

TENTAMEN (EXAMINATION)

9

Tentamensdatum/Examination date: 2017-01-02
(åå-mm-dd/yy-mm-dd)

AID-nummer
AID number

Ifylles av student

1	4	0	3		
---	---	---	---	--	--

Completed by student

Ifylles av vakt

		1	4	0	3
--	--	---	---	---	---

Completed by supervisor

Kurskod/Course code: TDDC88 Provkod/Exam code: TEN 1

Kursnamn/Course title: Programutvecklingsmetodik

Institution/Department: IDA

Inlämnat: antal blad 10 provformulär
Enclosed: number of sheets exam booklet

Markera behandlade uppgifter med X/Mark tasks attempted with an X

X här/here	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	X	X	X	X	X	X	X	X	X						
Erhållna poäng Points obtained	10	9	8	8	9	8	20	7	10						
X här/here	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Erhållna poäng Points obtained															

Anvisningar/Instructions

- Skriv AID-nummer, datum, kurskod och provkod på varje blad som lämnas in/
Write AID number, date, course code and exam code on every sheet that is handed in
- På varje papper får högst en uppgift lösas om inget annat anges/
Maximum one task per sheet unless otherwise instructed
- Skriv endast på papprets ena sida om inget annat anges/
Use only one side of each sheet unless otherwise instructed
- Numrera de papper som lämnas in/Number every sheet that is handed in
- Använd inte röd penna/Do not use a red pen/pencil

Sen inlämning
Late hand in

Klockslag _____
Time

Orsak _____
Reason

Σ Poäng/Points: 94(5) Betyg/Grade: 5!

Examinator/Examiner: Mes

AID-nummer: AID-number: 1403	Datum: Date: 2017-01-07
Kurskod: Course code: TDDC 88	Provkod: Exam code: TEN1

Blad nummer: Sheet number: 1

1. a) B, C ✓ 2

b) Functional:

R1: The member shall be able to select a car and be presented with information about the car. ✓

R2: The member shall be able to select one of his/her bookings and change the date for the booking. ✓

Non-functional:

R3: The system shall be implemented in java. ✓

R4: The portal shall contain one single page with all functions presented on that page. ✓

c) Traceability: That you give each requirement an identification number to be able to trace the requirement in Requirements spec. — Design — Development — Testing ✓ nice!

Stakeholder: A person or company with major interest in the project. Someone who is affected by the project and therefore want to affect it. ✓ 4

Process requirements: A requirement on the process to be followed in the project. Could be whether agile or waterfall method should be used. ✓

User story: A small text that describes a user's intended use of the system in the form.

"As a <role> I want <something> so that <benefit> ✓

Should have effort estimation and priority! ✓

2. a) B, D

2

b) Risk identification: "what can go wrong?" Identify potential risks and list them.

Risk analysis: "How bad is it?" Determine the probability that the risk occurs and the impact of this. Prioritize risks based on this.

Risk planning: Create plans for the risks if they occur. Make mitigation to lower probability and contingency plans to lower impact. Can also use risk avoidance and risk transfer.

4

Risk monitoring: Assess the risks, see if they are still there and if they have changed. Make adjustments based on the assessment

c) Advantages:

Short - Quick releases, continuously showing customer a working product.

Long - can finish many tasks and focus on testing to improve the quality of the product. -

3

Disadvantages:

Short - Hard to map requirements to iterations

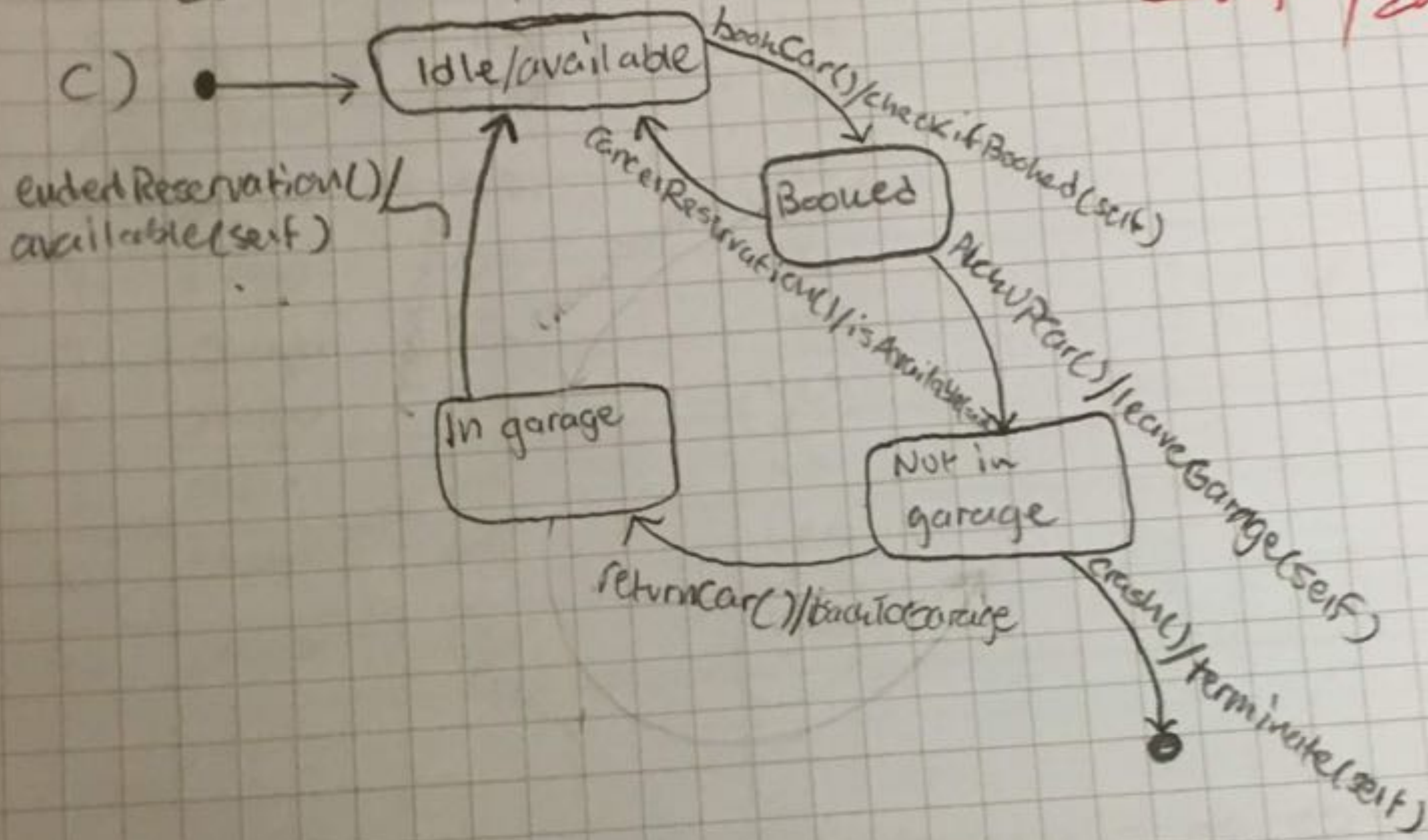
Long - Less frequent releases \Rightarrow harder to adapt to customers wishes or changes since the development process has come very far in each iteration.

3. a) B

b) • It eases communication between different state holders. It gives a high-level presentation of the system that is easier to understand.

• It enables reuse of the system, if a similar system is to be developed, that system will have common requirements → Modules can be reused.

Design? / Documentation?



1p

3p

4p

~~8p~~

4. a) A, C ✓ 2

b) Variable age = Current year - Persons birth year.

EC1: 41 = age, types of persons are dancer, singer, other.

EC2: 52 = age ✓

EC3: 65 = age

ID	type	age	Retired
1.	dancer	40	NO
2.	dancer	41	yes
3.	dancer	42	yes
4.	singer	51	NO
5.	singer	52	yes
6.	singer	53	yes
7.	Other	64	NO
8.	Other	65	yes
9.	Other	66	yes

Handwritten notes: A red 'NO' is written next to the 'Retired' column. Red circles are drawn around the 'yes' entries in the 'Retired' column for rows 3, 5, 6, and 9. A red checkmark is next to row 5.

Test cases for dancer around boundaries of 52, 65 will give yes as output.

Test cases for singer around boundaries of 41 will give no, and for boundaries around 65 will give yes.

Test cases for other around boundaries of 41, 52 will give no.

c) Baseline: Also called tags or release branch. Contains the latest working version of the software. ✓

Trunk: Contains the latest integrated version of the software. ✓ 3

Pull request: A pull request clones the public repository into the local repository and merge local changes. X ✓

Regression test: Verifies that previously developed software still performs correctly even after it was changed. The purpose is to make sure that the changes didn't introduce new faults. ✓

AID-nummer: AID-number: 1403	Datum: Date: 2017-01-07
Kurskod: Course code: TIDC88	Provkod: Exam code: TEN 1

Blad nummer: Sheet number: 5.

