Validation of Public Sector IT Architecture

July 11, 2012
Validation of Public Sector IT Architecture
(Validatie van ICT-architectuur in de publieke sector)

Open Universiteit in the Netherlands, School of Computer Science
Master thesis Computer Science

July 11, 2012

Chairman: Dr. Ir. Karel Lemmen, Open Universiteit in the Netherlands
Supervisor: Drs. Math Dicker, Open Universiteit in the Netherlands

T76318
Contents

Preface v
Abstract (English) vii
Samenvatting (Dutch) ix
1. Introduction 1
2. Problem definition and research plan 3
3. What is (good) architecture? 7
  3.1 Three visions on the quality of architecture 8
    3.1.1 Vitruvius 9
    3.1.2 Christopher Alexander 23
    3.1.3 James Scott 32
    3.1.4 Conclusion: architecture and its key quality aspects 36
    3.1.5 Conclusion with regard to the first and second research questions 43
  3.2 Operationalising the quality of IT architecture 45
    3.2.1 Quality in modern literature on IT and systems architecting 45
    3.2.2 Synthesis of the operationalisations found 55
    3.2.3 Conclusion: good architecture operationalised 59
    3.2.4 Conclusion with regard to the third and fourth research question 59
4. Conceiving the Architecture Validation Framework 61
  4.1 The outline for the framework 61
    4.1.1 The Gateway review method as an example for the framework 62
    4.1.2 Different contexts for different architects 63
    4.1.3 The outline 64
  4.2 Filling in the framework 64
  4.3 The manual 64
  4.4 The result: the scorecard IT Architecting 64
  4.5 The framework in the perspective of the literature: a reflection 65
5. Testing the Architecture Validation Framework 67
  5.1 The judgement asked of the experts 67
  5.2 Key quality aspects and operationalisations judged 67
  5.3 The scorecard judged 69
6. Conclusion 71
  6.1 Can we establish the quality of architecting? 71
  6.2 Can we improve the quality of architecting? 72
  6.3 Further research 72
Bibliography 73
Appendix 1. The scorecard IT Architecting 77
Preface

I've worked with information technology issues in the public sector for quite a few years now; it is my field of expertise. As I became gradually more and more involved in information technology applications I had nothing to rely on but my degree in Law, amateur experience in computer programming, and—I think—some common sense. I was frequently amazed at the difficulties and chaos that could arise if a seemingly simple good idea relying on IT had to be put in practice.

I decided to follow a few courses in the Computer Science curriculum of the Open University to better come to grips with the phenomenon. And then I took a few courses more, and so on. Having come to the end of the program, I had to choose a subject for a thesis. From both a theoretical standpoint and from my professional experience, I had come to the conclusion that the concept of architecture expresses best what is needed to improve the situation. I was not alone in that; the Dutch Court of Audit had reached the same conclusion. Unfortunately, there seemed to be considerable confusion about what it is: IT architecture. I therefore decided to dedicate my thesis to IT architecture.
Abstract (English)

The aim of this research is to improve the role of IT architecture in large public sector projects. This must prevent the frequent failures and overruns in both time and money that characterize these projects.

This research project consisted of the design of a solution for this problem based on an analysis and literature study. A prototype was constructed and judged by a team of experts.

The research was fundamental in nature in that we tried to grasp the essence of the concept of architecture, something that has existed for over 2000 years. The literature we chose did not focus on IT but rather the architecture (theory) for the building sector. The findings were compared with modern literature about IT architecture and also systems architecting beyond IT.

The prototype we designed, named ‘Architecture Validation Framework’, can be considered as a first version based on seven Key Quality Aspects of IT architecture that were identified and further operationalised.

A characteristic of the framework is that our literature findings motivated us not to construct a test for ‘the architecture’ as a finished set of documents, but a test focussing on the architecting process. We found that the earliest phase, when the commissioner is orienting himself regarding the possibilities, is of crucial importance. The framework is designed to be used in a way similar to Gateway reviews for projects. The framework is therefore not based on lengthy and waterproof audit methodology, but rather provides a quick scan on essential conditions for a successful architecting process. This scan must be done for the first time as early as possible and should be repeated at important moments in the development of the project, or as soon as doubts about the architecting process appear.

The experts who judged the framework unanimously concluded that the approach we chose and the Key Quality Aspects as identified including the operationalisations were valid considering the research aim. Their most important recommendation was to see the framework and accompanying document used not only as a basis for a review, but also as a document that can give commissioners a better insight into the conditions that must be fulfilled in order to make architecting successful, thereby making their role and their project successful.
Samenvatting (Dutch)

Dit onderzoek beoogt IT architectuur een belangrijkere en productievere rol te geven in grote projecten in de publieke sfeer. Daarmee moeten de frequentie mislukkingen en overschrijdingen in tijd en geld die deze projecten kenmerken worden teruggedrongen.

Het onderzoek was een ontwerponderzoek waarbij op grond van een probleemanalyse en literatuur een ontwerp is opgesteld voor een oplossing. Een prototype is gebouwd en beoordeeld door experts.

Het onderzoek had een fundamenteel karakter in die zin dat getracht is de essentie van het begrip architectuur te duiden, zoals dat al meer dan 2000 jaar bekend is. De gekozen literatuur betrof dan ook niet zozeer de recente IT literatuur, maar vooral literatuur over architectuur(theorie) in de bouwwereld. De bevindingen hieruit zijn geconfronteerd met moderne literatuur over IT architectuur en ook systems engineering, breder dan IT.

Het ontworpen prototype, getiteld Architectuur Validatie Raamwerk, kan beschouwd worden als een eerste opzet waarin een zevental sleutelkwaliteitsaspecten van IT architectuur zijn geduid en voorzien van operationalisaties.

Kenmerkend voor het Raamwerk is dat op grond van de studie van de literatuur niet gekozen is voor een beoordelingskader voor ‘de architectuur’ zoals die door een project als document wordt opgeleverd, maar een beoordelingskader voor het architectuurproces. Daarbij bleek met name de allervroegste fase, waarin de opdrachtgever zich oriënteert, van cruciaal belang. Het Raamwerk is ontworpen om te worden gebruikt op dezelfde wijze als waarop zgn. Gateway reviews op projecten worden gedaan. Het gaat niet om een langdurige en methodisch waterdichte audit, maar om een snelle scan op essentiële condities voor een succesvol architectuurtraject zoals die kan worden uitgevoerd door ervaren deskundigen. Die scan moet zo vroeg als mogelijk worden uitgevoerd en regelmatig herhaald op belangrijke momenten in het project of op het moment dat er twijfels ontstaan over het architectuurproces.

De experts die het Raamwerk hebben beoordeeld kwamen unaniem tot de conclusie dat de gekozen aanpak, sleutelkwaliteitsaspecten en operationalisaties goed gekozen waren gelet op het beoogde doel. Hun belangrijkste aanbeveling was om het Raamwerk en de bijbehorende toelichting niet alleen te zien als basis voor een review maar om het document te gebruiken om opdrachtgevers inzicht te verschaffen in de eisen die vervuld moeten zijn om architectuur tot een succes te maken en daarmee hun rol en hun project tot een succes te maken.
1. Introduction

In 1999 the Dutch government presented a new policy concerning IT aiming at more coherence in the use of IT for public purposes. Several years later, in 2006, the Court of Audit of the Netherlands published a report promoting IT Governance, indicating that the coherence sought in 1999 was still far from perfect. The Court of Audit recommended investing in Enterprise Architecture in order to improve coherence.

Problems remained, and in 2007 the Dutch parliament decided to ask the Court of Audit to further investigate the most important causes of problems in large IT projects commissioned by the state. The Court of Audit published its report in 2007, indicating that the main cause of the many failures is that projects tend to be too ambitious and too complex. Subsequent research by the Court into the possible remedies for the situation led to a second report. The main conclusion was that the government should invest in IT governance, primarily by appointing CIOs for the different ministries. These CIOs should be responsible for—among other things—correct use of Enterprise Architecture and portfolio management. Use of Enterprise Architecture in projects is supposed to improve coherence and quality, both functional and technical, thereby reducing risk. The missing actor in this scenario is obviously the architect. Who is he? It is clear that he must be an engineer if he is to take responsibility for the application of sophisticated technology. Yet, it seems obvious that he must be more than an engineer. If reliable and buildable design were a matter of applying proven design methodology and calculating solutions, there would not have been all these problems.

