FOI-R--1576--SE
February 2005
ISSN 1650-1942
User report

Niklas Hallberg, Richard Andersson, Lars Westerdahl

Quality-Driven Process for Requirements Elicitation: The Case of Architecture Driving Requirements

Command and Control Systems
SE-581 11 Linköping

SWEDISH DEFENCE RESEARCH AGENCY
Command and Control Systems
Box 1165
581 11 Linköping

FOI-R--1576--SE
February 2005
ISSN 1650-1942
User report

Niklas Hallberg, Richard Andersson, Lars Westerdahl

Quality-Driven Process for Requirements Elicitation: The Case of Architecture Driving Requirements

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Command and Control Systems Box 1165 581 11 Linköping	FOI-R--1576--SE	User report
	Research area code 4 C4ISTAR	
	Month year February, 2005	**Project no.** E79065
	Sub area code 49 Interdisciplinary projects regarding C4ISTAR	
	Sub area code 2	

Author/s (editor/s)	Project manager
Niklas Hallberg Richard Andersson Lars Westerdahl	Arne Worm
	Approved by Martin Rantzer
	Sponsoring agency FMV
	Scientifically and technically responsible Niklas Hallberg

Report title

Quality-Driven Process for Requirements Elicitation: The Case of Architecture Driving Requirements

Abstract (not more than 200 words)

Requirements engineering is a complex activity with the purpose to specify *what* systems should accomplish rather than *how*. A difficulty in requirements engineering is to reveal stakeholders' needs and to express them as requirements. This document presents a quality driven process for requirements elicitation used for derivation of structured high level architecture driving requirements. The objective of the quality driven process for requirements elicitation is to ensure that the stakeholders' true needs are identified and that they are used as the foundation for the specification of requirements.

Keywords

Requirements engineering, quality-driven process, architecture driving requirements

Further bibliographic information	Language English
ISSN	**Pages** 12 p.
	Price acc. to pricelist

Utgivare	Rapportnummer, ISRN	Klassificering
Totalförsvarets Forskningsinstitut - FOI	FOI-R--1576--SE	Användarrapport
Ledningssystem	**Forskningsområde**	
Box 1165	4 Ledning, informationsteknik och sensorer	
581 11 Linköping	**Månad, år**	**Projektnummer**
	Februari, 2005	E79065
	Delområde	
	49 Breda projekt inom ledning, IT och sensorer	
	Delområde 2	

Författare/redaktör	Projektledare
Niklas Hallberg	Arne Worm
Richard Andersson	**Godkänd av**
Lars Westerdahl	Martin Rantzer
	Uppdragsgivare/kundbeteckning
	FMV
	Tekniskt och/eller vetenskapligt ansvarig
	Niklas Hallberg

Rapportens titel (i översättning)

Kvalitetsdriven kravhanterings process för kravinsamling: Fallet med arkiteturdrivande krav.

Sammanfattning (högst 200 ord)

Kravhantering är en komplex verksamhet som syftar till att specificera vad system skall åstadkomma snarare än hur. En svårighet i kravhantering är att identifiera intressenters behov och uttrycka dessa som krav. Detta dokument presenterar en kvalitetsdriven process för kravinsamling, som används för insamling och strukturering högnivå arkitekturdrivande krav. Avsikten med denna kvalitetsdrivna kravhanteringsprocess är att säkerställa att intressenternas verkliga behov identifieras och att de används som grund för specificeringen av krav.

Nyckelord

Kravhantering, kvalitetsdriven process, arkitekturdrivande krav

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN	**Antal sidor: 12** s.
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

1 Introduction

In Sweden a major reorganization of the Swedish Armed Forces (SwAF) is ongoing since the beginning of the early 1990's. The objective is an organization that is cost effective by being flexible and adoptable to the new tasks and challenges that the SwAF has to meet. To be able to perform command and control in this new organization a new C^2-system is required, in which a modern information technology will be an important component. However, to be able to develop such an extensive information system an architecture framework is required. This architecture framework has to be based on the goals and requirements that constitute the foundation for the C^2-system. To identify the requirements relevant for the development of the architecture framework a process for determining those requirements is needed. For this purpose a process for elicitation of architecture driving requirements was developed, which is called the Quality driven process for the requirements elicitation.

In this report, the quality driven process for the requirements elicitation used for derivation of architecture driving requirements is presented. To put the process in a context, a short introduction of requirements engineering is presented. Thereafter, the quality tools applied in the process and the quality driven process for requirements elicitation are presented.

