

TENTAMEN (EXAMINATION)

7

Tentamensdatum/Examination date: 18-01-03
(åå-mm-dd/yy-mm-dd)

AID-nummer
AID number

Ifylles av student

		1	1	0	5
--	--	---	---	---	---

Completed by student

Ifylles av vakt

2	1	0	5		
---	---	---	---	--	--

Completed by supervisor

Kurskod/Course code: TDDC88 Provkod/Exam code: TEN1

Kursnamn/Course title: Programutvecklingsmetodik

Institution/Department: IDA

Inlämnat: antal lösblad 12 tentamensformulär
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	8	9	9	9	7	9	19	6	9						
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: 100 (15) Betyg/Grade: 5!

Examinator/Examiner: [Signature]

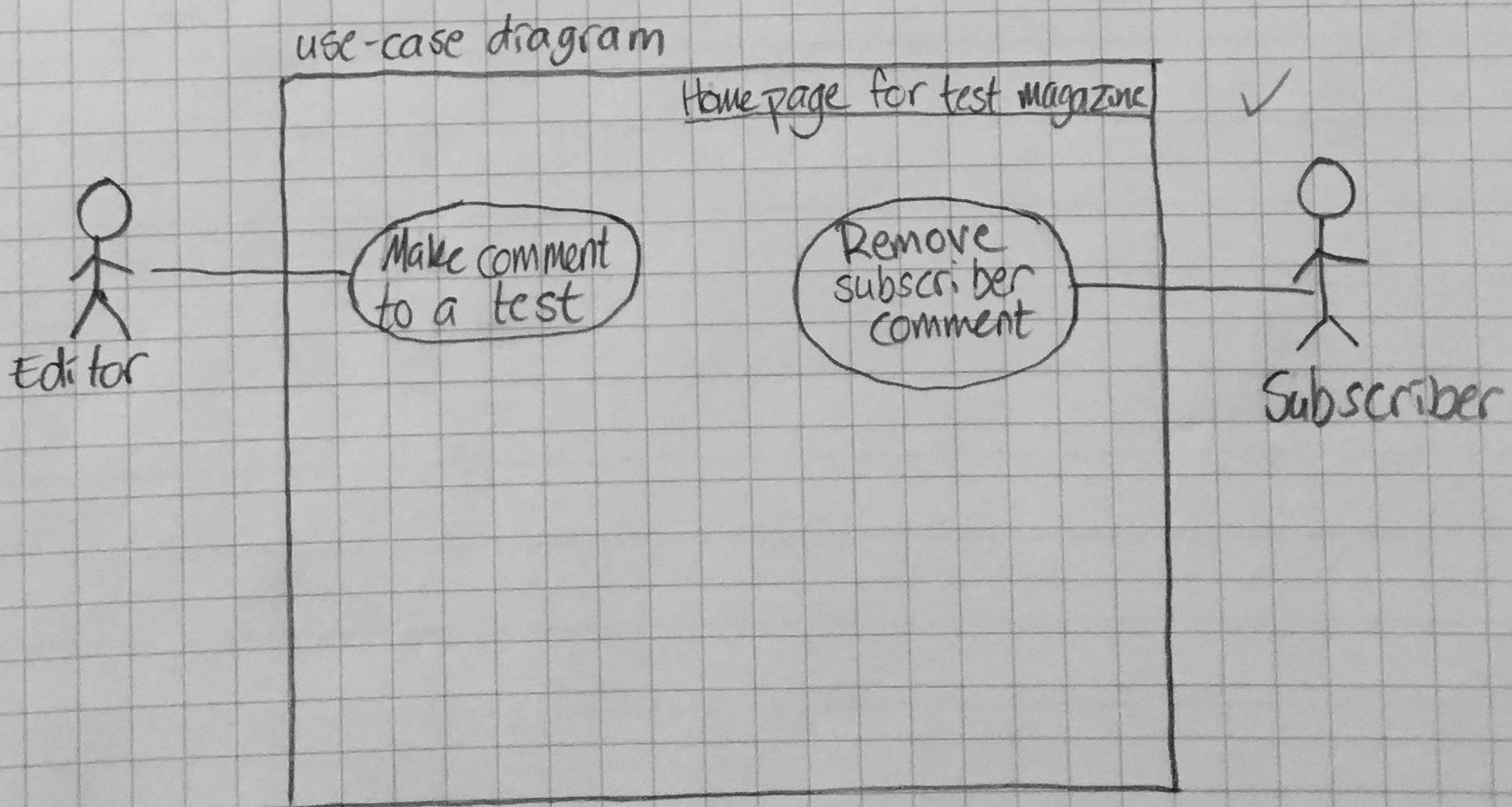
AID-nummer: AID-number: 1105	Datum: Date: 18-01-03
Kurskod: Course code: TDDC88	Provkod: Exam code: TEN1