We can conclude that the idea that good architecture is a necessary condition to improve the content of IT projects is well accepted. This study takes as a point of departure the hypothesis that an important problem with IT architecture is that it is not clear by what standards the architecture of a project is to be judged. Similarly, it is unclear what qualities an architect must have, other than being an engineer. Under those conditions, appointing a CIO will solve only part of the problem. The objective of this study is to find out how to establish in an objective manner if IT architecture is applied well in a given context.

2 The Court of Audit is a ‘High Council of State’ which investigates whether central government revenue and expenditure are received and spent correctly and whether central government policy is implemented as intended.
3 TK, 2005-2006, nr. 2.
4 Court of Audit, Lessons from government IT projects, part A, 2007.
5 Court of Audit, Lessons from government IT projects, part B, 2008.
6 Ibid., p. 58.
2. Problem definition and research plan

In this chapter the problem definition, research questions and research approach are explained.

Research problem
As the extensive research by the Court of Audit shows, large IT projects in the public sector often fail, even if a considerable amount of time and money has been spent on preparation.

The court of audit advises further investment in IT architecture in order to improve the situation, but does not indicate how to establish when a project has invested enough in IT architecture. We do not dispose of standards or methods to establish ex ante if a given project’s architecture is of sufficient quality. Checking compliance with specific and easily measurable standards and criteria is not enough.

Research scope
This research is of an interdisciplinary and exploratory nature. Taking Computer Science as its point of departure, it relies on architecture theory as developed for the building sector, cultural history, and even to a certain extent—given the fact that I take the public sector as a ‘case’—on political sciences. This means that I have used a broad spectrum of literature.

Research goal
This research addresses the gap between necessary and sufficient criteria for judging the IT architecture for a larger IT system developed for the public sector. The necessary criteria are generally accepted standards in the domain. The goal of this study is to develop an Architecture Validation Framework (AVF) for the validation of the architecture of such a system. ‘Validation’ means confirming suitability for purpose by applying a set of criteria. By ‘AVF’ we mean a coherent structure of quality aspects of architectures, criteria for judging those aspects, and methods needed to apply judgement. The term ‘framework’ is chosen to indicate that within the scope of this research we develop only a first version, although even if it is tested in concrete cases. Thorough validation of the framework is (far) beyond the scope of this research.

Research questions
In this paragraph the research questions are presented.

1. How can we define ‘IT architecture’, conceptually?

We will use literature about (the concept) architecture in the building sector and try to distil from that a concept of architecture that fits the application in IT in the 20th century. To start with, a few general indications about the terminology used are in place here. ‘Architecture’ generally in this study refers to a set of ideas underlying the major choices that determine further design and implementation of construction. Architectures may stabilize and standardize to the extent that they become implicit: the architecture of the automobile has not changed fundamentally for more than a century.

For example, the ‘Dutch Government Reference Architecture’ (NORA).
now. Architecture can be applied on different levels, and with specific scope. A ‘city plan’ can be considered architecture in that it presents the set of ideas underlying maps that guide future construction. A new building to be built in the city has an architecture of its own, but must take into account the city plan. The climate control system in the building also has an architecture; the scope is again the whole building, but this time only with regard to climate control. These different levels and scopes are not disjunctive. If the commissioner required the building to be heated exclusively with solar power, then the architect, hopefully, has shaped and positioned the building after having sought advice from the architect who will design the climate control system. If a high level architecture describes policy guidelines to be used in lower level architectures, then the term ‘Enterprise Architectures’ or ‘Reference Architectures’ is often used.

If the sets of ideas of the different levels and scopes are coherent, this enables effective and efficient design and construction. It is no wonder that the concept of Architecture therefore appeals strongly to the Court of Audit. If it advises ‘more architecture’ in the context of major IT projects then we can infer that they would like to see coherence between a. ‘sets of ideas’ on the level of the state and lower levels, and b. between different scopes, such as business plans and IT systems supporting the business plan, and c. between the IT architecture and the guidance needed by engineers and suppliers who must further design and implement these systems, thereby efficiently enabling more effective government.

The notion of ‘architecture’ is not necessarily vague (as is sometimes assumed) but different meanings are given in practice as we have seen. It is therefore necessary to describe in some depth which angle is taken in this study. The focus is on the specific role IT architecture has to play in larger organizations when concrete decisions have to be made. The perspective is that of an individual system or project in a greater context: there is a specific need or opportunity that relies heavily on IT and the question is whether or not to trust the architecture.

2. What are the key quality aspects of an IT project architecture that should play a role in validation?

3. How can we operationalise IT architecture, taking into account different levels of architecture and different approaches in current literature?

4. What are possible criteria per key quality aspect?

5. How can we operationalise these criteria so that they can be tested?

6. Is the AVF to be constructed on the basis of the aforementioned criteria, on the one hand, sufficiently general (fits different types of projects) and, on the other hand, sufficiently clear and concrete (can be used by any professional who meets certain competence criteria)?

7. What criteria can be formulated to establish if the AVF is successful in a test or not? We must be able to establish the presence or absence of architecture and a score on the different quality aspects respectively. Since there is no straightforward way to test such a framework, expert opinion will be the main approach.
Examination Committee
Chairman and examiner: Dr. Ir. K.A.M. (Karel) Lemmen.
Secretary and research supervisor: Drs. L.M.M. (Math) Dicker.
3. What is (good) architecture?

In this chapter the first four research questions are addressed, focussing on the concept ‘IT architecture’. In short: what is it, what determines if it is of good quality and how can we operationalise quality criteria?

There is no shortage of attempts to somehow domesticate the unruly concept of architecture in IT through definition. Smolander, for example, has attempted to identify uses of the architectural metaphor in IT and found four: architecture as blueprint (design for an artefact to be built), as literature (documentation of the construction of the system as built), as language (vehicle for communication with stakeholders), or as decision (that is, the basis and documentation of a rational decision-making process). Although this classification of uses of the concept is recognizable and may to a certain extent explain why different professionals take a different angle on the subject, it does not really help us to understand if there is an essential concept linking these phenomena or if architecture is just an attractive label, applied haphazardly to different things.

Corneliussen made an attempt to synthesize Smolander’s approach by mapping the first two types of architecture (‘blueprint’ and ‘literature’) to software architecture, and the latter two (‘language’ and ‘decision’) to business architecture. He reserved the term IT Architecturing for the process of aligning software and business architecture. However, he is not thinking along the traditional lines of ‘business IT alignment’. He emphasizes the importance of human interpretation and stakeholders’ comprehension of the relationship between the two. He sees IT architecturing as a sensemaking process (Weick), where technology is considered intentional and subjective in nature. Human understanding is required to mediate between various architectural semantics, corresponding with the different metaphors. The problem with Corneliussen’s synthesis is that in practice it is hardly different from the traditional Enterprise Architecture and IT-business alignment approaches that, so far, have not proven to be very successful.

Our intuition tells us that those who try to clarify architecture by dissecting it miss the point in that architecture is holistic in nature. Rechting, whom we will discuss later on, makes this particularly clear where he describes the architect’s activities spanning a spectrum from the design of leading edge spacecraft to the tactics of Congressional budget negotiations. There is another inescapable fact: architecture as a deliberate, goal-oriented activity – which is what we are after - requires an architect and a commissioner. We will therefore include a commissioner and an architect as entities for which we need criteria, as we want to make architecting processes more effective.

---


10 The Business IT alignment concept is similar to IT Architecture in that it is ubiquitous, considered by many essential, but at the same time hard to grasp, or maybe nonexistent. For the latter view see: Ciborra, C., ”De profundis? Deconstructing the concept of strategic alignment”, Scandinavian Journal of Information Systems, 1997, 9(1):67–82.

We will use a number of Venn-diagrams to demonstrate interdependencies. In our treatment of Vitruvius we will use three diagrams to clarify criteria for what constitutes architecture, criteria for what constitutes good architecture and the qualities of the Vitruvian architect. A fourth diagram is presented at the end of the chapter, showing an attempt to distinguish ‘architecture as such’ from ‘good architecture’ by identifying necessary and sufficient criteria for the former. The reader should be warned that although the diagrams are partly overlapping in the concepts used, the similarity of their appearance at first glance is to a large extent coincidental.

