Item 1

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

A	*Structured scenarios* detailing use cases should always include a basic path where everything goes as expected.
B	Since *agile requirements management* is based on user stories, stakeholders do not need to be frequently involved.
C	A *feature* is a distinguishing characteristic of the system.
D	*Generalization between actors* is not allowed to be used in a UML Use Case diagram.

1 b) *Scenario*:

The Broadcasting of the 45th Chess Olympiad necessitates a new online system (BCO). The BCO system records all moves made by each chess player with the help of an electronic chess board (ECB). The ECB immediately reports each move made by a chess player to the BCO server. Moreover, the ECB gives an alert if a player makes an illegal move. The ECB also reports if checkmate (or stalemate) is achieved on the board, when the game ends immediately.

The BCO system shall keep track of the remaining thinking time of both chess players using an electronic chess clock (ECC). The arbiters set a maximum thinking time for the players on the ECC. Once the game is started, the time of the white player starts decreasing until the player presses the flip button – when his/her opponent's time starts decreasing. The ECC indicates if a player exceeds his/her thinking time (i.e. the player is flagged), then the game is stopped with the loss of the respective player.

Viewers can watch the games online. In the BCO system, each move (and the time needed for the move) is broadcast on the Internet by the BCO server but delaying the transmission by 15 minutes. The BCO server reports the final result when a game is finished (a draw, or one of the players wins by checkmate, flagging or resignation).

Task: Create a *UML Use Case diagram* for the BCO system with key actors and 6-7 use cases. Give *appropriate names for all 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 your UML Use Case diagram(s) and write in the text which appendix contains the answer to this question. (4)

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

Task: Write *two user-level functional requirements* and *one non-functional requirement* for the BCO system. You shall follow the best practices for writing natural language requirements. Name the high-level *software quality* (sub-)*characteristic* (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	The *functional architecture* of a system is designed after the overall system architecture is completed.
B	Creating a small number of large subsystems enhances *safety*.
C	Lifelines in UML Sequence diagrams typically represent the actors and the components of the system.
D	When using generalization in UML, a class should be different from its superclasses and subclasses.

2 b) *Scenario*: (exactly the same as in Section 1 b)

The Broadcasting of the 45th Chess Olympiad necessitates a new online system (BCO). The BCO system records all moves made by each chess player with the help of an electronic chess board (ECB). The ECB immediately reports each move made by a chess player to the BCO server. Moreover, the ECB gives an alert if a player makes an illegal move. The ECB also reports if checkmate (or stalemate) is achieved on the board, when the game ends immediately.

The BCO system shall keep track of the remaining thinking time of both chess players using an electronic chess clock (ECC). The arbiters set a maximum thinking time for the players on the ECC. Once the game is started, the time of the white player starts decreasing until the player presses the flip button – when his/her opponent's time starts decreasing. The ECC indicates if a player exceeds his/her thinking time (i.e. the player is flagged), then the game is stopped with the loss of the respective player.

Viewers can watch the games online. In the BCO system, each move (and the time needed for the move) is broadcast on the Internet by the BCO server but delaying the transmission by 15 minutes. The BCO server reports the final result when a game is finished (a draw, or one of the players wins by checkmate, flagging or resignation).

Task: Draw a *Block (box-and-line) diagram* that represents the functional software architecture of the BCO system with the core components and relationships (4).

B I U á

0 / 10000 Word Limit

2 c) Consider that a modern web application is being developed for the BCO System (described right above in Section 2 b).

Task: Decide whether you would apply each of the following architectural styles for the architecture of this web-based BCO system. Give a short justification for each of your decisions, i.e. why the specific architectural style is applicable in the context of the BCO system, or why it is not meaningful. (4)

(i) *Model-View-Controller*, (ii) *Layered architecture* (Two separate questions!)