1. a) B. ✓

The editor removes a subscribers comment ✓
 b) Use Case 1: The editor logs in to the system with his/her username & password. The editor is now logged in & navigates to the page where subscribers' comments are shown. The editor clicks on the subscriber comment that he/she wants to remove and presses the button "remove". The chosen subscriber comment has now been removed by the editor.

Use Case 2: A subscriber comments on a test. ✓

A subscriber logs in to the system with his/her username & password. The subscriber is now logged in & navigates to the chosen test. The subscriber clicks on the button "comment" and shares his/her experience of the product by writing a comment in the text box. The subscriber clicks the button "Enter" and the subscribers comment is now added as a comment to the test.



AID-nummer: AID-number: 1105	Datum: Date: 18-01-03
Kurskod: Course code: TDPC 88	Provkod: Exam code: TEN 1

1. c) Unambiguous requirements: The requirement is consistently stated. Only has one meaning. ✓

Prototyping: Made in an early state of the process, a simplified version of the product. So the interface/function can be tested without being fully developed. Good for getting input/feedback in an early stage. ✓

Design Constraints: A non-functional Requirement, limiting the solution space. E.g. "The system shall be implemented in PHP". ✓

Requirements validation: A part of an iterative process in creating requirements. Includes for example, prototyping, simulation, model checking & acceptance testing. To validate the requirements. Tracability is important. ✗

↓
to check requirements
match stakeholders
needs

AID-nummer: AID-number: 1105	Datum: Date: 18-01-03
Kurskod: Course code: TDDC88	Provkod: Exam code: TEN1

Blad nummer: Sheet number: 3.

2. a) A.

b) 1. Risk identification

Input: Brainstorming risks, of the project, can be connected to organization, tools, requirements, people, technical etc.

Activity: Creating a list of all identified risks connected to the project.

Output: A list of all risks connected to the project

2. Risk analysis

Input: The list of possible risks connected to the project from step 1.

Activity: Analysing the risk and evaluate the risks in terms of probability & impact. Every risk gets a probability number between 1-4 which tells how high the probability is that the risk occur, where 1 is not likely & 4 is very likely. Every risk also gets an impact number between 1-4 which tells how "bad" it will be for the project if it occur. Where 1 is insignificant & 4 is catastrophic. From that, a "risk magnitude factor" is calculated, which is the probability factor times the impact factor. The list is then sorted on the risk magnitude factor, where the biggest number has the highest prioritization.

Output: A prioritized list

3. Risk planning input: the prioritized list

Activity:

What to do with the risks, three alternatives:

1. Risk avoidance - reorganize or make changes so the risk disappears (not always possible)
2. Risk transfer - put the risk on someone else, e.g. "outsource", insurance etc.

2. b).

3. Risk acceptance - live with the risk, accept it, you can

- Mitigate the risk - lower the probability that it occur
- Create a contingency plan - lower the impact that the risk will have on the project if it occur.

A choice need to be made for how to handle the risks at the prioritized list. Output: A plan for the risks on the list.

4

4. Risk monitoring

Constantly make assessment of the risks, has the probability/impact changed? Or have new risks occurred?

Input: The prioritized list and risk plan made above

Activity: Add &/or update risks on the list and update the risk plan.

Output: An updated risk plan & prioritized list of risks.

c)

daily SCRUM: A short meeting between the teammembers, one by one should answer the questions "what have been done", "what will be done" & "what problems did occur", since the last meeting, should be held while the members stand up.

Sprint planning meeting: Should be held before each sprint, should be decided what to be done in the sprint, a sprint plan. It is held by the product owner.

4

Sprint review meeting: The meeting to review the deliverable from that sprint, is held after each sprint.

Sprint retrospective meeting: Held by scrum master. Discuss what went well/wrong in the sprint & what can be improved. Is held after each springt.

19

3. a) A D.