2 Requirements Engineering

Requirements engineering refers to the activity in system development related to exploring needs and determining requirements. The aim is to specify *what* the systems should accomplish rather than *how* it is performed [1]. In the early era of requirements engineering, the focus was on the functional requirements and to formalize them in a requirements specification. Today, there are numerous approaches for requirements engineering, where also the non-functional requirements are given increased attention [2].

A difficulty in requirements engineering is how to reveal stakeholders' needs since they seldom are capable of expressing them in clear need formulations. Instead, they will provide descriptions of problematic situations that they have experienced, technical solutions that they believe can be useful et c. In quality approaches, such as *Total quality management* (TQM) and *Quality function deployment* (QFD) those descriptions are referred to as *the voice of the customers* [3]. In this approach, the concept of *statements* is used instead of *the voice of the customers*, since the term *customers* could be misleading. Statements are expressed by stakeholders or someone else as an observation, assumption or a remark of interest for the development of the future system. They can include descriptions about the stakeholders, the actual or wanted situation of use, experienced problems, business goals, and visions about the future system. A statement could be anything that can be used to reveal needs.

A need represents a lack of something required, desirable, or useful. It should be expressed from the point of view of the businesses, organizations and/or users that are the target of the development. Since those statements not only consist of needs, they must be *cleaned*, so that the actual needs are identified. When the actual needs have been determined, they can be used as a source for identification of requirements. Requirements are formulated as demands or restrictions on the future system. For instance, the system shall provide a specific functionality or the system shall have a specific feature.

Table 1: An example of how the VCT could be used.

Statement	Who	What	When	Where	Why	How	Need
Methods shall support and make possible that assignments are given in form of orders	The one in charge	Give order	All the time	Every-where	To get things done		***Support for giving orders***

Another difficulty in requirements engineering is related to the *requirement traceability problem*. There are several reasons for the necessity to trace requirements from their origin to design features and vice versa [4]. In larger information system development projects, several categories of stakeholders are affected by the design, and the different categories of stakeholders have different opinions of which functionality and features the system should have. Unfortunately, it is not possible to meet all specified requirements. Therefore, prioritization of requirements is necessary so that the most prominent functions and features will be implemented [5]. It exists several approaches for prioritization of requirements. However, they are not covered in this report.

3 The Quality Driven Process for Requirements Elicitation

The purpose of the quality driven process for requirements elicitation is to ensure that the actual needs of the stakeholders are identified and constitute the foundation for the determination of the requirements. Prerequisites for the process are access to relevant data that could be relied upon and participation of persons that are able to interpret this data. This section describes the tools and process of the Quality driven process for requirements elicitation as it was used for derivation and structure of high level architecture driving requirements.

3.1 Quality Tools

This section presents the tools included in the Quality driven process for requirements elicitation. The tools are; the Voice of the customer table (VCT), Affinity diagram, and Hierarchy diagram.

3.1.1 The Voice of the Customer Table (VCT)

The *Voice of the customer table* (VCT) (Table 1) is a tool for analysis of statements to reveal the actual needs [6]. There exist two different types of VCT, which could be used in sequence. The first is used to determine the needs, and the second for selection of solutions. In the process presented here, only the first VCT is used. The VCT can be used in participatory design projects to bridge the terminology gap between the user representatives and the professional developers of the design team. Although the work on the VCT can be experienced as time-consuming, the time is well spent when it comes to transforming the needs to requirements [3]. The VCT is a key tool for the qualitative analysis of the statements. The VCT constitutes an eight-column table; in the first column the captured statements are inserted, each for one row. The following columns describe an analysis of *who* asked for it, *what* do they want to do with it, *when* do they want to do it, *where* would they like to do it,

why would they do it, and *how* would they do it. Then the last column for the *need* can be filled in, hopefully with a correct understanding of what the statement really implied. In the Quality driven process for requirements elicitation proposed here, a minor modification is made, since an extra column for architecture driving requirements is included.

3.1.2 Affinity Diagram

It is common that a large number of entities (e.g. needs or requirements) are identified. However, to be manageable the number of entities must be limited. The entities can also be at different levels of abstraction, which also has to be handled. Duplication of entities, sometimes formulated differently, must be identified and removed. Therefore, the entities are scrutinized and categorized in *affinity diagrams* (Figure 1), which are created in a bottom-up procedure [7]. First, the entities are grouped in categories consisting of related entities. If necessary, an entity can be split into several sub-entities and duplicates are removed. The entities on a conceptually higher level can function as names for categories, thereafter the categories are grouped with similar categories.