Item 3

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

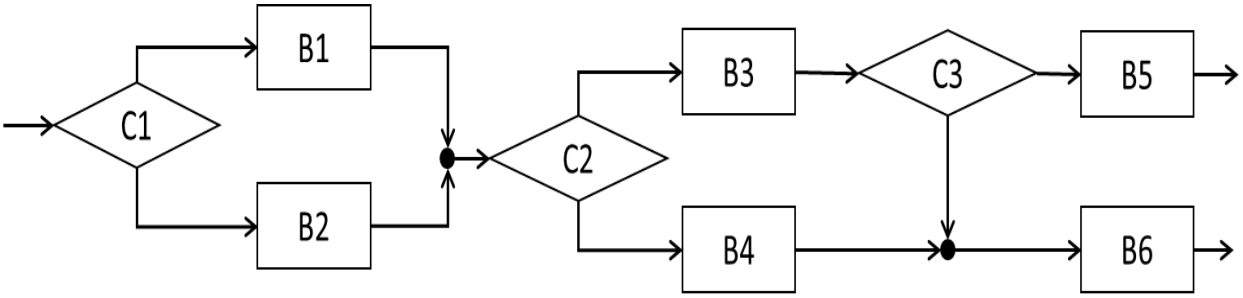
A — Since *black-box testing* establishes confidence, *white-box testing* is only required if *black-box testing* cannot be carried out.

B — *Test-driven development* (TDD) focuses on unit testing, while *behavior-driven development* (BDD) focuses on acceptance testing.

C — *Continuous delivery* is a prerequisite for *continuous deployment*.

D — Using *feature branches* for software configuration management ensures a linear version history.

3 b) Consider the following control flow graph created from a computer program where B1, B2, ..., B6 are block statements while C1, C2, C3 are conditional statements (2)

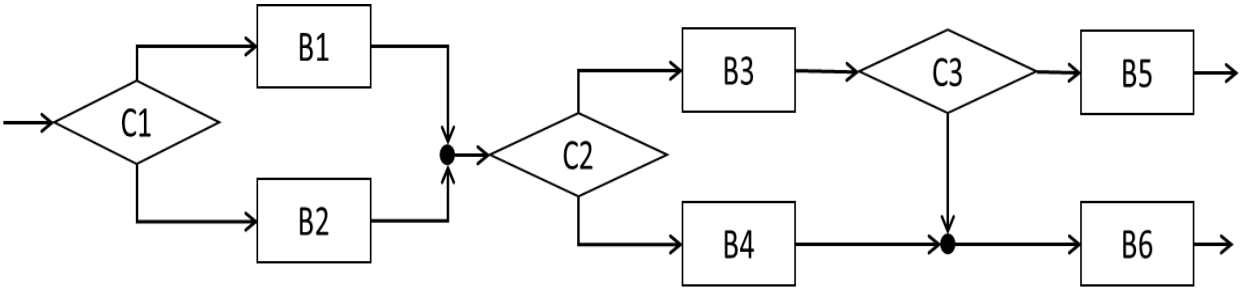


What is the minimum number of test cases needed to obtain *full branch coverage*? ☐

What is the minimum number of test cases needed to obtain *full path coverage*? ☐

(No penalties for incorrect answers)

3 c) Consider the following control flow graph (same as in Section 3 b).



A test suite contains a set of test cases where each test case consists of all the blocks and conditionals that are executed together. For example, a test case can be given as
#1: {C1, B2, C2, B3, C3, B5};

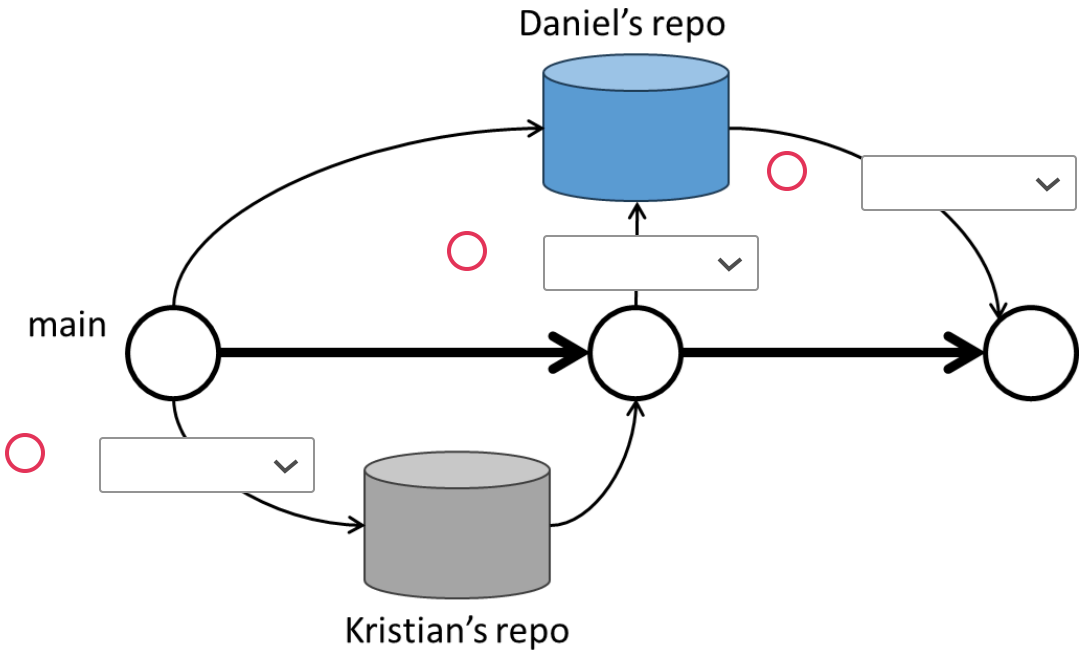
Select all properties that applies to each test suite.(3)

