UNIVERSITETET I AGDER

Emnekode : IS-402

Kandidatnr. : 2208

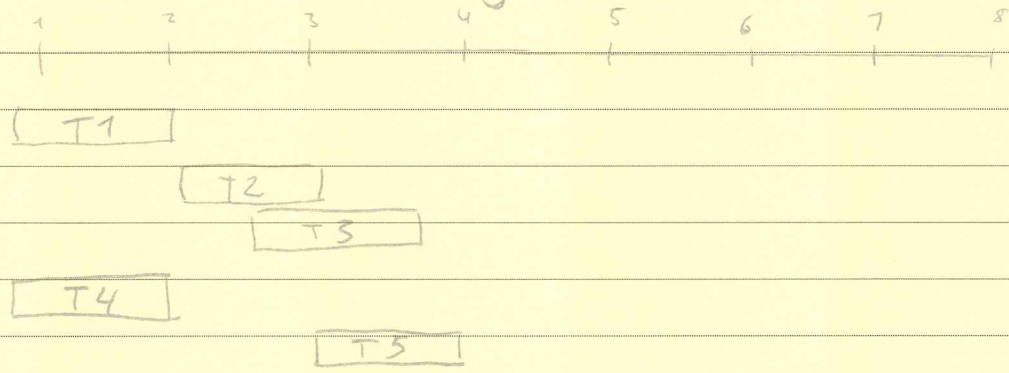
Dato : 25.11.2015

Ark nr. : 1 av 12

EXERCISE 1.

Gantt-Charts is a way of seeing and planning diffrent task in a project, its for getting a overview of what kind of task is important to a notter of when they should start in the whole project. You also see who is responsible for that task and can help see what person is working on what.

this is how a gantt-Chart can be setup.



as we see T1 and T4 is the first task that starts, but we can say that for to strat T2 we need to finish T1 and T4, same for T5, T2 needs to be finished. and like i said before, the tasks has given responsibility to given person(s).

this is one way of planning a project and deviding tasks.

UNIVERSITETET I AGDER

Emnekode : IS-402

Kandidatnr. : 2208

Dato : 25.11.15

Ark nr. : 2 av 12

EXERCISE 2.

When starting a projet, one of the main jobs we have to do early is to be able to know what we are building. this is where requirements come in to place. without them you can almost say you are working "blind". When talking ang communicating with a customer its ~~easy~~ actully hard to decsribe what he wants inn the product, thats why if we document it down with the customer the requriements, we can go trough them and see if there is some missunderstanding. this way its easier to know that we are building the rigth system. this will be easier to Validete with the customer, and verifi withe requreiments.

We have different kinds of requirements. Main functions, thet can be devided in to functinall requriement. And non-functinoll requirements -> quality and performance.

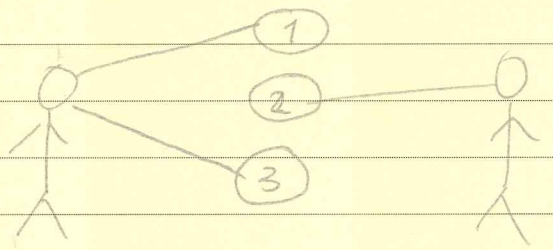
functionel requirements is for exampel "When clicting loginn i should be loggedinn" ~~with the~~ its a inntended innput and a expected output or a reaction to an event.

EXERCISE 2.

performancee requirements can be like runtime and effictiveness -> for exampel "the system should load inn 0,2 sec".

quality requirement can be usability "the design should be easy on the eyes".

these kind of requirement can be identified with user stories, this is a good way to see what kind of requirement it is, an exampel on this is -> "as a stundent i should have access to fronter, so i can have a overview of all my subjects". This way we can identifie what kind of requirements we need. ~~And also using use cases.~~ this is not the only tool, because it desent give us all the information we need, thats wlhy we can also use user_cases. here we can set up a UML-diagram with roles to see the interaction withe the system and other roles,



1. open assignment
2. deliver assignment.
3. correct assignment.

