Home exam 1DV600, 2016-03-19

Name: [Henry Pap]

Personal ID: [96-01-09-7398]

Lnu email: [hp222fq@student.lnu.se]

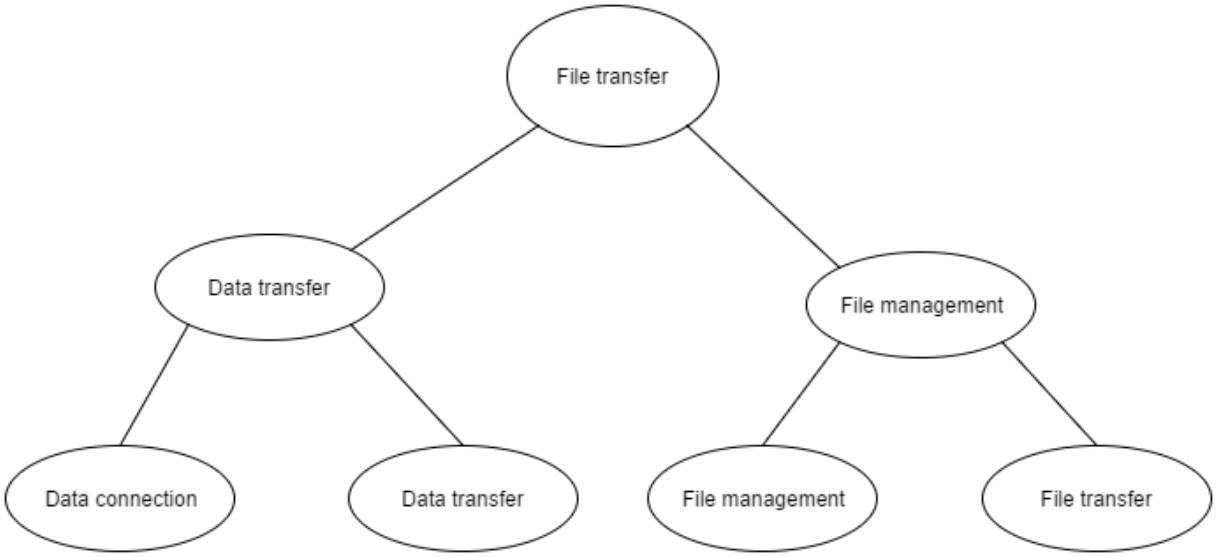
Task 1

a)

By planning iteratively the project will be divided into smaller projects. These smaller projects can then be planned with timing, design. And by planning incremental huge/bigger problems will be broken down into smaller parts “increments” when adding them all together a solution is built.

Combining these two methods we gain control, features/requirements will often be treated like increments which can be broken down into iterative projects. They will have a planning were due date and requirements will be set and when finished focus will be on the next iterative, when all is finished the increment is considered finished and first part of project is done.

Work breakdown structure
(using example from slides)



b)

What is a risk?

A risk can mean a lot, it can be failure in system, tools that is unstable, people with lack of knowledge or dependency. Basically everything than can contribute to time or money loss.

There is two good planning modules “top-down” and “bottom-up”. Top-down approach is planning ahead, this will “generate” a work breakdown structure. Bottom up approach is more agile it has a base plan (requirements) and also iterative plans which makes it easy or future changes to be made.

Top-down – plan everything to avoid risks

Bottom-up – lose plan so changes can be made to avoid risks.

Examples

Top-down

Clients and developers/leaders talk and sets the requirements. A big plan is made and the developers will follow it to the end. Clients can’t really make any changes during the process.

Bottom-up

Clients set up requirements and developers starts planning requirement. Together with clients developers build the projects as client want.

Task 2

a) SCRUM compared to OpenUp

Roles

Scrum has three major roles product owner, scrum master and development team while OpenUp has more such as stakeholders, analyst, architectures, developer's project manager and testers.

Artifacts

In scrum increments are produced from sprints also sprint backlogs and product backlog. In openup test cases, micro increments project plan, software design are produced.

Activities

While scrum is based around sprints were daily meetings, retrospective, review, planning is made OpenUp has these separate planning, tests, architect design and so on.

b) Pros and cons

SCRUM pros

- Effective use of time and money
- Large projects are split in sprints
- Developments are coded and tested during sprint reviews

Cons

- Adopting scrum framework in larger terms is challenging
- Only experienced team members can get scrum to work

OpenUp pros

- Gives structure to project
- Easy for beginner developers to follow plan
-

Cons

- Not much freedom for developers
-

Task 3

a)

Scenarios is a good way to understand the different needs in a project. It defines behaviours in different situation and the expected outcome of it (result).

Example a login system

- User enters correct username and password and press login button thus logging in.
- User enter false username or password and press login button, an error message popped up saying “wrong username or password”.
- User forgot password and presses the ‘forgot password’ button, user is taken to a form where email is required to be able to changing password.

These are just some scenarios but is shows how the system should be built and how to behave. Just by these scenarios we can clearly see we need some use cases for example a login, reset password and how these use cases should look e.g. login requires password, username field, a login button and a forgot password button.

b)

Requirements can be hard to capture, some requirements are functions some are performance related, FURPS+ come handy when it comes to categorising requirements. Functionality, Usability, Reliability, Performance, Supportability are the categories requirements can be grouped in, the plus is about design, implementation, interface and physical requirements. Together they make up a great template to understand what different requirements are.

Example our login system

Requirements	Following the FURPS+ model
Login form	Usability (could also be design but not the major thing)
Reset form	Usability
Database	Implementation
Error message	Supportability
Sending reset email	Functionality
Login	Functionality
Reset	Functionality

Task 4

a)

A test case is a combination of test in a specific part of the system were data like expected result, actual results, scenarios defines the test case.

Example

Back to our system

A test case to our system could be on the login system itself (not reset part), were input risks can be tested, database and error handle.

A test case would be an entire document with description and other information is set but the main thing would be the data and a table is a good way of showing it.

scenario	Test steps	Expected result	Actual result
User enters username containing false characters such as “<>#	Enter username: <hello>	Login form should display an error message	Login form displayed error message
User enter correct values	Enter username and password that exist in database	User should be logged in	User login
User enters username that does not exist	Enter username that does not exist in database	Database should tell system that user does not exist thus display a error message	Login form displayed error message

b)

Unit testing can be viewed as the smallest testable part of a system a method in a class for example. The test object is the unit itself and the objective is to detect defects in code.

Integration testing comes after unit testing were the test is focused on quality and functionality. The test object is large pieces of system if not the system itself and the objectives is to see how the pieces in the system works together.

Unit test vs integration test

Unit test is on a small level to see how methods work, it is an easy implementation that doesn't require much and is mainly for the developer. As for the integration test it shows a bigger perspective and require more resources like databases and does a more convincing job on how the system works, is a good way to show customers and other non-programmers that it works.

Example

Back to our login again

A unit case in our login system could be the filtering part were the input values is being checked for invalid characters. The integration test could be testing the whole login system, showing us how the different pieces corresponds together (like the test case example).