

ITEA



Information Technology for European Advancement

Task 1.8 - RM problem description (D 1.8.1b)

Version 01 - Public
Edited by Loek Bokhorst

Software Development Process for Real-Time Embedded Software Systems (DESS)

ITEA COMPETENCES involved:

- 1) Complex Systems Engineering**
 - 2) Communications**
 - 3) Distributed Information and services**
-

ITEA

1. INTRODUCTION	1
1.1 Purpose	1
1.2 Structure.....	1
1.3 Terms and abbreviations	1
2. REQUIREMENTS MANAGEMENT PROBLEM DESCRIPTION	2
2.1 Requirements Management	2
2.2 Requirements management for components	3
2.3 Problems with Requirements Management	3
2.4 Requirements coverage.....	4
2.5 Impact analysis.....	4
2.6 Requirements for Requirements Management	4
2.7 Illustrations of items to cover concerning Requirements Management	6
2.7.1 Component reuse	6
2.7.2 Users of components.....	6
2.7.3 Component information model.....	7
2.7.4 Determining the quality status	7
2.8 Tools integration	7
<i>State of the art problems with tools integration</i>	<i>7</i>

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to give a description of the current problems that exists with Requirements Management when developing innovative object-oriented component based software for embedded real-time systems.

This problem description forms the basis for finding solutions on how to deal with Requirements Management within the ITEA DESS development methodology.

1.2 Structure

The structure of this document is as follows.

Chapter 2 describes the current problems that exists with Requirements Management when developing innovative object-oriented component based software for embedded real-time systems. It contains the common insights by the ITEA DESS partners.

Chapter 3 describes the problems that each ITEA DESS partners currently encounters with Requirements Management in their development process.

Chapter 4 contains minutes of meetings and discussions that were held on defining the problem, starting with the minutes of a brainstorm session on this subject in Praag (oct. 2000).

1.3 Terms and abbreviations

CRS	Customer Requirements Specification
SRS	Systems Requirements Specification
Verification	Do we build the product right ?
Validation	Do we build the right product ?

2. REQUIREMENTS MANAGEMENT PROBLEM DESCRIPTION

2.1 Requirements Management

Requirements management includes: the structuring and administration of information, which is processed during the acquisition, derivation, analysis, verification, validation, baselining, updating and tracing of requirements over the entire product life cycle.