Given the great wealth of literature on IT architecture, it is hard to filter out sufficiently ‘basic’ principles that can help us to judge architecture. Therefore, we will in the next paragraphs attempt to go back to the roots of the concept of architecture: building architecture. In the following paragraph we will look at three authors, two who can be considered ‘architecture theorists’ and a third who takes the political science angle on the architecting process. Next, we will compare the findings with literature on the quality of IT architecture, in order to determine how we can connect our findings to the IT domain and operationalise and expand them accordingly. In the process we will validate our implicit assumption that ‘architecture’ in building and in IT refers to – on this level of abstraction – the same thing.

The reason we search for clues so far from our IT-home is that the concept of architecture in IT has not lived up to expectations. We do not expect the solution to be to make a ‘better copy’ in IT of existing building practices. Rather, the aim is to find out about the essentials of architecture, thereby helping us to define what is essential for IT architecture and what determines its quality.

3.1 Three visions on the quality of architecture

In this paragraph we will look for the roots of the concept of architecture in the building sector and city planning and its long history. We start with some introductory remarks on the treatment of our sources on building architecture vs. IT architecture and the fundamental nature of the bond between man and architecture.

Obviously, we are looking for fairly abstract characteristics of architecture, since we want to use these concepts in IT. For this reason it is useful to not only read and interpret the work of famous architecture theorists ourselves, but also to study scholars who have tried to identify and clarify essential elements of their theory. This will help us to both critically examine the source and at the same time to get the most out of it.

In the following paragraphs we will present, per author, a synthesis containing the findings. In order to facilitate further use of the results and assure traceability, we will also summarize the findings as a list of criteria, which are presented in a table giving references to the original sources. We will maintain the grouping in criteria for commissioner, architect and, finally, for architecture ‘as such’.

When did it start? Architecture is as old as mankind, much older than any conscious building technology. Man has always sought shelter from the elements, lived in groups and moved about quite a bit. Those factors alone necessitate conscious and relatively sophisticated design
of the use of natural shelter and additional construction. Christopher Alexander\textsuperscript{12} has described a slightly more advanced form of architecture practiced in the case of the hut of the Mousgoun, African tribesmen living in the northern section of French Cameroon. They have been established in that region long enough to develop a specific and, to a certain degree, standardized technology. The physical, social and psychological needs of the inhabitants are well balanced against the possibilities of the technology, which in turn works well only if the positioning of the hut with regard to other huts, the sun and wind is chosen carefully. And, as Alexander points out, in at least one respect this design practice is absolutely superior to modern practices: there is a perfect fit between the wishes of the inhabitant and the design: it is the inhabitant who is the architect.

Now the question is: is that all there is to it? Is the history of the development of architecture no different from that of fishing: first every family for itself, and gradually the better fishers specialized and developed more sophisticated methods in conjunction with more sophisticated technology? We will come back to this question later. First, we will explore some very old but definitively sophisticated thinking about architecture.

\subsection*{3.1.1 Vitruvius}

Vitruvius was a Roman army engineer who lived roughly between 75 and 15 BC. His great work is titled ‘De Architectura’ also known as ‘The ten books on architecture’. As an architect he made the humblest career possible, being a public servant, even if there are indications that he belonged to the elite who served the state rather than a superior officer\textsuperscript{13}. He was largely involved in the engineering of military equipment and infrastructure. He claims to have built one wooden temple, of which no remnants have been found, in Fano on the Adriatic coast.

Vitruvius is interesting for a number of reasons:

- Vitruvius has been a source of inspiration for over two thousand years. His book presents an all-encompassing view on the architect and his work, including a description of the required education.
- The role of the architect as he describes it is different from the modern building architect in that he is more involved with technology and the ‘meta technology’ (hoists etc.) of construction. The architect's responsibility is very broad in scope with Vitruvius, encompassing an idealized version of the ethics of a modern architect. Both elements are relevant for IT architecture.
- When it comes to the relation between IT architecture and its namesake in the modern building industry, it is often said that the difference in maturity between the respective industries prohibits meaningful comparison. Looking at architecture in the building industry in earlier days may therefore yield interesting insights.
- Although his book does not limit itself to the public sector, the perspective of the public sector is overwhelmingly dominant in the book: Vitruvius was himself a public servant and the book is written to support the emperor.
- Vitruvius’ intention was very similar to the purpose of this study: to give commissioners (in the case of Vitruvius the Emperor or those in his service) an instrument to judge if their architects deliver good work.


\textsuperscript{13} Peters, T., “Vitruvius Handboek bouwkunde”, (translation to Dutch of Vitruvius work, of which the original title read “De architectura libri decem”, Ten books about architecture), Athenaeum-Polak and Van Gennep, 5th print, Amsterdam, 2008, p.13.
Analysis of Vitruvius: four books about the Ten Books

We will look at Vitruvius through the eyes of a number of scholars who studied him. McEwen¹⁴ wrote two books that are relevant for this study. In the first one, “An Essay on Architectural Beginnings”¹⁵, she explores and interprets the oldest Greek sources on architecture, thereby reiterating the path that Vitruvius had followed 20 centuries before, as Vitruvius relied heavily on Greek sources and terminology¹⁶. Her second book, “Vitruvius: Writing the body of architecture”¹⁷, aims to clarify the most famous metaphor used by Vitruvius, that of architecture as a reflection of the human body. The book is a broad cultural historical analysis of Vitruvius and his work. Fleury wrote the first of a series of new and extremely well-documented French translations of Vitruvius’ ten books, in which especially the introduction is a rich source of information¹⁸. Peters wrote a new Dutch translation of the Architectura, of which the introduction contains an excellent analysis of Vitruvius’ person and work¹⁹.

The classical origin of architecture

Although the development of architecture out of practical necessity seems plausible, it does not explain how the design of something as sophisticated as the Greek temple of Hera at Samos built in the 8th century B.C could seemingly come ‘out of the blue’.²⁰ McEwen claims that (Western) architecture and (Western) philosophy (‘speculative thought’) developed in parallel, and were both the expression of ‘emerging Western consciousness’ and what she calls ‘the theoretical event’.²¹ ²² Her book is interesting as it presents a very focussed cultural history of the concept of architecture in relation to central concepts in the mental world of classical antiquity. Although the historical underpinnings of her theory are (as she readily admits) speculative, her theory and observations help to better come to grips with the more elusive side of the concept of architecture.

McEwen argues that at the time the temple of Hera was built knowledge was equated with a capability to construct and shape things. The talented individuals who built the temple of Hera demonstrated ‘Kosmos’ (divine order) through their skill; a ‘Technē’ (letting appear). Knowledge or wisdom could not be separated from the experience of the knower, just as he who knows how to dance cannot transfer this knowledge by ‘telling’ it to someone else. ‘Theoria’, which refers to seeing, is the revelation of the divine in things well made.²⁴ According to McEwen, the ambition and capability to create more sophisticated constructions appeared first historically and gave rise to gradually more sophisticated thinking. Thus, architecture was first, and philosophy followed. As Aristotle put it: ‘It is through wonder that

¹⁴ Indra Kagis McEwen holds an honours B.A. in English and Philosophy (Queen’s University), a professional degree in architecture, a Masters degree in architectural history and theory, and a PhD in Art History (McGill). She is an interdisciplinary scholar, specialised in political dimension of the history and theory of architecture.
¹⁷ McEwen, 2003, ibid.
¹⁹ Peters, ibid.
²¹ Ibid, p. 4, 123.
²³ McEwen, ibid, p. 124.
²⁴ Ibid, p. 125, 126.
men now began and originally began to philosophize.\textsuperscript{25} Gradually, however, knowledge was seen as the result of ‘seeing’ and no longer as the result of ‘doing’. With this shift came the wish to stabilize knowledge. Socrates, in a passage in the Meno, refers to the statues made by the mythical first architect Daedalus. These statues showed divine inspiration in the ultimate form: they moved and came to life. When Meno wonders why episteme (knowledge) should be preferred to doxa (right opinion) Socrates answers that Daedalus’ statues were of great value only when bound by chains, for if they were not bound they “would play truant and run away”, no matter how fine works of art they were. “Bound, they have the nature of knowledge and are abiding, like true knowledge, fastened by chains.”\textsuperscript{26}

The chapter further elaborating on Daedalus starts with the concept of the pattern that was used both in abstract and concrete form in ancient architecture. In Plato’s ‘Timaeus’ the universe is constructed according to a pattern (paradeigma), by a craftsman. McEwen states that the discovery of a pattern is an inherent feature of the human experience of making. The pattern is a ‘standard of rightness’ and measures the work and is measured by it.\textsuperscript{27} We will come back to the pattern concept later, when Christopher Alexander ‘rediscover[s]’ the architectural pattern, a concept that made possibly him more famous in software engineering than in the building industry.

