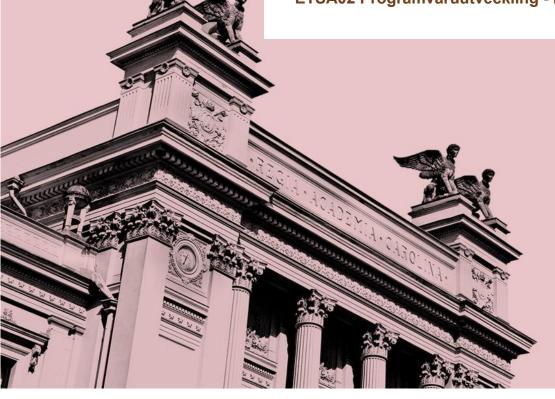


Exercise 3: Requirements specification

ETSA02 Programvaruutveckling - Metodik, VT18





Exercise 3 - Goals

- Hands-on work: start working on detailed requirements
 - break down a selected feature
 - specify detailed requirements
 - discuss requirements with other groups
- Output for projects:
 - concrete progress toward SRS v0.9 (Beta release)
- Feedback on Alpha Release



Exercise 3 - Agenda

- Introduction (10 min)
 - The purpose of requirements
 - Characteristics of good requirements
- Work in groups (35 min)
 - Specify detailed requirements for a selected feature
 - Send result to teacher for printing during the break
- Requirements workshop (45 min)
 - Printouts organized in four station areas in the classroom
 - Groups rotate to analyze other groups' requirements
 - » Report positive feedback and improvement potential on sticky notes

Requirements is about communication

- Who will read your requirements? What do they know?
- Internal:
 - developer to know what to build
 - tester to know what to test
 - sales engineer to know what to sell
 - project manager to overview progress
- External:
 - customers to know what to expect
 - Regulatory Body to enforce standards



The many faces of the SRS

- SRS is a contract
 - what the supplier promises the customer
- SRS is a fundamental description of the system
 - a target for the development organization
- SRS is input to planning
 - helps a project estimate effort and plan resources
- SRS is a goal for testing
 - without an SRS, testers' task less defined
- SRS is a cornerstone for high quality software
 - numerous studies show importance of SRS



Characteristics of good requirements

Correct = If the requirements are not correct, we risk spreading misinformation within project and to customers. Incorrect requirements are useless and potentially dangerous!

Unambiguous = Everyone understands it the same way! Can everyone read, discuss and agree on what it means?

Clear & Concise = Simply and clearly stated. Makes it easier for others (incl. non-team members) to understand.

Verifiable = If a requirement is not verifiable, determining whether it was correctly implemented is a matter of opinion.

Complete = Spec covers the full scope? Exceptions, error handling? Only reference to specific versions of documents.

Consistent = Are there requirements that contradict each other?

Design independant = Describes functionality from user perspective, not how to implement

Traceable = What motivates this req? Indicates if & how much it is needed. Useful when discussing scope &/ reqs changes

Modular = Can be changed without excessive impact.

