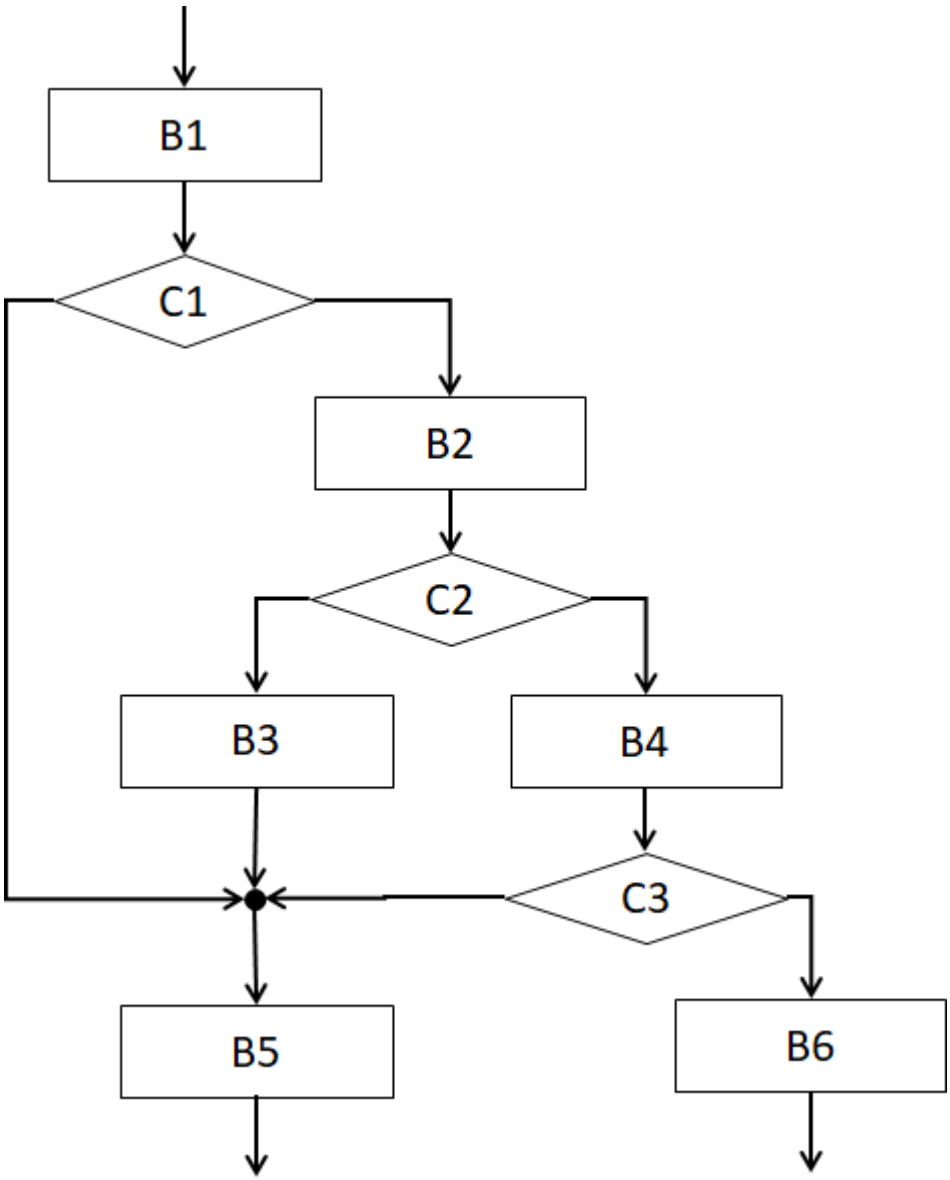
Task: Decide whether it is beneficial to apply each of the following architectural styles for the architecture of this web-based TMS system. Give a short justification for each of your decision, i.e. why the specific architectural style is applicable in the context of TMS, or why it is not meaningful. (4)
Client-Server
Pipes and Filter
(You should make separate decisions for these two architectural styles)

2 c) By attaching an illustrative example, briefly describe the following four concepts in UML Sequence Diagrams: *lifeline*, *synchronous message*, *return message*, *alt fragment*. (4)

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

A	*Equivalence partitioning* and *boundary value testing* derive identical test suites when used for testing the same functionality.
B	*Continuous integration* cannot be achieved without *continuous delivery*.
C	Specification-based (black-box) testing can reveal *missing* functionality.
D	Typically, there is no need for *stubs* in case of a *bottom-up strategy* in integration testing.

3 b) Consider the following control flow graph created from a computer program (where B1, B2, ... B6 are blocks, C1, C2, C3 are conditionals): (2)



The minimum number of test cases required to achieve *full statement coverage* is

☐.

The minimum number of test cases required to achieve *full branch coverage* is

☐.

3 c) Consider the control flow graph listed above in 3 b)
Task: Provide a *minimal test suite* that provides *full statement coverage*. Each test case shall contain all the blocks and conditionals that are tested together. (2)
For example, TC#1: {B1, B2, B3, B4, C2}.

3 d) Describe the *decentralized workflow* in Git which is an open-source distributed version control system. Mention *one advantage* and *one disadvantage* of this workflow. (4)

Sektion 4

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

- [ ] *Incremental methods* integrate one component at a time with the rest of the system.
- [ ] *Iterative development methods* focus on delivering documents, which can be reused in future projects.
- [ ] SCRUM prescribes *pair programming* as a mandatory way of working.
- [ ] *The classical waterfall model* requires an early commit to system architecture.

