

Functional requirements

Software Engineering - week 3

- Functional requirements specify the software functionality that the developers must build into the product to enable users to accomplish their tasks, thereby satisfying the business requirements.
- Sometimes called behavioral requirements, these are the traditional “shall” statements: “The system shall email a reservation confirmation to the user.”

Software Requirements, 2nd Edition (K.Wiegers)

Functional requirements

Software Engineering - week 3

- You should find and sketch about 15 different functional requirements from the domain.
- You can discuss with your exercise guide about meaningful requirements.
- You are able to sketch functional requirements from your use case model and requirements list given by the customer.
- It is also good to sketch requirements of your own.
- A use case could include several functional requirements. It is usual that you can find more than one requirement.
- **You should find more requirements than just the list from the customer!**

Functional requirements

Software Engineering - week 3

- Document description, origin and limitations.
Generate a unique ID all of your requirements.
- Backwards traceability depends on the origin from the requirements.
- Example description and limitations:
 - User can check his/her account balance from the system
 - User can check his/her account events from the system
 - User must get an event list on paper
 - The card must return to the user before money

Functional requirements

Software Engineering - week 3

- Your requirements have to be *unambiguous*.
- Be specific when using different words (for example user name: what do you mean by that?)
- Functions of the system are described in little pieces.
- If you find two different requirements from one description divide the requirements for two different requirements.
- Familiarize to Karl Wiegers example requirement documentation. You might find it valuable.

About unambiguous

Software Engineering - week 3

- A requirement is unambiguous if its possible to plan a method which demonstrates that the requirement is executed.
- For example sentences:
 - "Software works fine"
 - "Product is user friendly"
 - "Software produces a print normally in ten seconds"
- These ARE AMBIGUOUS words because "fine", "user friendly" and "normally" are not objectively interpret.

Software requirements

Software Engineering - week 3

- You and the users must agree on actions they can perform and response they should expect. This common understanding is captured in the requirements document.
- How the software responds to the agreed upon request is not addressed in the requirements document. For example, the requirements document does not include screen layouts, database schemas, descriptions of communication layers— in short, no statements of design of any sort.
- For example, it is a requirement for a word processing application to be able to open an existing file. It is a design issue whether to build a customized file selection tool or use a platform standard file selection tool.

Tasks for the third week

Software Engineering - week 3

- Functional requirements are divided to software requirements (about 20 different requirements).
- Think requirements from the viewpoint of the software.
- Software requirements cover the functional requirements and specify them.
- Different kinds of software and non-functional requirements:
 - Performance, interface, use, resource, verification, acceptance test, documentation, security, maintenance, safe operation

Tasks for the third week

Software Engineering - week 3

- Reference for the functional requirement(s) is needed. Where the software requirement comes from or what is the other source of the certain requirement?
- Familiarize to IEEE standard section 5.3.
- Ivy Hooks: Writing Good Requirements

Requirements - example

Software Engineering - week 3

- User requests event list from his/her account
 - Account events are read from relation database
 - Response has to come in one second
 - Database communication is handled in JDBC-interface
 - Telecommunication must be crypt with RSA.

Traceability matrix

Software Engineering - week 3

- Familiarize with the matrix from the document template.
- Make the matrix using functional and software requirements

Requirements traceability matrix Software requirements vs. functional requirements		Date: 7.3.2005 Page 1 / n
Project: ATM		
ID of functional requirement	ID of software requirements	Description of functional requirements
FR-1	SR-1 SR-2 SR-6	Tilitapahtumien hakeminen
FR-2	SR-2 SR-4	Rahan nostaminen tililtä

Some tips for the software requirements

- Simple and explicit requirements are response time/performance, reliability, acceptance/test requirements
- For example identification of administration or right verifications.

From use case to requirements

Software Engineering - week 3



FR-01: User can log in with his/her account to the server

OR-01: Maximum length of user name is 20 characters

- 
1. Use Case
 2. Functional requirement
 3. Software requirement

ID of functional requirement	ID of software requirement	Description of software requirement
FR-01	OR-01	Login to the server with user name and password

Examples

Software Engineering - week 3

FR-1	Priority	Locked	Priority	Stability
	Substantial	No	1	TBC
Origin	Customer requirement list			
Description	ATM reads the event list from the server.			

SecR-1	Priority	Locked	Priority	Stability
	Substantial	No	1	TBC
Origin	From the planning meeting in Oct 24 2002; FR-1			
Description	Telecommunication connection practice must be HTTPS and SSL 3.0.			

SecR-2	Priority	Locked	Priority	Stability
	Substantial	No	1	TBC
Origin	From the planning meeting in Oct 24 2002; FR-1; SecR-1			
Description	Telecommunication must be crypted in 1024 bit RSA.			

Software Engineering - week 3



Tasks for the third week

Software Engineering - week 3

- Tasks for the week 3:
 - Functional (user) requirements of the system
 - At least 15 different unique requirements
 - Software (non-functional) requirements of the system
 - At least 20 different unique requirements
 - Traceability matrix for the software vs. functional requirements