Walkthrough of the exam 2014-10-31 in reverse order

Kristian Sandahl

LINKÖPING UNIVERSITY

8. *Scenario:* The trains in Sweden must be on time. Therefore you have started developing a passenger delay minimization system. The idea is to keep track of the passengers with tickets in their mobile phones. When a train is approaching the station, passengers are reminded to go to the platform. Passengers get reminders at different times depending on i) their distance to the platform and ii) a prognosis of queues to stairs and elevators; we want a constant flow of people moving.

At the platform the passengers get information about where to wait for boarding the train. The system knows which doors of the train that is best for each passenger. Passengers can either get a map on the screen or be guided with a clicking sound, where the frequency becomes higher the closer they are to the correct boarding position.

If passengers send a “need assistance” message to the server, the train crew and the station hosts gets a list on their mobile units of people that need boarding assistance.

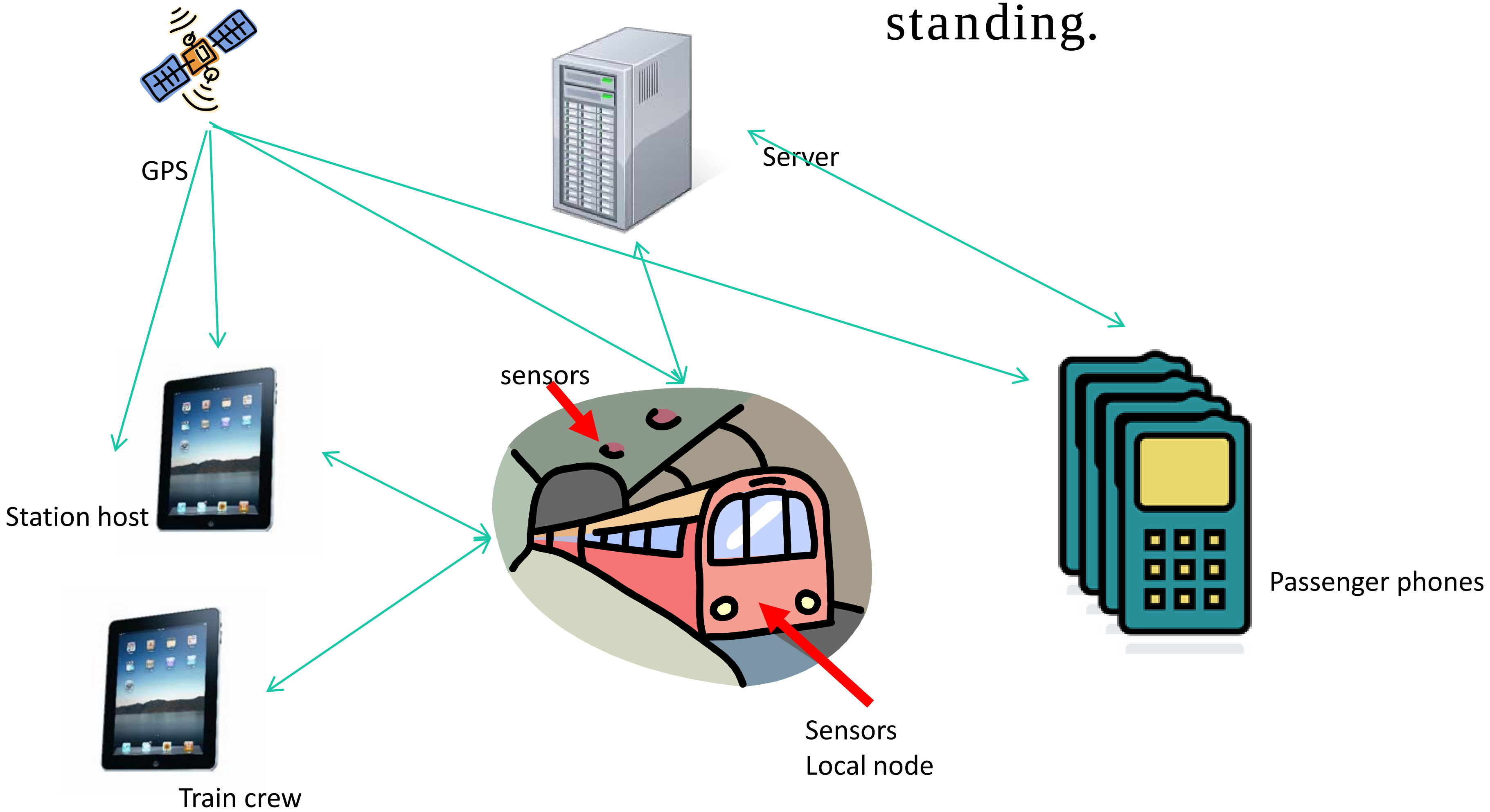
The people needing assistance are prioritized depending on how far they are from their boarding position. A station host or a crew member can select a person to help, then the other personnel do not need to bother about this person. When done, the helper checks the passenger on the mobile device, and a confirmation is sent to the chief crew member.

For safety reasons, you don’t want people to run around when the train is in motion at the platform. So, you implement a safety software according to the following specification: 5 seconds before the train reaches the first end of the platform, all information about boarding positions are erased from passengers’ phones. The information appears again when the train has been standing still for 2 seconds.

Task:

a) Identify about 5 *hardware nodes* that you need in order to *deploy* software to your system. (5)

Hint: always do a system sketch for your own understanding.