5.0) A, C

2

- b) • Management review: The goal is to check deviations from plans in order to be able to respond to those. Performed by management. 1
- Technical reviews: The goal is to evaluate conformance to specifications and standards. Performed by technical leadership and peers. 1
- Walk-through: The goal is to teach everyone about the product and present what defects are there. Is led by the author. 1
- Audit: The goal is to evaluate conformance to spec. and standards. Is performed by an external 3rd party source. 1
- c) • Maintainability = $1/(1+MTTR)$: Is a good predictor because it tells us how often the system needs to be repaired. If low maintainability, you should examine the system to see why it needs to be repaired so often. 2

MTTR = Mean time to repair.

- Cohesion: The relation between internal parts of a module. If a module has high cohesion, it does only what it is designed for. Therefore you know it will be easy to maintain since the internal parts are very associated to each other. 2

9

AID-nummer: AID-number: 1403	Datum: Date: 2017-01-07
Kurskod: Course code: TDDC 28	Provkod: Exam code: TEN 1

Blad nummer:
Sheet number:
6.

6. • variable: Year

EC1: Year = 2017 EC2: Year ≠ 2017

• variable: month

EC3: month < 1, EC4: $1 \leq \text{month} \leq 12$, EC5: month > 12

• variable: day

EC6: day < 1, EC7: $1 \leq \text{day} \leq 31$, EC8: day > 31

EC9: year = 2017
month = 01/03/05/07/08/10/12
day > 30 -

EC10: year = 2017
month = 02
day > 28 -

Test table:

Id	Input	Valid date
EC1	2017-01-01	Yes
EC2	2018-01-01	NO
EC3	2017-00-01	NO
EC4	2017-04-01	yes -
EC5	2017-32-01	NO
EC6	2017-02-00	NO
EC7	2017-02-23	yes -
EC8	2017-02-39	NO
EC9	2017-05-31	NO
EC10	2017-02-29	NO

AID-nummer: AID-number: 1403	Datum: Date: 2017-01-07
Kurskod: Course code: TDDC88	Provkod: Exam code: TEN1

Blad nummer: Sheet number: 7.

7.a) Sprint: Definition: A fixed time period where requirements are selected, prioritized and developed. In the end of the sprint, a working product should be done for a delivery. The team meets every day.

Benefits: The smaller, fixed sprints are a good way of focusing on a part of the software and it makes it easier to break down requirements into performable tasks. Also, the continuous deliveries to customers are good, to get feedback.

Problems: For short sprints it can be hard to map requirements to the actual sprint. It also requires great effort estimation to meet the tight deadline.

In waterfall: Sprints are not good for waterfall-model projects, since some phases can require longer time than 1-4 weeks in larger projects.

b) Product owner: Definition: This person represents the customer in the project. This person creates and makes decisions about the product backlog, the requirements of the system. Leads the Sprint Planning meetings.

Benefits: Represents the voice of the customer. Is highly involved in the process and that enables the rest of the team to handle changes quickly by communicating with the product owner.

Problems: Needs to be good at communication, since he/she represents the customer and their wishes. If changes in requirements are done late, it can be hard to respond to them.

In waterfall: It exists a similar role in waterfall projects, project owner. This person makes decisions about a project's future. So, a slightly modified product owner can work well in the waterfall model.

c) Product backlog: Contains all requirements for the project and is a prioritized list of these.

Benefits: Gives a clear view on what needs to be done, and makes it easier for the team to know which requirements to focus on in the current sprint.

Problems: If the requirements are written bad or are too large or vague, it can be hard to break them down into doable tasks.

Waterfall model: I wouldn't say the product backlog would fit into the waterfall model, since each phase have clear requirements in them and a clear plan. So it wouldn't make any sense to have a list with all requirements that you can select from. Though, if only for overview purpose, it is good!