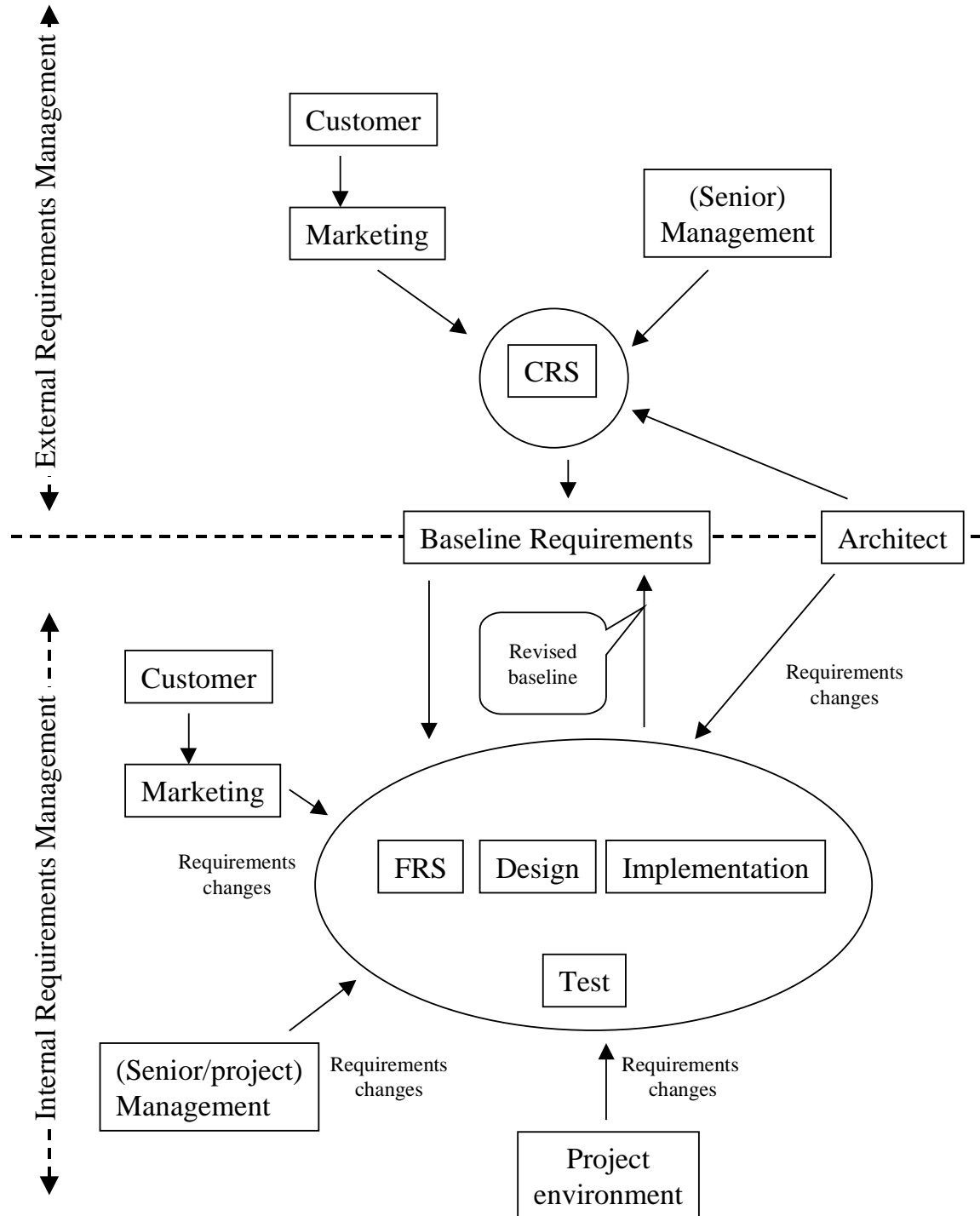


Figure 1 Requirements Management

ITEA

Goal of requirements management: Requirements are consistent throughout the whole development process.

Via verification and validation activities, the difference between the required and actual status of the requirements is measured and documented.

2.2 Requirements management for components

On component level, requirements management consists of the following:

- What are the requirements for the component ?
- Which of the requirements for the component are covered (implemented) ?
- Which of the requirements are not yet covered ?
- How much work is involved with implementing the missing requirements ?
- What is the required quality of the component ?
- What is the actual quality of the component ?
- ...