UNIVERSITETET I AGDER

Emnekode : 15-402
Kandidatnr. : 2208
Dato : 25.11.15
Ark nr. : 4 av 12

EXERCISE 2.

all of the ways of identifing user-stories/~~cases~~ can be sorted with MoSCoW, this is away of prioritiseng the requirements:

- Must-have
- Should-have
- Could-have
- Wont-have.

When documenting, you have something being prepared from the principle a product requirement document (PRD) that a agent can menge, this consist of

- Main Objectiv
- appliction domain an users
- main function (no detelts).
- perfomence req
- Quality req
- (most important) data
- Misc.

When we have this we can make a more detiled way, a function specificdon document (FSD). A FSD is a more detaild document specifing ~D product data aim, product development enviorment, testing, appendix. etc. this document should be written in a good way, have consisity, assesseble, Unanbiguos ~D have you coverd everything? nothing going agenst each other? easy to read? you could wrigth it in free prose.

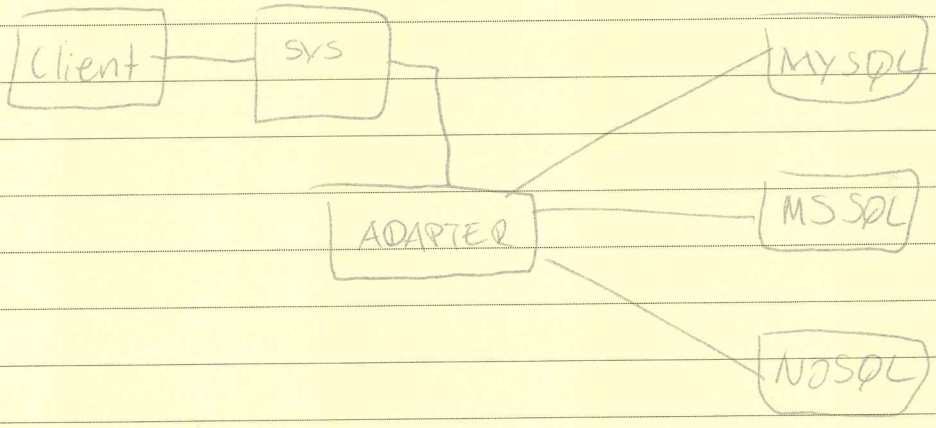
EXERCISE 2.

documenting is a easy and cheap way of getting the requirements down. One of the main aspects of a FSD is it can be a contract with the customer. this is to help you it something goes wrong or the customer says that "no that isent right". If you have FSD you can save yourself from a lawsuit. That's why good communications is important also.
But it can also help you to see and compare with the acctuall product when it's finished, it can be used again with diffrent projet in the future. And help for misunderstandings. Also give a pint point om what we are building.

Requirements is even important in an agile metlods!

EXERICE 3.

the first thing that caugth my eye was database and a commen interface, in this problem they have differenece of database and difference way of sorting them.
Since we can design a commen interface I would use Wrapper, this is a structual way that makes relationtion with entites.
A wrapper can be used to make a commen understanding af diffrent typus of systemes, like a database, if we have for example: MySQL, MSSQL, NoSQL, they are structural in a diffrent ways. Here we can use a wrappes to make a commen interface

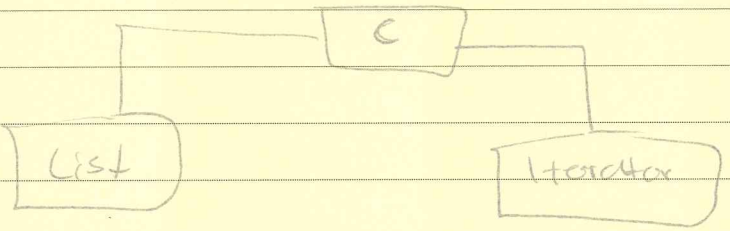


As we can se they go trough a adaptor that reads and makes a commen interface. this is also commen in merging situation. this is a good way for ~~[illegible]~~ having to make Access from one system when haveing more datasystems.

EXERCICE 3.

Im a lillle worried heee on this one, because the CRM has a list, so i don't know if you can use a iterator that can go trogh the list, as explained in the exercice different sorting modules. I have explained a wrapper because it's makes more sense when you are desribing with diffrent datas and a commen interface.

But as i said it can also be a iterator.



that goes trough the list ~~until~~ of that list, It iterets trough, for(element : ~~current~~).

UNIVERSITETET I AGDER

Emnekode : 15-402

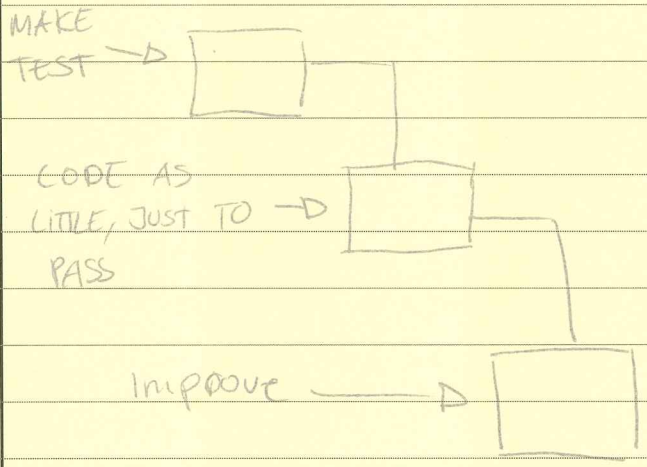
Kandidatnr. : 2208

Dato : 25.11.15

Ark nr. : 8 av 12

EXERCISE 4.

In some ways you may program first, then make a test case and test it. But here is wher TDD is different, you make the test case BEFORE you make the code. In TDD after you have made a test case, you only make as much code as possible for it to pass, after that you can improve it or refactor the code better. the whole idea is to make as little as possible. And then improve it.



One of the draw backs is that for GUI it's not a good way for testing when becoming to complexed, some have argueed that BDD is better for GUI.

EXERCISE 5

"SCENARIO OF PLAN-DRIVEN:
"We are making a big project for the goverment, its a system that holds all of the information on pays an task (like altinn). Security is a big priority and Objective is clear for everyone."
Since we are building a big proret and would have a lot of people on this project it would be smart for a plan-driven approach, with these kinds of project a good structure is important. When we have a lot of people, it may not all be expiriend, a plan-driven would help with that. In This senario the goal of the project is clear and chenges to it would be minimell. or forseen. on beeforhand. then a plan-driven approach isam be better.
Security is also a big priority, this would be a good way of planning very good beforhand on what needs to be donc. I would use a water fell-modell for this senario, its a good way of plannin before going trough the impementaik'on.

~~REQUIREMENTS~~

~~ANALYSIC/DESIGN~~

UNIVERSITETET I AGDER

Emnekode : IS-402

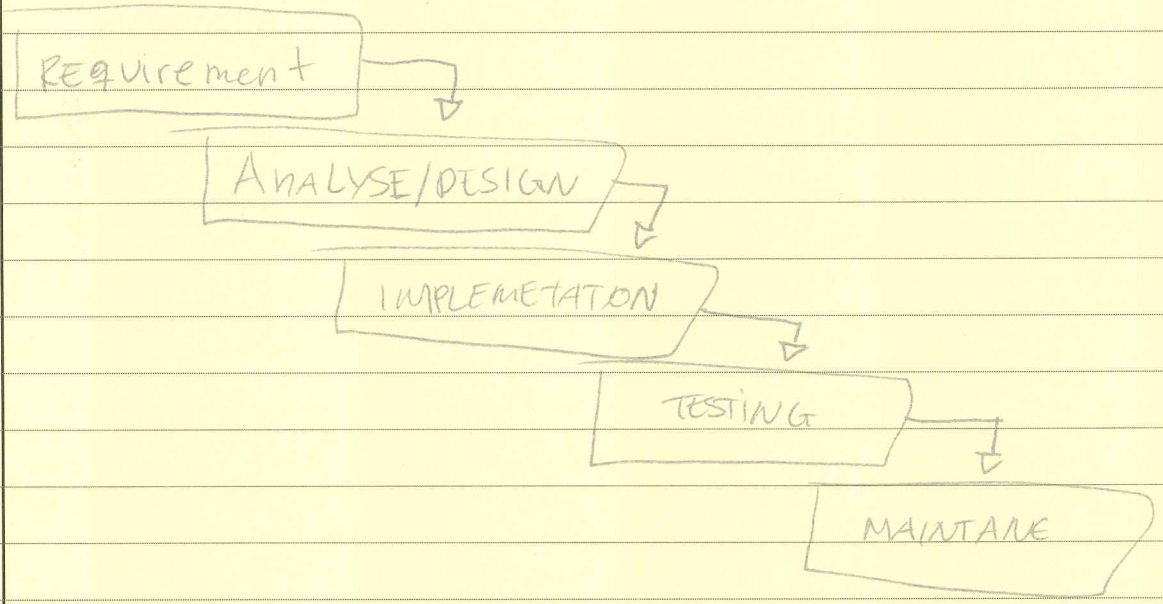
Kandidatnr. : 2208

Dato : 25.11.15

Ark nr. : 10 av 12

EXERCISE 5.

A watterfall modell is a step way approach it has usally 5 stages:



With this ~~modell~~ method after you finish ~~the~~ one of the stages you move on to the next one, this is way you can't go back ~~to~~ or "up". And why its a watterfall modell.
But you a plan-driven a approch as i said before is a godd way if you know "everything" on beforhand and changes is low, to plan ahead detaljed so that everyone know what to do. a lot of people, big project, maybe not everyone is a expert for security reasons it should neve be depoyed if not tested "good" enough, also if the customer isent around all the time planning good beforhand is good so you know what to do.

UNIVERSITETET I AGDER

Emnekode : IS-402

Kandidatnr. : 2208

Dato : 25.11.15

Ark nr. : 11 av 12

EXERCISE 5

SCENARIO OF AGILE:

" small project for a firm that wants a CRM-system, ~~and or~~ ~~not~~ with good comunication with the team."

When we are talking about an agile approach, we are talking about be able to make agile desicions. this is for exampel if changes needs to be done while in the ~~FFFP~~ develoment of the program. A agile you want to have good communiction. We have maybe around 6 (+,-) people In the project and everyone knows what to do.

I would use a extreme programming (XP). XP is a good way for developing if you have good comunication withe the customer, and they are allways avaible, and the team. You can set up a enviroment where you have pair programing, 2 people, one driver for tactical approch an one navigator, that has a bigger picture of ~~the~~ program, ~~It's~~ Its higher quality and can be the same cost can be a expert programmer and a junior programmer. they switch roles as they go. this is also a good way for giving tacit knowledge.

UNIVERSITETET I AGDER

Emnekode : IS-402

Kandidatnr. : 2208

Dato : 25.11.2015

Ark nr. : 12 av 12

EXCERCE 5

XP is using also TDD, as explained in excerice 4.

with a XP approach you would make a redse plan withe the customer for what we are building next from user-stories. Beware from every interall a product has to shown to the customer + delivered. this is why changes can happen and eadly changed, product is done, and if there is something else customer want it can be done. With XP we need to have the customer availble all the time. this is for missunderstandings or just needs somting clearer. With XP we have something called (20:80). this is 80% is actully development of the requirement and 20% time is maybe improvments or other. In XP every body owns the product/code. so every one can go inn and make changes, this is good and bad if you dont have to many people. XP is very agile and i would probably use this, plan-drien is about document, planing, ressorses. But Agile is about product, customer, people, this is my personall view, becouse you need experienc people with XP, plan-drien is better for big project and not thett experienced.
XP if it breat... fix it!