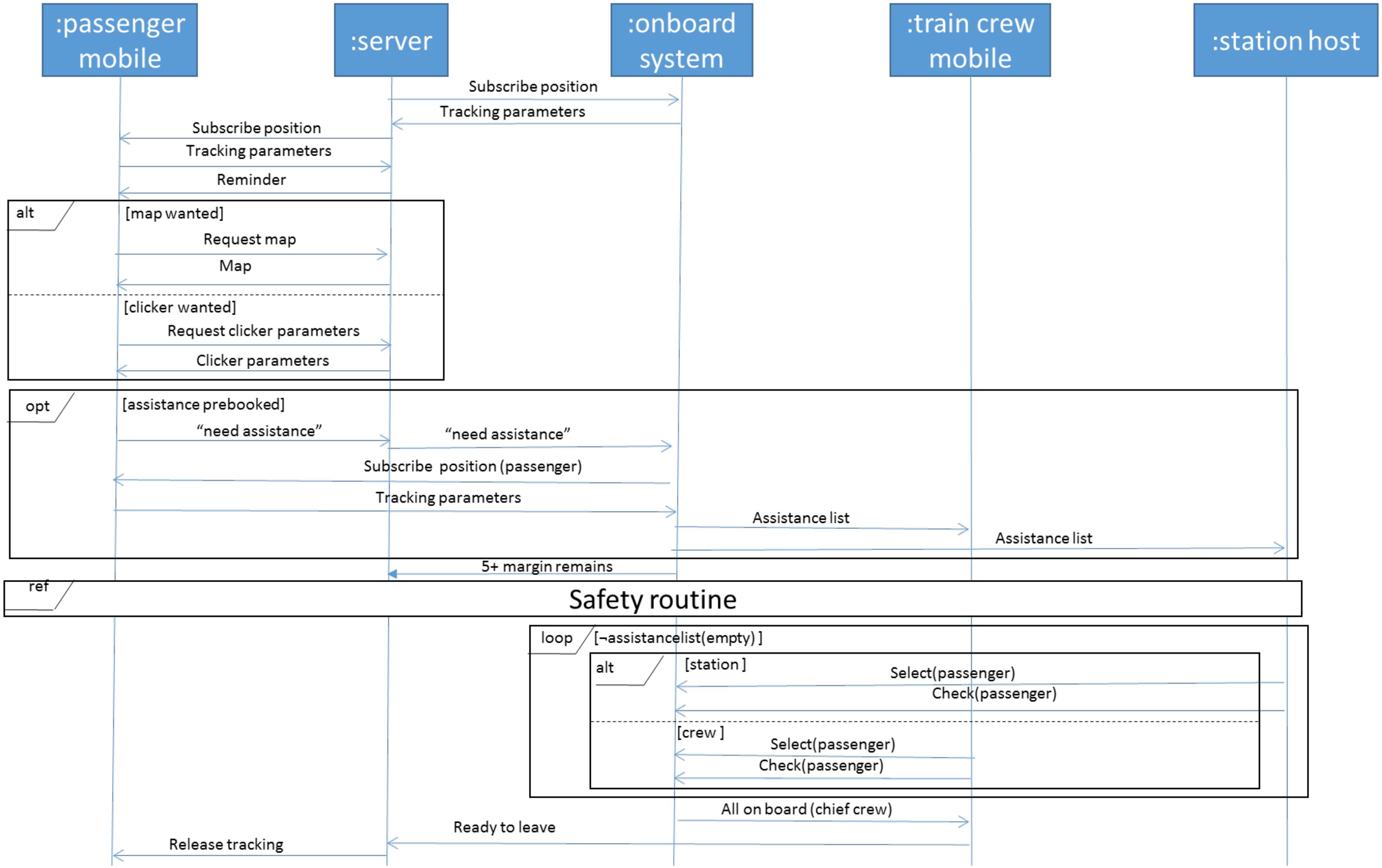


1p per sensible node

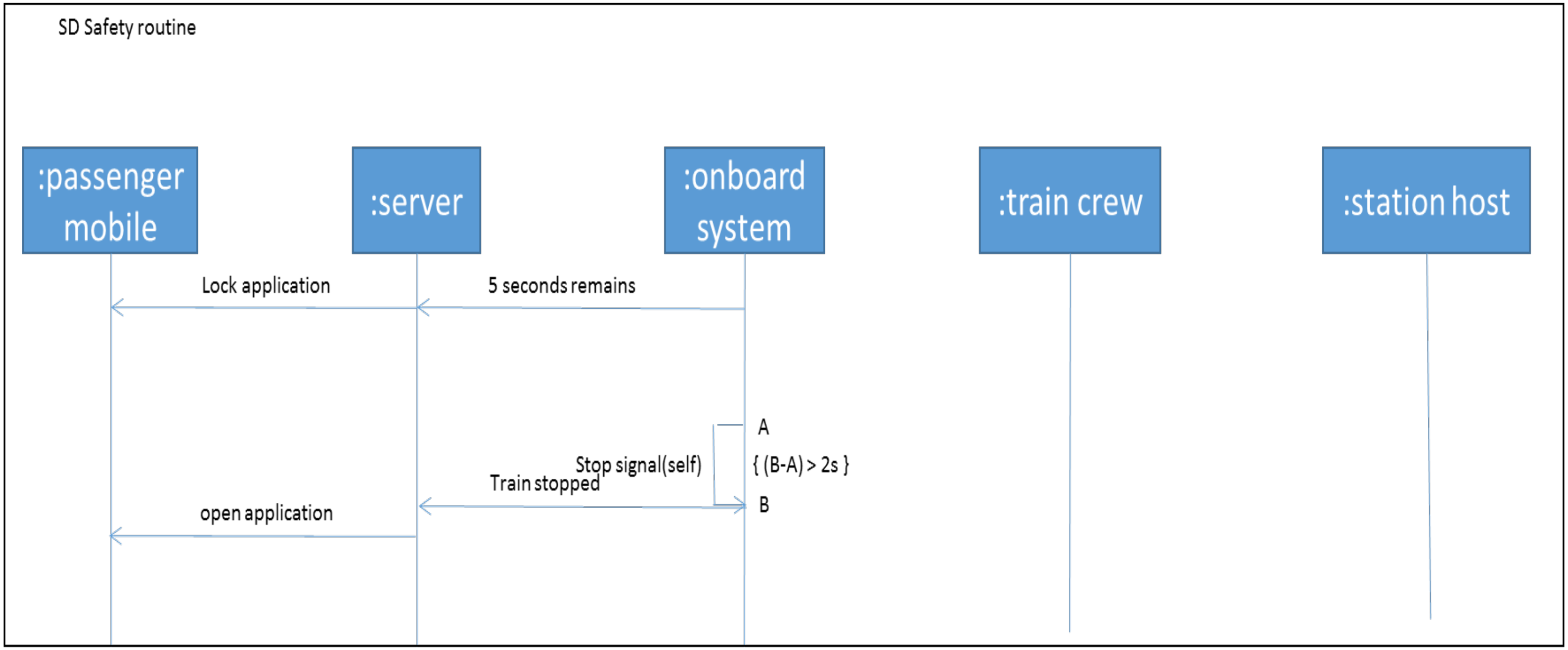
b) Draw a *UML sequence diagram* for the *messages* between the hardware nodes. Only passengers with a pre-booked assistance need are allowed to send “need assistance” messages. Use at least one *fragment*. (5)

Assumptions, the server requires a tracking subscription of trains and passengers that monitors their positions. For the passengers this is released when the train departs. No data is stored.