There are no negative scores for this question!

		All test cases are *valid (feasible)*	Test suite provides *full statement coverage*	Test suite is *minimal*
A	#1: {C1, B2, C2, B3, C3, B5}; #2: {C1, B1, C2, B3, C3, B6};	☐	☐	☐
B	#1: {C1, B1, C2, B3, C3, B5}; #2: {C1, B2, C2, B4, B6};	☐	☐	☐
C	#1: {C1, B2, C2, B3, C3, B6}; #2: {C1, B1, C2, B4, B6}; #3: {C1, B1, C2, B3, C3, B5};	☐	☐	☐

3 d) Select the appropriate labels from the drop-down lists to annotate the *centralized workflow* in Git which is an open-source distributed version control system. (3)

No negative score in case of incorrect answers!



Item 4

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

A	Risk management is only necessary for large organizations.
B	The risk “The users employed by the customer might dislike the system.” is avoided by letting user representatives participate in prototype development.
C	A project with one team can efficiently manage 3-10 project specific risks.
D	The risk “We will lack competence in developing a simulation model.” can have a contingency plan “Develop a data-driven model based on historical data.”

4 b) Explain the following concepts of project planning: *Phase*, *critical path*, *milestone*, and *tollgate*. (4)

B I U á

0 / 10000 Word Limit

4 c) Describe in detail what a *burn down chart* is, e.g. What are the values of the axes? How do you make the calculations? When do you update the chart? How do you handle vacations? The most important question: why is it useful? (4)

B I U á

0 / 10000 Word Limit

Item 5

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

A	A mature organization updates its processes whenever necessary.
B	ISO 9000-3 is a standard that provides guidelines for environmental management systems.
C	Usability is a sub-characteristic of Maintainability in ISO/IEC 25010 since a usable system rarely needs any changes.
D	In ISO/IEC 25010 Co-existence means that a (software) component can share the environment with other components.

5 b) Describe two different *metrics* or *measurements* that can be used to measure *reliability*. Use the same format to describe metrics or measurements as was introduced on the slides in the course. (4)

B I U á

0 / 10000 Word Limit

5 c) *Scenario*: You have developed a very good large language model (LLM) and have lot of customers who like you to provide them with applications around it. That is good, but internally the work is chaotic, all people make small contributions to almost all projects prioritizing work according to the earliest-deadline-first principle. No one seems to have an overview of which state the different projects are in. The customer satisfaction data have recently shown a large variance.

Task: Describe a *CMMI process area* that you think will help you the most. A description is typically 5-6 sentences; that covers *Introductory notes* and/or *Specific goals*. Describe also how this will help your company. (4)

B I U á

0 / 10000 Word Limit

Item 6

6) *Scenario*:

In the Chess Olympiad, national teams compete with other national teams in two categories (open, women) in a Swiss tournament with 11 rounds. In each round, each match consists of four individual games played with alternating colors (e.g., a member of Team A plays White on Board 1 and 3 and plays Black on Board 2 and 4). An individual game on a board is played between two players (showing their name, nationality and rating). National team captains need to announce who plays on which board 30 minutes before the start of the round. A win in an individual game counts 1 point, a draw counts 0.5 points, a loss scores 0 points. A team wins (and thus scores 2 team points) if the sum of individual scores exceeds that of the other team. If those individual scores are equal, then both teams get 1 team point. The current standing displays the teams with decreasing number of team points (most team points first) using individual game points as a tie breaker if two teams stand with exactly the same team points.