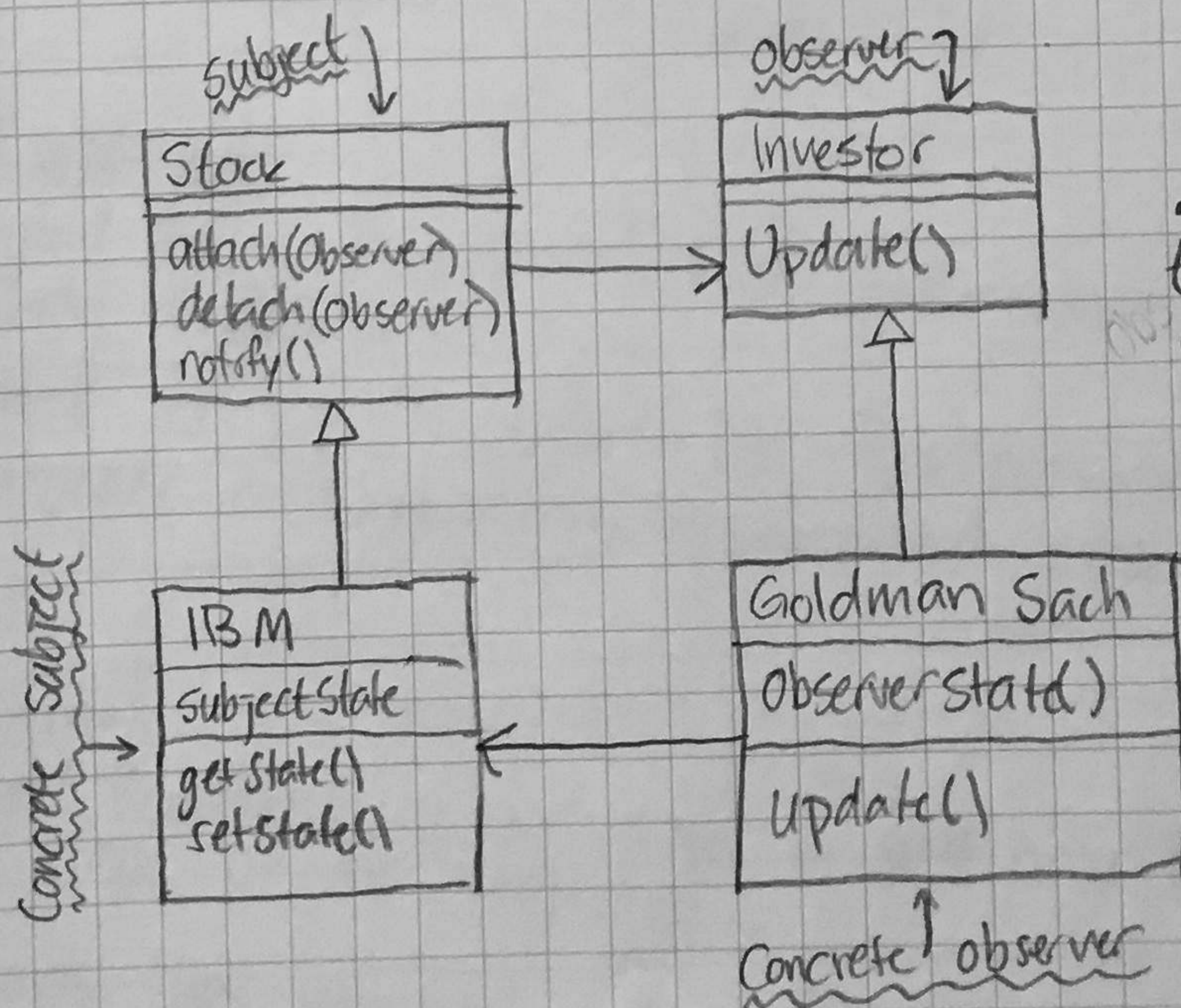
2

b) 1. Modularization means that you can replace parts and enabling changes easier. Without modules it will be a higher dependency & more difficult to make changes. That also mean that you can reuse some modules for other systems/projects which not would have been possible without modularization. To reuse modules means that money can be saved for later, & also that knowledge is saved & used.

2.

3. Modularization also means that it is easier to isolate faults. Modularization makes it easier to find & understand the faults, but also to fix them, since they can be isolated without the high dependency no modularization has. That means less money spend on finding & fixing faults. That also means that it is easier to create a system with less faults, which is important for example "security" of the system. High understandability

c) Observer, A design pattern where objects are affected by other objects. Ex of class diagram:



Applicability:

- When an abstraction has two aspects, one dependent on the other
- A change on one object requires change on others
- An object should be able to notify others without making assumptions about what these objects are.