Passengers already have software for displaying maps and using clicker in their phones.



:passenger mobile
:server
:onboard system
:train crew mobile
:station host
Subscribe position
Tracking parameters
Subscribe position
Tracking parameters
Reminder
alt
[map wanted]
Request map
Map
[clicker wanted]
Request clicker parameters
Clicker parameters
opt
[assistance prebooked]
"need assistance"
"need assistance"
Subscribe position (passenger)
Tracking parameters
Assistance list
Assistance list
5+ margin remains
ref
Safety routine
loop
[¬assistancelist(empty)]
alt
[station]
Select(passenger)
Check(passenger)
[crew]
Select(passenger)
Check(passenger)
All on board (chief crew)
Ready to leave
Release tracking


SD Safety routine
:passenger mobile
:server
:onboard system
:train crew
:station host
Lock application
5 seconds remains
A
Stop signal(self)
{ (B-A) > 2s }
Train stopped
B
open application

1p per reasonable handling of:
Set up trace
Load map or clicker
Need assistance
Safety routine
Train leaves

No fragment -1p
Semantic UML–problems -1p
Syntactic UML-problems -1p

c) Write *test-cases* for the software controlling the safety software. Identify *input* and *output variables*. The test cases shall fulfill the *boundary value testing* as is defined in this course. (10)

Assumption: time is non-negative

Input variables:

R = time remaining before arrival
S = time standing still

Output variable:

I = Information shown (Boolean)

Equivalence classes:

EC1: R > 5s
EC2: R <= 5s

EC3: S <= 2s
EC4: S > 2s

Test table:

Id	R	S	I	EC
1	6	NA	T	EC1
2	5	NA	F	EC2
3	4	NA	F	EC2
4	0	1	F	EC3
5	0	2	F	EC3
6	0	3	T	EC4

1p per correct EC
1p per correct TC

7. Two developers sit on a train and listen to a conversation of two *SCRUM-masters*. The train is quite noisy so they do not get all the details, but they decide to try to use SCRUM in their project anyway. However they do not agree about what they heard about the length of the *sprint*, the size of the *team*, the length of the *daily SCRUM meeting*, and who was doing the *estimates*. So, they send an e-mail to you with the following questions:

What would be the consequences, good or bad, if we:

a) … set the length of a *sprint* to 1 week compared to if we set it to 4 weeks?
b) … have *daily SCRUM* meetings of 15 minutes compared to if we *time-boxed* the meetings to 60 minutes?
c) … have a *team* size of 4 people compared to a *team* of 14 people?
d) … the *estimations* are done by the *SCRUM master* compared to if they are made by the *team*?

Help them by answering the questions. The motivations for the consequences are important and are the base for grading.
(20)

a) 1 week:
+ fast feed-back, short improvement cycles.
– interval can be too small for large requirements, sensitive to sickness leave, more overhead in deployment and more meetings per development hour

4 weeks:
+ still quite fast feed-back, can take quite large pieces (with 8 members 1280 hours), wrong estimates can even out each other
– hard to give the team changed instructions during so long time

b) 15 minutes:
+ meeting is short, everyone is alert, several problems might be remembered, no need for taking notes.
– if there are problems there is a risk that many people have to remain after the meeting, risk that the wrong people remain after meeting.

60 minutes:
+ lot of time to plan for solving issues
– often too long time for a daily meeting, needs note taking and chair person, risk that the meeting will take an hour even if it could be completed earlier, risk that late arrival becomes accepted.

c) 4 people:
+ easy communication, fast reaction to problems.
– hard to cover all competence needed

14 people:
+ lot of perspectives possible, less sensitive for losses.
– many communication paths, longer meetings, more overhead, risk of free riders

d) SCRUM master estimates:
+ fast estimates,
- not all competence used, dependent on a single person.

Team estimates:
+ much experience available, still possible to come to fast decision with planning poker instead of hidden-vote Delphi method, members start thinking of how they will perform the work.
– risk of lengthy discussion, requires much courage and honesty

5 p per correct question,

-2p if only one alternative is explicitly described.

More –p for unclear statements.

6. *Scenario:* Let us return to the description of the inventory system in problem **1 b)**. In that text you could read about some different end-user categories (truck driver, attendant, and cashier). Including these, write down a list of 5 different *stakeholders* of a project developing a new inventory system. Thus, you have to define at least 2 more *stakeholders*.

1 b) *Scenario:* An inventory system of a super market consists of a central server where the inventory is always kept up to date. At arrival, the goods are automatically registered by a bar code scanner, and the truck driver gets an automatic order of where to store the goods. Items of goods are transported from the store to the shop by shop attendants. They use a mobile device to either: i) manually enter the number of items they place in the shop, or ii) scanning the bar code for each item placed in the shop. For fresh food, the weight of food in the shop is recorded. The cashier scans the bar codes of items being sold and enters the weight of fresh food. Attendants regularly makes inventory of remaining items in the shop. If there is a large difference between actual and expected number of items in the shop, the chief attendant is notified.

For each *stakeholder*, write down one of their most prioritized *software quality factor*, together with a motivation of about 2 sentences for why this factor is important. Your solution must deal with at least three different quality factors to demonstrate that you know more than your favorite *quality factor*.

Example of accepted pattern of solution:

Stakeholder1	Quality factor1	Motivation1
Stakeholder2	Quality factor1	Motivation2
Stakeholder3	Quality factor2	Motivation3
Stakeholder4	Quality factor2	Motivation4
Stakeholder5	Quality factor3	Motivation5

(In the example stakeholders and motivations are unique, whereas Quality factors 1 and 2 occur twice. Still the solution contains 3 different quality factors, and is thus accepted)
(end example)

Examples:

Truck-driver	Correctness	If advised to an already occupied shelf, it will take lot of manual steps to sort out.
Shop attendant	Usability	The system must have a clear interface where it is obvious that the system has received the correct input.
Cashier	Performance	There are many repetitive tasks, the system cannot delay the input by the cashier
Developer	Testability	If the developers are given time to make a testable design, repetitive testing can be efficient. Much testing is needed since the system is business critical and handled by non-computer specialists.
Shop owner	Maintainability	The owner wants the investment in a New system pay off during a long period Without replacement.

