Business Value of Solution Architecture

Raymond Slot[1], Guido Dedene[2], and Rik Maes[3]

[1] Capgemini, Utrecht, Netherlands
raymond.slot@capgemini.com
[2] University of Amsterdam, Netherlands
guido.dedene@econ.kuleuven.be
[3] University of Amsterdam, Netherlands
maestro@uva.nl

Abstract. The theory and especially the practice of IT architecture have been developed quite vigorously the last years. However, hardly any quantitative data about the value of IT architecture is available. This paper presents the results of a study, which measures the value of IT solution architecture for software development projects. The study identifies ten architecture-related project-variables and correlates these with eight project success variables. Statistical analysis of 49 IT projects shows that the use of solution architecture is correlated with decreased budget and time overrun, increased reliability of project planning and increased customer satisfaction. The results of the study indicate that IT usage of solution architecture for custom software development projects leads to better project results. Also the limitations of the study are discussed.

Keywords: Business Value, Solution Architecture, IT project success.

1 Solution Architecture

1.1 Introduction

The theory and especially the practice of IT architecture have been developed quite vigorously the last years. International and national standardization organizations, such as *The Open Group* [1] and in the Netherlands the *Telematica Instituut* [2] are working on standardization of business and IT architecture and the effects of these efforts are reaching the end users. Various IT organizations, such as Capgemini, have developed their own architecture framework and are using it in the market [3].

Considering the activities that take place in the business and IT architecture world, it is surprising that the business case for these activities is for a large part nonexistent. There is little research done to quantify, in financial terms, the value of architecture.

The main subject of this article is to quantify, in financial terms, the value of solution architecture for organizations. Organizations invest in solution architecture. These investments include training of architects, development of architectures and implementation of architecture processes. Is the spending of this money justified? Approaches to information economics [4] and [5]do not include the effects of investing in business and IT architecture. The key-question this article addresses is “How to quantify the value of IT solution architecture for an organization?”

E. Proper, F. Harmsen, and J.L.G. Dietz (Eds.): PRET 2009, LNBIP 28, pp. 84–108, 2009.

1.2 Required Disciplines for IT Projects

In the literature, project management, analysis & design and software development & testing, attract a lot of attention and many methods and approaches have been devised for these activities. For instance, for a discussion on project management see Kerzner [6] and PRINCE 2 [7]. Analysis & Design and Software Development & Testing are described in various development methods, among which RUP [8] and DSDM [9] are well known. In addition, CMMI [10] defines stages and maturity levels for (software) development processes.

1.3 Role of Solutions Architecture

None of these approaches recognizes explicitly the role of solution architecture, although the DSDM Consortium has published a paper on the relationship between TOGAF [1] and DSDM [11]. See for another initiative in this direction the Enterprise Unified Process [12].

According to Piselo [13], about one third of custom software development projects fail, about half of the projects is late, over budget, or has reduced functionality and only one sixth of the projects is delivered on time, within budget and according to specification. Considering the lack of attention of the major software development methodologies for architecture, one could assert that this is one of the reasons for this poor performance. In this paper, we will study the effects of solution architecture on projects. Concretely, we will test the hypothesis whether the success of software development projects is correlated with the usage of solution architecture.

1.4 Development under Architecture

Enterprise architecture sets standards and guidelines, based on strategy, for the structuring of the organization. The enterprise architecture is implemented by many projects, each implementing its own part of the total design. The approach where project objectives are also determined by enterprise architecture objectives is called development under architecture. Wagter [14] formulates this as follows: "Development under architecture realizes concrete business goals within the desired time frame, at the desired quality levels and at acceptable costs. [..] When a project is developed under architecture, the project starts with a so-called Project Start Architecture [(PSA)]. A [PSA] is a translation of the overarching [enterprise] architecture principles and models to rules and guidelines tailored to the project. This provides the practical rules, standards and guidelines used by the project. Also, project design choices that influence other projects are described in the [PSA]." Based on this, we define "development under architecture" as follows:

A project is developed under architecture if standards, rules and guidelines of the enterprise architecture are incorporated in the scope of the project, and these are tailored towards the objectives of the project, as described in a solution architecture document. Furthermore, the solution architecture describes how the software built by the project should interact with its environment.

1.5 Project Success

An IT project can be considered as a process with a number of inputs (called project variables) and outputs (success variables). The approach to measure project and success variables we have chosen is derived from Wohlin and Andrews. [15]. They state: "If $pv_1, pv_2 \ldots pv_i \ldots pv_k$, are the project variables and $sv_1, sv_2 \ldots sv_i \ldots sv_k$, are the success variables, then the objective is to identify which project variables are good estimators for which success variables. Project variables describe key drivers and characteristics of the software project and can be measured (or estimated) before the project starts [or during project execution]. Success variables are measured when the project is completed." Figure 1 illustrates this.

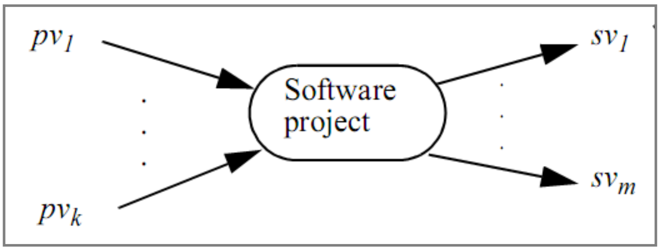

Fig. 1. Project and success variables (from Wohlin et al.)

Examples of project variables are: quality of the requirements, experience of the project manager, quality of the project architecture, etc. An example of a success variable is budget overrun.

Based on this model, we can define the effects of solution architecture on project success as follows:

Software development projects are more successful if we can identify solution-architecture related project variables, which are positively correlated with one or more success variables.

We selected ten architectural project variables and six success variables and we correlated each of the project variables with every success variable. For every significant correlation, the size of the effect will be calculated.

1.6 Project Variables

Jurgens [16] identified about 80 project variables or input variables to a project. He identified several categories of project variables: solution architecture-related variables, process-related variables, functional-requirement related variables, capability-related variables and IT-related variables. We selected the following architecture-related project variables:

Table 1. Overview of architecture-related project variables

Nr	Architecture-related Project Variable	Rationale for inclusion
1	Involvement of an architect in the technical budget calculation for the project	The technical budget is the budget for which the project should be built. Setting a technical prize includes considering various (technical) factors and price drivers, which is the expertise of the architect.
2	The general experience of the architect, who creates the project architecture, as reflected in the certification level of the architect.	Experienced architects have more insight oversight then less experienced architects.
3	The certification level of the architect, should match the complexity of the project.	Projects come in different sizes therefore and projects that are more complex should be linked to the more experienced architects.
4	The specific experience of the architect on the subject of the project.	It is probably advantageous for an architect to have experience with the specific topics of the project.
5	Quality of the solution architecture	The solution architecture is the guideline for the project.
6	Quality customer's domain architecture	The customer's domain architecture provides guidelines for the solution architecture.
7	Quality of the customers enterprise architecture	The customer's enterprise architecture is the guideline for the underlying domain and solution architecture.
8	Quality of the customers architecture governance process	A high-quality architecture governance process helps the project made the right decisions.
9	Presence of a controlling architect during the execution of the project	A controlling architect checks whether the project keeps itself to the solution architecture.
10	Compliancy testing between architecture and project during execution	If the compliancy is checked of project deliverables and the project as picture, then, any discrepancies between the two are known

1.7 Success Variables

For a selection of the success variables, we refer to Wideman [17]. He describes four dimensions of project success.

1. Internal Project Objectives (efficiency during the project)
2. Benefit to Customer (effectiveness in the short term)
3. Direct Contribution (in the medium term)
4. Future Opportunity (in the long term)

He gives the following key-questions and success variables for each of these dimensions.

Table 2. Dimensions, key questions and factors for project success according to Wideman

Dimension	Key-questions	Success variable
1. Internal	• How successful was the project team in meeting its schedule objectives? • How successful was the project team in meeting its budget objectives? • How successful was the project team in managing any other resource constraints?	• Meeting schedule • Within budget • Other resource constraints met

Table 2. (*Continued*)

Dimension	Key-questions	Success variable
2. Benefit to Customer	• Did the product meet its specified requirements of functional performance and technical standards? • What was the project's impact on the customer, and what did the customer gain? • Does the customer actually use the product, and are they satisfied with it? • Does the project's product fulfill the customer's needs, and/or solve the problem?	• Meeting functional performance • Meeting technical specifications & standards • Favorable impact on customer, customer's gain • Fulfilling customer's needs • Solving a customer's problem • Customer is using product • Customer expresses satisfaction
3. Direct Contribution	• Has the new or modified product become an immediate business and/or commercial success, has it enhanced immediate revenue and profits? • Has it created a larger market share?	• Immediate business and/or commercial success • Immediate revenue and profits enhanced • Larger market share generated
4. Future Opportunity	• Has the project created new opportunities for the future, has it contributed to positioning the organization consistent with its vision, goals? • Has it created a new market or new product potential, or assisted in developing a new technology? • Has it contributed additional capabilities or competencies to the organization?	• Will create new opportunities for future • Will position customer competitively • Will create new market • Will assist in developing new technology • Has, or will, add capabilities and competencies

Not all of the information for the success variables mentioned by Wideman where available for our survey. We were able to collect information about the following six success variables:

Table 3. Overview of success variables

No	Variable	Definition
A	**Budget**	Percentage under run or overrun for the project. We compare the actual project cost to the original project planning.
B	**Time**	Percentage under run or overrun for the project time. We compare the actual timeframe with the original, planned timeframe.
C	**Customer Satisfaction**	Customer's satisfaction assessment of project execution and result.
D	**Percentage Delivered**	The percentage of the intended results that are actually delivered by the project.
E	**Functional Fit**	The match between the required and delivered functionality; is the functionality delivered by the project in accordance with the planned functionality?
F	**Technical Fit**	The match between the required and delivered non-functional characteristics; is security, availability, performance, etc. of the delivered result is accordance with the planned characteristics?

2 Case Study Description

2.1 Objective and Approach

The objective of the case study is to test the hypothesis that software development projects are more successful when developed under architecture. This hypothesis is tested by correlating architecture-related project variables with project success variables. The hypothesis can be confirmed if we find significant correlations between architecture-related project variables and success variables.

2.2 Description of the Projects

Forty-nine projects were included in the study. These were all IT projects where software was developed, based on tailor-made specifications of customers. About half of the projects developed software for companies in the financial sector; the others are from industrial and governmental organizations. The types of projects are: transformation projects[1], merger and acquisition projects[2], single function integration projects[3] and lifetime extension projects[4].

2.2.1 Project Size

The project size was between € 50K and € 2,5M. The average project size is about € 700K, while the median size is about € 350K. See Figure 2 and Table 4 for an overview of the key figures of the projects within the survey scope.

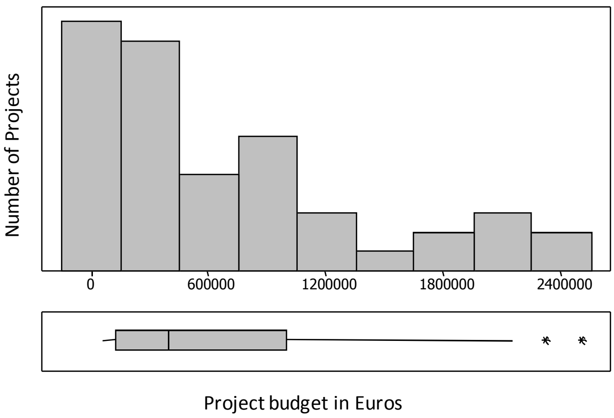

Fig. 2. Histogram and box plot of project size

Table 4. Key figures of the size of the surveyed projects

Characteristic	Value
Number of projects	49
Average project size	€ 695.000
Minimum project size	€ 50.800
First quartile	€ 125.000
Median	€ 349.000
Third quartile	€ 939.000
Maximum project size	€ 2.500.000

[1] Cross functional projects, such as CRM implementations.
[2] Rationalization or consolidation projects, shared service centre.
[3] New general ledger or HRM system.
[4] Web enabling of legacy, application integration.

2.3 The Null-Hypothesis

Based upon the definition of the success variables (see Table 3), the following null-hypothesis statements are formulated.

Table 5. Null-hypothesis statements

No	H_0 statement
I	Usage of solution architecture is not significantly correlated with the expected value of the actual project budget (as percentage of the planned budget).
II	Usage of solution architecture is not significantly correlated with the variance of the actual project budget (as percentage of the planned budget).
III	Usage of solution architecture is not significantly correlated with the expected value of the actual project timeframe (as percentage of the original timeframe).
IV	Usage of solution architecture is not significantly correlated with the variance of the actual project timeframe (as percentage of the original timeframe).
V	Usage of solution architecture is not significantly correlated with the expected value of customer satisfaction.
VI	Usage of solution architecture is not significantly correlated with the expected value of the percentage delivered.
VII	Usage of solution architecture is not significantly correlated with the expected value of the functional fit.
VIII	Usage of solution architecture is not significantly correlated with the expected value of the technical fit.

Remarks

1. Note that the null-hypothesis states that the project and success variables are *not* correlated. Rejection of the null hypothesis implies acceptance of the alternate hypothesis, i.e. that the variables are correlated.
2. The analysis identifies *correlations* between project and success variables and not *causal relationships*, because the type of statistical analysis that is used, is not able to identify causal relationships. However, by analyzing these correlations, we are often able to give meaning to the correlation and describe a causal mechanism that may be underlying the correlation.
3. For both the budget and time success variables, we have defined two H_0-hypothesis statements. The budget and time success variables are tested both with regard to the expected value (statements I and III) and the standard deviation (statements II and IV). The other success variables are tested only for expected value. The reason to test budget and time for standard deviation is that architecture may be correlated with an increase of reliability of the planned budget and time. A lower value of the standard deviation indicates less variance in the outcomes of the projects and thus a higher reliability of the planned figures.

2.4 Measurement Setup

The approach we have chosen to collect the required information is by means of interviews with the project manager. One of our main points of attention was to ensure optimal reliability of the basic information, which was collected using the interviews. The following procedure was observed, to minimize ambiguities and errors in the answering of the questions:

1. Carefully formulate the questions and answers, to make them as unambiguous as possible. We first set-up a test questionnaire, used this questionnaire several times and then define the final questionnaire based on the experiences of the test interviews.
2. Setting up guidelines how to interpret the questions, especially in situations where the answers were not clear-cut. The interviewers used experiences from earlier interviews in later interviews.
3. Analyze the answers for specific patterns or outliers, which could indicate for misinterpretations or ambiguities questions. Use these experiences to revise the instructions for the interviewers or/and to adapt the formulation of the question and answers.
4. The interviewers were trained in interpreting the answers to questions as univocally as possible, by organizing discussions between them about the interpretation of questions.
5. Check the answers of the interviewees where possible by independent means. Some answers could be checked by information from financial systems, others by crosschecking it with other people who work on the same project.

This procedure delivered 49 filled-in questionnaires.

3 Case Study Results

3.1 Summary of Results

The table shows the correlations for project and success variables that test significantly – the probability value (p-value) of the test is equal of smaller than the significance level, for a chosen significance level of 5%. P-values that are not significant are replaced by a dash.

From the table can be concluded that all H_0 statements are rejected, except for H_0-statement VII. Project variables 2 and 9 are not significantly correlated to a success variable; the other project variables are all significantly correlated with at least one of the success variables.

Table 6. Overview results null-hypothesis testing

			H_0 Statement							
			I	II	III	IV	V	VI	VII	VIII
Project Variables	1	Technical Calculation	-	0,2%	-	-	-	-	-	-
	2	Certification Architect	-	-	-	-	-	-	-	-
	3	Certification w.r.t. Project	-	-	-	-	0,0%	-	-	-
	4	Specific Experience Architect	-	-	-	-	5,0%	1,5%	-	-
	5	Project Architecture	-	2,4%	0,2%	-	0,8%	0,2%	-	0,3 %
	6	Domain Architecture	-	-	3,6%	-	1,9%	0,6%	-	-
	7	Enterprise Architecture	-	-	1,8%	3,5%	0,1%	2,6%	-	-
	8	Architecture Governance	0,3%	-	-	-	-	1,8%	-	-
	9	Controlling Architect	-	-	-	-	-	-	-	-
	10	Architecture Compliancy	-	-	1,0%	-	-	-	-	-

3.2 Interpretation of the Findings

In the following paragraphs, the findings are interpreted and explained. The interpretation follows a standard structure:

1. *Statement* – The formulation of the H_0 statement.
2. *Finding* – The actual findings from the analysis.
3. *Conclusion* – Conclusions that can be drawn from the findings.
4. *Significance* – Significance level. This level is equal or below the significant threshold of 5%.
5. *Interpretation* – Interpretation of the findings and the conclusion, which may provide additional reasoning or foundations for the conclusion.
6. *Consequences* – The size of the effect is explained, in terms of the effect on the success variable.

3.3 H_0 Statement I – Expected Value of Budget Overrun

3.3.1 H_0 Statement

Usage of solution architecture is not significantly correlated with the expected value of the actual project budget (as percentage of the planned budget).

Finding
Project variable 8 (*Quality of the customer's architecture governance process*) tests significantly. The other variables are non-significant. H_0 statement I is rejected.

Conclusion
The presence of an architecture governance process (either fully functional or limited in scope and responsibilities) is significantly correlated with a lower expected value of budget overrun, compared to a situation where there is no architecture governance process in the customer's organization present. The difference in expected value is 19%.

Significance
P = 0,3%

Interpretation
The presence of an architecture governance process implies that the organization is working with architecture and, therefore, is using solution architecture and higher-level architectures. The reverse situation is not necessarily the case; an organization may be defining solution architectures without having an architecture governance process. This finding shows that the presence of an architectural governance process has its own additional value.

Consequences
The average project size is € 700.000. A decrease the overrun with 19% will save on average € 130K per project, or about € 6M for the 49 projects that we have examined.

3.4 H_0 Statement II – Variance of Budget Overrun

H_0 Statement
Usage of solution architecture is not significantly correlated with the variance of the actual project budget (as percentage of the planned budget).

Finding
Project variable 1 (*Presence of architect during calculation of the technical price*) and project variable 5 (*Quality of the solution architecture*) tests significantly. The other variables are non-significant. H_0 statement II is rejected.

Conclusion
The presence of an architect during the calculation of the technical price is significantly correlated with a lower variance of the actual project budget, compared to a situation when there is no architect present during technical price calculation. The difference in the standard deviation of the project budget overrun percentage is 21 (13 versus 34).

The presence of a high-quality project architecture is significantly correlated with a lower variance of the actual project budget, compared to a situation when there is only a medium or poor quality or no project architecture present. The difference in the standard deviation is 18 (13 versus 31).

Significance
P = 0,8% (variable 1)
P = 2,4% (variable 5)

Interpretation
Presence of an architect during the calculation of the planned cost and the quality of the project architecture is correlated with an increase the reliability of the cost planning significantly. Reduction of variance is a major goal of the Six Sigma methodology [18]. When process variance is reduced, then the process becomes more predictable and overrun decreases. A major problem with custom software development is the lack of predictability of the actual cost. Both project variables *Presence of an architect during the calculation of the technical price* and *High-quality solution architecture* are correlated with a significantly improved reliability of the project budget planning.

Consequences
Piselo [13] states that only 16% of custom software development projects deliver their results according to plan. If the process variance is reduced, then this improves the process quality. For instance, we can calculate that only 13% of the projects with a high-quality solution architecture have more than 20% overrun, versus 38% of the projects with the medium or low quality solution architecture.

3.5 H_0 Statement III − Expected Value of Project Timeframe

H_0 Statement
Usage of solution architecture is not significantly correlated with the expected value of the actual project timeframe (as percentage of the original timeframe).

Finding
Project variables 5, 6, 7 and 10 (*Quality of the project architecture, Quality of the domain architecture, Quality of the enterprise architecture* and *Architecture compliancy testing*) test significantly. The other variables are non-significant. H_0 statement II is rejected.

Conclusion
Usage of solution architecture is correlated with a significant decrease in time overrun for projects. Four of the 10 project variables test significantly, which makes the project timeframe one of the success variables that correlate with multiple aspects of the use of architecture.

The presence of a high-quality project architecture correlates with a decrease in time overrun of the project, compared to a situation where there is a medium or poor quality project architecture present. The difference in overrun is 55% (71% overrun versus 16% overrun).

The presence of a high-quality domain architecture correlates with a decrease in time overrun of the project, compared with situation where there is medium or poor quality domain architecture present. The difference in overrun is 44% (49% versus 5% overrun).

The presence of a high-quality enterprise architecture correlates with a decrease in time overrun of the project, compared with situation where there is medium or poor quality enterprise architecture present. The difference in overrun is 46% (51% versus 5%).

The presence of an informal architecture compliance testing procedure correlates with a decrease in time overrun of the project, compared to the situation where there was no compliancy testing between architecture design and implementation. The difference in overrun is 56% (66% versus 10%).

Significance
P = 1,9% (variable 5)
P = 3,6% (variable 6)
P = 1,8% (variable 7)
P = 1,0% (variable 10)

Interpretation
It is interesting to note that four of the ten project variables correlate with the success variable. Probably, the same effect is measured multiple times, but from different angles. For instance, presence of enterprise architecture and the presence of the domain architecture denote probably the same type of architectural maturity of the customer's organization and both project variables may be an indication for a common underlying cause. Further indication of this is that variable 6 and variable 7 have almost the same expected values for time overrun. To understand this result more fully, it is necessary to analyze the interaction between project variables. However, the survey size is too limited to perform this type of analysis (see § *4.1*). Consequently, we have to limit ourselves to the supposition that interaction between project variables plays a major role in this result, without being able to quantify this interaction.

Overall, we can conclude that application of solution architecture is correlated with a substantial decrease in project time overrun.

Consequences
The average actual project timeframe for the projects that we have examined is one year – which includes on average 40% overrun. Consequently, application of architecture is correlated with a decrease of average project time of about four months.

3.6 H_0 Statement IV − Variance of Project Timeframe

H_0 Statement
Usage of solution architecture is not significantly correlated with the variance of the actual project timeframe (as percentage of the original timeframe).

Finding
Project variable 8 (*Quality of the enterprise architecture*) tests significantly. The other variables are non-significant. H_0 statement IV is rejected.

Conclusion
The presence of a high-quality enterprise architecture correlates significantly with a decrease of variance in the actual project timeframe, compared to a situation where there is medium or low quality enterprise architecture or no EA. The difference in the standard deviation of the percentage of the project timeframe overrun is 108 (115 versus 7).

Significance
P = 3,5%

Interpretation
The interpretation of this result is not very clear, because the difference in the standard deviation is quite large and the question is why we do not measure a correlation for domain and project architecture. Also, the sample size for one of the answers (answer 1 – *high-quality enterprise architecture*) is rather small – only 8 projects. The p-value for domain architecture is 11%, which could indicate a trend. However, the p-value for project architecture is 74%, which is nowhere significant. A possible explanation could be provided by the overall process maturity level of the organization. Higher process maturity may reflect itself in a high-quality enterprise architecture and this may influence the variance of project timeframe. We do not have information on process maturity of the organizations that are involved in the survey and, therefore, there is no way we can verify this theory. We suspect that this result could be spurious. Subsequent research may clarify this finding.

3.7 H_0 Statement V − Customer Satisfaction

H_0 Statement
Usage of solution architecture is not significantly correlated with the expected value of customer satisfaction.

Finding
Project variables 3, 4, 5, 6 and 7 (*Match of certification level of the architect to the level of the project*, *Specific experience of the architect*, *Quality of the project architecture, Quality of the domain architecture* and *Quality of the enterprise architecture*) test significantly. Project variables 2 and 8 (*Certification level of the architect* and *Quality of the customer's architecture governance process*) are close. H_0 statement V is rejected.

Conclusion
Usage of solution architecture is correlated with a significant increase in customer satisfaction. Five of the ten project variables test significantly, which makes customer satisfaction one of the success variables that correlates with multiple aspects of the use of architecture.

Matching the level of the architect with the level of the requirement correlates significantly with an increase of customer satisfaction, compared to a situation where the certification level of the architect was under project level. The difference is a customer satisfaction score of 4,1 versus 2,8 (on a scale of 1 to 5). This one project variable explains 51% of the total variance in the score.

Broad experience of the architect with the type of engagement correlates significantly with an increase of customer satisfaction, compared to a situation where the architect has only some experience with the type of engagement. The difference is a customer satisfaction score of 4,0 versus 3,6. This project variable explains 8,5% of the variance in the customer satisfaction score.

The presence of a high-quality project architecture correlates significantly with an increase of customer satisfaction, compared to a medium or low quality or no project architecture. The difference is a customer satisfaction score of 4,1 versus 3,5. This project variable explains 16,8% of the total variance of the customer satisfaction score.

The quality of the domain architecture correlates significantly with an increase of customer satisfaction. The customer satisfaction score is 4,2, 3,8 and 3,4 for respectively a high-quality, medium quality or low quality domain architecture. This project variable explains 12,5% of the total variance of the customer satisfaction score.

The quality of the enterprise architecture correlates significantly with an increase of customer satisfaction. The customer satisfaction score is 4,4, 3,9 and 3,4 for respectively a high-quality, medium quality or low quality enterprise architecture. This project variable explains 24,3% of the total variance of the customer satisfaction score.

Significance
P = 0,0% (variable 3)
P = 5,0% (variable 4)
P = 0,8% (variable 5)
P = 1,9% (variable 6)
P = 0,1% (variable 7)

Interpretation
Our supposition is that Customer Satisfaction score is the outcome of the comparison between the expectation of the customer and the actual outcomes of the project. If the outcome of the project is only mediocre, but customer expectation is low, then the

outcome of the project may still exceed customer expectation, and, therefore, customer satisfaction can be high. Customer satisfaction is the perceived discrepancy between expectation and realized results.

To understand the effect of perceived realized results to the customer, we analyzed the relationship between budget and time overrun with customer satisfaction. We find that budget overrun is not correlated with customer satisfaction.

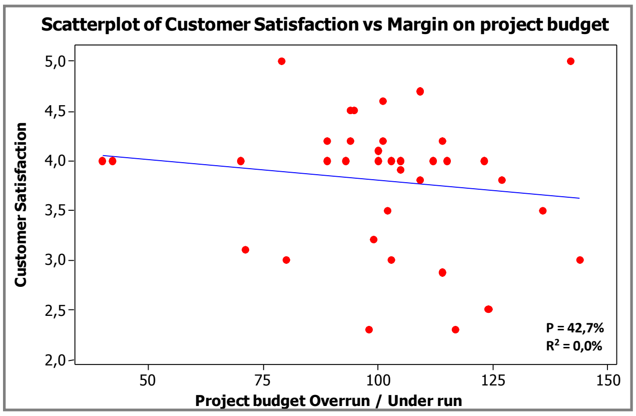

Fig. 3. Customer satisfaction as function of budget overrun

A p-value of 42,7% does not indicate a correlation. This lack of correlation can be explained, when we realize that budget overrun is not necessarily a problem for the customer. In the case of a fixed-price construction, the IT service provider is fully responsible for the budget overrun. In this situation budget overrun may be causing an increase of customer satisfaction, because the customer receives the required functionality, while the overrun costs are paid by the provider. Budget overrun can be correlated with both high or with low customer satisfaction, and is therefore not

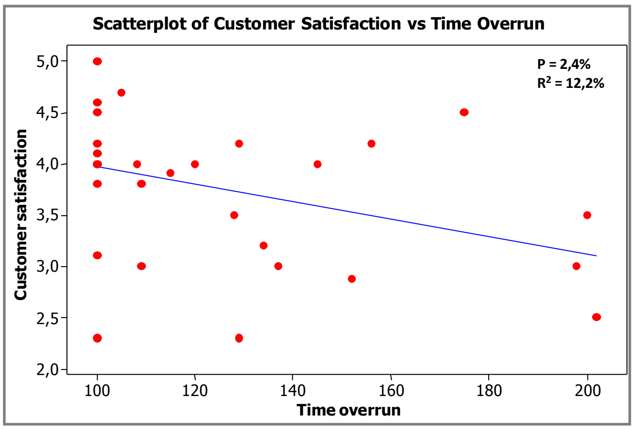

Fig. 4. Customer satisfaction as function of time overrun

related to the perceived value of the project for the customer. See Figure 4 for the analysis of time overrun and customer satisfaction.

We find that this correlation is significant and is described in the following formula:

$$Score = 4{,}84 - 0{,}0086 * Time\ overrun\ . \quad (1)$$

We find that increased time overrun is correlated with decreased customer satisfaction. Contrary to budget overrun, time overrun is always immediately experienced by the customer. When a project is delayed, then the customer is forced to adapt business schedules, the resource planning, interdependencies with other projects, etc.

A further indication that time overrun and customer satisfaction are correlated is also given by the correlations with the project variables 5, 6 and 7 (respectively quality of the project, domain and enterprise architectures) and customer satisfaction, because these three project variables also correlate with the expected value of the project timeframe (success variable 3).

The finding that time overrun and customer satisfaction are correlated, confirms the supposition at the beginning of this paragraph that customer satisfaction is related to the actual outcome of the project. Can we also find correlations that customer satisfaction is correlated to the expectation of the customer? Interestingly, project variable 3 (*Match of certification level of the architect to the level of the project)* does not correlate to budget or time overrun – but correlates with customer satisfaction. Our interpretation is that an architect whose level is matched with the level of the project, manages the expectations of the customer in such a way that it improves customer satisfaction, while less experienced architects does not have this competence.

Therefore, the correlation between customer satisfaction and both time overrun and certification level of the architect (project variable 3) supports our supposition and we can conclude that there are indications that customer satisfaction is influenced by the ability of the architect to manage expectations of the customer and by the time overrun of the project.

Consequences

Customer satisfaction is the result of a comparison of customer expectation and actual outcome of the project. The subjective elements of the customer satisfaction are co-determined by the experience of the architect. If the experience of the architect is lower than the level required by the project, then we find that this is correlated with lower customer satisfaction. The difference is 0,4 point, on a scale from 1 to 5. The objective elements of customer satisfaction are co-represented by the time overrun. The effect is a 0,2 point decrease in customer satisfaction for every 20% additional overrun.

3.8 H_0 Statement VI − Percentage Delivered

H_0 Statement

Usage of solution architecture is not significantly correlated with the expected value of the percentage delivered.

Finding

Project variables 4, 5, 6, 7 and 8 (*Specific experience of the architect*, *Quality of the project architecture, Quality of the domain architecture, Quality of the enterprise* *architecture* and *Quality of the customer's architecture governance process*) test significantly. H_0 statement VI is rejected.

Conclusion
Usage of solution architecture is correlated with a significant increase in percentage delivered. Five of the ten project variables test significantly, which makes percentage delivered one of the success variables that correlates with multiple aspects of the use of architecture.

Broad experience of the architect with the type of engagement correlates significantly with an increase of percentage delivered compared to a situation where the architect has only some experience with the type of engagement. The difference is 8% (92% versus 100%). This project variable explains 11,6 % of the variance in the percentage delivered.

An increase in the quality of the project architecture correlates significantly with an increase of percentage delivered. The difference is between low quality and high-quality project architecture is 12%. This project variable explains 16,9% of total variance of the percentage delivered.

An increase in the quality of domain architecture correlates significantly with an increase of percentage delivered. The difference between low quality and high-quality domain architecture is 13% (92% versus 105%). This project variable explains 13,8% of the total variance of the percentage delivered.

The increase in the quality of enterprise architecture correlates significantly with an increase of percentage delivered. The difference between low quality and high-quality enterprise architecture is 9% (49% versus 103%). This project variable explains 8,6% of the total variance of percentage delivered.

Improved architecture governance correlates significantly with an increase of percentage delivered. The difference between no governance and formal governance is 10% (94% versus 104%). This project variable explains 10,2% of the total variance of percentage delivered.

Significance
P = 1,5% (variable 4)
P = 0,2% (variable 5)
P = 0,6% (variable 6)
P = 2,6% (variable 7)
P = 1,8% (variable 8)

Interpretation
Five of the ten project variables correlate with the success variable percentage delivered. It can well be that the same underlying_effect is measured multiple times, but from different angles. For instance, the presence of enterprise architecture and the presence of the domain architecture may be linked by the architectural maturity of the customer's organization. To understand this result more fully, it is necessary to analyze the interaction between project variables (however, see § *4.1 Limitations of the Analysis*). We can conclude that application of enterprise architecture is correlated with a substantial increase in percentage delivered.

Consequences
Analyzing the differences in percentage delivered for the five project variables, we can conclude that usage of solution architecture is correlated with an increase of the percentage delivered of the project with approximately 10%.

3.9 H_0 Statement VII − Functional Fit

H_0 Statement
Usage of solution architecture is not significantly correlated with the expected value of the functional fit.

Finding
None of the variables tests significantly. H_0 Statement VII is not rejected.

Conclusion
The functional fit delivered by projects is not correlated with the use of solution architecture.

Interpretation
This result can be explained by considering the mechanisms of IT project development. It is the business that decides on the functionality of the project; i.e., business answers the *what* question. IT is responsible for building the solution; in other words, IT is responsible for the *how* question. It is therefore explicable that architecture is correlated with the quality of the transformation (as indicated by the other success variables), but not with delivered business functionality.

3.10 H_0 Statement VIII − Technical Fit

H_0 Statement
Usage of solution architecture is not significantly correlated with the expected value of the technical fit.

Finding
Project variable 5 (*Quality of the project architecture*) tests significantly. H_0 statement VIII is rejected.

Conclusion
An increase in the quality of the project architecture correlates significantly with an increase of technical fit.

Significance
P = 0,3%

Interpretation
This result is in line with the interpretation for statement VII. Architecture is correlated with the quality of the transformation, which includes the technical fit (performance, security, availability, etc.).

4 Conclusions and Recommendations

4.1 Limitations of the Analysis

4.1.1 The Role of Second-Order Effects

We found that multiple project variables may correlate with the same success variable. For example, H_0 statement III (Expected value of project timeframe) is correlated with the project variables 5, 6, 7 and 10 (*Quality of the project architecture*, *Quality of the domain architecture*, *Quality of the enterprise architecture* and *Architecture compliancy testing*). These variables are correlated with respectively 55%, 44%, 46% and 56% lower time overrun. Can we conclude from these figures that the project variable *Quality of the project architecture* (project variable 5) on its own is responsible for 55% decrease in time overrun? The answer is no, because there are multiple variables or combinations of variables responsible for the decrease in time overrun. See the example below, which illustrates the interaction between project variable 5 and variable 6, for H_0 statement IV -- *Variance of Project Timeframe*.

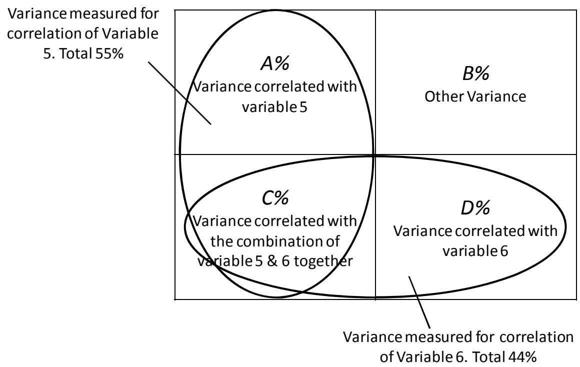

Fig. 5. Example of distribution of variance in project timeframe with two variables

Figure 5 shows a breakdown of the total variance in the success variable project timeframe for two project variables. The total variance is split out into four components: variance that can be explained by the combination of variable 5 and 6, variance that can be explained by variable 5, variance that can be explained by variable 6 and remaining variance that cannot be explained by either 5 or 6. From our measurements we know that A and C together is equal to 55% and that C and D together is equal to 44%, but we cannot determine the value of C, independent from A or D. In other words, we can determine the total variance for with variable 5 or the total variance for variable 6, but we cannot determine the combined effect of both variables, because this is equal to the measured variance of variable 5 (55%) plus the measured variance for variable 6 (44%) minus the combined effect (C%) which is unknown. In reality, we are not dealing with only two variables, but with multiple variables, and the number of second-order interactions between n variables increases quadratic with increasing n. (The number of second-order interactions between n variables is equal to

$n(n-1)/2$.) On top of this, there are third-order interactions, fourth-order interactions, etc.

4.1.2 Measuring Second-Order Effects

The questionnaire used to measure the project variables uses three-level multiple-choice answers. For example, for project variable 5 (the quality of the solution architecture) the possible answers are: high-quality, medium-quality and low-quality. With a survey size of 49 projects, the average sample size for one answer is about 16 (49 / 3) projects. This sample size is important because reliability of an answer depends on this size. To ensure a minimum level of reliability of the answer, a minimum sample size of 6 was chosen. When a sample size is 5 or smaller, then the sample is not used.

If testing a H_0 statement for second order effects, then we need to test it simultaneously for two project variables. In this case, the average sample size becomes $49 / 3^2 = 5{,}4$ projects. The minimum sample size is 6 projects, which means that the average sample size becomes less than the minimum sample size. In addition, variance in sample sizes means that some samples will be even smaller. For instance, a breakdown of the results of project variable 5 (Compliancy Testing) and project variable 10 (Project Architecture) gives the following results[5]:

Table 7. Sample sizes breakdown for project variables 5 and 10

		5. Compliancy Testing Answer			Total
		1	**2**	**3**	
10. Project Architecture Answer	**1**	3	13	8	**24**
	2	1	3	11	**15**
	3	0	0	6	**6**
Total		**4**	**16**	**25**	**45**

We find that only four of the nine samples sizes are equal or above the minimum sample size of 6. Five of the samples are less than six and cannot be used for testing.

The conclusion from this analysis is that we are not able to measure second order (or higher order) effects. Because of the limited size of the survey (49 projects), any tests for second-order effects are unreliable; the sample size is just too small. To test a H_0 statement simultaneously for two project variables – with the same average sample size of 16 – a survey size of 144 projects is needed.

4.1.3 Consequences

When drawing conclusions for a project variable, then we have to take into account that we are measuring not only a single variable, but we are measuring the effect of this variable combined with the interaction of this variable with other variables.

[5] The total number of projects is in this case is 45, because four respondents did not answer this question.

A consequence of this is that we cannot exactly determine which project variables are correlated with an effect. In the example of the previous paragraph, we cannot say that variable 5 is responsible for a 55% decrease of time overrun. We can only say that variable 5, in combination with the variables it interacts with, delivers a combined effect of 55%. However, we do not know the variables it interacts with and we do not know the size of this interaction.

In addition, we are not allowed to combine the results or draw conclusions from the combination of project variables. For example, in the above example we cannot say that a combination of project variable 5 and variable 6, delivers a specific result, or that variable 5 without variable 6 will deliver a different result. The project variables that we have measured are not independent from each other and influence each other in ways we are not able to determine.

However, we must also realize that measuring (only) first-order effects does not imply that these results are not reliable or are not real. The results are real and can be trusted; the limitation of the measurement is that we are not able to determine the exact, individual correlation of project variables with a success variable.

4.2 Results Summary

Usage of solution architecture within software development projects is correlated with the following effects:

Table 8. Overview of main results

(a)	19% decrease in project budget overrun
(b)	Increased predictability of project budget planning, which decreases the percentage of projects with large (> 20%) budget overruns from 38% to 13%
(c)	40% decrease in project time overrun
(d)	Increased customer satisfaction, with 0,5 to 1 point – on a scale of 1 to 5
(e)	10% increase of results delivered
(f)	Increased technical fit of the project results

These results demonstrate that usage of solution architecture is correlated with significant positive effects on software development projects. For instance, result (b) shows that solution architecture is correlated with a reduction of 25% of the percentage of projects with large overrun. This difference is substantial. This degree of improvement justifies the application of development of projects under architecture. With an average project size of € 700.000, this amounts to a saving of approximately € 140.000 for one out of four projects. Typical organizations have a dozen to several hundred IT projects running and on the average project portfolio, this will save millions Euros annually. The same considerations are true for the other results.

Of course, there is a cost associated with building up and maintaining the architecture processes and capability. These costs need to be balanced with the savings. Still, cost is only one of the aspects when taking the choice to implement an architecture function. There are other factors that are also positively influenced by architecture, which are not directly related to financial cost considerations, but are also important for the success of IT within an organization, such as increased customer satisfaction and decreased project time overrun.

It is interesting to note that the identified correlations are all positive: a 'better' value of a project variable correlates with a 'better' outcome of the success variable, for all identified significant correlations. This positive-positive trend gives an intuitive confirmation that the use of architecture is beneficial for projects; use of architecture does not counteract project objectives.

4.3 Other Research

4.3.1 Comparison to Standish Results

The Standish group [19,20]has published a top 10 of project success factors. The 2001 version of the report mentions the following main factors:

Table 9. Overview project success factors (Standish Group [20])

	Factor
1.	Executive support
2.	User involvement
3.	Experienced project manager
4.	Clear business objectives
5.	Minimized scope
6.	Standard software infrastructure
7.	Firm basic requirements
8.	Formal methodology
9.	Reliable estimates
10.	Other (Small milestones; Proper planning; Competent staff; Ownership)

Contrary to our findings, this list does not contain any design or architecture factors. An explanation for this could be that at the time of this research (1995-2000), architecture was not widely used or known. The value of architecture was not a topic for IT executives, project managers or project staff and was apparently not identified by the Standish researchers. We feel that architecture should be on this list, because our research shows that architecture is a major project success factor.

Other researchers do value the constructive role of enterprise architecture. For example, the National Research Council states in a review on FBI's Trilogy Information Technology Modernization Program, "if the FBI's IT modernization program is to succeed, the FBI's top leadership [..] must make the creation and communication of a complete enterprise architecture a top priority."[21]. This statement acknowledges the value of enterprise architecture for system development initiatives.

One of the other conclusions from the original Standish Chaos report [19] is that the success rate and the size of the project are linked. The lower the project cost, the higher the success rate. They provide the following figures:

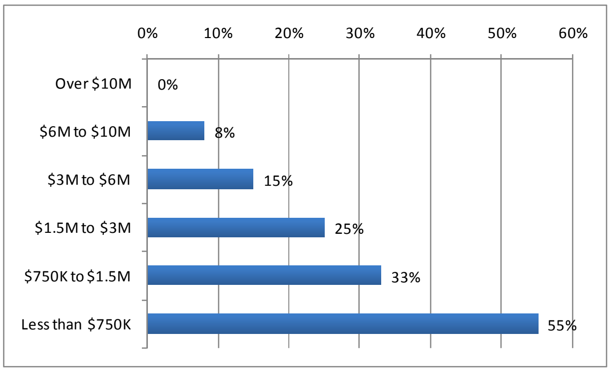

Fig. 6. Project success rates (Standish Chaos Report [19])

A correlation of the size of the project with budget overrun provides the following result:

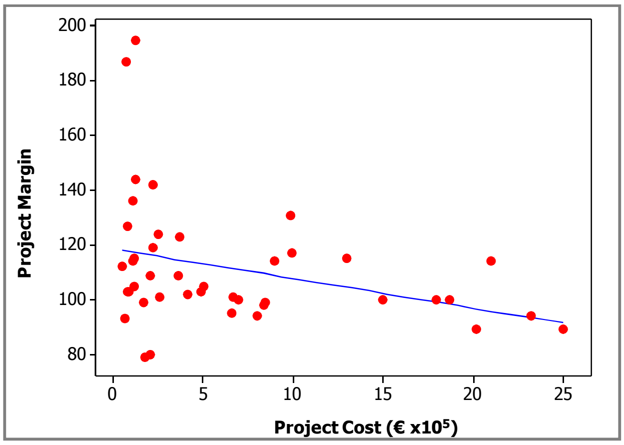

Fig. 7. Correlation between project cost and margin

The vertical axis describes the percentagewise budget overrun or under run. The horizontal axis describes the total cost of the IT project (x 10^5) in Euro's.

Table 10. Key figures for the linear correlation between project cost and margin

Figure	Value
p	0,035
r^2	8,4%
Corre-lation formula	*Project Margin* = 118,6 – 1,1.10^{-5} * *Project Cost*

The correlation that we find between project cost and project margin is that projects become *more* successful with increasing project size. This is a contradiction to the findings of the Standish report, because they find their projects become *less* successful with increasing project size. The figures are not exactly comparable, because Standish defines project success as a combination of on time, on budget and with sufficient functionality. Our correlation only considers cost overrun. Still, the trend is clearly contradictory.

In discussion with project and risk managers about the reason for our finding, the following explanations are given:

1. For small projects, the initial planning effort in determining the project cost is much smaller than for large projects. Therefore, the project cost estimations for small projects are less reliable and the complexity of the project may be underestimated.
2. For small projects, it is very difficult to overcome a project setback within the existing budget. If a small, two-month project has a setback which delays the project for one month, than the budget overrun in absolute terms may be small, but percentagewise the overrun is 50%. For large projects, this type of small setbacks can be absorbed within the existing project budget and the risk margins.
3. For large projects, you have the time to rethink (part of) the solution and learn from lessons earlier in the project. For small projects, if you are halfway through the project and then find out that the original solution needs adjustment, there is no time or budget to redesign.

These arguments provide an explanation for finding that increasing project size correlates with higher project success.

When trying to explain the discrepancy between our results and the findings from the Standish report, then we must realize that our maximum project size is € 2,5M, while the Standish report examines projects with maximum size over $ 10M. This variation in maximum size may explain the difference. The arguments above describe the reasons why small projects (< € 1M) have high overruns; these arguments are not relevant for projects above € 2,5M. It is possible that the decreasing trend reverts to an increasing trend for larger projects.

4.4 Applicability and Conclusions

4.4.1 Applicability of the Results

The study that we conducted was carried out in a rather uniform context. Individual projects are relatively small (less than € 2,5M) and all projects were executed in the context of a commercial IT service provider. Furthermore, the number of projects within the scope of the study is a rather limited. This type of research benefits from a large survey base. Several hundred projects would be better; several thousand projects would still be better. The question is then whether the type of effects and the direction of the effects that we have measured (architecture lowers budget overrun, lowers time overrun, increases the percentage delivered, etc.) are valid in the general situation. Based on this one study, we cannot provide definitive statements on this.

Nevertheless, one of the major findings of the analysis of § 4.2 is that: “a ‘better’ value of a project variable correlates with a ‘better’ outcome of the success variable, for all identified significant correlations. This positive-positive trend gives an intuitive confirmation that the use of architecture is beneficial for projects; use of architecture does not counteract project objectives." None of the identified correlations between architecture project variables and success variables counteracts project success; all correlations are in the same positive direction.

4.4.2 Overall Conclusion

We did not identify correlations between project variables and success variables with a negative effect on the success variable. To the contrary, we identified multiple positive correlations. We therefore conclude that the use of solution architecture is correlated with a substantial improvement of several key success variables. Based on this finding, our main conclusion is that we can confirm our initial hypothesis that for custom software development projects with a maximum cost of € 2,5M, the use of solution architecture is correlated with improved project results.

Acknowledgements

A survey like this cannot be conducted without the cooperation of many people. We would like to thank everyone who contributed to this research. Our gratitude goes especially to the following persons: André Weber, Bernard Hüdepol, Edgar Giessen, Edwin Kok, Frank Harmsen, Ger Donners, Goossen Foppen, Hans Schevers, Jack van Eijk, Jan Joosen, Jos Smit, Jouke Hopperus Buma, Kaeso de Jager, Maickel Sweekhorst, Marco Folpmers, Mark Grimberg, Mathieu Hagen, Max Stahlecker, Nico Toussaint, Ron Tolido, Rudolf Jurgens, Tom Vanderwiele, Ton Hardeman, Tonny Wildvank, Victor van Swede, Wil van Hamersveld, Wouter Schmitz and the project managers who took the time to fill in the questionnaire.

References

1. TOGAF: The Open Group Architecture Framework, Version 8.1 Enterprise Edition. TOGAF (2004)
2. Lankhorst, et al.: Marc: Enterprise Architecture at Work, Modeling, Communication and Analysis. Telematica Instituut, Enschede (2005)
3. Rijsenbrij, D., Goedvolk, H.: Integrated Architecture Framework. White Paper (1999)
4. Parker, M., Trainor, E., Benson, R.: Information Strategy and Economics. Prentice-Hall, Englewood Cliffs (1988)
5. Oirsouw, R., Spanderman, J., Vries, H.: Informatie Economie, Investeringsstrategie voor de Informatievoorziening (1993)
6. Kerzner, H.: Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley, Chichester (2003)
7. Office of Government Commerce: PRINCE2. Office of Government Commerce (accessed 2008), http://www.ogc.gov.uk/methods_prince_2.asp
8. Barnes, J.: Implementing the IBM Rational Unified Process and Solutions: A Guide to Improving Your Software Development Capability and Maturity. IBM Press (2007)
9. DSDM: Enabling Business Agility. In: DSDM Consortium, http://www.dsdm.org/10 (accessed, August 2008)

10. SEI: CMMI for Development Version 1.2. Carnegie Mellon University - Software Engineering Institute (2006)
11. DSDM and TOGAF: DSDM and TOGAF Joint White Paper (2003)
12. Ambler, S.: Enterprise Unified Process (EUP) (accessed, January 2009)
13. Piselo, T., Strassman, P.: IT Value Chain Management - Maximising the ROI from IT investments. Standish Report (2003)
14. Wagter, R., Berg, M., Luijpers, J., Steenbergen, M.: Dynamic Enterprise Architecture: How to Make It Work. Wiley, Chichester (2005)
15. Wohlin, C., Andrews, A.: Prioritizing and Assessing Software Project Success Factors and Project Characteristics using Subjective Data. Empirical Software Engineering: An International Journal 8(3), 285–308 (2003)
16. Jurgens, R.: The Added Value of Enterprise Architecture, M Sc Thesis., University Delft (2008)
17. Wideman, M.: Improving PM: Linking Success Criteria to Project Type, `http://www.maxwideman.com` (accessed July 02, 2008), `http://www.maxwideman.com/papers/improvingpm/dimensions.htm`
18. Pyzdek, T.: The Six Sigma Handbook, Revised and Expanded. McGraw-Hill, New York (2003)
19. Standish Group International: CHAOS: A recipe for success (1999)
20. Standish Group International: Extreme Chaos (2001)
21. McGroddy, J., Lin, H. (eds.): A Review of the FBI's Trilogy Information Technology Modernization Program, Washington DC (2004)
22. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley, Reading (2003)
23. IEEE: The IEEE 1471-2000 standard - Architecture Views and Viewpoints (2000)
24. Maes, R., Rijsenbrij, D., Truijens, O.: Reconsidering Information Management Through a Generic Framework, Amsterdam (1999)
25. Martin, J., Leben, J.: Strategic Information Planning Methodologies. Prentice-Hall, Englewood Cliffs (1989)
26. Capgemini: Architecture and the Integrated Architecture Framework, `http://www.capgemini.com` (accessed, December 2007), `http://www.capgemini.cm/resources/thought_leadership/architecture_and_the_integrated_architedcture_framework`
27. Kruskal, W., Wallis, W.: Use of ranks in one-criterion variance analysis. Journal of the Americal Statistical Society(45) (1952)
28. Levene, H.: In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling. Stanford University Press (1960)
29. Lindman, H.: Analysis of variance in complex experimental designs. W. H. Freeman & Co., San Francisco (1974)
30. Zachman, J.: A Framework for Information Architecture. IBM Systems Journal (1987)
31. Montgomery, D.: Introduction to Statistical Quality Control. John Wiley and Sons, Chichester (2005)
32. Moore, M., Kazman, R., Klein, M., Asundi, J.: Quantifying the Value of Architecture Design Decisions: Lessons from the Field. Software Engineering (2003)