As knowledge became abstracted from creation, McEwen states, “not only did the craftsman lose prestige and political influence, but also did the thinker. Indeed, for all his eloquence on the matter of civic harmony, Socrates met his death, in part at least, because he lacked political commitment.”\textsuperscript{28} Through De Architectura, Vitruvius tries to remedy this situation. He typically sees the architect as a combination of a craftsman and a thinker who deserves more political influence. Vitruvius’ professional attitude, as McEwen notes, is reminiscent of a time when the architect did not rely on explicit theory when architecting. Vitruvius calls the written work in which the theory of the building he constructed is articulated a commentarius, from the verb commentor, meaning “to think over.”\textsuperscript{29} He sees it as essential that an architect is trained in letters so as to reach a ‘better memory’ in his commentaries. Theory was a reflection, where lessons were drawn, adding not so much to an existing body of explicit theory but to documented experience.

\textit{Vitruvius’ message and intended audience}

Peters presents a realistic analysis of Vitruvius’ personal intentions. No doubt Vitruvius wanted to promote the position and status of architecture and the architect. It is very likely that he hoped that his book would bring him the fame he never acquired as an architect.\textsuperscript{30} At least in this respect his project was more than successful. Although Vitruvius in his introduction claims to write for the emperor, it is totally obvious, and in fact acknowledged by him, that he catered to a much broader readership: he considered his book ‘a not unwelcome gift to the whole world’.\textsuperscript{31}

\textsuperscript{25} Ibid, p. 125.
\textsuperscript{26} McEwen, ibid, p. 5.
\textsuperscript{27} McEwen, ibid, p. 41, 42.
\textsuperscript{28} Ibid, p. 130.
\textsuperscript{29} Ibid, p. 125.
\textsuperscript{30} Peters, ibid, p. 19, 167.
\textsuperscript{31} Ibid, p.34-35.
\textsuperscript{32} Ibid, p. 168.
Another reason for writing the book was Vitruvius’ frustration about certain colleagues. Vitruvius was painfully aware that recognition of architecture as an important discipline did not automatically mean that those in power would—or could—make optimal use of it. Opportunistic individuals with the right relations or impressive personality had all too often been given prestigious assignments or had been listened to regardless of real professional qualities: ‘...since ignorants are sooner the favourites of those in power than the true architects, and since I think one should not try to compete with such individuals, I prefer to promote the value of our knowledge through this book’³³.

Vitruvius generally stresses the necessity of not only integrity, but also generosity and unselfishness³⁴. Furthermore, he clearly saw architecting not only as a profession, but also as a calling supported by a broad intellectual education.

Architecting the Roman Empire
Where Peters explains the personal motivation of Vitruvius, McEwen in her book about Vitruvius takes a more abstracted view and tries to shed more light on the fit between the historical context and Vitruvius’ theory. A major theme in her book is the question of what was behind the most fundamental metaphor in his work, that of architecture as a reflection of the human body³⁵. McEwen concludes that in Vitruvius’ metaphor ‘body’ referred to the Roman Empire, the whole of the artefacts in which it manifested itself and to architecture as the shaping force. The empire was shaped by architecture, and architecture was, reciprocally, shaped by the body of the empire³⁶.

Fleury cites the observation by Martin³⁷ that urban planning, unlike architecture, is something that comes in waves. Fleury notes that in Vitruvius’ period there was a rare coincidence of high levels of both urban planning and building. The relative stability of the Roman Empire both required and enabled consolidating actions. Vitruvius’ approach to urban planning is therefore ‘ex nihilo’, building new cities from the ground up³⁸.

Architecture confirmed a political message. Temples played a central role as symbols of Roman culture communicating stability, even when the political situation in Rome or the region concerned could temporarily be volatile. McEwen also attributes the relatively prominent position of sundials in De Architectura to Vitruvius’ intention to promote architecture as supporting stability. The emperor had improved the calendar, thus establishing a better match with the sundial. The suggestion of concordance between the divine powers controlling the course of the sun and Augustan's reign improved the legitimacy of Roman government³⁹.

Vitruvius’ mission, according to McEwen, was based on a sincere belief that Roman occupations meant an extension of civilization and the mission of architecture was to support it. The order in which De Architectura treats its subjects is exactly the order of events followed by the urbanization of Gaulle and Iberia: choose locations, organize, build public

---

³³ Ibid, p. 84.
³⁵ McEwen, 2003, ibid, p. 5.
³⁶ Ibid, p. 299, 300-301.
³⁸ Ibid, p. xcvi.
buildings, build private buildings and finally construct ‘furnishings’ such as sundials and defence equipment.\textsuperscript{40}

\textbf{Firmitas, Utilitas, Venustas}

Vitruvius is especially famous for the triplet ‘Firmitas, Utilitas, Venustas’ (solidity/durability, functionality, attractiveness) as criteria for good architecture, which is especially of interest for this study.\textsuperscript{41}

Firmitas (solidity/durability) is not given much attention in De Architectura. Fleury notes that Vitruvius always seems to be saying both too much about technology (for the reader with a general interest in architecture) and too little (for someone who would like to apply his instructions concretely)\textsuperscript{42}. This is true and, in fact, only to be expected, given the intended audience. The reason that Vitruvius did throw in some technical detail here and there is probably that he wanted to stress the fact that the architect knows all about technology and is therefore a useful expert. Peters notes that Vitruvius’ positive approach to technology was rather unusual in intellectual circles in Rome. Technology was primarily associated with manual labour.\textsuperscript{43} A typical example of the Vitruvian attitude is his comment on the invention of the scales: it contributed to more honest trade practices. The example also shows how Vitruvius was keen to see the moral component of technology and its application.\textsuperscript{44}

Utilitas (functionality) is important in Vitruvius’ book. His short description of the concept refers primarily to layout. Yet, reading De Architectura, one of his most consistently stressed functional requirements is concern for health.\textsuperscript{45} Fleury sees this in part as a logical consequence of the fact that Vitruvius made deliberate and extensive use of older writers, and Plato and Hippocrates alone are rich sources of information when it comes to health. But he does note that Vitruvius’ health concerns seem to overrule all other concerns. For instance, when it comes to city planning, health takes a much more prominent place that economic or defence considerations. Peters considers it therefore a genuine concern of Vitruvius.\textsuperscript{46}

Venustas (attractiveness) is also an important topic. It has two sides. On the one hand, there are countless rules and guidelines in De Architectura, such as in book 3 and 4 about temples. On the other hand, Vitruvius is very concerned with concrete attractiveness for humans. The Latin word for beauty generally, according to McEwen, would sooner be ‘pulchritudo’ referring to a more platonic, ideal beauty. Venustas, consistently used throughout De Architectura, implies ‘pleasing to the senses’, but as a derivation from ‘Venus’ it also refers to a binding force protecting Roman dominance. The latter point is considered by McEwen to be the second reason for Vitruvius’ choice of vocabulary.\textsuperscript{47}

\textit{Further Vitruvian quality criteria}

The triplet ‘Firmitas, Utilitas, Venustas’ refers to quality criteria for the result of architecture, the work built. Vitruvius devoted far more text to the qualities of the architect and how he should perform his role.

\textsuperscript{40} Ibid, p. 283.
\textsuperscript{41} Peters, ibid, p. 38.
\textsuperscript{42} Fleury, ibid, p. xxxv.
\textsuperscript{43} Peters, ibid, p. 21.
\textsuperscript{44} Ibid, p. 21.
\textsuperscript{45} Ibid, p. 12.
\textsuperscript{46} Fleury, ibid, p. xcixii.
\textsuperscript{47} McEwen, 2003, ibid, p. 200-212.
Integrity and unselfishness are considered vital qualities; lacking these, ‘no work can really be completed’\(^{48}\). He also considers it essential that an architect be asked for an assignment and not offer himself\(^{49}\), since, otherwise, what else can he be after other than his own interest?