2p per motivated quality factor.
1p may be withdrawn if the motivation is unclear or it the choice of stakeholder/quality factor is very strange

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

A. The goal of an *inspection meeting* is to resolve and correct as many *defects* as possible, even if a few might not get documented in the inspection record.
B. The reason for collecting data about an *inspection* is twofold: to improve the development process, and to improve the inspection process itself.
C. A *walk-through* is less formal than an *inspection* and is effective in finding defects caused by *errors of omission*.
D. The *inspection leader* doesn't check for defects himself/herself.

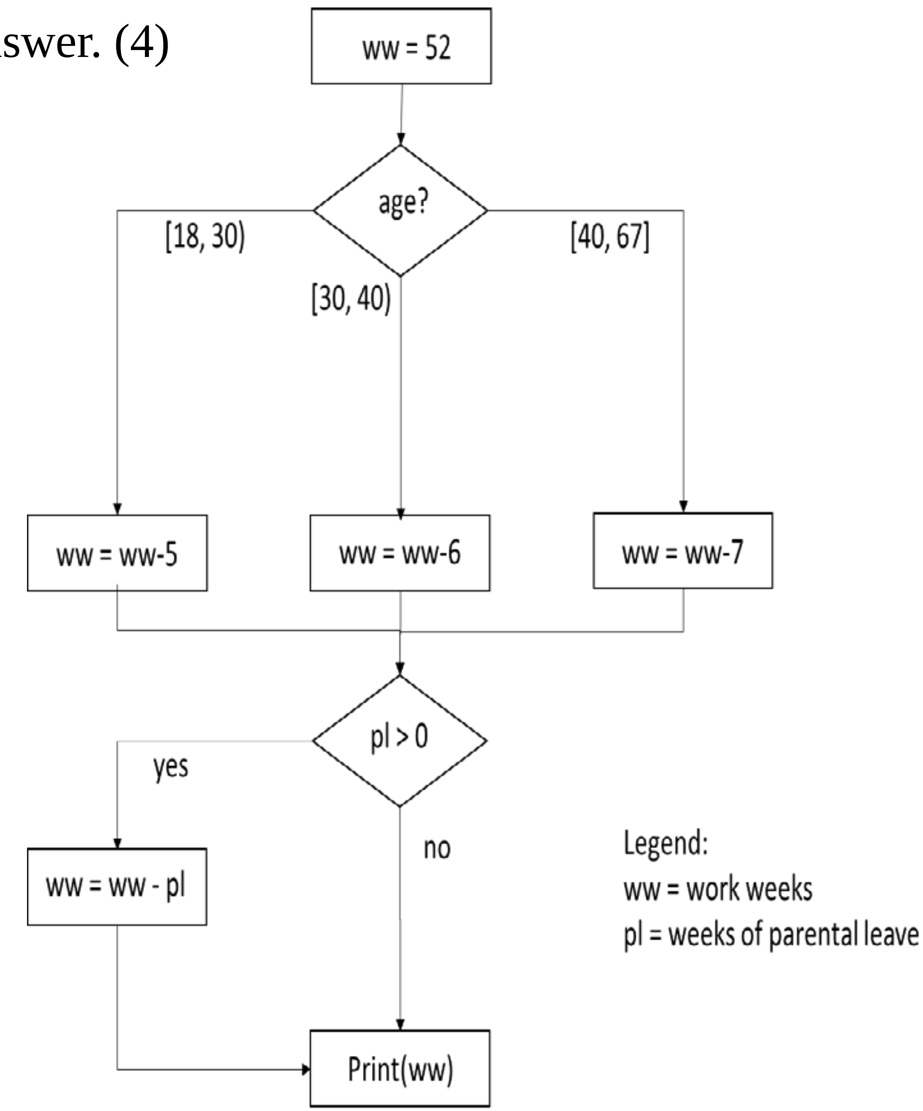
B, C

(A the goal is detection, not resolving, D all participants are inspectors)

5 b) Calculate the *cyclomatic complexity* of the flow graph used for illustrating problem **4 a).** Will the cyclomatic complexity change if you add a statement, for instance, to pay a bonus if pl > 0? Don't forget to motivate the answer. (4)

n = 8
e = 10
p = 1

V(G) = 10 – 8 + 2*1 = 4



Correct calculation 2p,
just an answer 1p,
Using decision point + 1 = 3, 0p since it is only applicable for binary decisions

Adding a statement can be answered in two ways both giving 2p:

1. Inserting new values: n=9, e=11, p=1 gives V(G)=4 no change
2. Observing that adding one node and one edge never changes the value.

Wrong calculation -1p
Wrong, or no motivation -1p

5 c) 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. Briefly describe the process areas and motivate how they can contribute to high *usability*. (4)

RD has specific goals supporting usability and other quality factors, e.g.:

- SP 1.1 Elicit stakeholder needs
- SP 1.2 Translate stakeholder needs into requirements
- SP 3.5 Validate requirements to ensure the resulting product will perform as intended in the end user's environment

It also mentions definition of external interface, but the major motivation is that ALL stakeholders are thought of.

Another PA can be Technical solution (TS) it has as a goal to develop selection critiera for solutions where usability ca be one. (This is actually one of very few places where the word "usability" is spelled out.

