- Business case: the role of the IT Architect -

Master Thesis
Software Engineering
University of Amsterdam

Date:	25 June 2004
Student:	H. Dekkers
Host organization:	Software Architecture group VU
Mentor VU:	Henk Koning
Mentor UVA:	Paul Klint
Version:	1.0
Status:	Final

SUMMARY

Studies show that many ICT projects fail to meet schedule and budget. A sizeable percentage of ICT projects runaway and don't complete at all. It is unclear to what extent initial goals are accomplished and if benefits exceed investments and operational costs. In my career I participated in a number of IT projects that can be considered as failures. Human aspects, rather than technical issues, were the main reasons for failure.

The master project is a literature survey. The main hypothesis is that a software architect can contribute to the success of a project by developing a vision and determining the feasibility before the start of a project, notably the business case. The literature has confirmed this hypothesis and enriched the notions of vision, feasibility and the role of the IT architect.

The hypothesis was derived after thorough analysis of three cases. This analysis was required to scope the research question to a root cause for project failure. This was found to be lack of vision and insufficient understanding of the risks involved. To sharply define the research questions, the assumptions about the role of the IT architect were captured in a list of responsibilities and in two viewpoints. The business case was identified as the object containing the vision and feasibility study. This led to the following research questions:

1. Does literature confirm that a business case for ICT projects should include a vision and feasibility study?
2. Does literature confirm the distinguished responsibilities of an IT architect in making a business case?
3. Does literature confirm the proposed viewpoints with respect to vision and feasibility?

The literature survey confirmed the importance of a business case for IT projects. The business case will prevent doomed projects from being carried forward. It will provide a framework for the successful implementation of the project by providing clear goals. It will also help mitigate risks by clarifying required adaptations to organization, project and system.

Literature also confirms the importance of vision. Vision portrays the value of the new system, captures the most important constraints and provides a solution respecting these constraints. One of the main contributions of vision is that it brings people together, provides direction for individuals, provides clear motivation and coordinates different activities. That's why a vision should be clear, and compelling.

There is much agreement in literature that the feasibility of a project needs to be considered when taking a go / no go decision. However there are many different views on how to determine feasibility. An intensive search did not come up with a suitable feasibility framework. Most frameworks were either too fine grained or too coarse. Analyzing the different frameworks, basically two questions came forward that need to be answered:

1. Is the organization capable of executing the project as proposed
2. Is the organization capable of adopting the resulting system

In both questions it is important to look at the whole, rather than considering aspects in isolation. A project simple for one organization might proof impossible for other organizations. This shows that organizational abilities have to be matched with project characteristics. These organizational abilities are defined by maturity of processes, competence of resources and by the culture of the company. Defining a new framework was not within the scope of this thesis, however the different frameworks and risk concepts were analyzed and these two questions were refined.

Finally literature confirms the role of the IT architect as the responsible person for developing the IT vision and having an important role in establishing the feasibility of the system. What's more, literature sees an important role for the IT architect in bridging the gap between business and technology by communication: sharing vision, facilitating decision making, resolving conflicts, finding consensus and gaining commitment.

CONTENT

1 Research approach

The master project is a literature survey consisting of three phases. The first phase focused on clarifying the research question and provided a clear direction for the literature study. The second phase was the actual literature survey. In the third phase the results of the literature survey were condensed and the essay was written.

1.1 Clarifying the research question

Much attention was paid to define the research questions. In order to stay motivated throughout the literature survey it was considered crucial that the focus of the research was close at heart. To get a result within three months the scope needed to be clear and concise. In order to determine what the literature survey brought the hypothesis was well defined.

The initial concern was how architecture can be used to increase the level of success of IT projects. To find a clear focus for the research questions three projects I've participated in were described and analyzed. The selection of projects focused on major projects that failed. The focus of analysis was on the important decision moments, assuming that better decisions would have improved project results considerably.

After elaborate analysis of the cases the choice was made to narrow the scope to the very early project decisions, to focus on missing or unreliable information and to focus on the possible contribution of IT-architects. Analyzing the three projects with these restrictions led to the 'business case' as a commonly recognized decision moment for which the research questions were formulated.

Analyzing the cases again led to the observation that two of the cases suffered from lack of vision and did not acknowledge the risks involved. This observation was further detailed and was the basis for the research questions.

1. Does literature confirm that a business case for ICT projects should include a vision and feasibility study?
2. Does literature confirm the distinguished responsibilities of an IT architect in making a business case?
3. Does literature confirm the proposed viewpoints with respect to vision and feasibility?

This phase took four and a half weeks. It took a lot of time to abstract vivid experiences; to establish a good working relation and to learn how to set up a literature survey.

1.2 Literature survey

The literature survey focused on the following fields:

- Architecture in general
- Business case in IT projects
- Business / IT alignment
- Feasibility and risks in IT projects

Literature sources mainly comprised of:

- Books from Henk Koning and the architectural group at the Free University of Amsterdam
- Books used in the master Software Engineering
- Papers from the internet, using ACM, Citeseer, Google, and SEI as main starting points

The strategy was first to get a good overview of existing sources and then to study and collect the contributions of relevant sources. To get a good overview many papers and books were read superficially and many internet searches were done. The search strings were logged.

The sources that proved to be relevant were read and a bibliography was made. The most promising sources were singled out and were studied in more detail. The literature survey took about six weeks.

1.3 Essay

The last phase focused on condensing the information, answering the research questions, and integrating information into a new hypothesis grounded in literature. This phase required discipline as many new ideas obscured the focus.

2 Cases

2.1 Introduction

This chapter describes and analyzes two projects I participated in. The projects are very close to heart and the description still reflects this closeness. The research questions were based on the description and analysis of these two cases. A third project that was described and analyzed played a less important role in this master project and is moved to the appendix. The cases are described in short followed by an analysis of the crucial decisions.

2.2 Case Sherlock

Period: 1/2/1995 – 1/10/1997
Refers to: New system development
Size: Approximately 100 entities

Synopsis

From 1994 to 1997 the system Sherlock was developed for a recently formed department, consisting of five smaller departments who operated pretty autonomously. The main functionality of the system consisted of tracking and tracing examined goods; reporting the chain of custody (who examined the goods, who had the goods in possession); reporting the chain of evidence (which evidence came from which items) and providing management information on the primary process. The main reason for implementing the system was the mislocation of several items during the past couple of years.

The project was conducted using the classical waterfall approach. The requirements study was done by an inexperienced engineer. There was no tooling used, other than a template for the requirements. Requirements engineering was done using interviews, task analysis, and feedback on prototypes. Freezing functional specifications took almost one and a half year (!), partly because the system was considered to be a threat to some of the responsible persons of the sub departments. To freeze the specs a seasoned software engineer was put onto the project, however with no prior project management experience.

The IT department did not have a lot of experience with software development. In the past most software development had been outsourced. Management had no IT background. To develop small projects the IT department had recently selected Progress, a 4GL, based on Client Server, Relational Database and Graphical User Interface environment. An external company was hired to assist with the implementation of Progress.

In the Progress environment it was easy and efficient to develop most of the functionality. But it offered poor support for re-use, event management and programming more complex functions. Bugs could only be found during manual testing and it was easy to introduce errors when correcting or changing functionality. There was not a lot of support and experience for this software environment. This slowed down the development process considerably.

During the acceptance tests and early production phase, it turned out that the program was not suitable for use. Summarized it took almost three years, a lot of people and money to develop a medium sized application that was poorly used.

Analysis of decision moments

During this project there were a number of moments and decisions that can be considered as crucial. Better decisions would have improved the results of the projects. These decisions are listed here:

Selection of project staff

Throughout the project, staff was selected poorly. The project team did not have the proper skills, especially in the areas of project management and human computer interaction. This resulted in two years schedule overrun and poor usability. The main selection criterion for staff was availability. There was not a good insight in the skills required for this project and the impact of selecting staff without these skills.

Feasibility of project

After the requirements engineering a go / no go decision was taken. This decision was taken by development management and management of the customer. The decision was very informal and missed important

information, specifically regarding the feasibility. There was no specific information asked, management trusted the judgment of the project team. On the other hand the project team did not know how to provide relevant decision information.

Usability / user acceptance versus satisfying requirements
Prior to the implementation of the new system, the department did not log any information. In order to satisfy the requirements users would have to log extensively while benefits for the users were minimal. The department management and project team acknowledged this dilemma, but decisions were not taken firmly. This resulted in changing requirements, however when the system was implemented user acceptance was still poor.

Process oriented versus modular
Within the department all kinds of items and cases were examined. Items varied from cars to the smallest traces of evidence. The requirements applied to all of these items. For the user interaction a choice had to be made between specific screens for each type of item, and general item screens. To limit development work and increase maintainability the second alternative was selected. However everybody felt uncomfortable with the resulting screens. This resulted in changing requirements, however when the system was implemented user acceptance was still poor.

Classical waterfall versus phased delivery
The users had experience with IT implementations. They had learned that if they wanted some functionality they should include it in the initial scope. Otherwise the features would never be implemented. The users did everything to enlarge the scope, causing delays. Problems and mistakes only became clear during acceptance tests.

2.3 Case CRM

Period: 1/6/2000 – 1/12/2003
Refers to: Selection and implementation of CRM package
Size: Approximately 3.000 users

Synopsis
The organization in question is a large call centre organization that works for other companies. It has approximately 6.000 employees world wide. The projects vary a lot, but the software used is basically the same. The environment mainly used is Edge, sometimes in combination with systems of customers. Edge is a 4GL with built in functionality for call centre applications (i.e. scripting, call registration, communication with telephone systems).

In the late nineteen-nineties CRM was a hype. As an early adaptor the organization was eager to offer CRM services without exactly knowing what this meant. This strategy was substantiated by employing a marketing strategist to develop the CRM concepts and secondly by acquiring a CRM tool demonstrating to customers that the organization was CRM enabled. Preferably the tool should have such a good reputation, that it would stimulate sales.

During the selection process it became clear that the CRM concepts available so far, were too abstract to translate to requirements for the selection process. It was decided that the tool would have to match the functionality and features of the current software environment. Also the new tool would have to be more efficient and enable higher quality than the current environment.

During the selection process it became apparent to the development organization that too much information was missing to make a substantiated choice. The project however was run by the CIO and pressure was high to make a choice without reservations. In the end the software development department conceded under condition of a pilot. During the pilot the development team ran into a lot of problems. The managers put these issues at the supplier, who promised to resolve them. While the pilot team was skeptical, management felt in control and decided to proceed with the implementation.

While working on the first release, the supplier put a new product on the market with a lot of nice features. Again there was a big decision: continue with current implementation or switch to the new product. The business architects were favorable towards adopting the new product and convinced the (new) CIO of this decision. Development conceded again, but temporarily managed to save the first project. However after some weeks it became apparent that the new product required skills of the programmers that were beyond the capabilities of the

development department. Months were lost trying to train staff in the new environment. One of the business architects switched to development to get the new product up and running, but to no avail.

After almost two years and lots of money the new environment was still far from being operational. The implementation team complained about to little feedback from operation. A decision was made to do a pilot project. The software was implemented and failed miserably. The implementation of the pilot cost ten times as much time and money than it had cost using the old environment. There were major problems and functionality was not much different than the functionality of the old edge environment. Within operational and general management this was reason to try to stop the project. IT management tried to counter this but after nine more months of hassling with few successes to show, the project was killed, as was the career of the development manager.

Analysis of decision moments

This case suffered from some of the same phenomena that were manifest and described in the e-learning case, in the appendix. There were solution pushers, information was concealed by the supplier and when the project was on a roll, too many people were committed and too much money was spent to stop it. There were a number of other major complications that were not apparent in the e-learning case. These are described below:

Shifting course of implementation

During the implementation several severe issues arose. These were reason to change the course of the project. The decisions were taken by highest management with an IT background. Yet there were strong differences in opinion. Positions were taken and defended. In the end the changed course meant that the project could not show success. Also the decision to change to the new web-based product was far less thorough than the initial selection process.