Task: Draw a *domain model in the form of a UML Class diagram* 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*. Give a *brief textual justification* (1-2 sentence) of your *key design decisions*. Append your UML Class diagram(s) and write in the text which appendix that contains the answer to this question. (10)

Item 7

7) *Scenario*: The BCO system keeps track of the remaining thinking time of both chess players using an electronic chess clock (ECC). After turning on the clock, the arbiters set a maximum thinking time for the players. A game is then started by a player by pressing the start button of the ECC. The time of the white player will be decreasing after that. A player can stop his/her clock, and at the same time, start his opponents' clock by pressing the flip button. The ECC indicates if a player exceeds his/her thinking time (i.e. the player is flagged), when the game is stopped with the loss of the respective player.

Task: Describe the state-based behavior of the electronic chess clock (ECC) by a *UML Statemachine diagram*. List all the *internal attributes*, *triggering events* (including timeouts) and *actions* which are used by the statemachine. Append the UML Statemachine diagram and write in the text which appendix contains the answer to the question. (10)

B I U á

0 / 10000 Word Limit

Item 8

8) Scenario:

To calculate the ranking of the teams in the Chess Olympiad, we are testing a function that compares the results of two teams:

```
int compareTeamAWithTeamB(int teamPointsA, int indiPointsA, int teamPointsB, int ind
iPointsB)
```

- The function returns -1 if the team points of Team A are greater than the team points of Team B, while it returns 1 if the team points of Team B are greater than the team points of Team A.
- If the team points for both teams are equal, then the function checks the individual points. Then the function returns -1 if the individual points of Team A are greater than the individual points of Team B, while it returns 1 if the individual points of Team B are greater than the individual points of Team A.
- If both team points and individual points are equal, then the function returns 0.

Task: We wish to carry out testing based on equivalence class partitioning, but there are some annoying dependencies imposed by the constraints for input variables *teamPointsA* and *teamPointsB* (as well as *indiPointA* and *indiPointsB*).

- By keeping the original output, you need to reformulate the comparison function with *only two input variables* derived from the original four inputs by *calculating the differences* between team points (of Team A and Team B) and individual points (of Team A and Team B).
- Then identify all valid equivalence classes for this new function
- Next, create a test table while performing strong equivalence class partitioning as it has been taught in the course. (You can omit test cases for invalid equivalence classes.)
- Finally, derive *three test cases for the original function* to cover all possible outputs.

Item 9

9) An interview technique with origins in psychology asks the interviewee to characterize a set of *elements*. The interviewee is asked to consider three elements at a time, and to identify a way in which two elements are similar to each other, but distinct from (and contrasted to) the third. One of these ways of identifying elements is called a *construct*.

In this exercise, the elements are the Method Frameworks: *Extreme programming (XP)*, *SCRUM*, and *Kanban*. An example of a construct is:

> *Construct*: Prescribes a daily stand-up meeting.
> Yes: XP and SCRUM
> No: Kanban
> *Comment*: A daily stand-up meeting takes about 15 minutes and has the purpose of keeping all team members updated on the state of work.

Now your task is to create five more constructs for the three elements of the Method Frameworks with a short comment (in 1-2 sentences) showing that you understand the construct. You can choose freely, but to get full credits, all three elements need to be alone at least once in a construct and contrasted to the other two elements. (10)

Item 10

10) Make a comparison between the review methods: *Inspection*, *Walk-through*, and *Peer-review* by answering the following questions:

a. What is the purpose of the review? (write a few keywords per method)
b. Who is involved in the review? (write a few keywords per method)
c. What artifacts can be reviewed? (write a few keywords per method)
d. How is the review performed? (write a few sentences per method)
e. What will be typical outputs from the review? (write a few keywords per method)

Your answer may include a table created in Wiseflow. (10)

0 / 10000 Word Limit

Item 11

Don't write anything here - this will be filled in by the Administrators

0 / 10000 Word Limit