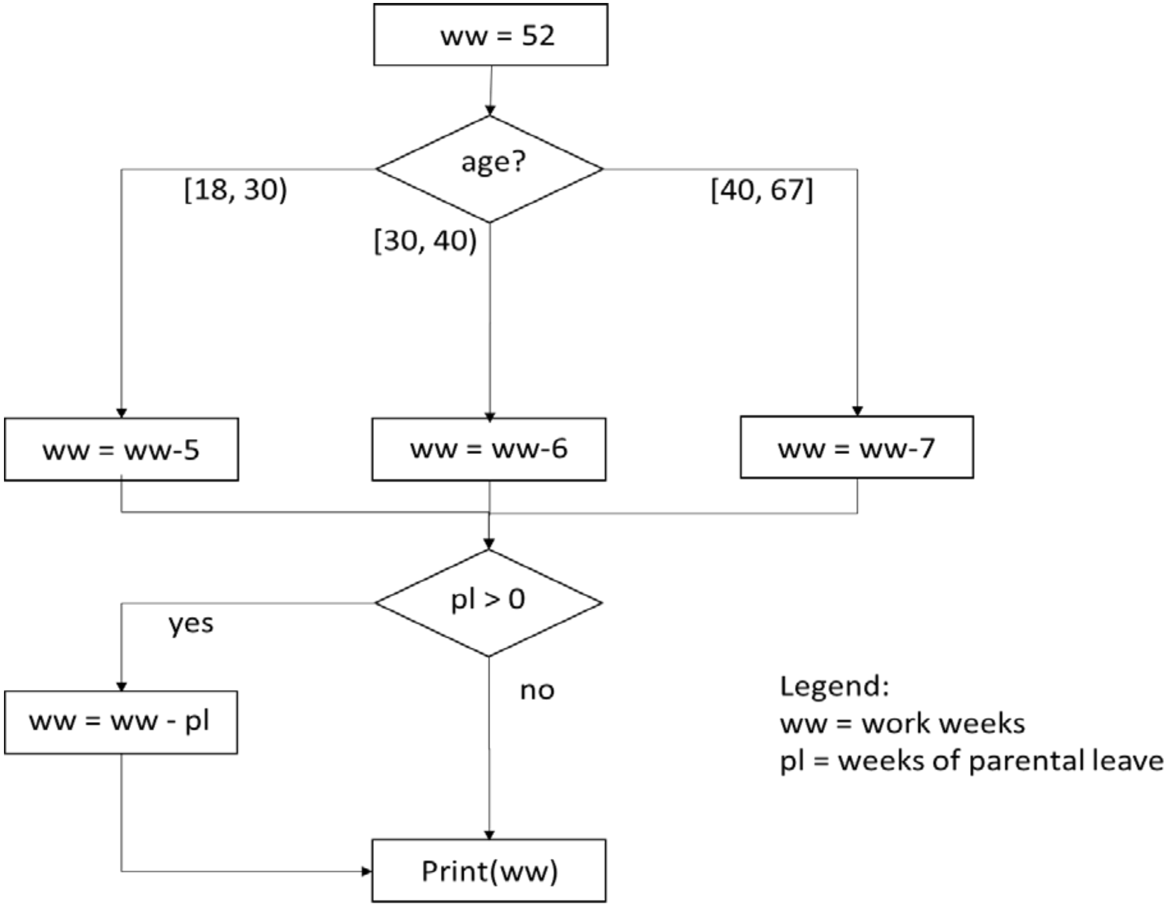
Configuration management CM) is explicit about managing user stories as configuration items.

Project management (PM) monitor product quality by periodic measurements.

Project Planning (PP) has a sub practice to plan stakeholder involvement.

…

2 p per PA description and motivation for usability



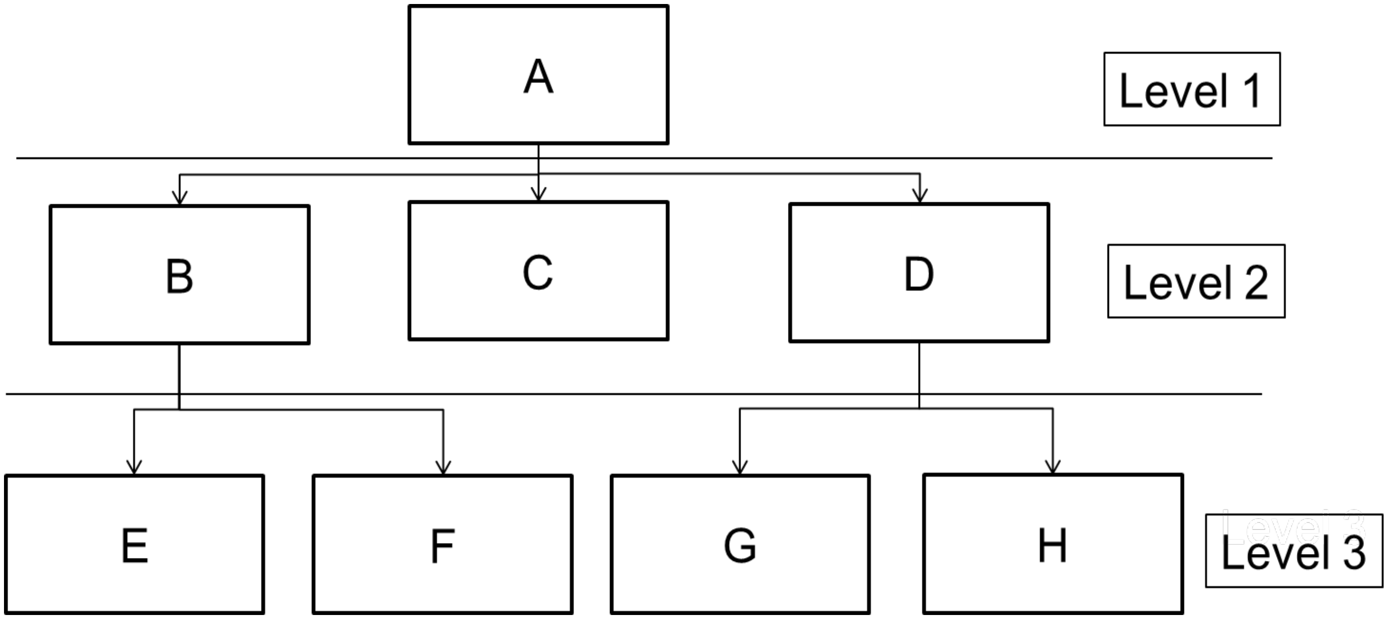
Id	age	pl	ww (expected)
1	25	0	47
2	35	10	36
3	35	0	46
4	50	0	45

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

A. The *test cases* 1, 2, 3, and 4 are **not** sufficient for *full path coverage* testing.
B. The *test cases* 1, 2, 3, and 4 are the minimum number of *test cases* for *statement coverage*.
C. The *test cases* 1, 2, and 4 give *branch coverage*.
D. Given that the flow graph follows the specification exactly, there would be 3 *equivalence classes* if we follow the rules from the lecture.

A, C

(B 3 is superfluous,
D for variable ww: 5 EC,
for variable pl: 2 EC if non-negative numbers assumed)



Changed answer of b) to the correct one.

a) How many *stubs* do you need if you perform *top-down integration testing*?

As many as non-root = 7

b) How many *drivers* would you need if you perform *bottom-up integration testing*?

As many as non-leaf = 3

c) How many *drivers* would you need if you perform *big-bang integration testing*?