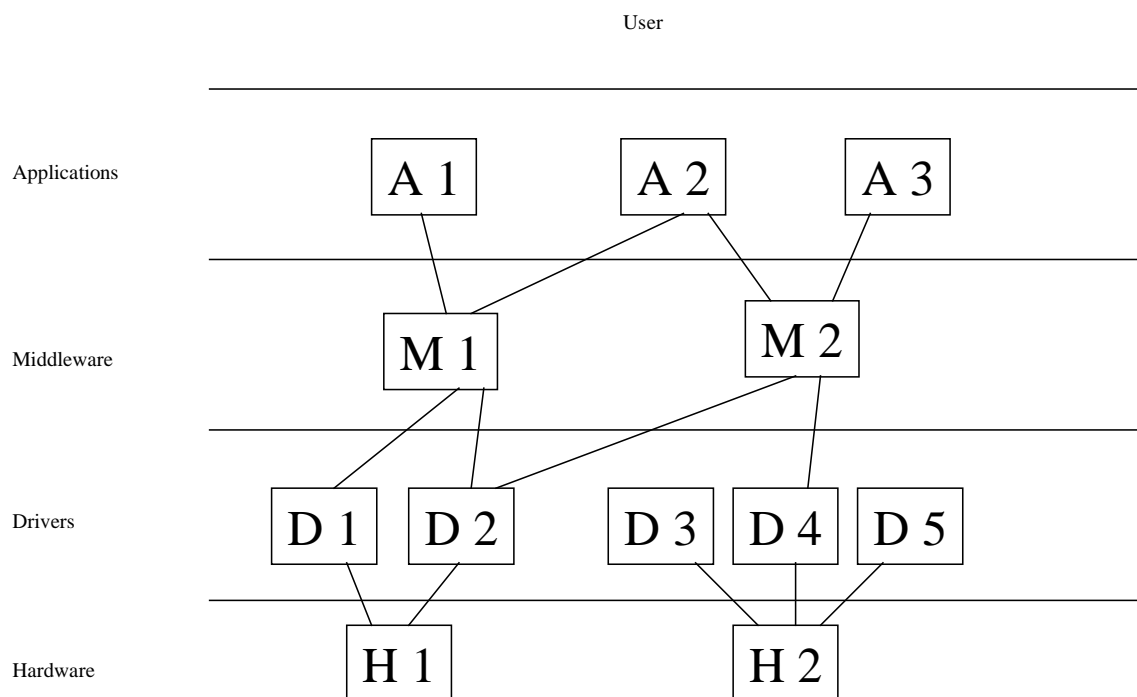


Figure 2: Example of a component based system

All answers to these questions depend on a version of a component. I.e. may differ for past-, present- and future releases of the component.

2.3 Problems with Requirements Management

For managing the requirements, information is needed about:

- Requirements coverage
- Impact analysis of new- and changing requirements

2.4 Requirements coverage

Concerning requirements coverage, following questions are relevant:

- How much of the 'higher level requirements' are covered, using a given set of components ?
- ...

2.5 Impact analysis

Concerning 'impact analysis', following questions:

- What is the impact of the higher level requirements on the requirements for the individual components ?
- What is the impact of adding or changing higher level requirements ?
- What is the influence on the system requirements when one component is replaced by another ?
- What is the importance of a component ? Perhaps one component is essential for multiple 'higher level requirements'. This component could get an higher importance (and therefore it could be decided to implement it earlier, and test it more thoroughly, ...)
- ...

2.6 Requirements for Requirements Management

In order to be able to manage requirements, traceability of requirements should be possible.

Traceability implies:

- Forward traceability: From 'higher level requirements' (as described in the CRS/SRS) to individual components to (partly) integrated components up to a complete system
- Backward traceability: From complete system, individual components up to 'higher level requirements' (CRS)

Note: With 'higher level requirements', the following is meant: requirements for platforms, (sub) systems or even entire product families

Traceability of requirements means that it is possible to verify whether requirements from one development phase are still incorporated completely in a next current phase. For example whether all requirements are still available in the design, all designed components are coded and all requirements are tested.

Additionally traceability must guarantee a proper change control of the requirements. No new requirements should be added in an uncontrolled way during the next phase.

Traceability should avoid that features are tested which are never specified. Traceability should make it possible to verify whether everything which is specified, and nothing more than that, is eventually available in the product.

Requirements should be kept consistent throughout the whole development process. Tool support is required.

