– First published in the Proceedings of the 6th IBIMA Conference on Managing Information in the Digital Economy, June 19-21, 2006, Bonn, Germany –

Success and Failure Factors for Software Architecture

Niina Hämäläinen, University of Jyväskylä, Jyväskylä, Finland, niina.hamalainen@titu.jyu.fi
Jouni Markkula, University of Jyväskylä, Jyväskylä, Finland, jouni.markkula@titu.jyu.fi
Tanja Ylimäki, University of Jyväskylä, Jyväskylä, Finland, tanja.ylimaki@titu.jyu.fi
Markku Sakkinen, University of Jyväskylä, Jyväskylä, Finland, sakkinen@cs.jyu.fi

Abstract

This paper provides a view of the software architecture development and management process. It reviews the literature and practitioners' experiences relating to the factors that cause success and failure for software architecture and classifies these factors into subgroups. This study demonstrates that the success of software architecture depends on multiple factors. Project management, organisational culture and communication, the skills of architects and architectural know-how, architecture methods and practices, the quality of system requirements and, finally, architecture solutions seem to affect the achievement of successful architecture.

1. Introduction

Currently, a concern of many ICT-service providers and user organisations in their system development work is software architecture. Another central issue in this development work is the quality of the system. Software architecture is a critical factor in the design and construction of any complex software-intensive systems. Software architecture has an impact on the quality of the system. On one hand, a good architecture can help ensure that a system will satisfy key requirements in such areas as performance, reliability, portability, scalability, and interoperability [10]. On the other hand, a bad architecture can be disastrous. It may prevent the achievement of goals that are set for the system.

Architecture evaluation is a way to increase the understanding of the quality of architecture. A variety of methods is being developed for the evaluation of software architectures. Evaluation methods developed during the last decade are, for example, SAAM [15], ATAM [16], ARID [8] and ALMA [4]. Evaluation objectives, criteria, as well as evaluation targets, examined by the software architecture evaluation methods, differ markedly. Evaluation objectives and use cases are discussed in some method comparisons (e.g. [2, 9]) and other studies (e.g. [13]). In spite of this discussion in various papers, evaluation criteria and metrics are presently neither established nor detailed yet. Nevertheless several evaluation criteria and metrics descriptions exist. Software architecture evaluation criteria are discussed for example by Hilliard et al. [11, 12] and Losavio et al. [18, 19]. One reason for the non-establishment of architecture evaluation criteria and metrics may be that common views on what is successful software architecture and what factors have an effect on achieving it do not exist. It is not clear what targets and factors should be evaluated and measured. However, successful architecture is a widely used concept.

Academia and practitioners have come to realize that a critical success factor for system design and development is finding a successful architecture. Although the idea of a successful architecture is not clearly defined, practitioners and academia have become increasingly interested in what makes software architectures succeed or fail. The identified success and failure factors help system development managers and architects make a number of critical decisions. These decisions relate, for example, to the selection of evaluation criteria and metrics for the quality assessment of architectures and architecture management processes.

It is generally known that the success of software architecture is typically influenced by factors at various levels. However, these factors are mainly discussed only in a few studies and reports organised and produced by some research institutes and the ICT industry (e.g. [21], [1], [5]). Thus, these factors are, as yet, far from having been fully investigated in detail.

Our study contributes to this field with an identification and analysis of success and failure factors of software architecture. Our research involved reviewing the relevant literature and practitioners' experiences on factors that cause the success or failure of software architecture efforts. The factors listed in the following section were distilled from various articles and empirical research on software architecture implementation. Moreover, in order to collect empirical data for the present study, we organised an interview for a focus group of practitioners from three ICT service provider and user organisations. Success and failure factors were then categorised into a number of subgroups representing various dimensions of change related to the development and management of software architecture. As a result, this study presents a number of factors related to software architecture success and failure.

This study consists of the following sections. Firstly, section 2 presents the research method used in this study. Secondly, sections 3 and 4 present the results of this study: success and failure factors for software architecture. Finally, section 5 summarizes the findings and presents areas for further examination.

2. Research method

In order to identify and analyse the success and failure factors for software architecture a series of the following research phases was used in this study.

Phase 1. The study of previous research and reports.
Firstly, a list of success and failure factors mentioned in previous research and ICT-industry reports was produced. Secondly, the list of factors was analysed and the similar factors were organised into groups. Finally, the preliminary system development areas to which similar factors were related were identified.

Phase 2. Empirical research: A focus group interview [17] of practitioners.
A semi-structured group interview with a focus group of practitioners from three ICT user and service provider organisations was organised. The goal of the interview was to collect success and failure factors from the practitioners.

Interviewees
Practitioners were specialists of the management of software and enterprise architectures. The companies and interviewees are described in the table below.

Table1: Interviewees in the focus group interview

Companies	*Interviewees*
Architecture consultation company Number of personnel 10 (year 2005)	3 system and sofware architecure consultants
Banking, finance and insurance company Number of personnel 11 974 (year 2005)	enterprise architecture architect
Telecommunication company Number of personnel 4989 (year 2005)	enterprise architecture architect

The arrangements for the interview
The participants from these companies were interviewed as one group in order for group members to influence each other by responding to ideas and comments of others [17]. This group influence came up and new aspects were brought out. However, some aspects may not have been brought out by interviewees due to confidentiality reasons.

We presented previous research results in the interview and in turn structured the interview according to them. The practitioners reviewed the previous study results based on their own practical experiences. In addition they were asked to add new factors to the results on the basis of their practical experiences.

Data collection
The interview was tape-recorded and videotaped. Notes were written during the interview session. Based on this data a list of system development areas affecting the success of software architecture and success and failure factors relating to these areas was produced.

Phase 3. Consolidation and analysis of results.
The results from the empirical study and previous research were combined. These results are presented in chapters 3 and 4. In the results, the factors identified in the literature review are marked with the literature reference (proportion of these factors 49 %). The factors identified purely from the interview data are marked with the marking [FGI] and these factors are without literature reference (proportion of these factors 27 %). The factors recognized both from the interview data and from literature are marked with both the literature reference and [FGI] (proportion of these factors 24 %).

3. Software Architecture Success Factors

In this study, we identified six system development areas that seem to affect the success/failure of software architecture. These areas are presented in figure 1. The success and failure factors, identified in this study, relate to these areas. In the following sections, we describe the success factors included in these areas. The failure factors related to these areas are presented in chapter 4.

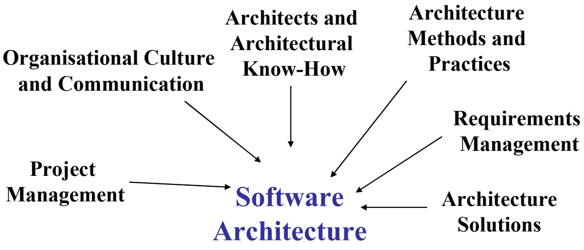
Organisational Culture and Communication
Architects and Architectural Know-How
Architecture Methods and Practices
Requirements Management
Project Management
Software Architecture
Architecture Solutions

Fig. 1. System development areas affecting the success and failure of software architecture.

Success Factors within Project Management
Project management offers time, staff and resources for architectural work. Software architecture success factors relating to the project management can be divided into factors relating to staffing, scheduling, planning and funding. In this study, we identified the following project management factors that promote the success of software architecture:

- *Clear aim of project*: The aim of the project is clear and reasonable [FGI = based on Focus Group Interview].

- *Strong management sponsorship*: The project and architecture work have strong management sponsorship [6]. Management offers time and funding for the project [FGI].
- *Clear milestones in the project*: Predetermined milestones are set in the planning stage to track the direction of the project [FGI].
- *Strong leadership:* Strong leadership specifically for the project [6].
- *Clearly defined teams and roles:* Project management teams are clearly defined. A good lead architect with a well-defined role and style [6].
- *Available knowledge / staff:* Market / business understanding is available [6].
- *Teamwork* [6].

Success Factors Related to the Organisational Culture and Communication

Organisational culture refers to the values, beliefs and customs of an organisation. Whereas organisational structure is relatively easy to draw and describe, organisational culture is less tangible. Organisational culture has an impact, for example, on how well the architecture will be adopted and followed. The success factors related to organisational culture are:

- *Status and role of architecture*: Architecture is woven into the organisational culture [6]. The role of the architecture and of the architectural descriptions is more instructive than supervisory [FGI].
- *Ownership:* Willingness to take ownership of architecture [6] [FGI].
- *Approving attitude towards architecture*: The project organisation is willing to follow architecture [6].
- *Training, teambuilding* [5]: The training of staff to design and manage architectures [FGI].

Successful communication between different groups can be seen as an effective exchange of information.

- *An effective and constructive communication culture relating to architectural issues:* Interpersonal and team communication [6]. The communication culture in an organisation is based on an open exchange of well-argued, even critical, opinions [FGI].

Success Factors Related to the Architects and Architectural Know-How

The personal skills of architects have an effect on the fluency of the architectural design process in collaboration with the stakeholders. Personal skills may also have an impact on architectural decision making. We identified the following skills of architects affecting the success of software architecture:

- *Practical experience:* Architects have practical experience on system development [21] or architects have the humility to discuss architectural solutions with the development team [FGI].
- *Domain knowledge*: Architects have at least a minimal knowledge on the problem domain [6, 21] [FGI].
- *System development knowledge:* Architects have knowledge on the system development method used and on how the architectural work is related to the method [FGI].
- *Capability to create architectural vision*: Architects have a capability to create a clear and compelling vision [6] that suits the organisation [FGI].
- *Conceptual thinking*: Architects are able to think conceptually and analytically [FGI].
- *Capability to argue rationally*: Architects are able to reason rationally, be critical of their own ideas, and put this rationality to use [FGI].
- *The ability to outline large entities* [FGI].
- *Communicative and social skills:* Architects can understand and combine views of the stakeholders [FGI]. Architects have communicative and social skills [21]. They are good communicators and listeners as well as good persuaders [6]. Moreover, they provides constructive feedback when it is needed [6]. They are also effective in selling and marketing architectural ideas [FGI]. These skills are important in spreading architectural knowledge, and explaining the urgency of architecture within an organization and a project team [21].
- *Project management skills*: Architects have good project management skills [6]. However, the project management skills needed depend on the scope of the project [FGI].
- *Humility:* The progress of architectural work is more important for the architect than personal merits [FGI].

Success Factors Related to the Architecture Methods and Practices

The software architecture management process contributes to the activities of capturing architectural requirements and understanding them, designing, analyzing/evaluating, realizing, maintaining, improving, and certifying the architecture as well as documenting it [3, 14]. The process model together with the methods and tools chosen to carry out architectural work, in turn have influence on this work. In addition, the standardization of the architectural concepts and of the descriptions in an organisation has an effect on the architectural practices. We identified the following factors relating to the architecture management process model, architectural methods and tools that affect the success of software architecture.

Architecture Management Process model:

- *Incremental and iterative development*: Deployed in phases / incrementally [6] [FGI].
- *Validation of requirements:* Validation of requirements during each step of the process [6].

- *The evaluation of architecture:* The evaluation of the architecture before it is implemented [FGI].
- *Life-cycle thinking in the architectural design*. The needs for change are taken into account in the architectural design [FGI].

Methods, tools and practices:

- *Suitable and effective methods and tools:* Architects should have effective tools at hand: methods that fit the specific requirements and situation of a company [21]. The methods should not constrain the architect in his work nor his creativity.
- *Well-defined limits for architects:* A well-defined field in which the architect is allowed to use his creativity in the architectural design and work [FGI].
- *Clear rules in the architectural decision making:* Clear rules on which architectural decisions can be made in the project and which decisions are made outside the project. Furthermore, clear definitions on which architectural decisions are made by architect and which are only prepared by him and which have to be decided by the project management. [FGI]
- *Change management* [FGI].

Standardization of architectural practices:

- *Standardization of architectural practices:* Standardisation architecture methods, descriptions, and terminology within the organisation [FGI].

Architectural specifications:

- *Clear and understandable architectural specifications*: Clear specifications including dependencies [6]. Architecture is understandable by all. That is, the architectural models and descriptions an architect produces, should be understandable and unambiguously interpretable by all stakeholders [6, 14]. Architectural models and descriptions are practical, easily translatable to the practice of software development and implementation. Otherwise the architecture will exclusively be used by the architects [21].

Enterprise architecture:

- *Defined and described enterprise architecture* [FGI]. Enterprise architecture is important in improving the adjustment of different projects to each other, and making sure information systems fit together, and into the entire architecture [21].

Success Factors Related to the Requirements Management

Architectural design and decision making is founded on identified requirements. Previous studies do not clearly highlight which factors in the requirements management advance the success of software architecture. However, the problems in requirements quality cause failure for software architecture like as described in the next chapter. Therefore, it is evident that the quality of the requirements and of the requirements management process advances the success of software architecture.

Three basic quality characteristics for the requirements of good quality are [20]:

- *Complete*
- *Agreed*: The requirements are correct, consistent, feasible, prioritized [FGI] and necessary.
- *Well-represented*. The requirements specifications are unambiguous, concise, traceable, non-redundant, organised [FGI], conformant to standards and verifiable.

Success Factors Related to the Architecture Solutions

Architectural choices and decisions are made in architectural design. Based on these decisions, the architectural specifications are produced. The following high-level success factors relating to architecture solutions are mentioned:

- *Simple architecture* [6]
- *Architecture solve the problem*: Solve at least the current [6] and impending [FGI] problems as well as change needs.

4. Software Architecture Failure Factors

The software architecture failure factors identified in this study are presented in this chapter.

Failure Factors related to the Project Management

Problems in staffing, scheduling, project planning and project funding complicate the architectural work. These kinds of problems are presented in the following section. In the interview of practitioners, we also noticed that some of these problems are more relevant for the service provider organisations than for the user organisations. For example, the lack of clear statement of the problem is more critical problem for the service providers than for the user organisations.

Problems and deficiencies in the project planning:

- *Not a clear statement of the problem*: The project lacks a clear problem statement or the project team has not provided a clear statement of the problem [1]. The organisation does not have time or willingness to define clearly the aim of the project [FGI = based on Focus Group Interview].
- *The project scope too broad*: The project scope is too broad [1]. The capability to divide the project into smaller entities/units may also be lacking [FGI].
- *No project, system or testing planning*: A project plan has not been put in place [1]. The project team has not written an overall architecture plan [1] and has not developed a system test plan [1].

No contingency plan has been provided [1]. No plan for moving to OO technology has been established. [1]

- *The lack of clear milestones in the project*: The direction of the project is not checked during the project. The only milestone is the end of the project [FGI].
- *No measures of success*: Measures of success have not been identified [1].

Problems in the scheduling:

- *No scheduling or unrealistic scheduling*: No project schedule is in place.[1] The deployment date is unrealistic [1] [FGI]. The focus is too much on getting positive results in the short term [21]. The project team has not put a hardware and installation schedule in place [1]. The project team has not allocated sufficient time for testing [1].

Problems in the project funding:

- *Funding not formalized*: Project funding has not been formalized [1].
- *Insufficient resources:* Insufficient resources have been allocated for building tasks. [1]

Problems and deficiencies in staffing:

- *Poor leadership*: No project manager/leader has been identified [1]. Poor leadership [6] Lack of control/authority [6].
- *Stakeholders unclear*: The stakeholders are not clearly identified [1] or they are difficult identify [7].
- *Lack of resources/talent:* The needed resource does not exist or project management is not able to offer it [FGI].
    - *Lack of domain expertise*: No domain experts have been committed to the project team [1].
    - *Lack of architect*: No architect exists [7] or failure to select software architects. Each layer has an architect assigned; however, a chief architect with responsibility for the overall architecture has not been selected [1].
    - *Lack of other resources*: For example the lack of points of view of end users or of administrator [FGI].
- *Lack of a quality assurance organisation*: A quality assurance organization has not been selected [1].
- *Lack of requirement team*: An independent requirement team has not been selected [1].

Failure Factors related to the Organisational Culture and Communication

The following aspects and factors relating to organisational culture and communication complicate architectural work:

- *Profit-centre and project culture:* Consideration of architectural issues only from the point of view of one's own profit centre or project [FGI]. Thinking too narrowly or short-sightedly [FGI].
- *Quarterly thinking:* Far-sighted architectural decisions are difficult to justify in the quarterly thinking [FGI].
- *"Turf" thinking:* Architectural decisions are formulated so that the decisions complicate the work of the decision maker as little as possible [FGI].
- *Organisational Politics:* Organisational politics drive the architectural decision making [6].
- *Negative Attitude towards Architecture and Architects:* The product team believes "we can solve it better ourselves" [6]. The designed architecture is not implemented. The product team implements its own ad hoc solutions [FGI].
- *Poor communication*: Poor communication inside/outside the architecture team [6]. The architecture team loses touch with the product team's problems [6].
- *Disparity in the perception of the architecture:* There are, for example differences in the perceptions between developers and architects [7].

4.3 Failure Factors related to the Architects and Architectural Know-How

Failure factors relating to the architects and architectural know-how are identified only briefly in previous research. However, the following factors are mentioned by previous studies and practitioners:

- *Unconvincing leadership by architects*: Architect or architecture team does not "sell" (lead) architecture enough [6].
- *Incapability to create an architectural vision* [6] [FGI].

Failure Factors related to the Architecture Methods and Practices

The following factors related to the architecture management complicate the architectural design.

Architecture management process, methods, tools and practices:

- *Attention focus on methods and tools, not on architecture*: Much time is spent on finding the best methods and modelling languages, which takes the attention away from the real purpose of architecture [21].
- *No architecture selection decision criteria*: The project lacks decision criteria to choose the software architecture [1].
- *No change management*: No modification (MR) tracking system in place [1] [FGI].
- *No iterative design*: The first version of the architectural design is implemented. The time is not used on architectural evaluations or on assessments of architectural alternatives [FGI].
- *The cutting down of the architectural design*: The time is focused on the coding rather than on the architectural design and evaluations [FGI].

- *Outputs not identified*: The expected outputs of the architectural work have not been identified [1] [FGI].
- *Outdated architectural documentation* [7].

Architectural specifications:

- *Essential architectural views / aspects not documented* [FGI].
- *Architectural descriptions are at too low a level or are not detailed enough* [6] [FGI]. Architectural specifications are class diagrams [7].
- *Architectural descriptions are at too high a level.* The architecture can not be carried out based on descriptions [FGI].

Enterprise architecture:

- *Enterprise architecture is not defined or described* [FGI].
- *Enterprise architecture is very heterogeneous* [FGI].

Failure Factors related to the Requirements Management

The following factors related to requirements quality complicate the architectural design and decision making:

- *Incomplete requirements:* Requirements are missing for a feature [1]. The existing environment (e.g. legacy systems) of system is not considered or described. An assessment of the size of the expected user community has not been done [1] Project lacks a clear statement of its data storage requirements. [1] Anticipated usage of the system was not clearly characterized. [1]
- *Unbalanced set of requirements* [7].
- *Requirements not prioritized:* The project team has not prioritized the requirements [1].
- *Requirements not documented*: No requirements documentation exists [1].
- *Requirements unclear:* Requirements not well-defined, not signed off, changing [6]. The team has not clarified some requirements. Requirements need to be clarified.[1]
- *Insufficient resources to support a new requirement have been allocated* [1].

Failure Factors related to the Architecture Solutions

The following factors relating to the architectural solutions are mentioned to be failure factors for the software architecture:

- *Architecture does not correspond to the requirements*: Does not solve the project teams problems [6]
- *Architectural decisions are based on the wrong interpretation of requirements:* The wrong interpretations of the regulations may lead, for example, to unnecessary complex architectural solutions [FGI].
- *Bad design / idea* [6].
- *Standards and standard components neglected* [7].
- *External structures drive the architecture*: Architecture follows customer's organizational structure [7]. Architecture depends on specifics of an operating system [7]. Architecture follows hardware design [7].
- *Exceptions drive architecture* [7].
- *Complex*: Too many components on every hierarchical level [7].

5. Conclusion

In this study, we identified and analysed success and failure factors for software architecture in system development work. This study demonstrates that the success of software architecture depends on multiple factors. Project management, organisational culture and communication, the skills of architects and architectural know-how, architecture methods and practices, the quality of system requirements and, finally, architecture solutions seem to affect the achievement of successful architecture.

Based on the analysis of the identified factors presented above, the main success factors and their relationship are presented in the figure 2.

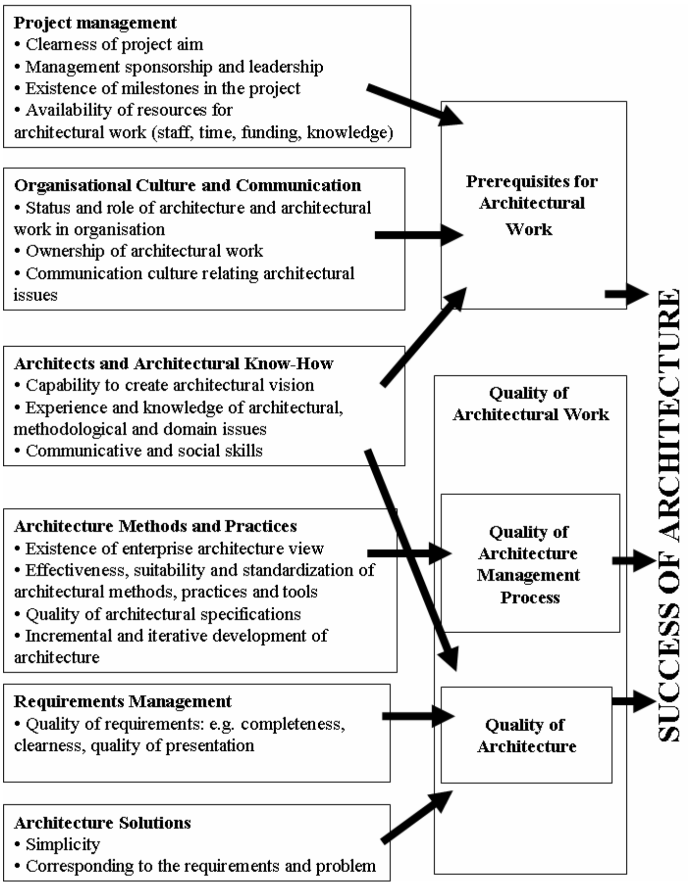
Project management
• Clearness of project aim
• Management sponsorship and leadership
• Existence of milestones in the project
• Availability of resources for architectural work (staff, time, funding, knowledge)
Organisational Culture and Communication
• Status and role of architecture and architectural work in organisation
• Ownership of architectural work
• Communication culture relating architectural issues
Architects and Architectural Know-How
• Capability to create architectural vision
• Experience and knowledge of architectural, methodological and domain issues
• Communicative and social skills
Architecture Methods and Practices
• Existence of enterprise architecture view
• Effectiveness, suitability and standardization of architectural methods, practices and tools
• Quality of architectural specifications
• Incremental and iterative development of architecture
Requirements Management
• Quality of requirements: e.g. completeness, clearness, quality of presentation
Architecture Solutions
• Simplicity
• Corresponding to the requirements and problem
Prerequisites for Architectural Work
Quality of Architectural Work
Quality of Architecture Management Process
Quality of Architecture
SUCCESS OF ARCHITECTURE

Fig 2. Main factors affecting the success of software architecture.

The results of this study can be used as a checklist by which practitioners in ICT service providers and user organisations undertaking, or planning to undertake, software architecture efforts can ensure that their software architecture–related efforts are comprehensive and well-implemented. These results can also help to decrease the chance of failure in architecture development.

A further outcome of this study is the development of software architecture quality management methods and process models, such as software architecture evaluation practices. This study shows for which targets architecture management evaluation criteria, metrics and methods could be developed and utilized.

Further research questions, raised in this study, include the question of which evaluation criteria and metrics are suitable for each success factor. In addition, the criticality of these software architecture success and failure factors in system development need to be assessed based on surveys directed to ICT service providers and user organisations. We are addressing this last question in our on-going research.

6. Acknowledgements

This paper is based on the work carried out in the AISA project (Quality Management of Enterprise and Software Architectures) financed by the the Finnish Funding Agency for Technology and Innovation (TEKES) and participating companies: Elisa Oyj, IBM Finland, OP Bank Group, SOK, and A-Ware Oy. We wish to thank the participating companies for their co-operation.

7. References

[1] Avritzer, A. and Weyuker, E. J. "Metrics to Assess the Likelihood of Project Success Based on Architecture Reviews," *Empirical Software Engineering* (4:3), September 1999, pp. 199 - 215.

[2] Babar, M. A., Zhu, L. and Jeffery, R. "A Framework for Classifying and Comparing Software Architecture Evaluation Methods," In Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), 2004.

[3] Bass, L., Clements, P. and Kazman, R. *Software Architecture in Practice*, Addison-Wesley, 1998.

[4] Bengtsson, P., Lassing, N., Bosch, J. and van Vliet, H. "Architecture-Level Modifiability Analysis," *Journal of Systems and Software* (69:1-2), 2004, pp. 129-147.

[5] Boehm, B. "Software architectures: critical success factors and cost drivers ", In Proceedings of The 16th International Conference on Software Engineering, Sorrento, Italy, 1994, p. 365.

[6] Bredemeyer Consulting. Software Architecting Success Factors and Pitfalls. Retrieved May 2006, http://www.bredemeyer.com/CSFs_pitfalls.htm

[7] Clements, P., Kazman, R. and Klein, M. *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley, 2002.

[8] Clements, P. C. "Active Reviews for Intermediate Designs" CMU/SEI-2000-TN-009, Software Engineering Institute (SEI), Carnegie Mellon University, 2000.

[9] Dobrica, L. and Niemelä, E. "A Survey on Software Architecture Analysis Methods," *IEEE Transactions on Software Engineering* (28:7), 2002, pp. 638-653.

[10] Garlan, D. "Software architecture: a roadmap ", In Proceedings of The Conference on The Future of Software Engineering, Limerick, Ireland, 2000, pp. 91-101.

[11] Hilliard, R., Kurland, M., J., Litvintchouk, S., D., Rice, T. and Schwarm, S. "Architecture Quality Assessment, version 2.0" The MITRE Corporation, 1996.

[12] Hilliard, R., Kurland, M. and Litvintchouk, S. "MITRE's Architecture Quality Assessment," In Proceedings of the Software Engineering & Economics Conference, 1997.

[13] Hämäläinen, N., Ahonen, J. and Kärkkäinen, T. "Why to Evaluate Enterprise and Software Architectures - Objectives and Use Cases", In Proceeding of the 12th European Conference on Information Technology Evaluation, Turku, Finland, 2005, pp. 213-222.

[14] IEEE "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems" IEEE Standard 1471-2000, 2000.

[15] Kazman, R., Bass, L., Abowd, G. and Webb, M. "SAAM: A Method for Analyzing the Properties of Software Architectures," In Proceedings of the 16th International Conference on Software Engineering, 1994.

[16] Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H. and Carriere, J. "The architecture tradeoff analysis method," In Proceedings of the Fourth IEEE International Conference on Engineering of Complex Computer Systems, ICECCS '98, Monterey, CA, 1998.

[17] Krueger, R. A. and Casey, M. A. *Focus Groups: A practical guide for applied research*, Sage Publications, Inc., 2000.

[18] Losavio, F., Chirinos, L., Lévy, N. and Ramdane-Cherif, A. "Quality Characteristics for Software Architecture," *Journal of Object Technology* (2:2), March-April 2003, pp. 133-150.

[19] Losavio, F., Chirinos, L., Matteo, A., Lévy, N. and Ramdane-Cherif, A. "ISO Quality Standards for Measuring Architectures," *The Journal of Systems and Software* (72), 2004, pp. 209-223.

[20] Pohl, K. "The three dimensions of requirements engineering: a framework and its applications," *Information Systems* (19:3), 1994, pp. 243 - 258.

[21] van der Raadt, B., Soetendal, J., Perdeck, M. and Vliet, H. v. "Polyphony in Architecture ", In Proceedings of the 26th International Conference on Software Engineering, 2004, pp. 533-542.