Education is considered by Vitruvius to be extremely important. A large part of the preface of the first book is dedicated to it\(^{50}\). For Vitruvius it is essential that the architect is broadly educated without trying to become a specialist in each field\(^{51}\). He should be trained in writing, geometry, drawing, history, philosophy, medicine, law, music and astronomy. Formal education should be accompanied by extensive experience\(^{52}\). One is useless without the other\(^{53}\).

Economic concerns are the responsibility of the architect. Controlling cost is important as Vitruvius explains in the preface to book 10, and the problem of cost overrun in building is ‘classic’\(^{54}\). He proposes to implement a regulation, which apparently had been the custom in a Greek city, to hold the architect personally responsible for cost overruns above a quarter of the estimate. The architect must also see to it that the design is economic in that it does not require materials that are hard to get or use materials or space inefficiently. The design must also fit the economic possibilities of the commissioner\(^{55}\), even if the commissioner has the final word on how ‘rich’ the choice of finishing materials should be\(^{56}\).

User interests are in good hands with Vitruvius. As a part of ‘venustas’, health is, as we have seen, a concern that we find throughout De Architectura, from climate concerns in city planning to avoiding lead in drinking water systems. It all sounds very modern\(^{57}\). The public interest generally is a guiding principle\(^{58}\). User influence is not expressed very clearly, even if the architect is supposed to be accommodating, and take advice from workmen and laymen seriously\(^{59}\).

**Analysis**

- Vitruvius considered the Roman Empire, especially in the period of relative stability under Augustus, as an ideal test case to prove that architecture could support a well-designed effort to improve the world by consolidating the Roman Empire. The era in which he happened to live offered a window of opportunity for a visionary. He believed in policy, and architecture was in his eyes a supreme policy tool. Vitruvius was, therefore, a good bureaucrat in the Weberian sense, extremely serious in his expertise, and laying great weight on integrity and loyalty. In turn, he hoped to improve the status of architecture and architects like himself by promoting their importance for the development of the Roman Empire.

---

\(^{48}\) Peters, ibid, p. 30.

\(^{49}\) Ibid, p. 168.

\(^{50}\) Ibid, p. 28-35.

\(^{51}\) Ibid, p. 33.

\(^{52}\) Ibid, p. 28.

\(^{53}\) Ibid, p. 28.

\(^{54}\) Ibid, p. 270.


\(^{56}\) Ibid, p. 188.

\(^{57}\) Ibid, p. 39, 236.

\(^{58}\) Ibid, p. 165.

\(^{59}\) Ibid, p. 30, 189.
The loyalty of the architect is, however, not entirely unproblematic. First, the commissioner cannot check if all the choices made by the architect are optimally in his interest. De Architectura basically tells him to hire a good architect. Second, the architect is supposed to take into account the interests of all who will use the building, which may not be wholly equivalent to the commissioners’ views on the matter. Vitruvius evades the issue subtly by implying that there is no difference. In the same paragraph where he explains his mission he states: “...as you (the emperor LF) will plan public and private buildings also in the future and since those must keep the memory of your accomplishments alive in future generations, I have formulated the rules that apply.” It is obvious in that case the emperor is supposed to build what citizens appreciate.

Vitruvius requires the architect to dispose of lots of explicit ‘scientific’ knowledge in different fields. We infer that he, in fact, wanted to improve the status and credibility of the ‘tacit knowing’ of the architect. The concept ‘tacit knowledge’ was developed by Michael Polanyi who studied scientific discovery. It comprises a range of conceptual and sensory information and images that play a role in an attempt to make sense of something. Many bits of tacit knowledge can be brought together to help form a new model or theory. Scientific knowledge, in which all faculties that are necessary for finding and holding scientific knowledge are fully developed, is the knowledge of approaching discovery. Architecture is also a discovery: you find a solution (model) through a process that is not, or only in part, deductive. Polanyi considers high personal involvement a necessary condition for success: “We must conclude that the paradigmatic case of scientific knowledge, in which all faculties that are necessary for finding and holding scientific knowledge are fully developed, is the knowledge of approaching discovery. To hold such knowledge is an act deeply committed to the conviction that there is something there to be discovered. It is personal, in the sense of involving the personality of him who holds it, and also in the sense of being, as a rule, solitary; but there is no trace in it of self-indulgence. The discoverer is filled with a compelling sense of responsibility for the pursuit of a hidden truth, which demands his services for revealing it. His act of knowing exercises a personal judgement in relating evidence to an external reality, an aspect of which he is seeking to apprehend. The requirements for application of tacit knowledge clearly match Vitruvius’ profile of the architect where the ‘knowledge of approaching discovery’ seems to apply equally well to architecting.

The technical detail in De Architectura was there to impress the reader and get the idea across that one had better hire a professional. But the instructions were, as Fleury rightfully noticed, in practice often too superficial to actually guide construction.

Vitruvius greatly values a broad education, but the reasons he gives for this are not always convincing. Knowledge of music is supposed to be important since the strings in instruments are comparable to strings in rock projecting war machines. Knowledge of history is important because the architect must be able to explain the story behind certain traditional ornaments. Study of philosophy is supposed to contribute to a noble

---

60 Peters, ibid, p. 28.
character, honesty, and reliability. The message that can be inferred is that the role of architect requires an intellectual background as broad as possible in order to understand as fully as possible the properties of the problem the architect is supposed to design a solution for, and the qualities of the solution. This background also helps to strengthen the authority of the architect when defending his design.

- Vitruvius’ concern for modesty and his idea that an architect should be asked and not offer himself also places the architect in the position of someone who is trusted as a person, almost as a doctor or a priest, not as a professional who delivers a completely verifiable product. That is understandable since, even in the best of cases, in those days it was rather obvious that a building project was ‘development’ of the type we know in software engineering today; much relies on judgement rather than verifiable engineering. The building process had its own dynamics and lots of detailing took place during construction. It is therefore not so strange that only afterwards, as McEwen noted, Vitruvius required the architect to carefully ‘think over’ what he had built. Another reason for insisting that the architect must be asked for an assignment may have to do with the fact that Vitruvius was aware that decisions based on tacit knowledge can only to a certain extent based on hard arguments. For these decisions to be acceptable, it had to be clear that there was no hidden agenda on the part of the architect and it obviously helps if the commissioner has explicitly placed his trust in the architect.

- Vitruvius paints a very realistic picture of the problem of political leadership facing ‘policy problems with technology-related solutions’: who to trust? In actual practice, political leadership, hardly able to make a defensible choice on substantial grounds, chose the expert who was judged to be the most credible for other reasons: convincing appearance and rhetoric, powerful relations. Often this type of advisor saw his own economic, social and political interests as a natural ingredient of his advice. Vitruvius sees professional integrity as crucially important.

- If the commissioner faced the risk of unreliable architects, the architect had to deal with commissioners with ‘strong opinions’. The very fact that the commissioner starts looking for an architect means that he thinks he needs construction to solve his problem or realize his vision. Very often a commissioner will have a rough idea of a possible design. His ‘requirements’ may be details of the construction he had in mind or reflections thereof. Vitruvius never specifically mentions this as a problem but he did wrestle with the phenomenon of the ‘do it yourself’ architect. The whole of De Architectura is about promoting the exclusive role of the architect as the independent authority in design. Yet, as he acknowledged in his sixth book about private buildings, he understands ‘homeowners’ who prefer to spend their money as they see fit on the basis of their own experience, rather than hiring an architect where it is very difficult for them to know in advance if this will prove useful. Even so, this is not an inconsistency on Vitruvius’ part for De Architectura is also about the problem of architecture as an immature market.

- Where Vitruvius does mention the do-it-yourself architect, he says nothing about the design-and-build concept. This is an indication that there was no such thing as a

\[\text{Peters, ibid, p. 168.}\]
'building industry’. This explains how an architect could reach such high status in Vitruvius’ days. He was more obviously indispensable for ambitious projects.

- Although the triplet seems almost the emblem of Vitruvius, he dedicated only a few sentences to it. This should be no surprise because Vitruvius was aware that the criteria of the triplet can only be applied after the fact, and therefore are not very useful for a commissioner who has building ambitions. The guarantee of quality in the Vitruvian view lies in the education, experience, and personal qualities of the architect, together with the right relation with the commissioner. Vitruvius’ operationalisation of good architecture is not a checklist for the finished product. Good architecture relies heavily on the criteria for what makes a good architect, the scope of his responsibilities, and the conditions for the architecting process. These are the true focus of De Architectura.