Consequences?

3 / ap

4. a) c. ✓

b) Full path coverage All possible path is executed, all decision branches and all statement executed. Requires 4 test cases, one which is [true, true], one [true, false], [false, true] & [false, false]

Test table

Input.

Test case	X	Y
1.	15	5
2.	10	5
3.	-5	0
4	0	10

[True, True], output: x=5, y=5 "you win" ✓

[True, false], output: x=0, y=5 "you lose"

[False, True], output: x=5, y=0 "you win"

[False, False], output: x=10, y=10 "you lose" 4

Ensures full coverage!

c) Bottom-up integration testing strategy

Advantages:

1. Is good for projects with strict performance requirements or realtime requirements. ✓

2. Is good for projects where the functions are complicated, errorprone or has high development risks. ✓

Disadvantages:

1. end-user feedback is postponed & will only be received in a late stage of the project. ✓

2. It demands that you put effort (time & cost) on writing drivers. ✓

AID-nummer: AID-number: 1105	Datum: Date: 18-01-03
Kurskod: Course code: TDDC88	Provkod: Exam code: TEN 1

Blad nummer:
Sheet number:
8.

6. ^(birthdate Last) ^{Question from patient}
input: age, doseNumber, suggestedDoseDate, timeSinceLastDose, presentDate
output: Reminder, answerToSuggestedDoseDate
EC1: age < 50 EC2: age ≥ 50
EC3: doseNumber = 2. EC4: timeSinceLastDose ≥ 3 years.
EC5: doseNumber ≥ 5

testtable:

EC	age	doseNumber	suggestedDoseDate	timeSinceLastDose	Reminder	AnswerToSuggestedDate
EC1	30	1	3 month from now	2 month & 1 day	Yes	No
EC2	60	4	1 year from now	1 year	No	Yes
EC3	20	2	1 week from now	9 month	No	Yes
EC4	30	4	1 year from now	3 years	No	Yes
EC5	80	8	1 day from now	2 years 11 month 1 day	Yes	Yes

AID-nummer: AID-number: 1105	Datum: Date: 18-01-03
Kurskod: Course code: TDDC88	Provkod: Exam code: TEN1

Blad nummer: Sheet number: 9.

7.

Metric 1:

- A rate of how useful the user experience the system to be.
- Let the test panel play the game for a relevant amount of time.
- A questionnaire, an answer to that from the testpanel & access to evaluate the answers. They answer the question "How easy to use did you experience that the product was in a scale 1-5"
- The mean value of all the answers.
- A high number of this metric indicates high usability since the user experience is important. If the user finds the system useful & easy to use it would indicate high usability.

Metric 2:

- The average number of clicks to get to a certain part of the system. e.g. "My profile" or "Inställningar"
- Observing a test panel which would be given the instruction to navigate to certain parts of the system.
- Observing & writing down the number of clicks.
- By taking a mean value of the observed values for each task.
- A low number of this metric indicates that it is intuitive and easy to navigate on the product, which indicates a high usability.

AID-nummer: AID-number: 1105	Datum: Date: 18-01-03
Kurskod: Course code: TDVC88	Provkod: Exam code: TEN1

Blad nummer: Sheet number: 10

7.

