



Emnekode : IS-402
Kandidatnr. : 220 2216
Dato : 25/11-2015
Ark nr. : 1 av 9

Ex. 1

Milestone planning is a very commonly used method in development project planning, and is used to set up different "goals" and "sub goals" for a project. These goals are used both as a way of helping the development team see that the project has progress, but also as a way of showing the customer how far the project has come, its "status quo", and what's next in the project. An example could be that the development team has just finished a module for a system and reports on the module and the system as a whole, as well as saying what part of the system is up for development next. Milestones should be well formulated and concrete, and should not go too much in detail on what needs to be done. Specific requirements are often stored somewhere else (functional specification document, product ~~prod~~ product backlog etc.).

Ex. 2

First off, developing a system for a customer without knowing what the customer wants seems to be a pretty bad idea, if not impossible. For example, a customer could want you to build him a vehicle. He wants a plane, but doesn't tell you, and you end up building a car instead. The customer ends up being disappointed, and you frustrated. It is important



Emnekode : 15-702
Kandidatnr. : 2216
Dato : 25.11.2015
Ark nr. : 2 av 9

Compdumer

Ex 2 cont.

It is important to gather requirements from the customer, and a good idea would also be to sort these in two groups, functional and non-functional.

Functionality requirements is about how the system works and what it does, and also how it looks, e.g. button placement, navigation, button functionality etc.

Non-functional requirements are more about the quality of the system, and what kind of requirements the system puts on the developer, and the development environment, and hardware and such. For example, non-functional requirements could be performance requirements such as hardware requirements, downtime percentage, response time, usability requirements etc.

To identify these requirements, one has a lot of options. For example, one could conduct interviews, be an observer in the natural working environment, hold focus groups, send out surveys and such. However, techniques based on face-to-face interaction works the best, as it lets the software provider observe body language, which is an important factor when pitching or receiving ideas.



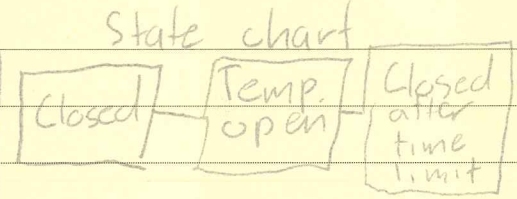
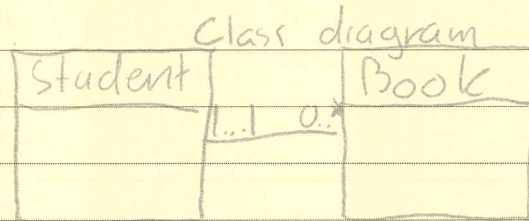
Emnekode : 15-402
 Kandidatnr. : 2216
 Dato : 25.11.2015
 Ark nr. : 3 av 9

Ex. 2 cont.

There are a number of ways to document requirements. Perhaps the most obvious ones are setting up user scenarios (stories about the use of the system in a fixed setting) or user stories (short, concrete sentences that where a user's role, an exemplary activity, and a given benefit of that activity is listed). A user story often looks like this:

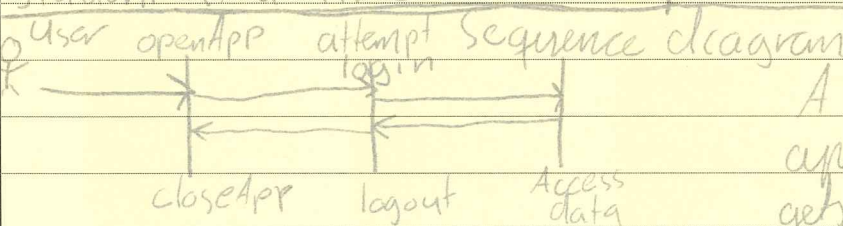
AS A <role>, I WANT <activity>, SO THAT <benefit>.

One could also use different types of diagrams. For example, a class diagram shows all objects in a system, and relationships between them. A sequence diagram sequentially explains the execution of a function, while a state chart shows when an object in a system changes its state. Illustrations below.



Library. A student can borrow zero or several books. A book can only be borrowed by one student at a time.

Example with hand-in folders in Frontier.



A user opens an application, logs in, gets requested data, logs out, and closes the application.



Emnekode : 15-407
Kandidatnr. : 2216
Dato : 25.11.2015
Ark nr. : 4 av 9

Ex. 2. cont. The functional spec. document should contain specifications on what the system should do, as well as information on how it should do things performance-wise. It is important that the document is:

- Comprehensive
- Unambiguous
- Accessible
- Complete
- Consistent
- Irreversible

And it should address considerations about different stakeholders concerned with the project. A functional specification document can be based on some of the points mentioned in a product requirements document, and can be part of a feasibility study.

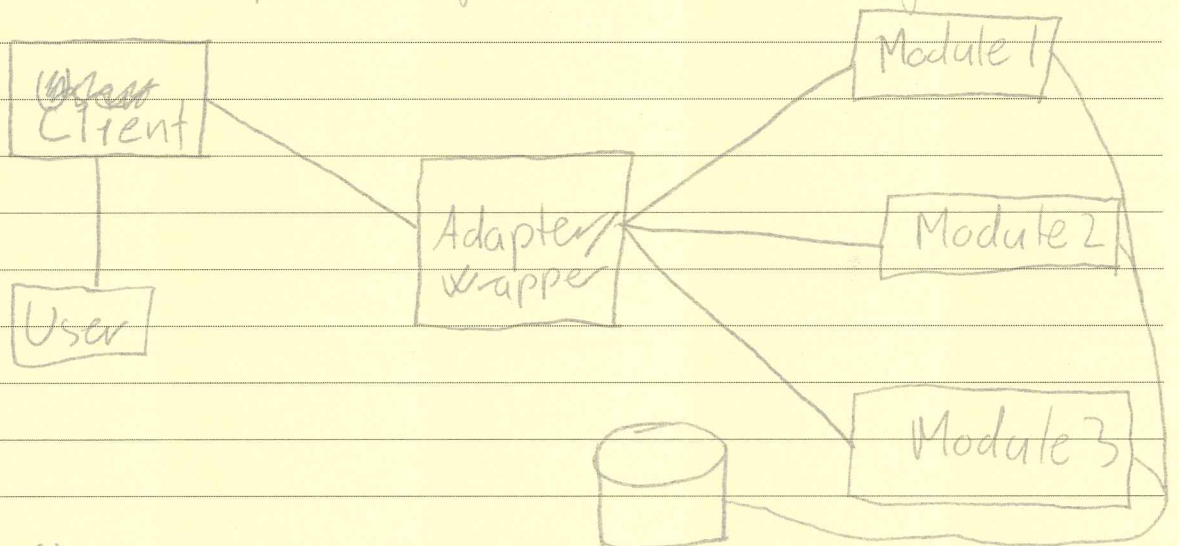
All in all, ~~the~~ developing without requirements is about as useful as walking paint dry on a wall, if not less. To make sure the customer ~~is~~ gets what he wants, requirements engineering is essential!! I have covered some factors that ~~need~~ can be considered when setting up requirements for a system, but not all. All projects are different, and not all techniques and methods work for all projects.



Ex 3

For bundling putting different subprograms together with a common common interface, the natural choice would be to use a wrapper. What a wrapper does is that it allows the user to control several different modules that might have different architecture, structure or design in general. An example here could be to control different types of databases by using a common tool (MySQL, NoSQL, JPQL). This can also be applied to routing tech-
~~ing~~ requires that for example use different kinds of collections (mesh, list etc etc.)

Sketched up it might look something like this:



The wrapper would here work much like a power adapter for laptops. Much like the adapter makes sure the laptop gets the right amount of power, the wrapper lets the user ~~control~~ get the controls for manipulating each module respectively through a common interface.



Emnekode : 15-402
 Kandidatnr. : 2216
 Dato : 25.11.2015
 Ark nr. : 6 av 9

Ex. 4

TDD is a testing approach often used in agile development contexts. For example, it's a mandatory part of extreme programming.

TDD is all about writing tests for the program being developed before writing the code. Also, when writing the code, one only produces enough code to fulfill the testing criteria.

The reason why TDD fits so well with agile development is that it always gives you a new piece of code to show the customer. This leads to constant continual collaboration between the provider and the customer, and makes the project open to change. TDD would not be recommended for large, plan driven projects, as a mid-project changes are not very well received in such projects.

A disadvantage by using TDD is that if one is programming bottom-up, one ~~would discover~~ could discover large errors, bugs or failures when testing of the entire system is being best conducted. One would then have to go back into the smaller components and units and provide a fix, which could lead to revealing even more defects, which could end up being quite a costly affair. This could actually also happen programming top-down. Regression tests are therefore important when using the TDD approach!!

Test 1: Test 2: Test 3: Test 4:

A A B A B C A B C D

Illustration shows ~~TDD~~ in theory. Test changes and additions!

Reg. tests



Emnekode : 15-402
Kandidatnr. : 2216
Dato : 25.11.2015
Ark nr. : 7 av 9

Ex. 5

As a given scenario, let's say a way that the CRM tool is only a module for a larger system, meaning most of the code has already been written in terms of compatibility and reuse. Therefore, the project does not become one of the biggest of scale, but a pretty big project nonetheless. The customer does not really have the time to provide any deep, concrete requirements, and he is not really sure what he wants in the end. Also, he is not entirely sure when he needs the system, the only thing he cares about is that it is done sometime sooner or later. However, you have a team of highly competent ~~user~~ experts ready for the job, flexible for any given timeframe. Also, you are eager to keep the customer up to date by informing on progress and changes along the way, even if the customer is busy running his business. Which method to choose?

Another scenario could be that the tool should be developed independently from scratch. This would make the project become large in scale, and with high criticality. The customer has a large number of specific requirements that need to be documented at a high level of precision. As he is busy with his business, changes along the way will not be tolerated. The customer has set a date for when he needs the tool, and this date can not be overrun. Your team is a mix of people, with some experts, and some less skilled people. The customer does not want



Emnekode : 15-402
Kandidatnr. : 2216
Dato : 25.11.2015
Ark nr. : 8 av 9

Ex. 5 cont. any updates along the way, he just wants a working product.

It's pretty obvious here that the first scenario is of the sort that fits agile development best, I would have recommend SCRUM, as it is a method tailored for working in increments, meeting requirements for reporting to the customer, while also being flexible to change and "bumps in the road". Also, having a team of experts on the case makes the team pretty self-maintaining and self-administering. As the project does not have a given time limit, using SCRUM is also a good idea, as increments can be added after needs. As the project is medium-sized, an agile approach like SCRUM would be a good idea.

For the second scenario, I would recommend using a plan-driven method like the waterfall model. The reason for this is mainly the project's lack of flexibility. There is a given timeframe along with a set of specific requirements. Also, the customer is not depending on how far the project has come as long as he gets his product by the deadline. Also, the team consists of a mixed group of people (some experts, some mid-range and maybe some some idiots), meaning that working after a specific plan may be to prefer.



Emnekode : 15-402
Kandidatnr. : 2216
Dato : 25.11.2015
Ark nr. : 9 av 9

Ex 5 cont.

The reason why the difference in scenarios needs to be addressed methodologically is to find the most efficient way to develop a product. Ineff. Inefficient projects can become timely, and thus costly for both the provider and the customer, which is not beneficial for ~~it~~ either ~~part~~ one. Also, working after a method that does not fit the organization, can lead bring more problems than benefits (eg. wrong product, demotivated employees, doubt in top level management etc.)