Prof. Dr. Detlef Kips, Dr. Martin Jung
Department Informatik, Universität Erlangen-Nürnberg

01.04.2019

Klausur: Praktische Softwaretechnik

Personal Information (Please use block letters!):

Name, Given Name:	______________________
Date of birth:	______________________
Matrikelnummer:	______________________

Please read the following hints and instructions carefully and confirm your understanding by signing below!

- Please put your student ID and a personal identification (with a picture of you) on the desk and have it ready for check of identity and participation!
- **No tools or resources besides writing equipment are allowed!**
- The solution must be given in the space provided. If the space is not enough for your solution, use the additional space at the end of this printout. Provide a short reference to the pages in the back at the original task. You may ask for additional empty pages, they will be provided by the supervisors and will be stapled to the printout. *You will only be allowed to hand in the stapled printout.*
- You may request scrap paper from the supervisors. *These may not be handed in after the test.*
- The working time is 90 minutes. You can reach 90 points.
- If you can't continue the test for health reasons, you have to prove this by providing a *„Erweitertes ärztliches Attest"* to the Prüfungsamt. In this case, signal the supervisor, request the corresponding form, and fill it out before leaving the test.
- **Check the printout for completeness! The printout has 17 pages, and an additional page with task information (not stapled). Check also if the print is readable.**

By my signature, I confirm the receipt of a complete test printout, and acknowledge the reading and understanding of the information given above.

Erlangen, den 01.04.2019 ....

(Signature)

This will be filled out by the teacher, NOT by the student!

Score:

Task	1	2	3	4	Σ
Score					

Grade:

In the following:

- Multiple-Choice-Tasks (1.a to 1.h), will be awarded at least 0 points. Generally, each error will result in deduction of a point. In particular, this means, that missing check marks are considered to be errors! In each sub-task (a-h), at least one statement is true, and you have to check all true statements.
- In your answers, especially when drawing diagrams, mind the **readability**. If we can't read the answer, we have to consider it wrong.
- In all tasks, only the final result is relevant, intermediate steps should be left out or put on scrap paper.
- We **strongly suggest** you to draft all diagrams on scrap paper and transfer the final result as a readable diagram to the printout.

Task 1 [24 points]

a) A workflow typically has ...

- ☐ input artifacts.
- ☐ complexity metrics.
- ☐ weighted transitions.
- ☐ responsible roles.
- ☐ output artifacts.
- ☐ supporting roles.

b) In a CRC Card Session …

- ☐ external use case scenarios may be used in order to find out appropriate class candidates.
- ☐ internal use case scenarios may be used in order to find out appropriate class candidates.
- ☐ the participants try to identify attributes and operations for every class candidate.
- ☐ the participants try to identify relationships to other classes for every class candidate.
- ☐ the participants try to identify the responsibilities of every class candidate.
- ☐ the participants try to collect as much information as possible about every class candidate.

c) An analysis class model typically ...

- ☐ contains information about real world concepts.
- ☐ associates real world concepts with each other.
- ☐ cannot be visualized as a class diagram.
- ☐ specifies which programming language it must be implemented in.
- ☐ specifies timed behavior.
- ☐ allows for snapshot instance diagrams to be derived from it.

d) The internal behavior of a use case's default scenario may be specified ...

- ☐ using a use case diagram.
- ☐ as a generic scenario.
- ☐ as an exemplary scenario.
- ☐ using a communication diagram.
- ☐ using a sequence diagram.
- ☐ using a data flow diagram.

e) When replacing one implementation of a component in a loosely-coupled system …

- ☐ the system’s use cases will change.
- ☐ black box tests will ideally not need to be re-written.
- ☐ all gray box tests must be re-written.
- ☐ white box tests may have to be re-written.
- ☐ other component implementations must be adapted.
- ☐ the new implementation must still satisfy that component's interfaces.

f) In a Code Review Meeting …

- ☐ the participants discuss how to fix the bugs found during testing the review object.
- ☐ the participants discuss their ideas how to improve the code.
- ☐ the author of the review object may explain her code to the other participants.
- ☐ the participants decide on the review object to be released or reworked.
- ☐ the participants have to agree on a common opinion concerning each finding.
- ☐ the moderator decides if there are different opinions concerning a finding.

g) Considering software testing ...

- ☐ a methodology is required to determine the outcome of a single test case.
- ☐ given the right strategy, everything can be tested.
- ☐ all parts of the software should be tackled with the same rigor.
- ☐ it is valid to exclude test cases from execution based on their priority.
- ☐ a test oracle is used to determine which test cases have to be executed.
- ☐ for critical parts of a software system "execute all" is the only valid regression test strategy.

h) In software configuration management ...

- ☐ test data, setup and results must be put under configuration control.
- ☐ feature branches are necessary to parallelize development.
- ☐ release branches are used to track customer deliveries.
- ☐ multiple integration branches are necessary if two or more teams work in parallel.
- ☐ centralized version control systems prevent concurrent changes on a particular source file.
- ☐ data and code have to be stored in separate repositories.

Task 2 [15 points]

a) Draw a simple example Petri Net realizing the Fork/Join-Pattern for dynamic multithreading!

b) Name four essential steps in project planning, in their logical sequence!

c) Name and draw five types of control nodes of UML activity diagrams and describe their respective meaning in one sentence!
(You may use drawing space on the next page.)

Task 3 [34 points]

You are tasked with developing a chat feature for the `StudOn` platform. You are given the following requirements:

REQ.1. Users have a nickname and an ID.

REQ.2. There are “channels” for users to converse in.