Unclear what information was needed and how solid the decision was

During the process it was not clear what information was required to make a well founded decision. Architects and management were convinced they were on top of the problem, only to find themselves wrong. The supplier had a different interest and was not fully open. The experience of IT staff and management was too little to assess if all information was on the table.

Outcome of decisions impact lots of people

This project had a major impact on the organization. Changes concerned lots of people. They were not part of the decision process. Their expectations were not managed. After the first (failed) implementations the general opinion was that CRM was a big mistake. This generated so much pressure that it caused the project to stop.

3 Research questions

3.1 Introduction

This chapter details the research questions. As the initial research concern was very broad, considerable attention is paid to the scoping of the research question. This is described in paragraph 3.3. The research questions concern vision and feasibility and the role of the IT architect. Paragraphs 3.4 and 3.5 make clear what is meant by vision and feasibility and to clarify the perceived role of the architect. Finally paragraph 3.6 lists the three research questions.

3.2 The Context of research question

The research questions are based on the cases described in the previous paragraph, the assumption that better decisions will improve project results and that architecture is a way to improve decisions. The projects concern development and implementation of large information systems.

3.3 Scoping the research questions

After elaborate analysis of the cases the choice was made to narrow the scope to the very early project decisions, to focus on missing or unreliable information and to focus on the possible contribution of IT-architects. This brought us to the 'business case' as a commonly recognized decision moment for which the research questions were formulated. The business case is typically used to make a go / no go decision, benefits are matched with costs and business needs are matched with risks.

Observation:
Examining the cases again led to the following observation. All cases concerned major changes involving new technology and staff that had no experience with similar projects. As a result there was no knowledge of the challenges that lay ahead. While business needs were clear, a vision how to satisfy these needs lacked. Problems that occurred were not foreseen and expectations were badly managed. As a result confidence in the project and in the (project) management was lost and the projects became subject to political forces. This leads to the following hypothesis.

Hypothesis 1: vision and feasibility part of the business case
The projects would have benefited from a vision that showed a way to satisfy business needs, and acknowledged organizational capabilities (skills, systems, processes), limitations of technology and project challenges. The vision is required to realistically determine costs and benefits and to assess feasibility. Only then it is possible to make a founded business case and make an educated go / no go decision. The vision will also provide a foundation and direction for the project and make it easier to manage, deliver results, overcome problems and manage expectations. This will make the resulting project less vulnerable.

Hypothesis 2: role of the IT architect in creating the vision and determining feasibility
Establishing the vision how the new system can satisfy business needs and determining part of the feasibility is the role of the IT architect.

Hypothesis 3: viewpoints assure the decision is made based on all relevant information
The vision and feasibility of the system are expressed by the IT architect in the form of an architectural description. As every business case is unique, the architectural description is situation specific. However from a more abstract level the questions that need to be answered for every business case are more or less the same:

- Will the project satisfy my business needs?
- Can the project be done as proposed?
- How much will the system cost?

Capturing these issues will prevent that relevant decision information is not being considered. This can be done in the form of viewpoints that can be used for creating business cases in different projects.

[IEEE 1471, 2000] defines the concepts of architectural definition, viewpoints, concerns and views as follows: "An architectural description is organized in views. A view represents the system from the perspective of a related set of concerns. Concerns are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders. Views adhere to a viewpoint, containing a specification of the conventions for constructing and using a view, a pattern or template

from which to develop individual views by establishing the purposes and audience for a view and techniques for its creation and analysis."

As becomes clear from this definition, viewpoints focus around concerns. As was analyzed there are two main concerns in the business case:

- Vision: How will the new IT system satisfy business needs?
- Feasibility: Can the project be done as proposed / is the vision realistic?

There is a strong relation between these two viewpoints. The vision focuses on the needs, solution and benefits based on identified constraints. The feasibility focuses on the risks of the project, and determines if the proposed vision is adequate and realistic.

How would the cases have been helped with the proposed business case?
The case Sherlock would have been helped by a business case. First of all information would have been considered that was now missing in the go / no go decision. This information included:

- Vision how to build and implement the system
- Solidity of estimation
- Insight in risks
- Clear description of benefits
- High level validation of requirements

When this information would have been available, the project would never have been started under the current conditions as the risks would have been considered too high in relation with the benefits. If the project would have been started, the business case would have provided a solid basis for management to intervene.

The same goes pretty much for the CRM case. There a business case was made, however missing vision and a decent feasibility study. In this case the following information would have become available in the decision: realistic benefits of the CRM solution; the business needs on a more concrete level; the high level requirements; a vision on how the CRM solution would materialize; feasibility information; and the capabilities of technology. Also this project would not have been started because benefits would proof too low in relationship to the risks. And if the project would have been started it would have benefited from a clear vision providing direction to the project and a commissioner eager to adopt the new system.

3.4 Assumed responsibilities of the IT architect with respect to the business case

This paragraph summarizes the assumed responsibilities of the IT architect. This list has been derived by contemplating the possible role of the IT architect in the described cases, had a business case been created. The main domain of an architect is considered the interaction between ICT and business on a strategic level, a conceptual, creative role in understanding how technology can satisfy business needs and determining the feasibility of the vision. This list will make it possible to see what we've learned from literature and if literature confirms the specific role of the IT architect.

Vision

- The IT architect should develop a vision and concept on how the new system will function within the company.

Business needs and foreseen benefits

- Assess if technology can satisfy the business needs
- Assess if technology can deliver the foreseen benefits

Feasibility study

- Knowledge of the productivity, capabilities and limitations of the technology, knowledge of track record of technology
- Determine how realistic the estimation and timeline is.
- The IT architect should be able to do a full risk analysis on the technical and operational issues, considering factors like:
    - Complexity and stability of requirements
    - Current and required skills of IT staff
    - Novelty of environment
    - Capabilities and limitations of environment
    - Showstoppers and dependencies

3.5 Proposed viewpoints for the business case

To make the perception of vision and feasibility concrete, two viewpoints are defined before reviewing literature. As has been stipulated in 3.4, there are two main concerns: vision and feasibility and consequently two viewpoints are proposed for use in the business case. To determine the contents of the viewpoints the cases were analyzed. The relevant terms / issues were identified and their relation was established. These were mapped to the attributes of the template. Some attributes in the viewpoints are marked not applicable (n.a.) for it was found they were on a too concrete level for this project at this moment.

Attribute	**Meaning**
Title	Business vision on new system
Stakeholders	Higher management, budget keepers, project team, operation management, marketing
Concerns	How will the new system function within the company? Can the technology satisfy the business needs? Can the technology deliver the foreseen benefits?
Type of information	This view describes changes in the organization after introduction of the new system (impact on the primary production process, on the sales process; changes in needed skills). It focuses on desired improvements and how they can be realized. There is an estimation of costs and financial benefits, and an assessment of the contribution to overall business goals.
Presentation media	Slide presentation
Architecture concepts	Business functions, work processes, business needs, capabilities of technology.
Stakeholder oriented terms to be used	Business problems, change process, lead time, business strategy, business mission, strengthen sales, increase in revenues, lower costs, earn back time, return on investment
Outline of text	n.a.
Stakeholder oriented graphics to be used	n.a.
List of diagrams	n.a.
Techniques	Using templates

Attribute	Meaning
Title	Feasibility of new system
Stakeholders	Higher management , budget keepers, project management, operation management, IT management
Concerns	Can this project really be done as proposed?
Type of information	Given the proposed new situation, as laid out in the business view, this view gives an account of a search for technological or organizational issues that can cause the project to fail. Data is sought and presented about the used technology that supports or questions the estimations of time and money and presumed ability of the IT and user organization to perform the proposed activities. Special attention is given to risky parts of the project plan, e.g. where there is a high dependency on specific resources.
Presentation media	Slide representation
Architecture concepts	Features, limitations, track record / experiences of/with used technology, skills, risks, changes in development process, changes in infrastructure and support process
Stakeholder oriented terms to be used	Trade offs, features, out of the box, customizable, rigid, flexibility, track record, success factors, skills, experience, resources, risks, predictability, single point of failure, dependencies, alternatives, feasibility
Outline of text	n.a.
Stakeholder oriented graphics to be used	n.a.
List of diagrams	n.a.
Techniques	n.a.

3.6 Research questions

1. Does literature confirm that a business case for ICT projects should include a vision and a feasibility study?
2. Does literature confirm the distinguished responsibilities of an IT architect in making a business case?
3. Does literature confirm the proposed viewpoints with respect to vision and feasibility?

4 Results

4.1 Introduction

This chapter is an anthology of literature concerning the research questions. The chapter is concluded with answers to the research questions. Paragraphs 4.2 and 4.3 discuss architecture in general, paragraph 4.4 contains the findings regarding the business case, paragraphs 4.5 and 4.6 are about vision and feasibility, paragraph 4.7 contains contributions with respect to the responsibilities of the IT architect and paragraph 4.8 is about viewpoints used in the business case. The size of the paragraphs about vision and feasibility reflects the variety of views found in literature. Paragraph 4.9 contains the conclusions about the research questions based on literature. Also a new hypothesis is made about the role of the IT architect in the business case, again in the form of a list of responsibilities and viewpoints, this time the hypothesis is grounded in literature.

4.2 IT architecture what are we talking about?

There are many definitions of IT architecture in computer science, varying in focus, scope and purpose. The IEEE defines architecture as the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution [Florijn et all, 2003]. In this definition the focus is a software system.

This definition applies both to existing systems and to blue prints for systems to be developed. With respect to existing systems Verhoef has defined software architecture as that part of the system that is the hardest to change [Florijn et all, 2003].

[Bernhard et all, 2002] positions architecture as an instrument to manage change. Architecture is defined as a description that will be realized. This description should be controllable and accomplishable and concern the fundamental aspect of the focus of change. Fundamental is defined as acceptable (understandable) and lean.

[Bass et all, 2003], [Hofmeister et all, 1999], [Kruchten, 2000] see architecture mainly as an instrument to master design complexity in the process of developing an information system.

The concept of architecture is also applied at an enterprise level. [Wagter et all, 2001] distinguishes three levels of architecture within an enterprise:

- Business architecture, concerned with product / services, processes and organization
- Information architecture, concerned with data and applications
- Technical architecture, concerned with middleware, platforms and networks

4.3 Purpose of architecture in information systems development

It is widely recognized that there are many problems in developing IT solutions, while IT has grown more and more important to organizations.

[Bernhard et all, 2002] gives the following overview of IT problems:

- Unpredictable projects: 39% of projects over 10.000 function points run away, only 26% finishes within schedule.
- Business processes are supported insufficiently
- Total cost of ownership is too high
- Responsibilities are unclear

[Bernhard et all, 2002] has identified the following source of these problems:

- Knowledge gap between business and ICT
- Conflicting interests of involved parties
- Complexity of software
- Isolated change processes

To [Bernhard et all, 2002] architecture has the following purposes:

- Bridge between vision, goals and realization
- Agreement between stakeholders
- Guarantee feasibility
- Manage process

[IEEE 1471, 2000] also positions architecting as being concerned with developing satisfactory and feasible system concepts as a response to the problems in the field. [IEEE 1471, 2000] situates the architectural level of systems development as the earliest period of decision making and evaluation.

[Smolander, 2003] positions architecture in the solution space, as an instrument that makes possibilities, limitations, and constraints explicit and more understandable. It may serve as reference during early stages of development including requirements engineering. Architecture also affects the perceptions of the problem.

Architecture design can be regarded primarily as a means for coping with complex solutions and technology, reached through communication between diverse stakeholders with varying skills and experience. Architecture development can be characterized as a process seeking boundaries, finding consensus, and identifying commonalities across organizational borders. [Smolander, 2003].