4 b) Describe how the *Delphi method* for effort estimation works. Write down a difference between the *Delphi method* and *Planning Poker*. (4)

B I U á

0 / 10000 Word Limit

4 c) Explain each of the four different *risk planning* methods: *Risk avoidance*, *Risk transfer*, *Risk mitigation*, *Contingency planning*. (4)

B I U á

0 / 10000 Word Limit

Sektion 5

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

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

5 b) Describe how you measure *Cyclomatic Complexity* V(G) of code. Identify two relevant quality factors that can be assessed using V(G). Remember to motivate your answer. (4)

B I U á

0 / 10000 Word Limit

5 c) What is the purpose of a *software inspection*? Select *three key roles* in software inspections and describe their *main responsibilities*. 1-2 sentences per role is enough. (4)

B I U á

0 / 10000 Word Limit

Sektion 6

6. *Scenario*: (Same as in Section 1) The ticket management system (TMS) of UEFA is orchestrating the sales of tickets for football fans of the UEFA 2024 European Championship. Each match at UEFA 2024 is played on a specific date and in the stadium of a host city between the home team and the away team. The number of available tickets for a specific match depends on the capacity of the stadium in the host city. TMS allows fans to make bids for match tickets in multiple rounds, where a predefined number of tickets gets sold in each round for each match. In each round, a fan can make a bid for several matches, but at most 4 tickets can be requested for each match by the same person. Once the bidding deadline of a round has passed, the system randomly distributes the tickets available for sale in the given round among the fans who made a bid. TMS notifies if the ticket application of a person has been approved for a given match by sending an email. In such a case, the respective fan is required to buy all the requested tickets for that match by a given deadline using an external credit card payment service, otherwise, the tickets will be reallocated.

Task: Draw a *domain model in the form of a UML Class diagram* for the GRADS 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 5 key attributes* mentioned in the description above *together with their type*. Append the UML Class diagram and write in the text which appendix that contains the answer to this question. Give a *brief textual justification* (1-2 sentence) of your *key design decisions*. (10)

Sektion 7

7. *Scenario*: (Same as in Section 1) The ticket management system (TMS) of UEFA is orchestrating the sales of tickets for football fans of the UEFA 2024 European Championship. Each match at UEFA 2024 is played on a specific date and in the stadium of a host city between the home team and the away team. The number of available tickets for a specific match depends on the capacity of the stadium in the host city. TMS allows fans to make bids for match tickets in multiple rounds, where a predefined number of tickets gets sold in each round for each match. In each round, a fan can make a bid for several matches, but at most 4 tickets can be requested for each match by the same person. Once the bidding deadline of a round has passed, the system randomly distributes the tickets available for sale in the given round among the fans who made a bid. TMS notifies if the ticket application of a person has been approved for a given match by sending an email. In such a case, the respective fan is required to buy all the requested tickets for that match by a given deadline using an external credit card payment service, otherwise, the tickets will be reallocated.

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

B I U á

0 / 10000 Word Limit

Sektion 8

8, *Scenario*: A speed control and reporting system has the following characteristics: If the owner of the vehicle is 26 years old or more, and the vehicle is **not** faster than 80km/h, then nothing happens. If the vehicle is travelling faster than 80 km/h, but not faster than 90km/h, then the owner will be warned. If the vehicle is travelling faster than 90km/h, then the owner will be fined. If the owner of the vehicle is less than 26 years old, then each respective speed limit is reduced by 10 km/h (with limits of 70km/h and 80km/h).

Task 1: Identify the *valid* and *invalid equivalence classes* for a high-level functionality *carPassed(int speed, int ageOfOwner)* which returns 0 if nothing happens, returns 1 in case of a warning and returns 2 in case of a fine. The two input variables can be treated independently from each other.
Task 2: Design an appropriate test suite using *weak equivalence class testing.* One test case should look like TC#1: (64, 32): 0 where 64 is the speed of the vehicle, 32 is the age of the driver and 0 is the expected value to be returned. (10)

B I U á

0 / 10000 Word Limit

Sektion 9

9. Select five concepts from the list below. You need to select at least one concept from each group for full credits.

For the five selected concepts, describe:

a) How it is used in software development;

b) How it contributes to software quality;

Group A

- Pair programming
- Refactoring
- Test-first programming
- User story

Group B

- Burn-down chart
- Sprint review meeting
- Sprint retrospective meeting

Group C

- Limit Work In Progress
- Lead time
- Kanban board

(10)

Sektion 10

10. Describe how you will obtain data for each of the following usability metrics. For each of the metrics also write down how it can be measured by using a low-fidelity (lo-fi) prototype. Examples of lo-fi prototypes include paper sketches, hand-drawn wireframes, and simple interactive mockups of the graphical user interface. Typical software tools can be PowerPoint, Figma, Photoshop, etc.

If you don’t think that a particular metric can be measured on lo-fi, write so with a motivation.

Metrics:

a) Number of different commands invoked by users.

b) Time to complete a predefined task.

c) Time spent in understanding error messages.

d) Frequency of help and documentation use.

e) Percent of favorable/unfavorable user comments.

(10)

0 / 10000 Word Limit

Sektion 11

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

0 / 10000 Word Limit