Furthermore, De Architectura makes clear that Vitruvius saw the elements of the triplet in practice as interrelated. For example, his constantly expressed concern for health cannot easily be categorized into one of the three. Functionality seems to fit best, but attractiveness is very close, for instance where Vitruvius describes how a family visiting the theatre should be protected from (unhealthy) draughts. In the same vein his story about the city of Salapa that had to be moved away from the marshes tells there is a link between attractive functionality and durability.

- Vitruvius’ concern for health, although often expressed in a fairly businesslike manner, gives the impression of a highly ethical and humanitarian professional attitude. It is important to note though, that his concern for a healthy and pleasant built environment can also be seen as ‘strategic’ in the sense that the Romans thus could consolidate their position in the occupied areas. Elites, whether local or imported from Rome, would appreciate the quality of life and so remain—or become—loyal Roman subjects. Tacitus, for example, strikes a cynical note where he describes how the ‘most salutary measures’ undertaken by the Romans in Britain led to appreciation and adoption of the Roman lifestyle by the native elite: ‘The simple natives called all this humanitas, when it was really a facet of their enslavement’.

- Vitruvius was, all in all, a visionary with a strong academic instinct, but also a true romantic who wrote ‘a master narrative if ever there was one’.

**Vitruvian Architecture and IT**

What remains now is the question of to what extent our analysis of what constitutes the essential Vitruvian view on architecture can serve as a basis or a source of inspiration for criteria to judge 21st century IT architecture. In the beginning of this chapter we have accepted the assumption that architecture in building and IT is comparable. We also assumed that the building architecture practiced in Vitruvius days, including the role of the architect, showed similarities with the IT industry today. This assumption is confirmed on the following grounds:

---

64 Cited in McEwen, 2003, p. 151.
65 McEwen, 2003, p. 301. The term ‘master narrative’ refers to a dominant historical view that explains the relation between events; it ‘makes sense’ of history. The concept, originally named ‘grand narrative’, was identified by Jean-François Lyotard in his classic 1979 work “The Postmodern Condition: A Report on Knowledge”.

• The profession exists and is recognized as such but its credibility is not uncontested. Its content is considered by many to be ‘out of the ordinary and obscure’.

• Vitruvius’ architect was still closer to technology than modern building architects tend to be, who leave technical detail to construction specialists. Even if few ‘IT architects’ claim the full breadth from business ambition to details of implementation, we see that a ‘lead architect’ must be able to involve himself in quite a few technical issues since so little is really proven. This in part explains why there are many ‘levels and scopes’ of architecture in IT.

• There is the same ambiguity among political leadership and higher management with regard to architects: on the one hand, they are not terribly happy with what architects have to offer. On the other hand, they are eager to follow the advice of people who may not even claim to be architects, but who provide them with the arguments supporting their preconceived notions.

• The poor quality of the estimates of building costs produced by unqualified architects is highly recognizable. IT architects often hardly consider it part of their expertise.

There are differences as well. Vitruvius had to deal with do-it-yourselfers, but not with the design-and-build concept, which is obviously rather dominant in the IT industry, be it in the form of large system integrators or application suite providers. If IT-architects want to aspire to the role of a Vitruvian architect, they must be convincing to commissioners in a role that is so thoroughly human and individual that no ‘industry’ can match it.

Overall, we see clear similarities between building architecture in Vitruvius’ days, and IT architecture today. The problematic practices Vitruvius describes are obviously not terribly helpful for IT architecture today. It is Vitruvius’ vision that can serve as inspiration in any case, accepting its historic fame as sufficient proof of its value. We can go a bit further, however, because we can see that important aspects of Vitruvius’ vision are practice in modern building architecture:

• Professional status and academic education are well established.
• Construction technology and calculation have improved in such a way that the architect can concentrate on his core business, namely, the added value of architectural choices, while maintaining responsibility for sound further design, construction and cost.

In Vitruvian terms, in modern building architecture Firmitas is taken care of, and Utilitas also. That is not to say that the modern building architect is exactly what Vitruvius would have wanted him to be:

• ‘Venus at’ in the broad Vitruvian sense is not guaranteed to say the least.
• In urban planning today political power and the profession still struggle with the division of roles.
• Few architects aspire to the broad education Vitruvius had in mind.
• The very high ethical standards Vitruvius considered essential have been ‘absorbed’ largely by standardization and regulation, so the problems he associated with a lack of ethics are mitigated. However, since not all ethics can be effectively addressed by regulation, seriously unethical designs can be proposed ‘fully in compliance with applicable regulation’. Not every building architect today is so successful that he can

66 Peters, ibid, p. 138.
afford to live entirely by Vitruvian standards, nor are those that could afford it always thus inclined.

**Conclusion Vitruvius**

We will follow the research questions: what is architecture, when is it good architecture and how do the criteria found translate to IT? We will use a number of diagrams for clarification. Finally, we will summarize the findings in a table.

**What is architecture?**

Even if Vitruvius did not ‘define’ architecture, the concept is crystal clear and compelling for any reader of De Architectura. According to Vitruvius, people with power (be it public power, economic power or both) who intuitively feel that they can reach non-trivial goals by changing the physical world with technical means need help: the commissioner’s ‘design’ must be translated into a set of rough descriptions of structure, functionality, technology, feasibility and cost, accompanied by a coherent rationale matching the commissioner’s intentions. The architect is also responsible for planning of transformation, cost calculation, organizing construction, overseeing construction and evaluation. The architect must be a well-educated and experienced professional, independent, of high integrity, and take into account the fundamental interests of all who are somehow involved in the process and its result. The commissioner’s most important role is to choose a good architect and create the conditions for a good architecting process.

What is most characteristic of architecture, as Vitruvius describes it, is that it involves the application of a broad spectrum of knowledge, which we have interpreted in modern terms as ‘tacit knowledge’. The clients’ non-trivial needs, as perceived and analysed by the architect, must be matched with available technologies to create an attractive solution. The following diagram maps these elements to the universe of attractive designs. With ‘attractive’ we mean not just ‘appealing to the senses’ but generally attractive including functionality and durability.
We can further analyse the relative importance of the different elements.

- The weakest criterion is the commissioner. If a well trained and experienced architect (like Vitruvius) happens to be working in the public service and develops a highly innovative and attractive solution, would that not be architecture? We think it is architecture but without adequate sponsorship it is unlikely that the architecture will be successful.

- The formally trained architect has stronger cards. If an amateur develops an attractive design, we are inclined to call this ‘art’ rather than architecture. Inversely, what if an architect hired by a commissioner for a problem that any engineering firm would solve in more or less the same way, we would not be inclined to consider this architecture, but design.

- Induction seems to be as strong as the architect, but we are inclined to consider it slightly stronger: if a design, based on a brilliant vision, is made by someone who does not fully qualify as an architect would that not be more architecture than a flawlessly engineered solution?

**What is good architecture?**

According to Vitruvius, architecture must be functional, durable and attractive; attractive, not only to the commissioner, but also within a wider context. In the following adapted diagram we have left out the commissioner since he is not essential for architecture as we saw in the last diagram. This time the universe is all designs, not just attractive designs, which
considerably changes the view. Good architecture (Vitruvian) is now at the heart of the diagram.

![Diagram showing Vitruvian 'postconditions' of good architecture]

Figure 2. Vitruvian 'postconditions' of good architecture

Consistent with the importance we gave to 'induction' we consider a result that was not designed by an architect, but is attractive, functional and durable, to be architecture, be it 'primitive' architecture. What about a design based on induction, made by an architect, but lacking one or more of the triplet elements? We have dubbed it ‘failed architecture’ but the use of the word ‘architecture’ may seem slightly inconsistent with our analysis of ‘architecture’: if the design is not attractive (in the broad sense: appealing and not obviously flawed in functionality and durability) it is not architecture. However, given the fact that the architect involved is a serious professional who applied induction, we feel that ‘failed architecture’ does more justice to the situation. To put it differently: everything was in place for a true ‘act of architecture’ (process) but it did not result in true architecture (product).

However, since Vitruvius saw the criteria for what constitutes a good architect, his commission and other conditions for the architecting process as what a commissioner should be looking for. The following diagram is therefore a better representation of Vitruvius’ message. It contains the most important advice of Vitruvius to the commissioner: only if the architect fulfils all requirements can he be successful. The commissioner should sooner look for an ‘unfortunate’67 (see figure) with the right qualities than rely on those who offer themselves even if they have an impressive personality or powerful relations.