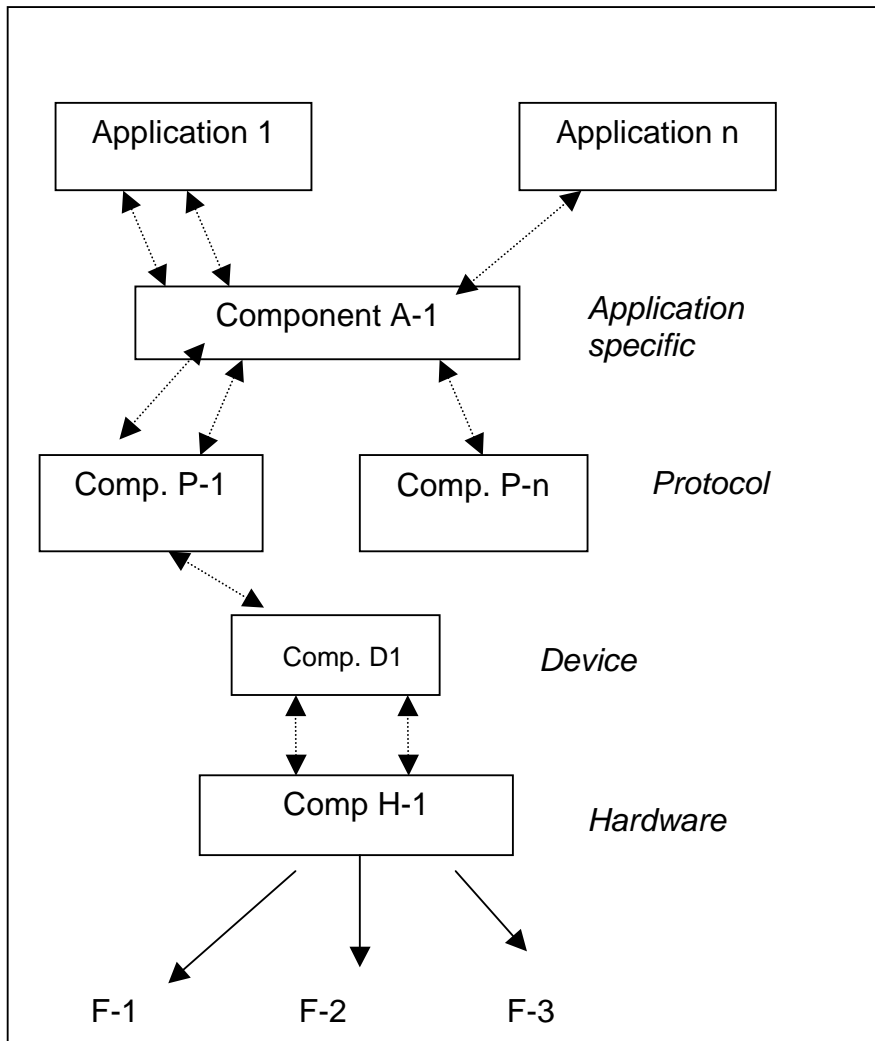
ITEA

The different verification checks are shown below:

- Requirements specification <==> Acceptance test specification
 - Design <==> Integration test specification
 - Code <==> Component test specification
-
- Requirements specification <==> Design
 - Design <==> Code
 - Component test specification <==> Integration test specification
 - Integration test specification <==> Acceptance test specification

For traceability, it is required that requirements are uniquely numbered and that this numbering remains unique during all phases of the development cycle.

2.7 Illustrations of items to cover concerning Requirements Management



2.7.1 Component reuse

When a component is reused for 100% in a different product or in the product family, then all related requirements, (requirements) constraints, quality status, development status, actual realised behaviour in terms of constraints (memory, performance) is reused and should be traceable in order to perform an adequate impact analysis.

2.7.2 Users of components

When a component is re-used in different products or in a product family, then changes to a component can influence the requirements (e.g. performance, ...) of all products or the whole product family.

2.7.3 Component information model

Especially when the requirements, including the constraints, are changed, this may have impact on aspects like the release plan of the component: schedule, its contents (which services are provided etc.), on which platform(s) or application(s) the component is validated or will be validated. The platform includes the system’s architecture and hardware (requires list).

2.7.4 Determining the quality status

Impact of requirements on :

- Actual performance in terms of resource usage
- Validation and testing metrics and reporting
- Test specifications and test-plan
- Test coverage (will show which provided services have been tested)

2.8 Tools integration

As seen from the requirements management perspective, the information needed for the management activity is contained in the several tools. These tools are listed hereafter, together with a short description of this information.

Tool	Related RM information	Format
Configuration management	Component label Development history	
Planning/tracking	Development costs Planning	Hours, \$
Validation and testing	Realised quality (of a system, component, requirement)	
Problem tracking	Problem reports / change requests Status of problem reports / change requests	
E-mail	Any correspondence on the requirements	
Design tool		

State of the art problems with tools integration

- The requirements management tool should provide links to this kind of information.
- The notation of the information is not specified
- The depth and automation of the linking of information is not provided, which may lead to too much manual labour and errors.