AID-nummer:
AID-number: 1403

Datum:
Date: 2017-01-07

Blad nummer:
Sheet number:
9

AID-nummer:
AID-number: 1403

Datum:
Date: 2017-01-07

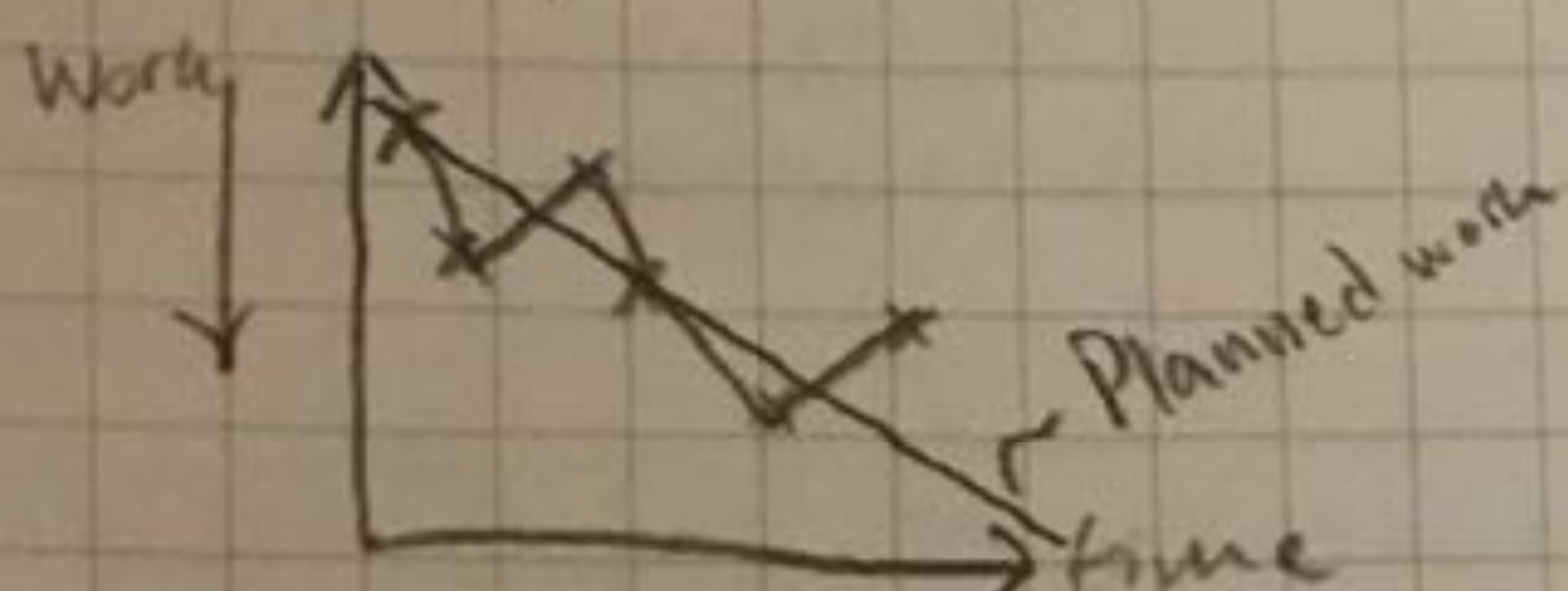
Blad nummer:
Sheet number:
8.

Kurskod:
Course code: TDDC88

Provkod:
Exam code: TEN1

7. d) Burn down chart:

Definition: A diagram that shows completed work compared to planned work for each day.



Benefits: You can see how you are doing, and how many resources you have spent, which is good from a cost- and time perspective. 4

Problems: Some work might be partly finished or almost done, so the diagram might not give a fair view always.

Waterfall: Not applicable, since in agile projects, you want to spread the workload evenly throughout the project which is not always the case in waterfall projects.