None – since big bang testing doesn't use drivers or stubs. Just ready-made units.

d) If it costs x times more hours to code a *driver* compared to a *stub*, when is it more advantageous to use *bottom-up integration testing* compared to *top-down integration testing*?

Number of stubs > x * number of drivers

7 > 3x

7/3 > x (we also buy 2,33)

4c) Explain the input, output, and functions being processed of a *build and integration server* as it is used in *continuous integration*. (4)

Input: (Recently committed) <u>Source code</u>
Output: Object code, <u>Test result</u> (web, mail, SMS)
Processes: <u>Automated build</u> (compiling, linking), running deployment scripts, integration, (smoke) <u>test</u> (on unit and integration level)

1p per correct answer, for 4 p all underlined answers shall be present

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

A. The *Façade design pattern* shows a flexible way of working with *inheritance*.
B. The *Observer design pattern* can be used in an auction system to allow buyers to keep track of the currently highest bid.
C. The *Strategy design pattern* is usable when you have a single intelligent algorithm that you want to use for many different purposes.
D. The *Façade design pattern* can be used to implement a *layered architecture*.

B, D

(A, is describing Strategy,
C Strategy is to use several algorithms for the same purpose)

3 b) Describe the concepts *coupling* and *cohesion*. Do we want high or low coupling amongst the components in an architecture? (4)

Coupling is the number of dependencies between different modules or classes.
The degree of coupling is the number of dependencies divided by the number of possible dependencies.

Cohesion is the number of dependencies within sub-parts of a module or methods in a class.
The degree of cohesion is the number of dependencies divided by the number of possible dependencies.
Cohesion can be of many different strengths.

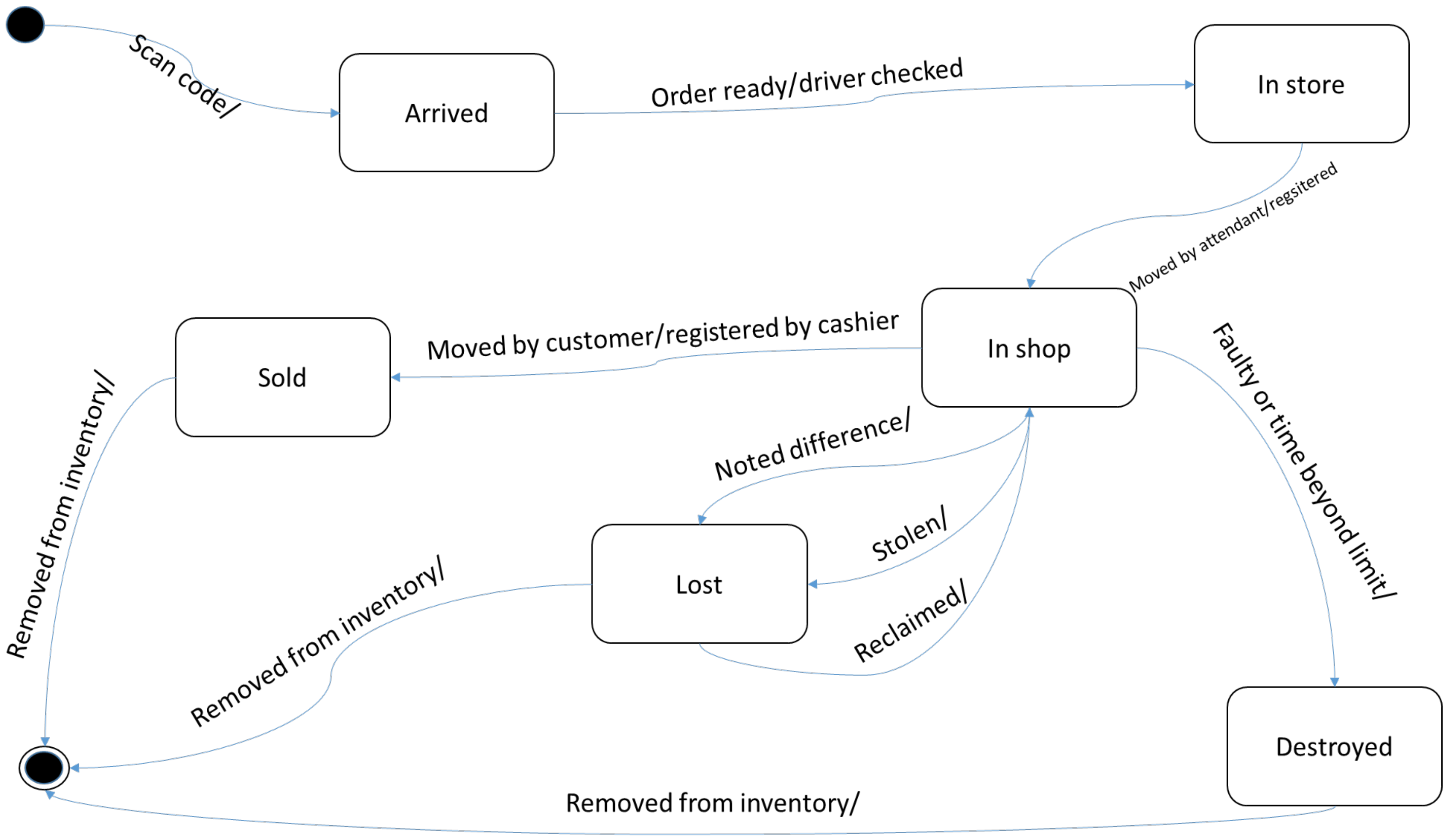
We want to have low coupling since the software shall be easy to understand, test, and change.
It is often a necessary property for safety-critical systems, or other types of dependable systems.
Low coupling can be a sign of cohesive modules, but it is not necessary.

Description of coupling 1p
Description of cohesion 1p
Low coupling 1p
motivation for low coupling1p

3 c) Draw a *UML state chart* of an item in the supermarket, described in the scenario in problem **1 b)** above. Besides the information in the scenario, an item can be:

1. Stolen, never found again, and thus erased from the inventory.
2. Stolen, reclaimed by the police and put back to its proper place in the store.
3. Destroyed if the customer or shop attendant finds a defect.
4. Destroyed if it has been too many days in the shop.