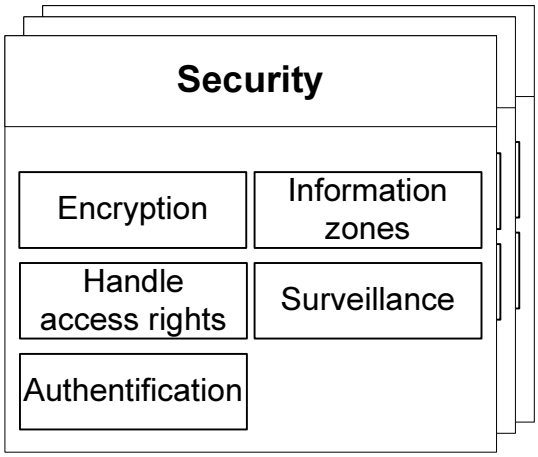

Figure 1: An example of an Affinity diagram.

3.1.3 Hierarchy Diagram

Hierarchy diagrams (also called tree-diagrams) is a notation used to display the entities and their relationships graphically (Figure 2). The notation provides an overview of the entities, categories of entities and the relationship between them, which is useful for review, restructuring, and final adjustments. The aim is to create a structure of mutually independent entities, where the entities on the same hierarchic level are at the same level of abstraction.

3.2 Process

This section presents the Quality-driven process for requirements elicitation. The prerequisites for the process are a number of data sources or data in which statements can be found that implicitly include the needs of a future system. Participation of stakeholder representatives that take active part in the work is useful, especially when it comes to the third and vital step in the process of determining the needs. To provide traceability between data sources, statements, needs and architecture driving requirements is essential. To visualize this, extra columns are inserted in the tables. The outcome presents the relationship between statements, needs and architecture driving requirements. The Quality driven process for requirements elicitation includes the following activities; (1) Data collection, (2) Identify

statements, (3) Determine needs, (4) Analysis of needs, (5) Determine architecture driving requirements, and (6) Analysis of architecture driving requirements.

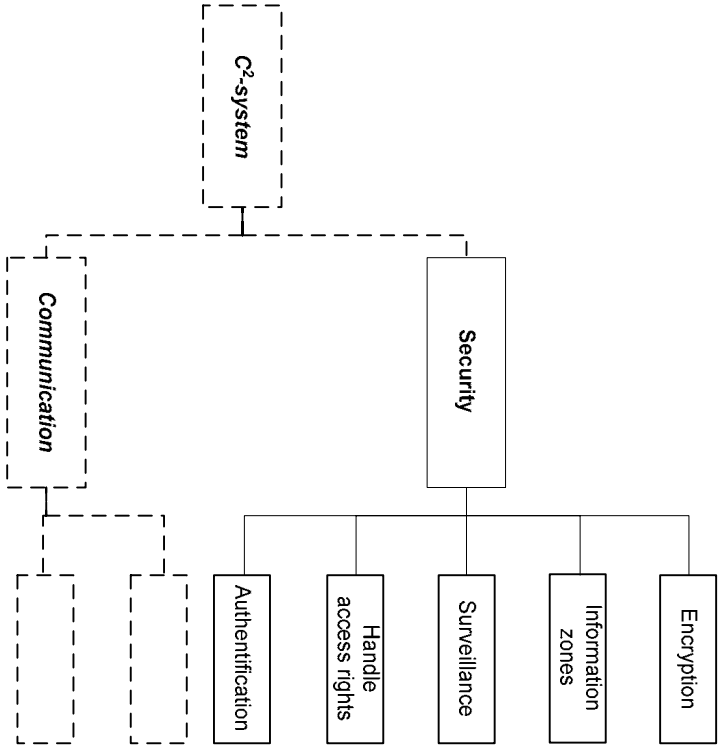

Figure 2: An example of a Hierarchy diagram.

3.2.1 Data collection

The objective of the data collection is to collect data about stakeholders, use cases, context of businesses, business goals, the future systems et c. For this purpose, several operational methods are possible, e.g. interviews, workshops, participatory observations, questionnaires, and document studies. Different data collections methods have different strengths and weaknesses. Hence, it is of advantage to use combinations of data methods. For instance, interviews provide rich information, which makes it possible to get a deeper understanding. Meanwhile, using questionnaires means that data could be acquired for a larger population. However, since the answers cannot be as detailed as during interviews it is of advantage to have participation of stakeholder representatives when interpreting the answers. Understanding the answers requires an understanding of the language and formulations of the population. The best approach is to use both qualitative data (e.g. acquired through in-depth interviews) and quantitative data (e.g. acquired through questionnaires), since together they provide both a wide and deep basis for determining the needs. The outcome of the first step is textual descriptions, e.g. about the stakeholders, the actual or wanted situation of use, and the future system.

3.2.2 Identify statements

The objective of the second step is to identify statements that contain information of interest for the specifying of the requirements. The identification of statements is performed by

careful reading through collected data, i.e., text documents, and identifying relevant statements. Performing this task, knowledge about the domain is of advantage for the selection of statements. The outcome of this step is a set of statements.

3.2.3 Determine needs

The objective of the third step is to determine a set of needs, based on the identified statements. For this purpose an analysis is performed based on the VCT. The use of VCT should be seen as support for this analysis, and all columns do not necessarily have to be filled in. The outcome of this step is a set of needs.

3.2.4 Analysis of needs

The objective of the fourth step is a structured set of mutually independent needs. The first part of this analysis is performed based on affinity diagrams, where the needs are categorized. The categories are then divided into more specified categories, until a coherent structure is achieved. During this analysis duplicates are removed and need formulations are improved. In the second part of the analysis, the structure of needs is viewed as a hierarchy diagram used to perform a final adjustment of categories and to find probable cases of missing (not explicitly expressed) needs. If a need is found to be missing, it is inserted into the structure. The structure should be arranged so that needs on each hierarchic level is at the same level of abstraction. The outcome of the fourth step is a hierarchic structure of needs.

3.2.5 Determine architecture driving requirements

The objective of the fifth step is to determine a set of architecture driving requirements, based on the identified needs. The architecture driving requirements are identified based on a detailed analysis of the needs. Several architecture driving requirements have already been identified during the analysis of statements with the VCT. Those architecture driving requirements are used as an inspiration to determine requirements based on the needs. The outcome of this step is a set of architecture driving requirements.

3.2.6 Analysis of architecture driving requirements

The objective of the sixth step is a structure of mutually independent architecture driving requirements. The first part of this analysis is based on affinity diagrams, in which the architecture driving requirements are categorized. The categories are restructured into more specific categories, until a coherent structure is achieved. During this activity, duplicates are removed and formulations are improved. In the second part of the analysis, the structure of architecture driving requirements is viewed as a hierarchy diagram to perform a final adjustment of categories and to find missing architecture driving requirements. If architecture driving requirements are suspected to be missing, they are formulated and inserted into the structure. The outcome of the sixth step is a structured set of architecture driving requirements.

3.3 Outcome

The outcome from the quality driven process for requirements elicitation are;

- The voice of the customer table
- The set of structured needs
- The set of structured architecture driving requirements

- Traceability between data source, statements, needs, and architecture driving requirements.

If the process is applied under given prerequisites the achieved architecture driving requirements will be of high quality.

4 References

[1] Siddiqi, J., & Shekaran, M. C. (1996). Requirements Engineering: The Emerging Wisdom. *IEEE Software*, 13, 15-18.

[2] Lauesen, S. (2000) Software requirements: styles and techniques. Addison-Wesley.

[3] Hallberg, N. (1999) *Incorporating User Values in the Design of Information Systems and Services in the Public Sector: A Methods Approach*, Linköping Studies in Science and Technology, Dissertation No. 596.

[4] Gotel, O. C. Z., & Finkelstein, A. C. W. (1994). An Analysis of the Requirements Traceability Problem. In *Proceedings of the First International Conference on Requirements Engineering* (pp. 94-101). Los Alamitos, CA: IEEE Computer Society Press.

[5] Karlsson, J., Wohlin, C., & Regnell, B. (1998). An Evaluation of Methods for Prioritizing Software Requirements. *Information and Software Technology*, 39, 939-947.

[6] Mazur, G. (1992). Voice of the Customer Table: A Tutorial. In *Transactions from the Fourth Symposium on Quality Function Deployment* (pp. 105-111). Novi, MI: GOAL/QPC.

[7] Holtzblatt, K., & Jones, S. (1993). Contextual Inquiry: A Participation Technique for System Development. In D. Schuler, & A. Namioka (Eds.), *Participatory Design: Principles and Practices* (pp. 177-210). Hillsdale, NJ: Lawrence Earlbaum.