Metric 3

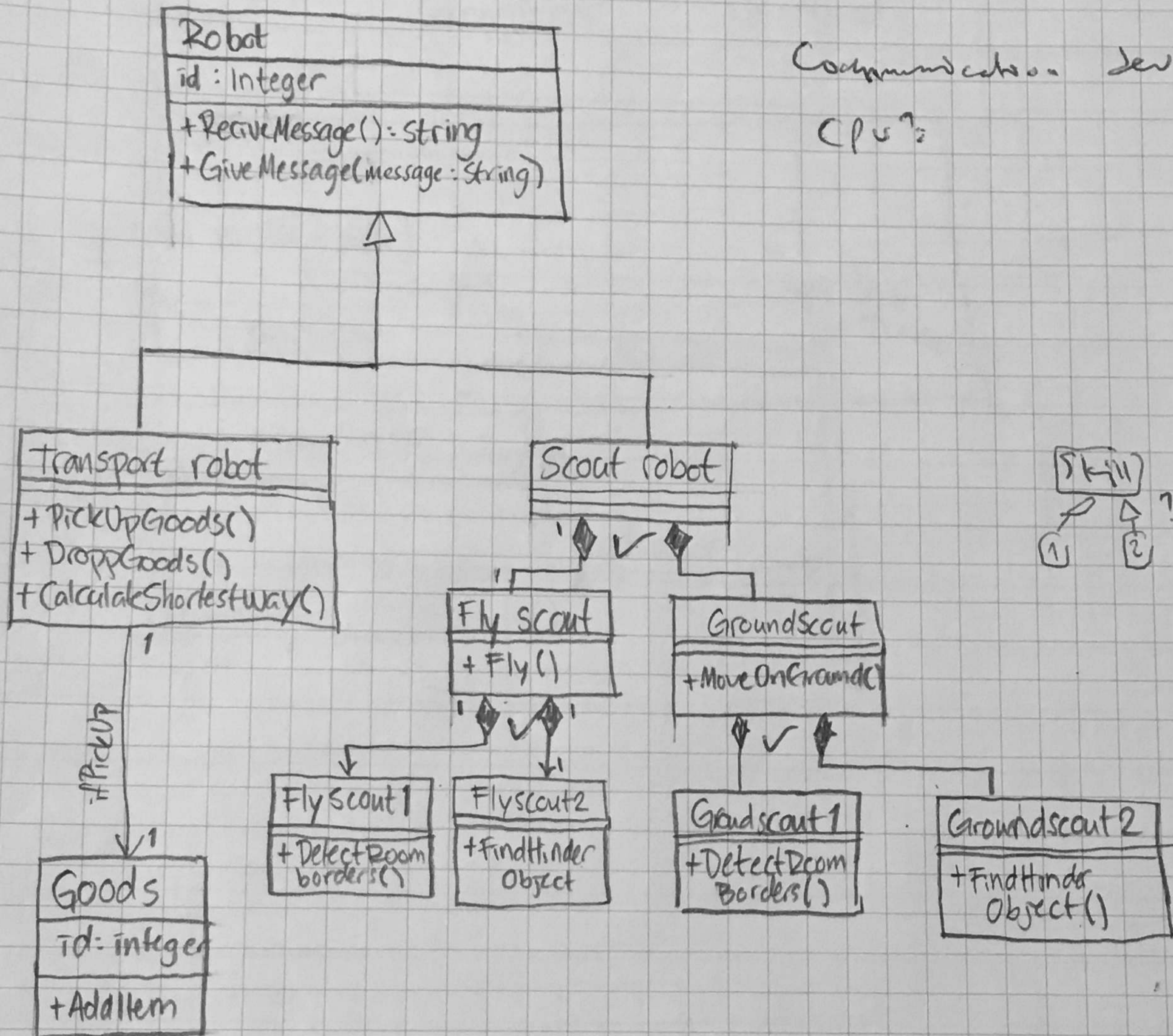
- Time to perform a task on the product.
- After a certain part of the product is finished, tests can be made of these individual parts. By asking the panel to perform a task, e.g. create an account. or change password. It can be made for different parts in different stages.
- Asking the panel to perform the certain task & clocking them, writing down the results.
- Mean value of time to execute a task.
- A low number of this metric indicates high usability since tasks can be executed quick & that indicates that it is easy to understand how to use the system, since the time to understand how to do the task also is included.

Metric 4

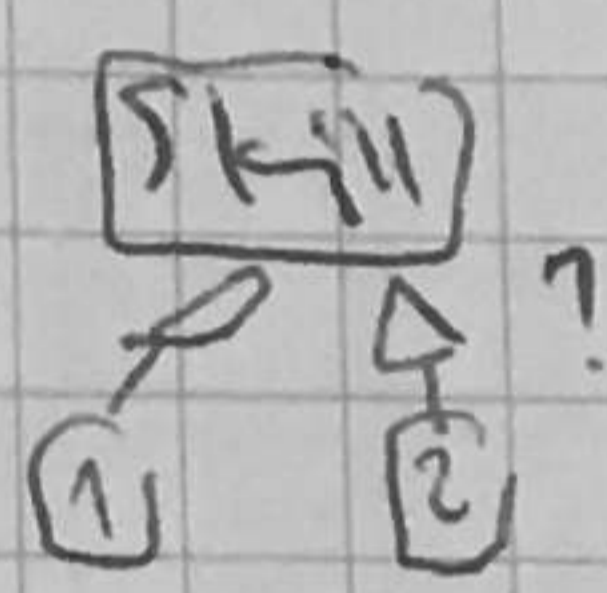
- Loading time vs. playing time compared to
- Load a game and measure loading time, measure average playing time for each game
- information about the above.
- By dividing loading time with playing time
- A low number of my metric indicates high usability since it

4
19

8.