Don't forget to write clear and logical information on the *transitions*.
You may make more assumptions of the system than those given in the problem description, but in that case you shall explain your assumptions in the solution.
(4)



Scan code/
Arrived
Order ready/driver checked
In store
Moved by attendant/regsitered
In shop
Moved by customer/registered by cashier
Sold
Faulty or time beyond limit/
Removed from inventory/
Noted difference/
Stolen/
Lost
Reclaimed/
Removed from inventory/
Destroyed
Removed from inventory/

Many solutions apply, we accept most assumptions made if they are written down.

Total list of requirements:
An item can be:
1. Arrived to the dock (1b)
2. Put in store (1b)
3. Put in the shop (1b)
4. Being sold (1b)
5. Missing without explanation (1b)
6. Stolen, never found again, and thus erased from the inventory. (3c)
7. Stolen, reclaimed by the police and put back to its proper place in the store. (3c)
8. Destroyed if the customer or shop attendant finds a defect. (3c)
9. Destroyed if it has been too many days in the shop. (3c)

-0,5 per missing requirement

UML syntax problem -0,5p per problem, we accept ovals instead of rounded corners, but not straight corners
UML semantic problem -1p per problem (even if multiple occurrences)

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

A. An advantage with the *waterfall model* is that it is simple and easy to understand.
B. An advantage with an *iterative model* is that the development team gets “lessons learned” and can improve the process while the project is running.
C. A drawback of the *waterfall model* is that it starts with the least important requirements.
D. A drawback of an *iterative model* is that the workload is uneven, all testing is left for the last iteration.

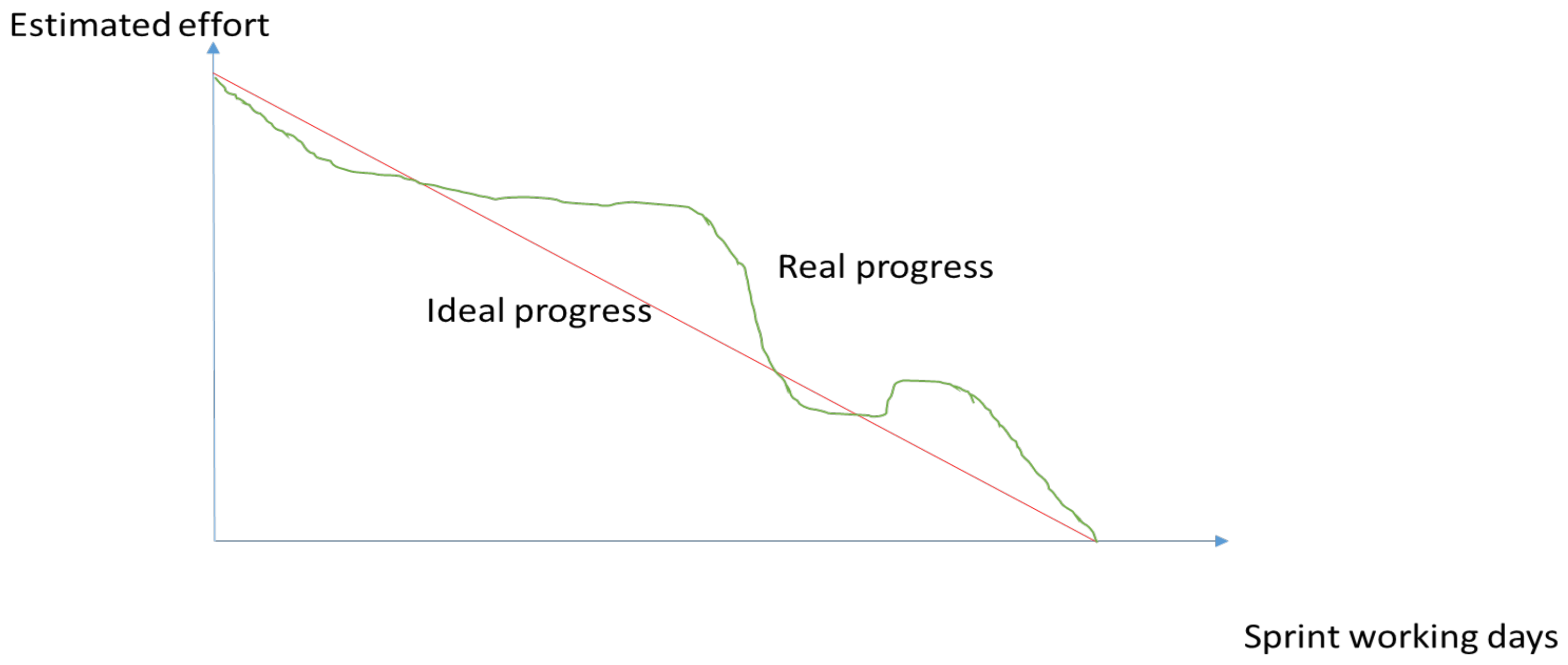
A, B

(C Waterfall can start with any requirement,
D can of course happen as accident, but then it can be questioned if the iterative model was really followed)

2b) Explain the concept of a *burn-down chart*, for instance from SCRUM. Make an example and describe how it shall be interpreted. Also, explain shortly how it can be used by a development team. (4)

The burn-down chart shows the progress of the team in a sprint. It depicts remaining work in estimated effort as a function of sprint working days. Ideally it shall grow steadily down to zero at the end of the sprint. This can be used by the team to:

- Observe the "velocity" of development. Action can be taken during the sprint if the velocity is too slow.
- It is also a baseline measure which can be used to aid estimation of coming sprints.
- It also serves as a baseline to evaluate process improvements.
- If easily accessed it gives the team an easy way to inform itself about the state of the sprint.



4p sensible description, including 2 facts similar to the above

Estimated effort can be replaced by estimated story points or estimated time.

No example or diagram: 3p

Only diagram 1p

2c) Write down four *risks* of a project: one *general*, one *project-specific*, one *indirect*, and one *direct risk*. Briefly describe the project you are thinking of. (4)
Hint: If you are taking a project course right now, use that one. If you are not, use your "own project": Finish my courses for this semester.

1p per sensible risk
A general risk can happen to almost all projects, ex: "People get sick", "The customer will change the requirements."
A project specific risk can only happen to our project, ex: "<person> will get sick due to <specific reason>", "our hardware might be delayed"
An indirect risk is impossible to control for the team, ex: "The server room of our cloud service will burn.", "Our customer gets bankrupt."
A direct risk can be handled by the project, ex: "the migration to the new OS will take too long time", "we will miss some important test-cases"

1 a) Consider the following requirements of an app for an inventory software:

I: The app shall support hand-written input.
II: When the user has finished drawing a character, suggested interpretations shall be shown to the user in maximally 0.2 seconds.
III: The movement of the user's pointing device shall be displayed as a curve with a thickness of 6 pt.
IV: The movement of the user's pointing device shall be displayed as a curve with a thickness of 1 mm.