---

67 Peters, ibid, p. 84.
How do the Vitruvian criteria translate to IT?
The broad intellectual education as a requirement is new for IT. The message is that technology is only one aspect, broad analysis of the nature of the problem and context, and understanding of the economic, practical and ethical aspects of the solution is equally important. The holy grail of ‘business IT alignment’ seems to suggest that present day IT architects are unable to bridge the gap between their knowledge of what technology can do, the real needs of the client and what must be done to realize robust improvement. The remedy is to involve more people (‘business consultants’). Vitruvius’ approach is rather to see to it that the architect has the necessary capability to analyse the problem personally.

The idea that the architect is responsible for the cost and timely realization of a project is clearly generally accepted in Vitruvius’ time, even if problematic in practice. This may seem unspectacular, but it is amazing that this division of roles has existed for more than 2000 years, although it is clear that in real life it has more often than not been difficult to do justice to the concept. If IT architecture could aspire to the classical role (with credibility) it would clearly improve its usefulness for commissioners tremendously. We are not referring to the cost of just plain programming or hardware, but to the total cost of realization of a project, including system integration as well as exploitation.

Also new for IT is the norm that the design must be ‘attractive’ in the Vitruvian sense: ‘pleasing to the senses’ and promoting well-being and considering health. Attractiveness in
the Vitruvian sense is a concept that can be applied in IT architecture, especially in the public sector. The essence is that all users of the system feel that the design was made taking into account their well-being, regardless of the main purpose of the construction. That requires more than a pleasant graphical design of the user interface.

Summary of the results
In the following table we summarize the results.

<table>
<thead>
<tr>
<th>Table 1. Criteria derived from Vitruvius</th>
<th>Quality aspect or related role</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commissioner</td>
<td>Display ambition, intuitive idea about possible construction</td>
<td>Peters, p. 27-28</td>
</tr>
<tr>
<td></td>
<td>Hiring an architect of proven quality and good reputation</td>
<td>Peters, p. 168</td>
</tr>
<tr>
<td></td>
<td>Determines how luxurious the construction shall be</td>
<td>Peters, p. 188</td>
</tr>
<tr>
<td>Architect</td>
<td>Professional, trained in theory and practice, of high integrity, and modesty</td>
<td>Peters, p. 30</td>
</tr>
<tr>
<td></td>
<td>Intellectual training broad in scope: writing of commentaries, philosophy, law, geometry, physics, astronomy, medicine</td>
<td>Peters, p. 29</td>
</tr>
<tr>
<td></td>
<td>Is asked for a commission and does not ask for one</td>
<td>Peters, p. 167</td>
</tr>
<tr>
<td></td>
<td>Is responsible for the contract with the builder</td>
<td>Peters, p. 32</td>
</tr>
<tr>
<td></td>
<td>Appreciates advice from builders and laymen</td>
<td>Peters, p. 189</td>
</tr>
<tr>
<td></td>
<td>Is responsible for the prevention of cost overrun</td>
<td>Peters, p. 270</td>
</tr>
<tr>
<td>Architecture</td>
<td>Must be strong and durable</td>
<td>Peters, p. 38</td>
</tr>
<tr>
<td></td>
<td>Must be functional</td>
<td>Peters, p. 38</td>
</tr>
<tr>
<td></td>
<td>Must be attractive</td>
<td>Peters, p. 38</td>
</tr>
<tr>
<td></td>
<td>Must not be detrimental to the health of users</td>
<td>Peters, p. 39</td>
</tr>
<tr>
<td></td>
<td>Fits the economic status and possibilities of the commissioner</td>
<td>Peters, p. 38</td>
</tr>
<tr>
<td></td>
<td>Is in compliance with building code</td>
<td>Peters, p. 32</td>
</tr>
</tbody>
</table>

3.1.2 Christopher Alexander
Twenty centuries after Vitruvius another architecture theorist tried to get to the heart of the concept of architecture: Christopher Alexander. This time, the link with IT is unproblematic: many books on software engineering mention Alexander’s name at least once, referring to his ‘pattern’ concept. The pattern, though, is in fact no more than a phase, an experiment in Alexander’s long academic quest for good architecture. Alexander taught architecture for almost forty years at the University of California, Berkeley. His academic career started with his dissertation, which will be discussed first. Next, we will discuss Grabow who, as a fellow academic in architecture, investigated the nature of Alexander’s architecture theory. Alexander’s work must be seen in the perspective of the ‘crisis of modern architecture’ for which he tried to provide a solution. Interestingly, IT also had a design crisis a little later (‘the software crisis’) and it is therefore not surprising that IT people have looked in his direction for inspiration.
Notes on the synthesis of form

In his dissertation “Notes on the Synthesis of Form”\(^{68}\), written in the early sixties, Alexander tried to develop a fundamental theory of design. Being trained as a mathematician originally, he did not feel comfortable with a design practice that seemed to rely on an intuitive and improvised confrontation of functional demands and aesthetic ideas. The following diagram\(^ {69}\) shows his analysis.

![Diagram](image)

Figure 4. Alexander’s conception of the design process

The first model depicts the building practice of ‘simpler civilizations’ as described earlier: the inhabitant builds on the basis of tradition, adapting perfectly to his needs and the surroundings. The second model is the actual situation: an architect forms an image for himself, a mental picture, of the context, and translates that into a mental picture of the solution, which then materializes in a design and construction. The third model first makes a formal picture of the mental picture, basically checking the mental picture and leaving out all sorts of bias based on language and experience. Sets of requirements, the result of decomposition, are the result.

---


\(^{69}\) Ibid, p. 76.
Then this formal picture translates into a formal picture of the solution. It consists of ‘constructive diagrams’ representing solutions for the sets of requirements. The following diagram⁷⁰ is presented by Alexander as an example. It shows a constructive diagram of what an intersection of roads should look like after reconstruction in order to alleviate congestion. The flow of traffic is modelled in the direction and width of the twelve arrows. The same information could have been presented in a table, but the constructive diagram shows immediately what the solution should look like.

The last and essential step from requirements to solution is still intuitive, as Alexander acknowledges, but at least it is a process ‘out in the open, under control’,⁷¹ ‘Notes’ (obviously, given Alexander’s background as a mathematician) offers mathematical support for his method, based on set theory. It is interesting that Alexander does not see design as an optimization problem. Requirements need to be sufficiently fulfilled⁷².

Ten years later Alexander had mixed feelings about his dissertation, as can be read in the preface of the 1971 paperback edition of ‘Notes’⁷³. With hindsight he sees the constructive diagram as the essence, and the mathematical method for ‘deducing’ such a diagram as artificial: the input consists entirely of subjective human observations. He feels that it is better then to look for fitting constructive diagrams, displaying both the essence of the problem and solution. It is useful to cite the passage in the preface where he comments on mechanical application of design methods:

‘I want to state publicly that I reject the whole idea of design methods as a subject of study, since I think it is absurd to separate the study of designing from the practice of design. In fact,

---

⁷¹ Ibid, p. 78.
people who study design methods without also practicing design are almost always frustrated designers who have no sap in them, who have lost, or never had, the urge to shape things. Such a person will never be able to say anything sensible about ‘how’ to shape things either.

Thus, in the end Alexander follows Vitruvius where the latter states:

‘Architects who have become architects by diligent practice without studying the books, never managed to acquire the authority their labours warranted. Those however who only rely on theory and learned books have not been aiming at reality but at its shadow. Those who have studied and practiced extensively have thus fully armed themselves and accomplish their work efficiently and with authority.’

The dissertation is impressive as an intellectual achievement, but is also rather confronting for the profession. In a compact and extremely well-written and readable essay of 130 pages with a few simple sketches, he denounces long-standing practices. The footnotes show a man who is extremely well read and very conscious of epistemological considerations.

After ‘Notes’

Alexander has built relatively little. He was a typical academic who experimented for the sake of the development of theory. Fairly early in his career he was given the opportunity to experiment with a guided version of ‘homemade architecture’. In 1975 Alexander was contracted by the Mexican government to set up a process for the design and construction of low-cost housing where the future owners were involved in the design. Even if there was an overall technical architecture, all individual houses were to be different. In this project three elements worked together jointly that are normally separated: requirements, design and building. The project was a success, even though Alexander threw in the use of highly innovative low cost, light weight armed concrete as vaults. One of the lessons Alexander drew from this experiment was that he had underestimated the importance of craftsmanship: the quality he looked for could not be reached if the actual construction process was of limited quality.