[Hofmeister et all, 1999] positions architecture in the field of developing large software systems. Keywords are managing complexity, resolve trade-offs between conflicting requirements, quality software, acceptable time to market, integration of system components. This view is shared by [Florijn et all, 2003], [Bernhard et all, 2002], [Bass et all, 2003], and [Smolander, 2003].

Another important goal of architecture is validation. Architecture enables verifying if all requirements and preconditions are met in the early stage of developing complex, large scale IT systems [Florijn et all, 2003].

Documentation is considered an important purpose of architecture, stipulated by many sources including [Smolander, 2003], and [Bass et all, 2003].

[Dikel et all, 2001] states that architecture enables companies to select the best mix of in-house development, and products and services available in the market place. This enables attractive competitive solutions that take advantage of strengths of many groups within and outside a company.

[Wagter et all, 2001] sees architecture as a means to maintain cohesion between processes, organization and IT.

4.4 Use of business case

A business case is an established go / no go moment before the start of a project [Kruchten, 2000], [OGC, 2004], [Holcolmb et all, 2000]. There is not one conception of a business case, though there are strong similarities. A business case typically consists of business needs, a cost benefit analysis and a risk assessment. Other aspects that tend to be included in a business case are: vision, business context, project objectives, project plan.

A business case is not only used to see if a project is economically viable, but is also typically used to verify alignment with company strategy. The business case also structures the process of organizational decision making.

Perhaps the strongest advocate of a business case for IT projects is [Remenyi, 1999]. [Remenyi, 1999]: the main value of a business case is to make a grounded decision, give adequate direction to the project and prevent a doomed project from being started. The business case is regarded as an evaluation moment, central to IT quality management. The value of financial forecast is put into perspectives. Keynes is quoted in saying that people only have limited ability to predict the future.

The comprehensive business case presented in [Remenyi, 1999] has the following aspects:

1. Clearly expressed business objective and set of outcomes, expressed at different levels of abstraction. In this book macro (high level statement following a template), meso (describes all benefits) and micro (numbers) level are distinguished.
2. List of stakeholders and beneficiaries, assessing their interests and position
3. Statement how the proposed IT expenditure will support the corporate strategy
4. Evaluation of the appropriateness of the technology and operational plan
5. Evaluation of risks

Some sources directly link architecture to business cases. [Wagter et all, 2001] sees the strategic dialogue between business management and architects as one of the key architectural processes. This process results in a

business case. The creation of a joint vision by IT and business is considered essential in this process. Feasibility and risks are much less of an item. In [Wagter et all, 2001] the business case includes:

- Sketch of the rough solution
- Impact analysis
- Financial analysis
- Project proposal

[Gordijn, 2002] presents an approach called e-business idea exploration. This approach has the same objectives as a business case. Using architectural tools an idea is developed into a vision, using models, and then validated. The validation focuses on technical and commercial feasibility.

4.5 Vision

Vision is widely recognized as an important concept providing direction, cohesion and motivation. Vision not only depicts the new situation, but also makes clear why this is beneficial and why it is important that the project is conducted. The vision identifies problems, abstracts these into constraints and provides a solution. The vision will make clear how the project result is integrated in the environment.

Definition of vision:

[Kruchten, 2000] defines vision as the stakeholder and user needs and the high level features of the system, based on cost benefit analysis. The vision describes what the system will do, captures expectations of different stakeholders, is written from the customer's perspective, focuses on essential features, acceptable levels of quality, scoping, operational capabilities, and inter operational interfaces outside system boundary. Vision is seen as the contractual basis for requirements.

[Gordijn, 2002] refers to vision as:

- How value is created and perceived
- How processes are arranged
- How the system will be structured

The definition of vision in [Dikel et all, 2001] makes clear how important it is to abstract from problems. Vision is defined as "identifying and conveying a substantive connection between seemingly unrelated use cases that connect expected use of architecture with tangible goals of the user that can be satisfied". Key in this definition is finding a solution that satisfies different needs, requiring an abstraction of some sort.

Constructing a vision:

[Dikel et all, 2001] quotes Thompson for his three step approach for projecting architecture vision:

1. Articulate compelling customer value clearly and concisely
2. Map the value to a small set of specific solvable problems
3. Translate these problems into a minimal set of constraints

[Hofmeister et all, 1999] starts information systems development with global analysis. This analysis focuses on the domain, requirements, skills, risks and concerns organizational, technological and product factors. During global analysis design strategies are formulated that are used in designing the system and developing the other views. From a more abstract point of view this global analysis phase can be considered as creating a vision by taking the following steps:

- Identify needs and constraints based on analysis of organizational, technological and product factors
- Choose consistent set of design strategies

A vision is based on the business needs. To determine the business needs, the following classification given by Parker and Benson in [Wagter et all, 2001] can be used:

1. Return on investment
2. Strategic Match (relation to strategic goals)
3. Competitive advantage
4. Management information (potential management information about core activities)
5. Competitive response (competitive deficit when project isn't executed)
6. Strategic IS architecture (relation to IS strategy)

Business / IT alignment:

From a project perspective it is important that the vision of the project matches with the (vision of the) company. An information system in itself is not a goal for most organizations. Information systems typically support new

ways of working. Most problems and constraints are found in the context of the business system. This context includes organizational processes, organizational behavior, future trends on the market place, future available software systems.

[Remenyi, 1999] quotes Chandler and Pascale to define business strategy and make the notion of strategic match more concrete. Chandler: "Strategy can be defined as the determination of the basic long term goals and objectives of an enterprise and the adoption of courses of action and the allocation of resources necessary for carrying out these goals." Pascale: "Strategy pertains to a firms plan of action that causes it to allocate its scarce resources over time to get from where it is to where it wants to go."

In this context it is important to realize that development and implementation of large information systems is in fact strategy execution. Only justified by how this will contribute to the vision of the company. However it is also important to realize that strategy and vision are also shaped by such a project, as the projects provide more and better information about opportunities, constraints and so on.

To establish if project vision is in line with business and business vision, concepts of the field business / I(C)T alignment can be used. [Henderson et all, 1993] has presented the Strategic Alignment Model (SAM). [Henderson et all, 1993] presents the strong notion of strategic alignment achieved by strategic fit and functional integration. Strategic alignment means that internal arrangement (structure, processes, skills) should be in line with external position of the company (product – market). [Henderson et all, 1993] distinguishes between Business strategy, IT strategy, Organizational infrastructure and I/S infrastructure. Functional integration is achieved on a strategic level between Business and IT and on an operational level. SAM has been subject to further research as shown in [Maes, 1999], [Wieringa et all, 2002], [Aerts et all, 2003]. Based on SAM [Maes, 1999] has created a generic framework describing the aspects of the domain of information systems development and the relationship between them. This provides a good framework to determine if the vision addresses contains all relevant aspects. The framework has three focus points: Business, Information / Communication and Technology and makes an abstraction on levels of strategy, structure and operation.

Alignment is in itself a broad and widely used management instrument [Maes, 1999]. Labovitz and Rosansky consider the alignment of strategy, customers, people and processes essential for the growth and profit of any company [Maes, 1999].

[Maes, 1999] has some reservations to go along with the compelling notion of alignment. His main arguments are that this concept has not been proven and that a strong case can be made that innovation springs from lack of alignment.

Other arguments to question the importance of strategic alignment between Business and IC/T alignment presented by [Maes, 1999] include:

- Business developments depend on many more factors than merely ICT.
- ICT development have a greatly autonomous character; acknowledged in [Wieringa et all, 2002] and [Aerts et all, 2003]. This can be seen as the motivation for companies to have an IT strategy as stipulated by [Henderson et all, 1993].
- ICT infrastructures are in themselves primarily dependent on the rigid installed ICT base

4.6 Feasibility

Feasibility has been defined as do-able. More specifically is the vision realistic and does it address all constraints. A first review of literature shows an abundance of risk factors, sometimes organized in risk frameworks. Unfortunately there are many different frameworks all using a different organization and constructed from different perspectives. To make it more problematic, most risk frameworks are intended to manage and mitigate risks rather than to determine feasibility of a project.

This paragraph has the following organization:

- Discussion of some of the relevant risk frameworks
- Summary of critical risks experienced by practitioners
- Discussion of risk detection methods
- Evidence that stresses the importance of organizational factors on success and failure of projects
- Establishing organizational abilities

Risk frameworks:
[Remenyi, 1999] presents the classification of McFarlan. This refers to failures of conceptualization: poor ideas that were not likely to succeed from the beginning:
1. Failing to meet customer requirements
2. Requiring behavior that is not ingrained in existing users
3. Incapable of evolving
4. Disrupting organization abilities
5. Lowering entry barriers for competitors
6. Undertaken before all tactical and strategic resources are in place

Parker and Benson identify four types of risks in a project that need to be considered when taking the go / no go decision [Wagter et all, 2001]:
1. Project or organizational risk: capability of company to adopt changes associated with project
2. Definitional uncertainty: to what extent can requirements be precisely defined
3. Technical uncertainty: availability of required technical knowledge and skills
4. Infrastructural risk: level to which project requires additional infrastructural investments not directly related to the project

[Carr et all, 1993] presents the very elaborate risk framework of SEI that identifies 64 risk factors, organized in three main categories: Product Engineering, Development Environment and Program Constraint and 13 sub categories.

To determine and predict the organizational adoption of new system, models and notions from innovation diffusion theory and theory about organizational change processes can be used. This literature survey has focused on innovation diffusion theory in the context of organizations. In this respect [Gallivan, 2001] and [Heemstra et all, 2001] provide relevant contributions. [Gallivan, 2001] finds that success of organizational adoption is for a large part determined by managerial intervention (e.g. communication, training), subjective norms (e.g. peers, clients) and by facilitating conditions (of the innovation, organization or individual). [Gallivan, 2001] presents a model that shows how eight organizational "patterns" (like bureaucracy) influence adoption.

[Heemstra et all, 2001] distinguishes five characteristics of innovations that help predict the speed, level, or success of adoption of the innovation. Important is that [Heemstra et all, 2001] stipulates that it concerns the individual perception of the person making him / her to adopt rather than objectively measurable characteristics.
- Benefits (innovative technique is better than predecessor)
- Match (matches with norms, skills and way of working)
- Complexity (relative easy to understand and use)
- Easy to experiment and try
- Visibility and easy to communicate results and benefits

Risks in practice
[Moynihan, 1997], [The Standish Group, 1995], [Keil et all, 1998], and [Addison et all, 2002] have done research to the reasons for failures as perceived by project managers. These researches have been done using interviews, and questionnaires.

The results presented in [The Standish Group, 1995] show the following causes of failure:
- Incomplete requirements: 13,1 %
- Lack of user commitment: 12,4 %
- Lack of resources: 10.6 %
- Unrealistic expectations: 9.9 %
- Lack of executive support: 9.3 %
- Changing requirements: 8.7 %
- Lack of project planning: 8.1 %
- Absence of need: 7.5 %
- Lack of IT management: 6.2 %
- Technological illiteracy: 4.3 %

The reasons of failure in decreasing order of importance as found by [Moynihan, 1997]:
1. The client's knowledge/understanding/clarity regarding the requirements/problem to be solved
2. The existence/competence/seniority/commitment of the project patron/owner
3. Level of IT competence and experience of the customer/users
4. Need to integrate/interface with other systems
5. Scale/coordination complexity of the project (numbers of disciplines, need to share resources, need to subcontract, and so on)
6. Main source of control over the project (developer versus client versus third parties)
7. Level of change to be experienced by the client (to procedures, workflow, structures, and so on)
8. The need to satisfy multiple groups of disparate users versus the need to satisfy one group of similar users
9. Who we will be working through: users versus the IT department, individuals versus committees
10. Developer's familiarity with platform/environment/methods
11. Developer's previous experience with the application

Most of the important risk factors found by [Keil et all, 1998] and [Addison et all, 2002] are acknowledged in [The Standish Group, 1995 or [Moynihan, 1997], except for the following two risks:
- Unclear or misunderstood scope / objectives
- Misunderstanding requirements

[Moynihan, 1997] compared the results of the interviews with the risks frameworks of [Carr et all, 1993] and a risk framework constructed by Barki, Rivard and Talbot after a wide review of literature. This analysis led to the following observations:
- The experience of the project managers reflect most of the risk factors identified in literature
- For one and the same risk concept different project managers considered different facets important. Though the differences appeared subtle, these were distinct and important.
- The project managers identified risk themes that were not present in literature. This could be explained by differences in context.
- Literature identified several risk areas not mentioned by the project managers. In the case of the SEI list, this concerned mainly the more technical risks. This could be explained by the context of the project managers, who operated in areas with relatively low technical complexity.

Based on these observations, [Moynihan, 1997] doubts attempts to catch all risks in one risk taxonomy.

Risk detection:
Most risk methods assume that the organization has adequate knowledge of the risks and that the risks can be identified by using a structured method [Carr, 1999]. Identification of risks is usually done in the form of checklists and group sessions. [OGC, 2004] and [Pandelios et all, 1999] present a comprehensive method to identify and analyze risks. [OGC, 2004] stipulates to first select an appropriate risk framework. [Pandelios et all, 1999] uses the risk framework and questionnaire presented in [Carr et all, 1993].

Architecture plays an important role in risk detection. [Bernhard et all, 2002] claims that potential conflicts and problems appear automatically when working out the viewpoints. [Bass et all, 2003] considers architecture captured by views to be the first artifact that can be analyzed to determine how well its quality attributes are being achieved and that can be subject to evaluation and risk assessment. [Bass et all, 2003] describes several methods to evaluate architectures to determine feasibility, focusing if the resulting system will satisfy its requirements.

Risk detection usually proceeds until all stakeholders have sufficient confidence [Bernhard et all, 2002]. [Bass et all, 2003] however warns against too much optimism, and considers completeness as an important criteria to stop examining solutions for feasibility.

Organizational factors:
Some sources confirm the importance of organizational factors like process, culture, maturity and patterns for the successful undertaking of a project. [Dikel et all, 2001] discerns organizational (anti-) patterns beneficial or detrimental to developing information systems using architecture. [Wagter et all, 2001] illustrates typical examples of organizational dynamics that frustrate the incorporation of working under architecture.

[Smolander, 2003] quotes Mathiassen: "The process of information system development is affected by the experience and competences of the development group, the considered object systems, the dynamics of the objectives and especially the social and technological environments in which the change takes place."

Field research presented in [Smolander, 2003] finds that E-business development seems to necessitate organizational changes, integration of information systems and rather painful formulation of common goals. Areas of concern that need to be supported by development processes include:

- organization change
- communication between stakeholders
- systems integration
- objective formation
- evolutionary development

[Bass et all, 2003]: "Architecture influenced by architects background and experiences, functional and quality goals of customer and developing organization (vision and structure are given as example) and by stakeholder. An architecture is only as good as its ability to be communicated to and understood by its stakeholders."

Organizational capabilities:
Feasibility is not just determined by the risks, but also by the ability of the organization to deal with these risks. A project that might be simple for one organization might proof impossible for another. There are a lot of models that describe organizational abilities to develop software. This is a wide research field and is not included in this literature survey. For a comprehensive overview refer to [Heemstra et all, 2001]. He describes various ways to assess ability of an organization, looking at processes (CMM), resources (skills, team, and so on) and culture. Another way to determine organizational ability to successfully conduct a project is by looking at organizational patterns and their effect on project outcome, as described by [Dikel et all, 2001] and [Gallivan, 2001].

Conclusions feasibility:
With respect to feasibility the literature survey has uncovered many views. It is hard to find cohesion. Taking some distance, the following observations can be made:

1. There are many different risk frameworks, all with different organizations, having different perspectives and distinguishing different factors.
2. Field research and risks frameworks show a tremendous diversity of acknowledged risks.
3. Validity of risk frameworks has not been established.
4. Most risks frameworks are taxonomies that focus on completeness and lack abstraction. As a result root causes of risks, or risk patterns are hardly ever identified by using these frameworks.
5. [Carr et all, 1993] claims that the number of risks and level of risks in itself do not say anything about the likelihood of a project to succeed or fail. That requires determining the impact of the risks, as well as the organizational abilities to mitigate the risks.
6. Different sources confirm the importance of organizational and environmental factors in the success and failure of projects.
7. By modeling processes and systems feasibility from a technical and business process perspective can be established

None of the risk frameworks discussed above can be readily used for the feasibility viewpoint. Analyzing the different risk frameworks brought forward two basic questions that need answering to establish feasibility:

1. Is the organization capable of executing the project as proposed
2. Is the organization capable of adopting the resulting system

Analyzing the literature material as summarized in this paragraph for keywords and mapping them to these two questions led to the following refinement:

Is the organization capable of executing the project as proposed?

- Organization: Is the organization committed enough?
- Organization: Are the expectations realistic?
- Organization: Is the company culture favorable to this kind of project?
- Task domain: Is the task domain clear?
- Task domain: Is it stable?
- Task domain: Is there a clear need for the new system by people from the task domain?
- Project and technology: Is the project team skilled and capable enough?
- Project and technology: Does the project team understand the domain?
- Project and technology: Are the resources and the timeline sufficient?
- Project and technology: Is the project organization mature enough to deal with the complexity?
- Project and technology: Has the technological solution been proven in the target infrastructure and in the task domain?

Is the organization capable of adopting the resulting system
- Organization: Is the organization favorable to changes?
- Organization: Is the organization skilled in changing?
- Change: What is the attitude of the users towards the change?
- Change: How complex is the change to the users?
- Change: Does the change match their expectations?
- Change: How beneficial is the change to the users?
- Implementation project: Is there room to train and inform staff adequately?
- Implementation project: Can the system be changed to fit the organization?

4.7 Responsibilities of IT architect

The different views on architecture are reflected in the opinions about the responsibilities of the IT architect. [Wagter et all, 2001] sees an advisory role of the IT architect in the creation of the business case / scenario. However [Wagter et all, 2001] is not specific to which aspects of the business case the IT architect contributes. [Kruchten, 2000] states that the project manager is responsible for identification of risks and the process of describing business. The IT architect is more concerned with leading technical activities. However [Bernhard et all, 2002] believes that the role of the IT architect in RUP should include the making of the business case and the creation of a vision.

All sources agree that the development of a vision is one of the key responsibilities of an architect. [Hofmeister et all, 1999]: to be a successful architect you must be a visionary. This means you must know in advance what the system will look like when its done, what it will accomplish, and how it fits in with the rest of the company's technology, and business objectives.

[Dikel et all, 2001] and [Wagter et all, 2001] stress the importance of involving the IT architect in the creation of the (company) vision, enabling strategic alignment. [Dikel et all, 2001] "Architects develop the (architectural) vision by mapping tangible future value to constraints, by clarifying risks and by designing architecture so that it can remain.

[Florijn, 2003]: developing vision, dealing with (conflicting) non functional requirements requires priorities and making decisions to define the target architecture are the primary tasks of an architect.

Some sources like [Wagter et all, 2001] make a distinction between the roles of a business architect, an information architect and a technical architect. [Gordijn, 2002] sees the role of an IT architect primarily in engineering the information system viewpoint, and establishing the IT costs and technical feasibility. The business oriented viewpoints are engineered by business developers and business process (re)designers.

The pivotal role of the IT architect between IT and business is acknowledged by [Dikel et all, 2001]. Here Thompson is quoted: "Being an architect is much more about understanding how to balance business, organizational dynamics and technology than it is about technical gears.".

[Hofmeister et all, 2000] lists a number of responsibilities of the IT architect in the process of developing a project. These include the following aspects:
- stay up to date with company's technology as well as what's new in the market
- get users perspective
- communicate vision
- key technical consultant
- bridges gap between domain and software engineering
- identify risks involved

4.8 The use of viewpoints in the business case

[IEEE 1471, 2000] confirms that architecture addresses vision and feasibility and that architecture is about the earliest period of decision making, ergo the business case. [IEEE 1471, 2000]: "An architectural description shall identify the concerns considered by the IT architect in formulating the architectural concept for the system. At a minimum, the concerns identified should include the following:
- The purpose or missions of the system
- The appropriateness of the system for use in fulfilling its missions
- The feasibility of constructing the system
- The risks of system development and operation to users, acquirers, and developers of the system"

Many sources in literature see the architectural viewpoint as the first artifact that can be analyzed to determine the ability of a future system to fulfill its requirements (functional and quality attributes), its global properties, integration of COTS components and impact of changing requirements [Bass et all, 2003], [Hofmeister et all, 1999]. The viewpoint being used for this purpose is called the conceptual view or structure view.

[Smolander, 2003] claims that viewpoints can be used to enhance agreement and enable common understanding amongst a wide group of stakeholders. With this respect viewpoints are very suitable to support a decision taking process like making the business case.

[Smolander, 2003] and [Gordijn, 2002] describe that modeling the solution using viewpoints is useful for developing the (business) vision, understanding the problem domain and capturing business needs.

Literature mentions a number of viewpoints that can be used in the process of a business case. [Bernhard et all, 2002] defines a business viewpoint, used in making the go / no go decision before the start of a project. This viewpoint describes the necessity of the application, vision, cost and benefits and contains the first structure of application. It gives direction to all subsequent viewpoints and decisions.

In his process of idea exploration, identical to making a business case, [Gordijn, 2002] distinguishes three viewpoints: a value viewpoint, a business process viewpoint and the information systems viewpoint. The viewpoints focus most of all on economic feasibility.

[Gordijn, 2002] makes a strong case for a limited set (no more than six) of predefined viewpoints eliminating the time consuming step to identify appropriate viewpoints. However most sources believe that viewpoints should be defined situational. [Smolander, 2003] finds in his research that perceptions on architecture, choice of viewpoints, use of architecture depend on organizational factors. Architecture is not so much concerned with selecting right and perfect description, but to accommodate for a process suitable for the organization to design architecture. Architecture is emergent through conflicts and constraints.

[Smolander, 2003], and [Remenyi, 1999] find a stakeholder analysis very suitable for exploring organizational perceptions on the proposed change. The architectural viewpoints are typically suited for this purpose.

4.9 Conclusions

This paragraph will focus on answering the research questions as specified in paragraph 3.7.

Does literature confirm that a business case for ICT projects should include a vision and a feasibility study?
Yes, many sources in literature confirm the importance of vision and feasibility information to make the early go/no go decision in a project. However not all sources refer to this moment as the business case. The sources in literature that define the business case confirm that vision and feasibility are important components of the business case.

Does literature confirm the proposed responsibilities of the IT architect with respect to the business case?
Most proposed responsibilities are specifically confirmed in literature. Some of the proposed responsibilities are not mentioned in literature, these include: knowledge of current and required skills of IT staff, complexity and stability of requirements, determine realism of estimation and timeline. This can be explained by the level of detail of the proposed list. Literature does confirm that an architect should be able to determine feasibility given constraints, while constraints include timeline, estimation, available skills, and so on. This implies for example that the IT architect is capable of determining solidity of estimation. This way the other unmentioned responsibilities can be inferred.

Literature also stresses the role of the IT architect in understanding the domain, getting user perspective and balance between business, organizational dynamics and technology. Further literature stresses the role of the IT architect in facilitating communication and decision-taking processes. These aspects are not well covered in the proposed responsibilities.

The newly proposed list strongly resembles the previous list, however it is more abstract and specifically includes communication.

Proposed responsibilities of the it architect grounded in literature:
Vision
Develop a compelling vision on the new system, showing how this system will satisfy business needs and can be developed and implemented within constraints.
- Understand business domain
- Know possibilities and limitations of technology
- Stay up to date with technology available in the market
- Identify and validate constraints
- Model solution

Feasibility study
Determine if the organization is capable of developing and implementing the system as proposed in the vision.
Determine if the system will indeed satisfy the business needs.

Bridge gap between business and ICT
- Communicate and share vision
- Facilitate decision making
- Resolve conflicts
- Find consensus and commitment

Does literature confirm the viewpoints with respect to vision and feasibility used in the business case?
There are only a few sources in literature that describe viewpoints used in the business case. [Gordijn, 2002] presents three viewpoints (value, process, information systems), [Bernhard et all, 2002] presents one (business view), just like [Hofmeister et all, 2000] (conceptual view). Analyzing the viewpoints reveals vague similarities between the business viewpoint presented in [Bernhard et all, 2002] and the value viewpoint of [Gordijn, 2002]. These viewpoints model the business aspects and relate to the proposed vision viewpoint. There are strong similarities between the information systems viewpoint of [Gordijn, 2002] and the conceptual viewpoint of [Hofmeister et all, 2000]. Both sources see feasibility as one of the major concerns in this viewpoint and there is a strong relation with the proposed feasibility viewpoint.

Most sources in literature however do not advocate predefined viewpoints. Yet [Gordijn, 2002] makes a strong case for predefined viewpoints in the business case in order to win time.

The most important aspect is that the literature survey has confirmed the two main concerns of the IT architect in the business case. No other concerns were revealed that are the focus of the IT architect. With this in mind I still feel comfortable with the two viewpoints proposed in the hypothesis. However the literature study did change my perception with respect to the content of the viewpoints. This will be discussed in the following sections.

Does literature confirm the proposed viewpoint with respect to vision?
The vision viewpoint is basically the same as the proposed viewpoint in 3.6. However most terms have gotten a more precise and clear meaning. The most important change is that this viewpoint should both capture system behavior and system composition. The system composition makes it possible to determine if the resulting system will meet all requirements.

The role of the IT architect in this specific viewpoint is scoped to:
- Business needs: Capture business needs, rather than to determine them.
- How will system satisfy business needs: Understand and advise how business processes will be organized when the new system is in place; understand potential of the new system and determine the features of the system.
- How will the system be developed and implemented: Develop the high level view of the components; show how the system will be integrated in the environment; advise on project plan.

Attribute	Meaning
Title	Business vision on new system
Stakeholders	Higher management , budget keepers, project team, operation management, marketing
Concerns	How will the new IT system satisfy business needs? How will the new system be developed and implemented?
Type of information	***Business needs:*** The reasons and urgency for the different stakeholders to change. Typical reasons are strengthening the competitive position or enhance continuity. Strengthening competitive position is about market share, product service offerings, price, image, and so on. Continuity is about the company's ability to change, solvability, dependencies, cohesion and so on. ***How will system satisfy business needs:*** High level view of new organizational processes and the required system behavior in form of essential features of the system. ***How will the system be developed and implemented:*** High level view of system components focusing on biggest risks and uncertainties, enabling reasoning about costs (project costs and costs of ownership) and the systems ability to fulfill requirements and meet constraints. High level project plan, determining the resources and skills required and impact on the organization. Vision how results will be integrated in the environment, both in the organizational processes as in the technology infrastructure and in the information and communication (infra) structure.
Presentation media	Slide representation
Architecture concepts	Business needs and functions, work processes, capabilities of technology.
Stakeholder oriented terms to be used	Business problems, business objectives, market share, change process, lead time, business strategy, mission, increase in revenues, lower costs, earn back time, competitive response, competitive advantage, dependencies, features, quality, scoping, operational, cohesion, strategic match, architectural fit, functional fit, design strategies, scenarios, business value, (position of) stakeholders, impact analysis, project proposal, cost benefits, alternatives
Outline of text	n.a.
Stakeholder oriented graphics to be used	n.a.
List of diagrams	n.a.
Techniques	Using templates, stakeholder analysis and scenarios

Does literature confirm the proposed viewpoint with respect to feasibility?
This viewpoint has undergone some drastic changes. The concern has remained the same, but the literature survey has provided a good understanding of the content of a feasibility study. This is reflected in the block: "Type of information".

The role of the IT architect with respect to this viewpoint focuses on:

- Engineering: Project and technology issues; communicate what organizational behavior is required to make this project successful.
- Adoption: Describe nature of the change for users. If the change affects members of the technical department, the IT architect focuses on all adoption issues with respect to the IT department.

Attribute	Meaning
Title	Feasibility of new system
Stakeholders	Higher management , budget keepers, project management, operation management, IT management
Concerns	Can this project really be done as proposed?
Type of information	***Is the organization capable of executing the project as proposed*** This focuses on the characteristics of the project (expected problems, level of uncertainty, complexity, scale, required quality, integration in infrastructure) and the ability of the organization to successfully overcome these characteristics (sufficient and competent resources, maturity of processes, perception of successful project, current infrastructure). The questions address issues about organization, task domain, project and technology. - Is the organization committed enough? - Are the expectations of the organization realistic? - Is the company culture favorable to this kind of project? - Is the task domain clear? - Is the task domain stable? - Is there a clear need for the new system in the task domain? - Is the project team skilled and capable enough? - Does the project team understand the domain? - Are the resources (staff, time, money, means) sufficient? - Is the project organization mature enough to deal with the complexity? - Has the technological solution been proven in the target infrastructure and in the task domain? ***Is the organization capable of adopting the resulting system*** This focuses on the likeliness of different stakeholders to adopt the new system and change their ways of working and the ability of the organization to manage and guide the adoption, given the nature of the change. The questions address issues about organization, change and project. - Is the organization favorable to changes? - Is the organization skilled in changing? - What is the attitude of the users towards the change? - How complex is the change to the users? - Does the change match their expectations? - How beneficial is the change to the users? - Is there room to train and inform staff adequately? - Can the system be changed to fit the organization?
Presentation media	Slide representation
Architecture concepts	Features, limitations, track record / experiences of/with technology, skills, risks, changes in development process, infrastructure and support process
Stakeholder oriented terms to be used	Trade offs, features, out of the box, customizable, rigid, flexibility, track record, success factors, skills, experience, resources, risks, predictability, single point of failure, dependencies, alternatives, feasibility, cost of ownership, process maturity, organizational culture and attitude, size of change, probability, disrupting organizational abilities, integration
Outline of text	n.a.
Stakeholder oriented graphics to be used	n.a.
List of diagrams	n.a.
Techniques	n.a.

5 Evaluation

Introduction:
This chapter contains the evaluation as prescribed by the template. It consists of an evaluation of the positive and negative aspects, a reflection of the research approach and is concluded by a self assessment.

Positive:
I started this master study with the drive to learn new ways to improve results of software projects. This master's project has given me a new perspective on problems and ways to overcome these. I can relate this to my experiences and when I will do a new project I will certainly approach this differently then before. Software Architecture has provided me with tools and instruments to support this new way of working.

During this master's project, I have learned a lot about software architecture, business / IT alignment and risk management. I can apply these concepts and see their place in the big picture.

The methodical approach of Henk Koning to conduct this literature survey was very challenging for me. Especially defining a clear, well scoped focus that I was passionate about. My usual approach to problems is very broad, keeping all options open and trying to see relations rather than to look at aspects in isolation.

Another very interesting method of Henk Koning was a creative approach to map conceptual models of researchers to other domains and put them in a different context. This brought forward new interesting ideas. It was interesting to see how much can be done with concepts and models of others when you truly open yourself up to them.

I enjoyed doing the literature survey tremendously. Beforehand I had some reservations about the idea of having to do this for six weeks in a row. But because the issue was so close at heart, I found it very interesting and enjoyable to learn what others thought about the subject.

Negative:
Finding a clear focus for the research was very hard. Time was short and it was difficult to distance myself of projects that were so very close at heart, that I had analyzed and discussed a thousand times. To do this required energy and room in my mind which I didn't have at the time. Fortunately Henk Koning came to the rescue and helped me finding my focus.

Time was short, during the research I constantly felt pressured. Especially when the first phase took twice as long as planned and the first weeks of the literature study did not show any promising results. At the same time I had to pay attention to important issues in my private life. Thanks to the clear direction and coaching of Henk Koning I managed to keep focused on the results and put the work in that was required. The prize of finishing the study more than compensates thirteen weeks of working in a row, with hardly any day off ☺.

Another frustrating issue was that it proved hard to find relevant sources. I examined dozens of papers and books to determine their relevancy. The processing of this information cost a lot of energy and made it hard to sleep at night. Another hard thing was that many sources discussed findings from a perspective that was different from mine. Grasping the underlying, implicit conceptual model took time and rereading.

During the first weeks of the literature study dozens of papers and books were superficially read to find relevant sources. The enormous amount of information made it hard to keep focused and it took a lot of energy to process and structure all information. Creative processes were unleashed and nights were haunted by phrases from different sources.

Reflection research approach
The research approach worked very well. Finding the clear focus that was close to my heart made sure the literature survey was very enjoyable and never became a drag. The only problem was the selection of cases that were so close to me that it was hard to take some distance. The step by step approach proved to be very useful and effective. This made all information good to manage and see the progress. Keeping explicit notes made it possible to tie conclusions and concepts to the right sources and to re-assess the search method.

Self assessment

Scale items: -, +/-, +

Quality of research result: 8 _

- \+ Sources were of good quality and were appropriate
- +/- Not sure whether all relevant sources were found
- \+ Got the intent of the author
- \+ Managed to combine and converge different streams of information
- \+ Research questions were answered conform recent insights
- +/- Help of Henk was required to get good result
- \+ Gained new insights.

Quality of essay: 8 _

- \+ Written in English
- \+ Well written considering time and amount of information
- \+ Size of paper is within boundaries
- \+ Size of results reflects complexity in literature
- +/- Found it hard to determine how thesis would be read and what information was sufficient

Difficulty of research question: 7 _

- \+ Research question addressed three fields: architecture, business/it alignment, risk management
- +/- If the research question had been scoped more to either vision or feasibility, more depth could have been achieved
- \+ Lot of time and effort was put into the research question to get a good focus and define it sharply. This was successful, and the rest of the research benefited tremendously.
- +/- Research questions focused on literature survey only. It would have been more challenging if I could have created something myself. However, this would have required a more narrow scope, as time was already very short.

Relevance of master: 8

- \+ Could use several books and papers studied in the masters
- \+ The course Architecture was the driving force behind the master's project
- \+ The research questions were "strategically aligned" ☺ with the masters, its mission being "How to improve software projects."
- \+ The feasibility study required a good overview over software engineering and the major problems, this was provided for in the masters.
- \- Research confirmed the importance of organizational factors for failure and success of projects, and showed several ways how to identify and mitigate these. This was not addressed in the master.

Appendix A: Bibliography

Elaborate bibliography

[Dikel et all, 2001] D. Dikel, D. Kane, J. Wilson, *Software architecture: Organizational principles and patterns,* 2001, ISBN 0-13-029032-7, published by Prentice Hall, 281 pages

Getting value from software architecture is not just a matter of technology. Success often depends on organizational factors. However as architectural issues maybe obscure to executives so are organizational aspects to IT architects and IT practitioners. This book describes five interrelated principles that help align organization and IT. These principles are Vision, Rhythm, Anticipation, Partners and Simplification (VRAPS). The book not only defines these principles and provides explanatory cases but also describes related organizational (anti-) patterns and their effect on architecture and business.

Dikel goes beyond traditional Business / IT alignment. Dikel acknowledges organizational patterns that help or frustrate the success of architectural development. These patterns can provide a way to assess the capabilities of the organization to implement new technology. This can serve as instrument for the feasibility study.

How does it position Architecture?
Architecture enables companies to select the best mix of in-house development, and products and services available in the market place. This enables attractive competitive solutions that take advantage of strengths of many groups within and outside a company. This is the strategic importance of IT architecture, as it provides a basis for strategic partnerships.

Does literature confirm that a business case for ICT projects should include a vision and a feasibility study?
The business case as such is not mentioned in this book. However vision is seen as crucial to successful architecture. Thompson is quoted for his three step approach for projecting architecture vision:

1. Articulate compelling customer value clearly and concisely
2. Map the value to a small set of specific solvable problems
3. Translate these problems into a minimal set of constraints

Dikel goes on to define vision as: identifying and conveying a substantive connection between seemingly unrelated use cases that connect expected use of architecture with tangible goals of the user that can be satisfied. (Architectural) Vision is considered an important driver for make / buy decisions and selecting technology of other companies.

Dikel discusses feasibility superficially: "Impossible solutions are prevented by:

- Analyze and articulate risks, in particular analyze how proposed solution maps to the constraints
- Test the assumptions
- Verify if solution is already applied in practice"

Does literature confirm the distinguished responsibilities of an IT architect in making a business case?
"Architects develop the (architectural) vision by mapping tangible future value to constraints, by clarifying risks and by designing architecture so that it can remain.". Dikel quotes Thompson: "Being an architect is much more about understanding how to balance business, organizational dynamics and technology than it is about technical gears."

Furthermore Dikel stresses the importance of intense communication between business managers and architects. Dikel goes as far to say that the main quality of an architect is to be able to understand and share the business vision. The IT architect should be hired by the business managers, rather than by technical managers.

Does literature confirm the proposed viewpoints with respect to vision and feasibility?
Dikel does not describe any viewpoints.

[Gallivan, 2001] M. Gallivan, Organizational adoption and assimilation of complex technological innovations: Development and application of a new framework, The DATA BASE for advances in Information Systems – Summer 2001 (Vol. 32, No. 3); pp. 51-85; Retrievable via:. http://delivery.acm.org/10.1145/510000/506729/p51-gallivan.pdf?key1=506729&key2=1310008801&coll=portal&dl=ACM&CFID=23146367&CFTOKEN=66498591

A new hybrid framework is presented to explain innovation adoption and diffusion processes, occurring within an organizational context, driven by an authority figure. First the author reviews traditional models. Most focus on voluntarily adoption of individuals and are not suitable to explain the processes within organizations. The author makes clear that most organizational adoptions of innovations are in fact a two stage adoption process: the primary authority adoption decision (the go decision) and the secondary adoption and organizational assimilation process (implementation).

The focus of this paper is on the secondary adoption processes. The success of secondary adoption processes is for a large deal determined by managerial intervention (e.g. communication, training), by subjective norms (e.g. peers, clients) and by facilitating conditions (of the innovation, organization or individual). This influences the assimilation process for individuals for which the author uses the model of Cooper & Zmud: (initiation, adoption, adaptation, acceptance, routinization and infusion). The author also distinguishes eight themes (organizational "patterns"), emerging from field studies, that help or frustrate adoption. The model and themes can be used in a feasibility study to predict level of success and failure.

How does it position Architecture?
This paper does not talk about architecture.

Does literature confirm that a business case for ICT projects should include a vision and a feasibility study?
This paper assumes that the decision has been made to use the new technology and does not go into detail about it.

Does literature confirm the distinguished responsibilities of an architect in making a business case when new technology is involved?
Not applicable

Does literature confirm the proposed viewpoints with respect to vision and feasibility?
The identified factors determining success of the implementation can be taken into account during the business case and should be part of the vision and be analyzed during the feasibility study. Also the organizational themes and their impact on the implementation can be of use when determining the feasibility.

[Gordijn, 2002] J. Gordijn, Value-based Requirements Engineering: exploring innovative e-commerce ideas, 292 pages, 2002, SIKS Dissertation Series No. 2002-8

This book presents an approach to explore an innovative e-commerce idea with the aim to understand it thoroughly and to evaluate it for potential profitability. The approach is based on a model for requirements engineering. The approach is lightweight, enabling idea exploration in one or two weeks. Graphical conceptual modeling is applied, enabling evaluation and common understanding. Three viewpoints are used in this model, a value viewpoint, a business process viewpoint and the information systems viewpoint. Scenarios are used to capture a value proposition, to gain common understanding and to evaluate an e-commerce idea. At last the approach is economic value aware. The model is illustrated based on a project carried out in the field of online news article provisioning.

Gordijn has defined a process, instruments (viewpoints, scenarios) and a value ontology for the exploration of ideas to establish their feasibility from a profitability perspective and from a technical perspective. Gordijn offers guidelines to construct and validate models.

Gordijn's mental model for establishing profitability of the idea is the value chain. For all participants in the value chain, including the end customer, the profitability of different operational scenarios is determined. This enables the prediction of customer behavior in different scenarios.

Gordijn sees two important goals to use models for idea exploration:

1. Enhancing agreement and common understanding amongst a wide group of stakeholders
2. Enabling validation in term of evaluating economic feasibility.

Like Smolander, Gordijn uses models as a means to determine not only the solution but also to understand and gather the requirements (explore the idea).

.

How does it position Architecture?

Architectural techniques and models enable the exploration and validation of ideas. Complexity is reduced by isolating different concerns in separate viewpoints. Technical and financial feasibility is the focus of validation.

Does literature confirm that a business case for ICT projects should include a vision and a feasibility study?

Gordijn's approach to idea exploration is very suitable for making a business case. An idea is developed into a vision, modeled and then validated. The validation is focusing on technical and commercial feasibility. Gordijn focuses on radical change, from no organization to a new organization.

Does literature confirm the distinguished responsibilities of an IT architect in making a business case?

Gordijn sees the role of an IT architect in engineering the information system viewpoint, establishing the costs and technical feasibility. The business oriented viewpoints are engineered by business developers and business process (re)designers.

Does literature confirm the proposed viewpoints with respect to vision and feasibility?

Gordijn makes a strong case for a limited set (no more than six) of predefined viewpoints eliminating the time consuming step to identify appropriate viewpoints. Gordijn distinguishes three viewpoints: a value viewpoint, a business process viewpoint and the information systems viewpoint. The viewpoints of Gordijn are organized differently then the viewpoints proposed by us. The viewpoints focus most of all on economic feasibility. Technical feasibility is also mentioned as a goal, but this is done implicitly. By designing the model it becomes clear if it is doable or not.

In the value viewpoint Gordijn shows how value is created for the different stakeholders. In the business process viewpoint and information systems viewpoint the costs and technical / operational feasibility is addressed.

[Wagter et all, 2001] R. Wagter, M. van den Berg, J. Luijpers, M. van Steenbergen, *DYA*, 2001, ISBN: 90-72194-62-4, published by Tutein Nolthenius, 201 pages

Speed and cohesion are crucial to gain a competitive edge. Architecture can be used to achieve these qualities by aligning business and IT. [Wagter et all, 2001] sees architecture as an instrument to manage the IT organization. [Wagter et all, 2001] acknowledges that companies find it hard to incorporate the concept of working under architecture, one of the reasons being the tendency of architectural processes to become bureaucratic. When architects are unable to find a balance between speed and cohesion, aligned to the priorities of the organization, architecture might be perceived as a limitation rather than an asset and become obsolete. [Wagter et all, 2001] introduces a number of principles and the following three key architectural processes:

- The strategic dialogue is about aligning business and ICT, culminating in business cases.
- Architectural services are the structural architectural processes that support both the strategic dialogue and the development process. These processes are about providing and developing models and regulations. The main principle underlying these processes is "just enough, just in time", enabling dynamic processes.
- Development under / without architecture is about developing the ICT solution focusing on the entire life cycle. In order to meet time lines required by business, offensive and defensive scenarios are distinguished in which architectural principles are temporarily ignored.

The book describes a number of organizational patterns that frustrate alignment between business and ICT and the adoption of working under architecture. The authors argue that an organization needs to acquire new patterns, with a central role for architecture. This is only an option if architecture really contributes and is not perceived as an obstacle. To this effect the IT architects needs to share the

business vision and not just focus on IT strategy; work under the principle just enough / just in time; when necessary (temporarily) neglect architectural principles in order to satisfy more pressing business needs; and have a controlling, and stimulating role in development.

How does it position Architecture?
[Wagter et all, 2001] sees architecture as an instrument to manage the IT organization and maintain cohesion between processes, organization and IT.

Does literature confirm that a business case for ICT projects should include a vision and a feasibility study?
[Wagter et all, 2001] sees the strategic dialogue between business management and architects as one of the key architectural processes. This process focuses on producing a business case. Creating a joint vision by IT and business is essential in this process. Feasibility and risks are much less of an item. [Wagter et all, 2001] is more concerned that architects are perceived as an obstacle who are more looking for the impossibilities rather than the possibilities.

Does literature confirm the distinguished responsibilities of an IT architect in making a business case?
[Wagter et all, 2001] confirms the importance of the role of the IT architect and of architecture in the creation of a business case. The role of the IT architect is considered advisory and is not specified further.

Does literature confirm the proposed viewpoints with respect to vision and feasibility?
[Wagter et all, 2001] does not specify any architectural viewpoints.

[Remenyi, 1999] D. Remenyi, IT investment: making a business case, 1999, ISBN: 0-7506-4504-0, published by Butterworth-Heinemann, 210 pages

[Remenyi, 1999] argues that before doing any IT investments a business case should be made, emphasizing what IT investments can do to support more efficient and effective business process and practices. A well constructed business case can be an important tool to manage the IT investment. A professionally produced business case consists of:

1. Clearly expressed business objective and set of outcomes, expressed at different levels of abstraction. In this book macro (high level statement), meso (describes all benefits) and micro (numbers) level are distinguished.
2. List of stakeholders and beneficiaries, assessing their interests and position
3. Statement how the proposed IT expenditure will support the corporate strategy
4. Evaluation of the appropriateness of the technology and operational plan
5. Evaluation of risks

Remenyi puts the value of a business case in perspective. He quotes Keynes in saying that we have only limited ability to predict the future. The value of a business case is to make a grounded decision that will give adequate direction to the project and prevent a doomed project from being started.

Remenyi presents practical (traditional) models to ensure a functional and strategic fit between IT and business. Remenyi also focuses on value of the project, in order to predict whether this project is the most promising one from a financial perspective. Organizational aspects are indirectly considered using the stakeholder analysis.

Risks are part of the business case. Remenyi presents a risk framework, underlying the notion that risks usually don't occur isolated but in patterns. More interestingly are the six categories distinguished by McFarlan, referring to failures of conceptualization: poor ideas that were not likely to succeed from the beginning.

1. Failing to meet customer requirements.
2. Requiring behavior that is not ingrained in existing users.
3. Incapable of evolving
4. Disrupting organization abilities
5. Lowering entry barriers for competitors
6. Undertaken before all tactical and strategic resources are in place.

How does it position Architecture?
Remenyi does not mention architecture in his book except in his 3 x 3 risk framework. Remenyi perceives architecture here most of all as addressing the technological properties.

Does literature confirm that a business case for ICT projects should include a vision and a feasibility study?
Remenyi is a strong advocate of a business case before the start of a project. His comprehensive business case has a somewhat different organization than the proposed viewpoints but addresses a many of the aspects mentioned in the proposed viewpoints.

Does literature confirm the distinguished responsibilities of an IT architect in making a business case?
Remenyi does not mention the role of an IT architect.

Does literature confirm the proposed viewpoints with respect to vision and feasibility?
Remenyi uses lots of models, however does not use viewpoints.

Standard bibliography Architecture in general

[Bass et all, 2003] L. Bass, P. Clements, R. Kazman, *Software architecture in practice,* April 2003, second edition, ISBN 0-321-15495-9, published by Addison-Wesley, 528 pages

[Bass et all, 2003] positions architecture as a means to design large information systems. Architecture provides a foundation for the system to be made, ensuring that it satisfies functional and quality requirements. [Bass et all, 2003] also sees "business qualities", like time to market, cost and benefits, and system lifetime as aspects that can be achieved through architecture. When designing the architecture conflicts between requirements arise and can be communicated. In this way architecture functions as tool for decision making and communication between stakeholders. [Bass et all, 2003] discusses all architectural instruments and issues to ensure proper architectural design. These instruments and issues include scenarios to capture requirements, design tactics, architectural patterns, views, evaluation methods like ATAM and CBAM, methods and issues to select and incorporate COTS components and issues concerning software product lines.

[Bernhard et all, 2002] P. Bernhard, M. Brattinga, J. Campschroer, W. Hordijk, A. Ligthart, R. Steetskamp, R. Verver, J. Vis, *Applicatieontwikkeling onder architectuur*, 2002, ISBN: 90-440-0667-3, published by Ten Hagen Stam, 206 pages

Application development under architecture gives an overview of architecture, based on experience of the authors. The book describes the need for architecture, elaborates some definitions of architecture, presents an architectural framework and then focuses on how architecture is used in developing applications.

[Florijn et all, 2003] V. Clerc, J. van Ekris, G. Florijn, H. Koning, G. Leih, M. Maat, F. Niessink, *Software architectuur – overzicht en compendium*, 2003, ISBN: 90-440-0752-1, published by Ten Hagen Stam, 233 pages

This book presents an overview of the backgrounds and developments in IT architecture focusing on Software Architecture. The first part of the book contains a global overview, the second part is a compendium of terms from IT architecture.

[Hofmeister et all, 2000] C. Hofmeister, R. Nord, D. Soni, *Applied Software Architecture,* 2000, ISBN 201-32571-3, published by Addison-Wesley, 397 pages

Designing a large software system requires juggling differing perspectives and goals, and evaluating differing options. Applied Software Architecture provides practical guidelines and techniques for producing software designs. It gives an overview of software architecture basics and a detailed guide to architecture design tasks, focusing on four fundamental views of architecture: conceptual, module, execution, and code. Four real-life case studies reveal the insights and best practices of skilled software architects in designing software architecture.

[IEEE 1471, 2000] *IEEE Recommended practice for architectural description of software intensive systems*, 20 pages.

> The earliest period of decision making and evaluation is increasingly referred to as the architectural level of systems development. At this moment there are no non-ambiguous definitions for terms used in the field of architecture. This document contains a consistent set of definitions for architectural concepts. It presents a conceptual model that relates several concepts used to describe architectures. These concepts include views, viewpoints, concerns, stakeholders.
>
> An architectural description is organized in views. A view represents the system from the perspective of a related set of concerns. Concerns are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders. Views adhere to a viewpoint, containing a specification of the conventions for constructing and using a view, a pattern or template from which to develop individual views by establishing the purposes and audience for a view and techniques for its creation and analysis.
>
> Architecting is concerned with developing satisfactory and feasible system concepts. It can be used to predict the fitness for use of a system. Other uses of architectural descriptions include:
> - Expression of system and its evolution
> - Communication among system stakeholders
> - Evaluation and comparison of architectures in consistent manner
> - Planning, managing and executing activities of systems development

[Smolander, 2003] K. Smolander, *On the Role of Architecture in Systems Development,* 2003, ISBN 951-764-735-2, published by Lappeenranta teknillinen yliopisto, 97 pages

> This thesis provides empirical results for the strong role of software and system architecture in the functioning of a systems development organization. Important findings are that selection and forming of the viewpoint model depends largely on the situation at hand related to business, technology, and organization. Further empirical research shows that architectural descriptions are used not only as basis for further design and implementation, but also as means for communication, interpretation and decision-making. Architecture design can be regarded primarily as a means for coping with complex solutions and technology, reached through communication between diverse stakeholders with varying skills and experience. Architecture development can be characterized as a process seeking boundaries finding consensus, and identifying commonalities across organizational borders.

Standard bibliography Business case in IT projects

[Gordijn et all, 2001] J. Gordijn, H. de Bruin, H. Akkermans, *Scenario Methods for Viewpoint Integration in e-Business Requirements Engineering*, proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 7, p.7032, January 03-06, 2001 10 pages. Retrievable via: http://www.hicss.hawaii.edu/HICSS_34/PDFs/INSDM03.pdf

.

> This short paper stresses the importance to first understand the profitability and feasibility of new e-Business ideas on a high level before detailing the requirements of the required system. To this purpose three viewpoints are used:
> - A business value viewpoint modeling the business in a value oriented way
> - A business process viewpoint showing the operational processes
> - A system architecture viewpoint demonstrating the feasibility of the high level requirements
>
> Scenarios are used to integrate the viewpoints and can be used in an iterative process to further detail requirements and design and model the system. An extension of UCM is used as notation for the scenarios.

[Holcolmb et all, 2000] L. Holcolmb, R. Thomas, M. Tiemann, *Architecture alignment and assessment guide*, CIO Council NASA, October 2000, 39 pages; Retrievable via: http://www.cio.gov/archive/arch_align_assess_oct_2000.pdf.

> To make sure that government agencies select new IT technology based on business needs, rather than because it is available, new regulations were put in place. This paper presents a validated approach to integrate enterprise architecture and IT capital planning, satisfying these regulations. The tools and

techniques presented are collectively called architecture alignment and assessment. The paper defines enterprise architecture and provides a three stage phase: selection, control and evaluation.

[Kruchten, 2000] Philippe Kruchten, *The Rational Unified Process: An Introduction,* 2000, ISBN 201-70710-1, published by Addison-Wesley, 278 pages

This book describes the Rational Unified Process. The process is designed to tackle common problems when developing software. Six best practices are combined in a detailed, manageable software process. At the heart of these practices are Use Cases as representation of requirements and UML as modeling language. Support is provided in the form of training, templates, guidelines, examples, descriptions and tools. RUP can be tailored or extended to fit the needs of specific organizations. The six best practices are: iterative software development, requirement management, component-based architectures, visual software modelling, continuous verification of software quality and control of changes to software.

[OGC, 2004] http://www.ogc.gov.uk

The Office of Government Commerce (OGC) is an independent Office of the Treasury. It is responsible for a wide-ranging program which focuses on improving the efficiency and effectiveness of central civil Government procurement. The OGC presents best practices for project and program management including the field of software development and selection of IT systems. The site shows a mature risk and value management approach and has clearly identified the business case as important decision moment. The following table is used to define the business case.

Business case perspectives	**What to check**	**Who provides specialist input**
Strategic fit	Business need and contribution to the strategy	Corporate planning, policy makers, business analysts, technical experts
Options appraisal	Value for money, options, costs, benefits, risks, dependencies	Finance, departmental economists, procurement
Achievability	Internal capability, skills and resource; assessment of departmental/supplier capability and understanding of the project; implementation plan	HR, project or program management offices, change management office, procurement advisers
Commercial aspects	Marketability, commercially sound approach to the potential deal, robust procurement strategy	Purchasing, procurement, commercial advisers, HR
Affordability	Availability of funding	Finance, corporate planning

Standard bibliography Business / IT alignment

[Aerts et all, 2003] A. Aerts, J. Goossenaerts, J. Wortmann, D. Hammer, *Architectures in context: on the evolution of business, application software, and ICT platform architectures*, pages 781 – 794, Information Management, Elsevier. Retrievable via: http://tmitwww.tm.tue.nl/research/icta/documents/AertsGoossenaertsHammerWortmann-IandM.pdf

This paper distinguishes between the business domain, the application software domain and the ICT platform domain. It analyses historical developments and shows that they experienced parallel developments. The parallelism can be explained by mutual influence and alignment. Six radical changes are distinguished in recent history, each resulting in a new dominant design and intensified integration between the three domains. The changes have led to increased complexity in all domains. To respond to changes adequately, architecture is required to ensure alignment between the domains.

[Heemstra et all, 2001] F. Heemstra, R. Kusters, J. Trienekens, *Softwarekwaliteit*, ISBN 90 440 0241 4, January 2001, published by Ten Hagen & Stam, 342 pages;

[Heemstra et all, 2001] combines a comprehensive set of the latest and most relevant theories and research results to answer the following four questions:

1. What is meant by quality of software
2. How is determined what this quality should be
3. Which factors influence quality of software
4. How these factors can be influenced.

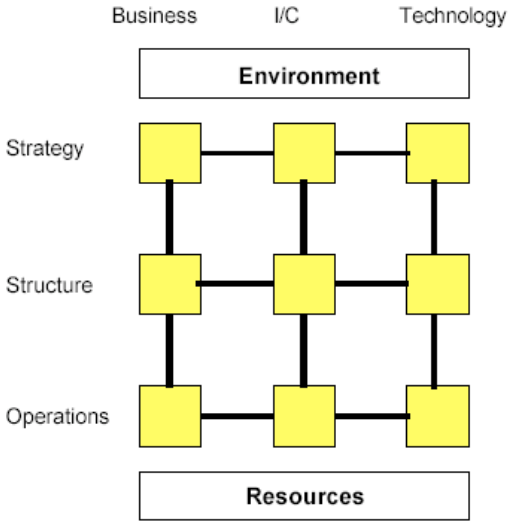
They find that quality can be increased by defining clear goals, continuous improvement, evaluation and measuring and by creating the right conditions. Quality of software is influenced by process, resources and (organizational) context.

[Henderson et all, 1993] J. Henderson, N. Venkatraman, *Strategic alignment: Leveraging information technology for transforming organizations*, pages 4 – 16, IBM Systems Journal, Vol 32, 1993. Retrievable via: http://www.research.ibm.com/journal/sj/382/henderson.pdf.

The strategic alignment model is presented, offering a framework to understand the potential of IT in tomorrow's organization. The framework distinguishes four domains of strategic choice: business strategy, information technology strategy, organizational infrastructure and processes, and information technology infrastructure and processes. The concept of strategic alignment is based on two building blocks: strategic fit (the interrelationship between external and internal components) and functional integration (integration between business and functional domains). As driving forces for alignment, Henderson and Venkatraman denote business strategy as driver, and IT strategy as enabler, both directly from a strategic perspective or indirectly through operational processes. All these alignment scenarios require different management practice.

[Maes, 1999] R. Maes, Reconsidering information management through a generic framework, 27 pages, PrimaVera Working Paper 99-15. Retrievable via: http://imwww.fee.uva.nl/~maestro/PDF/99-15.pdf.

Maes presents a generic framework for information management, that can be used for the definition, investigation and positioning of information management. This model builds on the strategic alignment model proposed in [Henderson et all, 1993], but adds a column by splitting up IT in I/C and Technology, substantiated by the differences in nature. Also a row is added by splitting up operations in structure and operations.



Maes tests the framework by applying different concepts and matching it with requirements for frameworks. He concludes that the framework holds and can be used for further research.

[Wieringa et all, 2002] R. Wieringa, H. Blanken, M. Fokkinga, P. Grefen, *Aligning application architecture to the business context*, Proceedings 15th International Conference on Advanced Information Systems Engineering (CAiSE03); pp. 209-225; Klagenfurt/Velden, Austria, 16-20 June 2003. Retrievable via: http://wwwhome.cs.utwente.nl/~fokkinga/mmf2002f.pdf.

Alignment of application architecture to business architecture is a key problem in designing, acquiring and implementing information systems. This paper elaborates the framework of [Henderson et all, 1993] to analyze the alignment problem. Based on this framework a design approach is presented. Architectural guidelines are derived from this framework and a design approach is presented that shows alignment between Business and IT.

[Wieringa et all, 2003] R. Wieringa, H. Blanken, P. van Eck, *Project Graal: towards operational architecture alignment*, presented at Landelijk Architectuur Congres 2003, 12 pages, Retrievable via: http://is.cs.utwente.nl/GRAAL/vaneck_etal_lac2003_abstract.pdf.

The paper presents a framework for architecture alignment that can be positioned between approaches for software architecture and strategic alignment models. The Graal model distinguishes five service layers: business environment, business processes, application systems, implementation platform, and physical network. These layers are described at different levels of refinement from abstract (maps to strategic) to detailed. Case studies show that design decisions at the application level and the business process level are motivated in terms of how these contribute to the business mission. Software platform and physical network infrastructure were motivated by reference of other projects carried out concurrently, by market developments and by contingencies.

[Wieringa et all, 2004] R. Wieringa, H. Blanken, P. van Eck, *Architecture alignment in a large government organization: a case study*, Proceedings of the CAiSE Forum, CAiSE, June 7 - 11, 2004, Riga, Latvia, 10 pages, Retrievable via: http://wwwhome.cs.utwente.nl/~patveck/papers/wieringa_etal_caise04forum.pdf.

> IT architecture is viewed as the structures present in the entire information technology support used by an organization. A case study of an operational IT architecture process is reported, focusing on the relationship between IT architecture and business context. Major findings are that application architecture is designed by aligning applications to the business process structure and that IT infrastructure architecture is designed by aligning it to technological trends rather than to business goals and problems. Application alignment usually starts with designing business processes based on business strategy. These business processes are the basis for architecture of the entire application layer, which in turn is the basis for the architecture of each individual application. Infrastructure design depends on overall business goals, current problems, current systems and current technology trends, but it has the tendency to focus on technological rather than business considerations.

Standard bibliography Feasibility and risks in IT projects

[Addison et all, 2002] T. Addison, S. Vallabh, *Controlling software project risks – an empirical study of methods used by experienced project managers*, Proceedings of SAICSIT 2002, pages 128 - 140; Retrievable via: http://portal.acm.org/citation.cfm?id=581525&dl=ACM&coll=portal.

> This paper reports significant risks and the controls utilized to reduce the occurrence of these risks or minimize their impact. Risk factors were identified in literature. During an empirical study their importance and frequency of occurrence was determined. Also the activities of project managers to control the risks were identified. 14 risks were researched. The project managers ranked unclear or misunderstood scope / objectives as the most important risks, followed by misunderstanding requirements and failure to gain user involvement. From the 14 identified controls, the most frequently applied were clearly assigning responsibilities to team members, developing and adhering to project plan and involving management.

[Carr et all, 1993] M. Carr, S. Konda, I. Monarch, F. Ulrich, C. Walker, *Taxonomy-Based risk identification*, 94 pages, Technical Report CMU/SEI-93-TR-6, ESC-TR-93-183. June 1993. Retrievable via: http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf.

> This report presents a risk framework used to construct a questionnaire to identify risks. The use of a questionnaire to identify risks is based on the assumption that software development risks are generally known by the staff and can be uncovered by a structural method covering all key development and support areas of the project.
>
> The risk framework, called software development risk taxonomy, is organized in three major classes: product engineering (the technical aspects), development environment (methods, procedures, tools) and program constraints (contractual, organizational and operational factors). These classes are further defined in elements, characterized by attributes. As an example: Requirements is an element of the class Product Engineering, one of the characteristics of Requirements is Stability. For Stability of Requirements the following questions are part of the questionnaire:
>
> - Are the requirements stable?
> - Are the external interfaces changing?

[Keil et all, 1998] M. Keil, P. Cule, K. Lyytinen, R. Schmidt, *A framework for identifying software project risks*, Communications of the ACM, November 1998/ Vol. 41. No. 11 pages 76 - 83; Retrievable via: http://portal.acm.org/citation.cfm?id=287843&dl=ACM&coll=GUIDE&CFID=23150144&CFTOKEN=97822689.

> Three panels of project managers were assembled in different parts of the worlds to identify risk factors. Eleven risks were commonly identified, the most important ones being lack of top management commitment, failure to gain user commitment and misunderstanding requirements. The introduction of new technology was not considered a top risk. One explanation was that these risks were fully understood and provisioned for in the project plan. The risks are categorized in a framework matching perceived importance of risks with perceived level of control by project managers. Resulting in categories: customer mandate, scope and requirements, environment and execution. For all these categories risk strategies are provided.

[Moynihan, 1997] T. Moynihan, *How experienced project managers assess risk*, pages 35 – 41, IEEE Software, 0740-7459/97. Retrievable via: http://csdl.computer.org/dl/mags/so/1997/03/s3035.pdf

In [Moynihan, 1997] 14 experienced application systems developers are interviewed to see if their experiences about software project risks matched with the findings in literature. The focus of the research was the factors considered by the project managers when planning new development projects. These factors were matched against two sources in literate:

- Barki, Rivard and Talbot build a comprehensive inventory of variables related to software development risk, based on a wide review of literature.
- The SEI taxonomy-based risk identification instrument

After the study the following conclusions were drawn:

1. The experiences of the project managers reflect most of the risk factors identified in literature
2. For one and the same risk concept different project managers considered facets important. Though the differences appeared subtle, these were distinct and important.
3. The project managers identified risk themes that were not present in literature. This could be explained by differences in context.
4. The risk sources in literature named several risk areas that were not mentioned by the project managers. In the case of the SEI list, this were mainly the more technical risk. This could be explained by the context of the project managers, who operated in areas with a relative low technical complexity.

[Pandelios et all, 1999] G. Pandelios, S. Behrens, R. Murphey, R. Williams, W. Wilson, *Software Risk Evaluation (SRE) Team Member's notebook (Version 2.0)* , 164 pages, Technical Report CMU/SEI-99-TR-029, ESC-TR-99-029, december 1999. Retrievable via: http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr029-app.pdf.

This report presents a process description of how to manage risks. This process consists of the phases: Risk Identification & Analysis, Interim Report and Mitigation Strategy Planning. The questionnaire presented in [Carr et all, 1993] is used as basis for Risk Identification. Interesting aspects are:

- Only that part of the Risk Questionnaire is used that fits the project profile
- Risks are scored in a risk matrix matching impact with probability
- Adequate attention is paid to consolidation and presentation of the risk information
- Time is spent to analyze the root cause of the risks, assuming that there may be causes resulting in many of the risks and the identification of these sources and the mitigation of them is crucial.
- The method is very thorough and takes roughly ten days.

[The Standish Group, 1995] The Standish Group. Chaos. 1-8 pages. Retrievable via: http://www.projectsmart.co.uk/docs/chaos_report.pdf

The focus of the latest research project at The Standish Group has been to identify:

- The scope of software project failures
- The major factors that cause software projects to fail
- The key ingredients that can reduce project failures

The total sample size was 365 respondents and represented 8,380 applications. The figures for failure were equally disheartening in companies of all sizes. Only 9% of projects in large companies were successful. At 16.2% and 28% respectively, medium and small companies were somewhat more successful. Currently, the 365 companies have a combined 3,682 applications under development. Only 431 or 12% of these projects are on-time and on-budget.

Opinions about why projects are impaired and ultimately cancelled: ranked incomplete requirements and lack of user involvement at the top of the list.

A framework for managing software projects and making them a success is a holy grail. Research at The Standish Group also indicates that smaller time frames, with delivery of software components early and often, will increase the success rate. Shorter time frames result in an iterative process of design, prototype, develop, test, and deploy small elements.

Appendix B: Case E-learning

This is the description of the third case. This case started with a business case with a clear vision and good feasibility study. Therefore this case was not used determining the research questions. However in retrospect the following observation can be made. The approach towards this project was fine from a technical perspective. However the organization needed quick success and could not cope with uncertainties, and the time that was required to mitigate risks. Had this been considered during the go / no go moment, perhaps the project was not started. Or by explicitly discussing the project "rhythm", the expectations of the management board could have been managed better giving the project more room to evolve.

Project info

Period: 1/4/2002 – 1/9/2003
Refers to: Phased implementation of e-learning
Size: for approx 3.000 employees

Synopsis

The organization is a large call centre organization that works for other companies. It has approximately 6.000 employees world wide. The projects vary a lot, as does the training required for these projects. The business needs for training are pretty clear: get employees as fast and effective as possible through the training; develop and change a course quickly; focus on knowledge and general rules; develop skills during work.

Because the unpredictable number of trainings, problems in the training department, and customers demanding higher quality; the organization was eager to find ways to reduce training time, and increase flexibility while increasing quality. E-learning was considered a possible solution.

To determine if E-learning was suitable, a business case was made; the characteristics of the organization were listed, and known benefits of E-learning were checked to see if they applied to the organization. The research was hampered by the fact that E-learning had not yet been implemented by a similar company. But the organization had always been an early adaptor and did not consider this to be a show stopper.

A thorough research gave a nuanced picture. Based on available data E-learning could yield benefits but risks were considerable and investments would take some years to profit. To learn as quickly as possible if the risks could be overcome, it was determined to out source the pilots. This way seasoned e-learning companies could show that e-learning would pay-off within the organization.

Real customer projects were selected for the pilot. The e-learning implementers could use the training budget of these projects to demonstrate they could do it cheaper while maintaining the level of quality. To establish if the level of quality was at least the same, a group of people were trained traditionally and a group of people were trained with E-learning. The people were then tested and the results were compared.

During the project, a large reorganization took place. The managing director had to show his projects contributing to the reorganization goals. E-learning was one of these projects. During the project the managing board was changed and the project fell under another manager. This manager had a different and more optimistic view on e-learning and was not so much interested in reducing risks but focused on implementing e-learning as quickly as possible. When the early pilots demonstrated that e-learning was not as profitable as foreseen, this outcome was not gladly accepted. The results were contributed to bad partners and poor implementation. After some new changes in management the project was finally derailed.

Analysis of decision moments

In this project a couple of very interesting phenomena took place, that troubled the outcome.

Solution pushers

Implementing E-learning had sponsors who were biased in favor of e-learning. They considered it to be the solution. They were influencing the decision process and pressuring the implementation. They ignored possible risks and considered other sounds as opposition. When e-learning proofed to be not the solution, they considered this as a threat to their position. Stakeholder interests are not just functional but also personal and political.

Hiding information

For a proper decision the right information is required. In this case it required a good understanding of both e-learning as well as the business specifics. Given the time limit it was considered impossible for the project team to acquire the appropriate knowledge of e-learning. And because the interest of suppliers was to get e-learning implemented within the organization their information was not objective. An approach was chosen to put the risk (and benefits) with the supplier. This was a good solution with respect to putting responsibilities at the right place, but took some time to flourish and demanded skills from the suppliers they did not have. In the end it turned out that the suppliers had been to optimistic and that E-learning did not yield the benefits initially perceived.

Shifting focus and limiting possibilities

The decision about e-learning was considered of major tactical importance. The decision involved many persons on level of the management board. Most of them had no experience with technology, nor with implementation of technology. At first the focus of the management board was to make a go / no go decision based on a business case. After this decision, they shifted the focus to getting it done. However logical and normal, this is potentially dangerous. First the premise is wrong, namely that the management board is in control. Technological solutions have their limitations. They show fully during and after implementation. They can't be controlled by the management board. If it becomes clear that the technological solution does not solve the initial problem, the process should be stopped. But this is hard to communicate and accept when the train has started rolling.

The second drawback is that when the push is on getting it done, it refers to implementation of the tool and not so much on getting the initial problem solved. The initial problem is lost out of sight, overshadowed by implementation problems. And maybe after many years, if someone has an interest, it is found that the implementation was successful in the sense that e-learning was implemented but not successful in solving the initial problem.