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

A. The requirements III and IV are *consistent*.
B. The requirement II is a *functional requirement*.
C. The requirement I is a general property of the app called a *feature*
D. The requirement III is a *functional requirement*.

C, D

(A they specify different thickness under same condition,
B it is about response time)

1 b) *Scenario:* An inventory system of a super market consists of a central server where the inventory is always kept up to date. At arrival, the goods are automatically registered by a bar code scanner, and the truck driver gets an automatic order of where to store the goods. Items of goods are transported from the store to the shop by shop attendants. They use a mobile device to either: i) manually enter the number of items they place in the shop, or ii) scanning the bar code for each item placed in the shop. For fresh food, the weight of food in the shop is recorded. The cashier scans the bar codes of items being sold and enters the weight of fresh food. Attendants regularly makes inventory of remaining items in the shop. If there is a large difference between actual and expected number of items in the shop, the chief attendant is notified.

Task: Create a *use-case diagram* for a new inventory system consisting of two different *actors* and two different *use-cases*. Don't forget the use case texts. Only logging in and logging out are basic functions, not to be considered as use-cases. (4)

Diagram of 4 use-cases and 4 actors. A full solution requires 2 use-cases and 2 actors.

Register goods in shop:

The shop attendant starts the mobile device, and make initializations.
The shop attendant fetches goods in the store.
The shop attendant identify the shelf in the shop.
The shop attendant puts the goods in the shelf and register the amount at the mobile device in the appropriate way.
The shop attendant might repeat the steps to fetch goods.
The shop attendant checks that synchronization has been made.
The shop attendant logs out and may put the mobile device in a charger.

Make inventory:

The shop attendant starts the mobile device, and make initializations.
The shop attendant goes to the shelf in the shop.
The shop attendant identify the goods in the shelf and register the amount at the mobile device in the appropriate way.
The system updates the current amount.
If the difference to the expected amount is large the chief shop attendant gets a messages on his/her mobile device.
The chief shop attendant might approve updating the current amount, or mark the inventory for future investigation.
Both the shop attendant and the chief shop attendant check that synchronization has been made.
Both logs out and may put the mobile device in a charger.

Place goods in store:

The truck driver starts the mobile device and puts it in the position in the truck.
The truck driver reads the first order in the list and acknowledges the order.
The truck is driven to the dock and the goods is identified.
The goods is loaded and driven to the correct place in the store.
Once off-loaded, the truck driver checks the order.
The truck driver might repeat the steps to fetch goods.
The truck driver checks that synchronization has been made.
The truck driver logs out and may put the mobile device in a charger.

Register sold goods:

The cashier starts the terminal and make initializations.
A new purchase is registered.
The goods arrival on the conveyor belt is registered in the appropriate way depending on the goods.
The cashier receives payment and returns a bill to the customer.
The cashier repeats the steps with new purchases.
The cashier checks that synchronization has been made.
The cashier logs out.

Two actors and two ovals with a verb phrase plus a system boundary are needed in the diagram.
Texts of 3-5 sentences similar to the examples are needed.
2p per good use-case
at least 3 sentences per use-case
actors are roles, not user1 and user2
actor can be a sub-system
Two single use-case diagrams minus 2 credits.
Only a correct diagram 1 credit. Diagram missing minus 1 cr.
Use-case name: a verb phrase

1 c) Good requirements have many properties. Explain the meaning of requirements being: *inspected*, *unambiguous*, *testable*, and *feasible*. (4) Hint: by "explain" we mean that a student like you who has not taken the course shall be able to understand what you wrote. (1-2 sentences per concept)

1p per described concept

Inspected – the requirements have undergone an inspection made by competent people.
Unambiguous – the requirements can only be interpreted in the intended way.
Testable – it is possible to test the fulfilment of the requirements with available resources.
Feasible – it is possible to implement the system with available resources.

Questions and Answers Session

During your study many questions might arise. Collect your questions and come to this occasion.

Tuesday, October 20, 12:00-14:00

Alan Turing, E-house

Wednesday, October 28, 12.30-14.00

Alan Turing E-house

Pass condition

To pass the exam (alternatives)

1. a) at least 4 credits in all areas in fundamentals **and**
   b) at least 50 credits in total
2. a) at least 4 credits in at least 4 areas **and**
   b) at least 60 credits in total

Part I: Fundamentals

- Requirements
- Planning and Processes
- Design and Architecture
- Testing and SCM
- Software Quality

10 credits per area. Max 50 credits.

Part II: Advanced

50 credits, distributed over 2-5 questions.

- argue, compare, and analyze different concepts and techniques.
- construct and/or design solutions to larger problem.
- explain more advanced and specific topics.

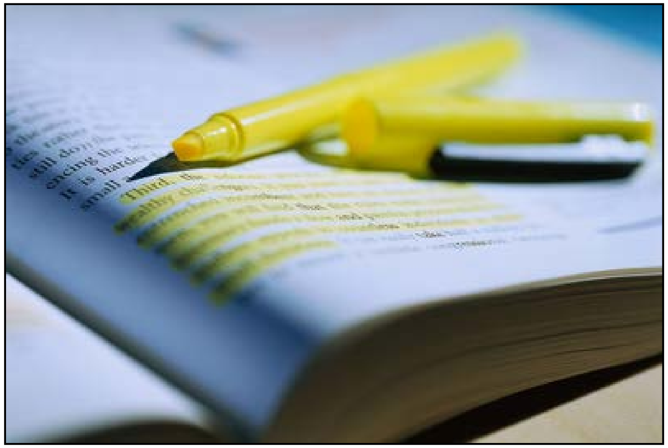
Grades if pass condition is met

Total credits	Mark
0-49	U
50-66	3
67-83	4
84-	5

Aids

Allowed aids

- Two sheets of **handwritten** A4 papers (can write on both sides)
- One volume of dictionary to or from English or an English wordbook.



Explicitly forbidden aids

- Textbook
- Machine-written pages
- Photocopied pages
- Pages of other format than A4
- Electronic equipment

Thanks for listening!

www.liu.se

LINKÖPING UNIVERSITY