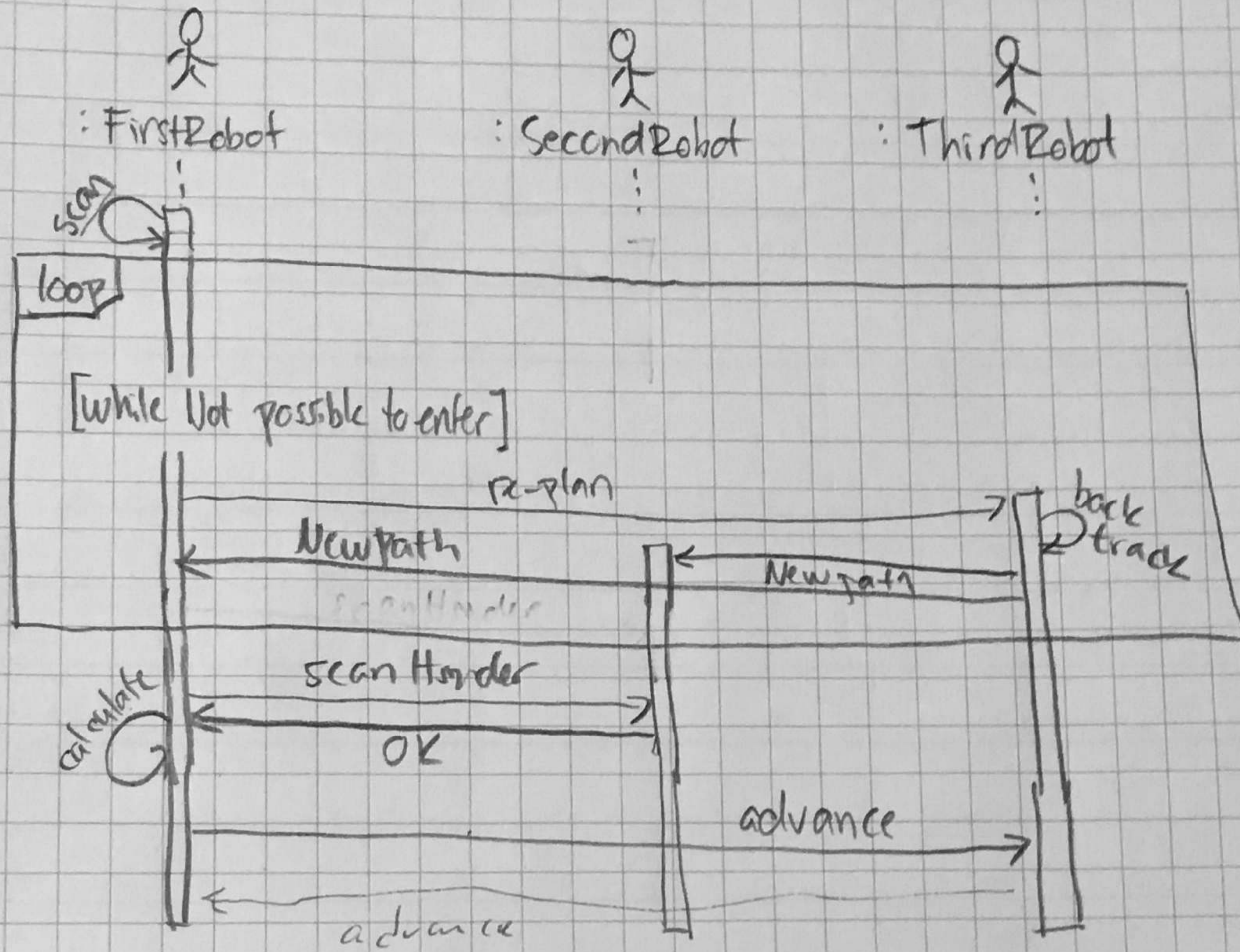


Communication service?
CPU?



6p

9.



9