REQ.3. Every channel groups one or more users together.

REQ.4. Users can be members of any number of channels.

REQ.5. One of a channel's users is its “owner”. A user can own any number of channels.

REQ.6. Channels contain any number of messages.

REQ.7. A message is strictly contained by exactly one channel and cannot exist without it.

REQ.8. A message consists of a timestamp and some text content.

REQ.9. Every message is written by one user called its “author”. A user can be the author of any number of messages.

REQ.10. Users keep track of unread messages in a collection called “inbox”. Messages can be referred to by multiple users' inboxes.

REQ.11. Users can only be authors of messages if they are members of the channel in which the message was written.

REQ.12. The owner of a channel must be one of its members.

a) Visualize the requirements REQ.1 through REQ.12 using a UML class diagram. Take note of all necessary classes, associations, properties, multiplicities, compositions, aggregations and constraints! You need not specify data types.

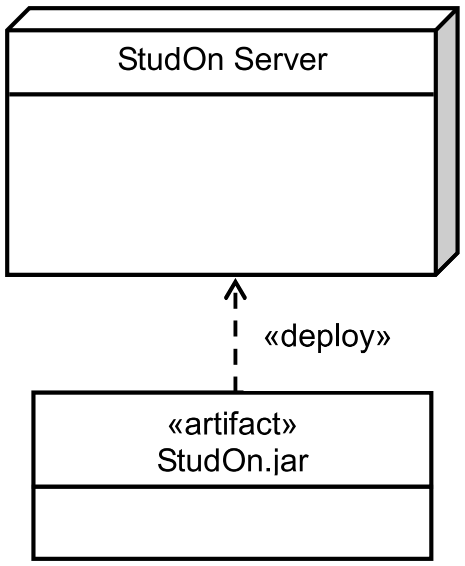
b) Visualize the following exemplary interaction scenario as a UML sequence diagram. It concerns a group chat scenario where a non-owner member named “Bob” posts a message. This message shall then be broadcast to the inboxes of all other users so that “Charlie” and “Alice” can read it. Note that the information about registered members is stored using channels.

   (1) There are three users: Alice, Bob and Charlie.
   (2) There is one channel that these three users are members of.
   (3) Bob sends the message to the channel.
   (4) After the message was received by the channel, all recipients (“Alice, Bob, Charlie”) are determined.
   (5) The message will be dispatched in parallel to the inboxes of all three channel members. It is strictly unspecified in which order these calls take place.

c) A senior colleague insists you incorporate the Observer pattern into your design. Which roles does the Observer pattern specify and which classes of your model would you assign these roles to?

d) Currently, **`StudOn`** is a server-side application as depicted in the following UML Deployment diagram. Since your chat application has drawn a lot of attention, there is now the demand for a mobile chat app. Extend the given diagram (or draw a new one including the given elements) to depict the following deployment aspects:

(1) There is a mobile device.
(2) The new app (called "**`StudChat`**") is deployed to the mobile device as a file named "**`StudChat.apk`**".
(3) The mobile device is connected to the **`StudOn`** server via a GSM connection path.
(4) "**`StudOn.jar`**" manifests a component called the "**`Application Server`**".
(5) Similarly, "**`StudChat.apk`**" manifests a component called "**`ChatApp`**".



Task 4 [17 points]

You are a member of an agile software development team. The product backlog is shown below, as is the sprint board. The first sprint has run exactly according to plan, all the capacity has been used. All tasks of a story need to be performed in order, e.g. ***Architecture*** has to be completed before ***Design*** may start. A story can be delivered once all tasks are done. You had one week of preparation, and the sprints are 3 weeks long, i.e. Sprint 2 is in week 5-7, Sprint 3 in week 8-10 and so on.

a) Determine the velocity according to sprint 1 (see page 15).

b) Assuming a constant velocity of the team, fill out the sprint plans for sprints 2 and 3! Make sure to fulfill the customer requirement "Deliver Story C in week 9".

c) Determine the number of sprints you will need overall, based on what you currently know!

d) During sprint 2, the following changes happen:
   (1) A team member gets pulled into another project, you find a good replacement. The velocity from now on (including sprint 2) is 15.
   (2) The effort estimation with the new team member after sprint 1 leads to the following changes:
      - "***B-Design***" is updated to 5
      - "***B-Integration***" is updated to 9
      - "***D-Design***" is updated to 3
   (3) A high-priority change request (effort "10") needs to be productive in week 11.

   How does the planning of sprints 2 and 3 look now, and how many sprints will the project now need?

Velocity according to sprint 1 (4.a):

Number of sprints required (4.c):

Number of sprints required after change (4.d):

Product Backlog:

Story	Task	Estimation	Story	Task	Estimation
A	Architecture	5	**C**	Architecture	2
A	Design	6	**C**	Design	1
A	Code & Test	12	**C**	Code & Test	8
A	Integrate	6	**C**	Integrate	3
B	Architecture	5	**D**	Architecture	3
B	Design	6	**D**	Design	8
B	Code & Test	15	**D**	Code & Test	10
B	Integrate	10	**D**	Integrate	3

Sprint Backlogs according to 4.b

Sprint 1 (week 2-4)	A-Architecture	(5)	
	A-Design	(6)	
	B-Architecture	(5)	
	C-Architecture	(2)	Planned Capacity:

Sprint Backlogs according to 4.d

Sprint 1	A-Architecture	(5)	
	A-Design	(6)	
	B-Architecture	(5)	
	C-Architecture	(2)	Planned Capacity: