Success and Failure Factors for Software Architecture

AISA Project Report

Version: 1.0	Date: 11.1.2006
Author: Niina Hämäläinen	Status: Final

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.

Contents

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 (Garlan 2000). 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 (Kazman et al. 1994), ATAM (Kazman et al. 1998), ARID (Clements 2000) and ALMA (Bengtsson et al. 2004). 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. (Dobrica and Niemelä 2002; Babar et al. 2004)) and other studies (e.g.(Hämäläinen et al. 2005)). 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.

(Hilliard et al. 1996; Hilliard et al. 1997) and Losavio et al. (Losavio et al. 2003; Losavio et al. 2004). 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 by a few studies and reports organised and produced by some research institutes and the ICT industry (e.g. (van der Raadt et al. 2004), (Avritzer and Weyuker 1999), (Boehm

1994)). 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 summaries 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 (Krueger and Casey 2000) of practitioners.

A semi-structured group interview with a focus group of practitioners from three ICT user and service provider organisations was organised. Practitioners were specialists of the management of software and enterprise architectures. The goal of the interview was to collect success and failure factors from the practitioners. 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 basis of their practical experiences. 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 empirical study and previous research were combined. These results are presented in chapters 3 and 4.

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.

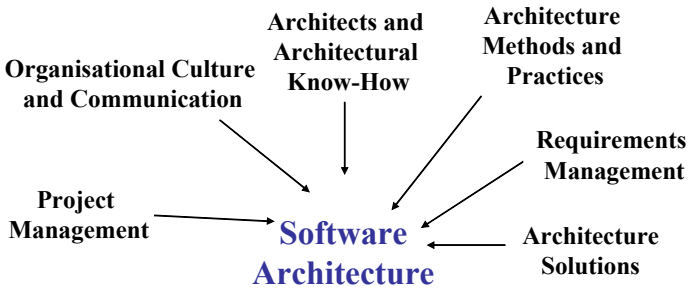

Figure 1. System development areas affecting the success and failure of software architecture.

3.1 Success and Failure 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 project management factors that promote the success of software architecture, which are displayed in Table 1.

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.

Table 1. Success and Failure Factors related to project management

Success Factor	**Adapted from**	**Failure Factor**
Clear aim of project The aim of the project is clear and reasonable.	[FGI = based on Focus Group Interview]	***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. The organisation does not have time or willingness to define clearly the aim of the project. [FGI = based on Focus Group Interview], (Avritzer and Weyuker 1999)

Success Factor	Adapted from	Failure Factor
Strong management sponsorship The project and architecture work have strong management sponsorship. Management offers time and funding for the project.	(Bredemeyer Consulting 2000), [FGI]	
Clear milestones in the project Predetermined milestones are set in the planning stage to track the direction of the project.	[FGI]	***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]
Strong leadership Strong leadership specifically for the project.	(Bredemeyer Consulting 2000)	***Poor leadership*** No project manager/leader has been identified (Avritzer and Weyuker 1999). Poor leadership (Bredemeyer Consulting 2000) Lack of control/authority (Bredemeyer Consulting 2000).
Clearly defined teams and roles Project management teams are clearly defined. A good lead architect with a well-defined role and style.	(Bredemeyer Consulting 2000)	

Success Factor	Adapted from	Failure Factor
Available knowledge / staff Market / business understanding is available.	(Bredemeyer Consulting 2000)	***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 (Avritzer and Weyuker 1999). *Lack of architect*: No architect exists (Clements et al. 2002) 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 (Avritzer and Weyuker 1999). *Lack of other resources*: For example the lack of points of view of end users or of administrator [FGI].
Teamwork	(Bredemeyer Consulting 2000)	
		The project scope too broad The project scope is too broad. The capability to divide the project into smaller entities/units may also be lacking. (Avritzer and Weyuker 1999), [FGI]

Success Factor	Adapted from	Failure Factor
		No project, system or testing planning A project plan has not been put in place. The project team has not written an overall architecture plan and has not developed a system test plan. No contingency plan has been provided. No plan for moving to OO technology has been established. (Avritzer and Weyuker 1999)
		Stakeholders unclear The stakeholders are not clearly identified (Avritzer and Weyuker 1999) or they are difficult identify (Clements et al. 2002).
		Lack of a quality assurance organization A quality assurance organization has not been selected (Avritzer and Weyuker 1999).
		Lack of requirement team An independent requirement team has not been selected (Avritzer and Weyuker 1999).
		Funding not formalized Project funding has not been formalized (Avritzer and Weyuker 1999)

Success Factor	Adapted from	Failure Factor
		Insufficient resources Insufficient resources have been allocated for building tasks. (Avritzer and Weyuker 1999)
		No measures of success Measures of success have not been identified. (Avritzer and Weyuker 1999)
		No scheduling or unrealistic scheduling No project schedule is in place.(Avritzer and Weyuker 1999) The deployment date is unrealistic (Avritzer and Weyuker 1999) [FGI]. The focus is too much on getting positive results in the short term (van der Raadt et al. 2004). The project team has not put a hardware and installation schedule in place (Avritzer and Weyuker 1999). The project team has not allocated sufficient time for testing (Avritzer and Weyuker 1999).

3.2 Success and Failure 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:

Table 2. Success Factors related to the organisational culture and communication

Success Factor	**Adapted from**	**Description**
Status and role of architecture	[17], [FGI]	Architecture is woven into the organisational culture. The role of the architecture and of the architectural descriptions is more instructive than supervisory.
Ownership	(Bredemeyer Consulting 2000), [FGI]	Willingness to take ownership of architecture
Approving attitude towards architecture	(Bredemeyer Consulting 2000)	The project organisation is willing to follow architecture
Training, teambuilding	(Boehm 1994)[FGI]	The training of staff to design and manage architectures.
An effective and constructive communication culture relating to architectural issues		Successful communication between different groups can be seen as an effective exchange of information. Interpersonal and team communication (Bredemeyer Consulting 2000). The communication culture in an organisation is based on an open exchange of well-argued, even critical, opinions [FGI].

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

Table 3. Failure Factors related to the organisational culture and communication

Failure Factor	**Adapted from**	**Description**
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 (Bredemeyer Consulting 2000)
Negative Attitude towards Architecture and Architects		The product team believes "we can solve it better ourselves" (Bredemeyer Consulting 2000). 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 (Bredemeyer Consulting 2000). The architecture team loses touch with the product team's problems (Bredemeyer Consulting 2000).

Failure Factor	Adapted from	Description
Disparity in the perception of the architecture		There are, for example differences in the perceptions between developers and architects (Clements et al. 2002).

3.3 Success and Failure 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:

Table 4. Success Factors related to the architects and architectural know-how

Success Factor	Adapted from	Description
Practical experience	(van der Raadt et al. 2004), [FGI]	Architects have practical experience on system development or architects have the humility to discuss architectural solutions with the development team.
Domain knowledge	(Bredemeyer Consulting 2000; van der Raadt et al. 2004) [FGI]	Architects have at least a minimal knowledge on the problem domain.
System development knowledge	[FGI]	Architects have knowledge on the system development method used and on how the architectural work is related to the method.
Capability to create architectural vision	(Bredemeyer Consulting 2000), [FGI]	Architects have a capability to create a clear and compelling vision that suits the organisation
Conceptual thinking	[FGI]	Architects are able to think conceptually and analytically.

Success Factor	Adapted from	Description
Capability to argue rationally	[FGI]	Architects are able to reason rationally, be critical of their own ideas, and put this rationality to use.
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 (van der Raadt et al. 2004). They are good communicators and listeners as well as good persuaders (Bredemeyer Consulting 2000). Moreover, they provides constructive feedback when it is needed (Bredemeyer Consulting 2000). 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 (van der Raadt et al. 2004)
Project management skills	(Bredemeyer Consulting 2000), [FGI]	Architects have good project management skills. However, the project management skills needed depend on the scope of the project.
Humility	[FGI]	The progress of architectural work is more important for the architect than personal merits.

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 (Bredemeyer Consulting 2000).

- *Incapability to create an architectural vision* (Bredemeyer Consulting 2000) [FGI].

3.4 Success and Failure 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 (Bass et al. 1998; IEEE 2000). 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 (Bredemeyer Consulting 2000) [FGI].

- *Validation of requirements:* Validation of requirements during each step of the process (Bredemeyer Consulting 2000).

- *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 (van der Raadt et al. 2004). 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 (Bredemeyer Consulting 2000) Architecture is understandable by all. That is, the architectural models and descriptions an architect produces, should be understandable and unambiguously interpretable by all stakeholders (Bredemeyer Consulting 2000; IEEE 2000). 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 (van der Raadt et al. 2004).

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 (van der Raadt et al. 2004).

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 (van der Raadt et al. 2004).

- *No architecture selection decision criteria*: The project lacks decision criteria to choose the software architecture (Avritzer and Weyuker 1999).

- *No change management*: No modification (MR) tracking system in place (Avritzer and Weyuker 1999) [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 (Avritzer and Weyuker 1999) [FGI].

- *Outdated architectural documentation* (Clements et al. 2002).

Architectural specifications:

- *Essential architectural views / aspects not documented* [FGI].

- *Architectural descriptions are at too low a level or are not detailed enough* (Bredemeyer Consulting 2000) [FGI]. Architectural specifications are class diagrams (Clements et al. 2002).

- *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].

- *A very heterogeneous enterprise architecture* [FGI].

3.5 Success and Failure 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 (Pohl 1994):

Table 5. Success and Failure Factors related to the requirements management.

Success factor	**Failure Factor**
Complete requirements	***Incomplete requirements*** Requirements are missing for a feature (Avritzer and Weyuker 1999). 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 (Avritzer and Weyuker 1999) Project lacks a clear statement of its data storage requirements. (Avritzer and Weyuker 1999) Anticipated usage of the system was not clearly characterized. (Avritzer and Weyuker 1999) *Unbalanced set of requirements* (Clements et al. 2002).
Agreed Requirements The requirements are correct, consistent, feasible, prioritized [FGI] and necessary.	***Requirements not prioritized*** The project team has not prioritized the requirements (Avritzer and Weyuker 1999).

Success factor	Failure Factor
Well-represented requirements The requirements specifications are unambiguous, concise, traceable, non-redundant, organised [FGI], conformant to standards and verifiable.	***Requirements unclear*** Requirements not well-defined, not signed off, changing (Bredemeyer Consulting 2000). The team has not clarified some requirements. Requirements need to be clarified.(Avritzer and Weyuker 1999)
	Requirements not documented No requirements documentation exists (Avritzer and Weyuker 1999).
	Insufficient resources to support a new requirement have been allocated (Avritzer and Weyuker 1999).

3.6 Success and Failure 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 and failure factors relating to architecture solutions are mentioned:

Table 6. Success and Failure Factors related to the architecture solutions.

Success Factor	Failure Factor
Simple architecture (Bredemeyer Consulting 2000)	***Complex*** Too many components on every hierarchical level (Clements et al. 2002).
Architecture solve the problem Solve at least the current (Bredemeyer Consulting 2000) and impending [FGI] problems as well as change needs.	***Architecture does not correspond to the requirements*** Does not solve the project teams problems (Bredemeyer Consulting 2000)

Success Factor	Failure Factor
	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 (Bredemeyer Consulting 2000).
	Standards and standard components neglected (Clements et al. 2002)
	External structures drive the architecture Architecture follows customer's organizational structure (Clements et al. 2002). Architecture depends on specifics of an operating system (Clements et al. 2002). Architecture follows hardware design (Clements et al. 2002).
	Exceptions drive architecture (Clements et al. 2002).

4 Discussion

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.

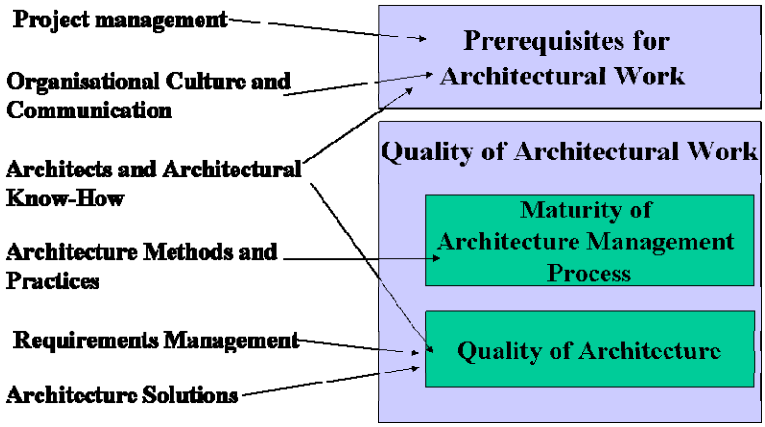

Figure 2. The main areas affecting architectural success.

Restrictions and limitations in this study

There are some limitations in this study. Corresponding success and failure factors were combined from different sources. Limited number of success and failure factors was considered in this study. However, the results give an image of the factors affecting architectural success.

5 Conclusion

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, well-implemented, and have the minimum chance of failure.

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.

References

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

Babar, M. A., L. Zhu and R. Jeffery (2004). A Framework for Classifying and Comparing Software Architecture Evaluation Methods. Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), IEEE Computer Society.

Bass, L., P. Clements and R. Kazman (1998). Software Architecture in Practice, Addison-Wesley.

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

Boehm, B. (1994). Software architectures: critical success factors and cost drivers. The 16th international conference on Software engineering, Sorrento, Italy, IEEE Computer Society Press.

Bredemeyer Consulting. (2000). "Software Architecting Success Factors and Pitfalls." from http://www.bredemeyer.com/CSFs_pitfalls.htm.

Clements, P., R. Kazman and M. Klein (2002). Evaluating Software Architectures: Methods and Case Studies, Addison-Wesley.

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

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

Garlan, D. (2000). Software architecture: a roadmap. The Conference on The Future of Software Engineering, Limerick, Ireland, ACM Press.

Hilliard, R., M. Kurland, J., S. Litvintchouk, D., T. Rice and S. Schwarm (1996). Architecture Quality Assessment, version 2.0, The MITRE Corporation.

Hilliard, R., M. J. Kurland and S. D. Litvintchouk (1997). MITRE's Architecture Quality Assessment. Proceedings of the Software Engineering & Economics Conference.

Hämäläinen, N., J. Ahonen and T. Kärkkäinen (2005). Why to Evaluate Enterprise and Software Architectures - Objectives and Use Cases. Proceeding of the 12th European Conference on Information Technology Evaluation, Turku.

IEEE (2000). IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. New York, USA, The Institute of Electrical and Electronics Engineers.

Kazman, R., L. Bass, G. Abowd and M. Webb (1994). SAAM: A Method for Analyzing the Properties of Software Architectures. Proceedings of the 16th International Conference on Software Engineering: 81-90.

Kazman, R., M. Klein, M. Barbacci, T. Longstaff, H. Lipson and J. Carriere (1998). The architecture tradeoff analysis method. Proceedings of the Fourth IEEE International Conference on Engineering of Complex Computer Systems, ICECCS '98. Monterey, CA, IEEE Computer Society: 68-78.

Krueger, R. A. and M. A. Casey (2000). Focus Groups: A practical guide for applied research, Sage Publications, Inc.

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

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

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

van der Raadt, B., J. Soetendal, M. Perdeck and H. v. Vliet (2004). Polyphony in Architecture. Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society.