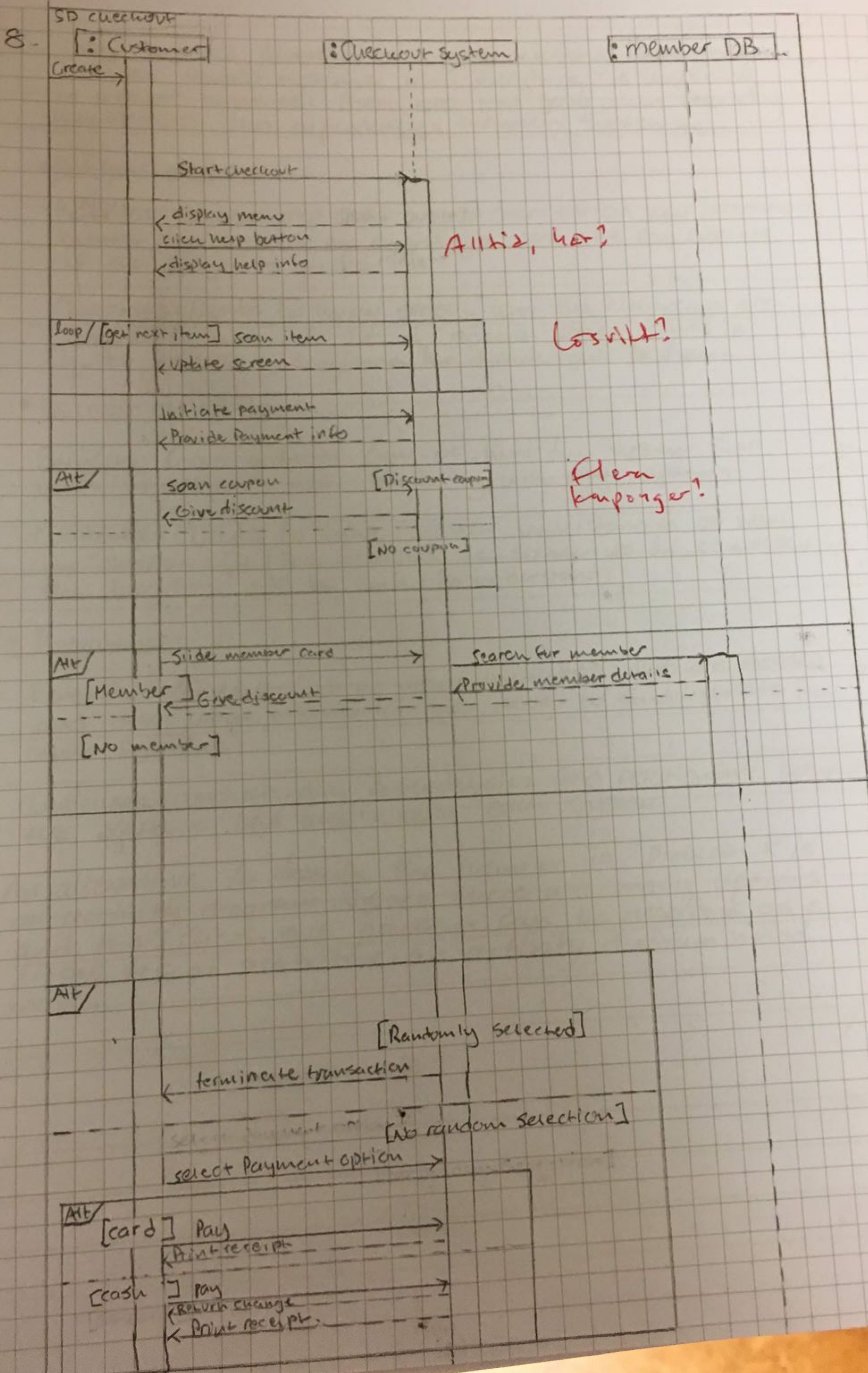
e) Retrospective meetings:

Definition: At this meeting, the processes are reviewed and evaluated. Often done by the end of each sprint and involves all team members. Seeks to improve the processes used. 4

Benefits: A great way to find good and bad things for how processes are followed. Makes it possible to improve continuously.

Problems: All members need to be well aware of the processes that are followed and the processes need to be well defined. Otherwise the purpose of the meeting will not be fulfilled.

Waterfall: Reviewing processes is always necessary and the retrospective meeting would be beneficial in the waterfall model as well, maybe after each phase or so.



9. UML Sequence diagrams are good describing interactions between several systems when everything is happening in a linear sequence. They are less good when describing relationships and dependencies between systems. +

• Designer of diagram:

Benefits: The designer gets a great overview on how interactions between the systems are supposed to work and can evaluate if the design is good or not. +

Challenges: Needs to know exactly how the interactions are supposed to work. A lot of special cases results in quite extensive diagrams for the designer to draw. +

• Reviewer of the diagram:

Benefits: Gets a better view of how the system shall work in real life when interacting with other systems. +

Challenges: Needs to have a clear understanding on what the system he/she reviews is supposed to do, otherwise, the diagram will be hard to understand. +

• Programmer:

Benefits: Gives a hint on what methods that are needed and also what information you need to send/retrieve from other systems. +

Challenges: Understanding dependencies and relationship between the systems and also within each system. (+)

An alternative to describe the behavior in problem 8 is an activity diagram. Since there are many options and the checkout procedure can be described as a linear set of activities, an activity diagram would work. + +

10!