An early publication is worth mentioning here, one of the few in which Alexander explicitly addresses city planning. In ‘A city is not a tree’ Alexander explains that the only reason that modern cities are laid out as ‘trees’ in the mathematical sense, with a strict separation of functions, is the fact that the complexity of a semi lattice, what a living city is and should be, according to Alexander, cannot be grasped by the human mind. Because the complexity of a semi lattice cannot be encompassed in any convenient mental form, the mind has an overwhelming predisposition to see trees wherever it looks and cannot escape the tree conception. A city should be ‘a receptacle for life’ mixing different functions. Alexander uses the example of Carnegie Hall and the Metropolitan Opera House in Manhattan that were

---

75 Peters, ibid, p. 28.
78 We have not further investigated the validity of this claim. The popularity of the ‘mindmap’ (laying out all facets of a problem in a treelike fashion in order to generate oversight) can be considered an indication of its credibility.
not built side by side. Each found its own place, and creates its own atmosphere. The influence of each overlaps the parts of the city which have been made unique by it.

Grabow - Christopher Alexander: The Search for a New Paradigm in Architecture

Grabow’s book⁷⁹ was published in 1983, but the content dates from the end of the seventies, so relatively early in Alexander’s career⁸⁰. It is, apart from Alexander’s written work, based on more than 100 hours of interview. Grabow was interested in paradigm changes in architecture, using Kuhn’s theory.⁸¹ Kuhn’s theory, in short, predicts that a field of science develops via revolutions, since there is too much vested interest to absorb more than marginal changes in the dominant theoretic perspective. Successful ‘revolutionaries’ make their point by explicitly denouncing the existing views and bringing evidence with them for a new perspective. Alexander seemed to fit this picture. Grabow therefore considered Alexander’s theory a candidate that could possibly change the science of architecture. The first element, denouncing existing practices, was certainly accomplished by in good hands with Alexander. The second point was more problematic. Since Alexander, as we have seen, had lost confidence in the mathematical approach, this (obviously promising) evidence was lost. Alexander claimed to have a new objective measure for good architecture, but its name alone, ‘quality without a name’, is not very promising from the point of view of evidence⁸². Moreover, he did not provide much in the way of measuring tools. His strongest card is the fact that the vast majority of people have a clear preference for—put simply—older architecture over new architecture. Thus there is a quality in architecture that we cannot identify but that we can measure indirectly via the appreciation of users. Alexander’s theory in those years addressed how to reach this quality⁸³. His instrument at the time was the ‘pattern’. An Alexandrian pattern is a solution to a problem, but it is a ‘richer’ concept than IT generally takes it to be. To give an example: you could consider a good piece of apple pie, baked by your grandmother for the occasion of a simple family visit, a pattern for the solution to hungry grandchildren. It is a solution to this nasty problem, but it is much more, it is a source of all sorts of positive sensations, associations, and emotions, and it stimulates and structures a pleasant social event. Such an apple pie definitively has the quality without a name. The recipe for this pattern is more complicated, however, than what is scribbled on the piece of paper grandma hands you.

By now, almost 30 years later, it is clear that Alexander did not start a scientific revolution, although his work has always received high academic credits⁸⁴. Alexander himself considers his quest successful. A remarkably clear statement by Alexander in this respect is found in the foreword that he wrote in 1996 for a book about the use of patterns in software engineering⁸⁵:

[80] Alexander himself calls it a biography in his foreword of the book, and in fact it is no doubt the best biographic source available about Alexander.
[82] “There is a central quality which is the root criterion of life and spirit in a man, a town, a building, or a wilderness. This quality is objective and precise, but it cannot be named.” From: Alexander, C., “A Timeless Way of Building”, Oxford University Press, 1979, p. 19. Alexander does not explain the choice of this uninformative name, but he may have meant something along the lines of Ben Okri (in: Astonishing the Gods): “When you name something it loses its existence to you. Things die a little when we name them”.
[83] It is a process which brings order out of nothing but ourselves; it cannot be attained, but it will happen of its own accord, if we will only let it. From: The Timeless Way of Building, p. 3.
[84] Alexander was elected fellow of the American Academy of Arts and Sciences in 1996, is a fellow of the Swedish Royal Society, has been the recipient of many architectural prizes and honours including the gold medal for research from the American Institute of Architects, awarded in 1970.
[85] Foreword to Richard Gabriel’s book “Patterns of Software, Tales from the Software Community”.

“We have begun to make buildings which really do have the quality I sought for all those years. It may seem immodest, to presuppose such success, but I have been accurate, painfully accurate in my criticism of my own work, for thirty years, so I must also be accurate about our success”.

Reading Grabow’s book and especially the interviews we can gather that an important condition for the quality without a name is the freedom the architect must have to do what he considers right from a perspective that may be broader than what the commissioner sees as his interest. This is in line with Grabow’s conclusion which starts with a reference to Geoffrey Scott, a historian of architecture, who around 1914 first coined the term ‘the architecture of humanism’. Scott considered the human urge to seek congruence between our own needs and feelings and the external world a deep-seated and universal instinct. We try to shape the world so that we create a “setting where we should be neither lost nor thwarted”86. We recognize fitness of the environment when we see it. Scott used the word ‘instinct’ to stress that it was something innate in humans. He considered the whole of Greek mythology ‘one vast monument to this instinct’, thus giving Vitruvius, who considered Greek sources important for the training of an architect, credit for his educational vision. Scott believes that the problem of modern architecture is that industrial society ignores this instinct, creating inauthentic architecture. Grabow sees this as a general problem of the 20th century sophisticated industrialized society, not confined to architecture.

According to Grabow, the most valuable element of Alexander’s work is his claim to the objective nature of the ‘fitness of the environment’, however intangible the criteria for designing or measuring the result may be (‘quality without a name’).

The OOPSLA speech
The keynote speech87 for the ’96 OOPSLA88 conference is the best source for Alexander’s own view on the significance of his theory for IT. Formally speaking, even if software engineering was the focus of the conference, Alexander’s speech provides a useful contribution from the point of view of architecture.

Patterns are, in Alexander’s view, mainly used in IT for documentation and communication of construction detail, which is useful for improving efficiency. In Alexander’s words, the essential idea behind the pattern was a ‘moral’ one: making things that have a higher value for human beings. The construction process should become ‘generative’ of the intended quality through using patterns. In software engineering the pattern is a proven solution to an engineering problem. This solution may very well help to construct a system that has the ‘quality without a name’ but the relation is more indirect. Two elements are considered important for Alexandrian patterns. Firstly, it is important to let solutions grow via ‘structure-preserving transformations’, keeping certain essential ideas and elements intact. Secondly, the process must be managed in a way that is respectful of this process of growth.

Regardless of the extent to which Alexander appreciated the use of his pattern concept by software engineers, his final conclusion in the speech is that IT has an important role in

87 When Alexander was invited for the keynote speech for OOPSLA ’96 by Jim Coplien, Alexander asked what was expected from him. The answer was that it didn’t matter as long as it was about patterns, because people would always find it interesting to hear the story from the man behind the pattern concept.
88 OOPSLA: Object-Oriented Programming, Systems, Languages & Applications. It is an annual ACM conference, mainly taking place in United States.
improving the world since more and more processes depend on IT. He takes building as an example. In a typically Alexandrian analysis, he presents some calculations about the quantity of design needed to bring the world to an acceptable level of architecture. The obvious conclusion is that this can only be done with the help of automation. He envisions a paradigm shift where the role of traditional architects is taken over by people who support automated design processes. For this to happen though, he considers it necessary that IT-professionals abandon their ‘mercenary’ attitude and take responsibility.

**Analysis**

- Alexander has tried to tackle the design problem by deduction. The very attempt alone has clarified the issue of design in a fundamental way. His self-acknowledged and complete failure has cleared the way for a more realistic view of what architecting is about: inductive discovery of a solution, much as Polanyi described it for scientific knowledge. Given the fundamental nature of his analysis, the conclusion holds for IT as well.

- Although Alexander is very much concerned with the ‘attractiveness’ of architecture for users, aesthetics is just one aspect of the evolution of a design. The design process is based on adaptation (to the terrain, to users) rather than on aesthetic vision. On the other hand, as Grabow has pointed out, an important feature of Alexander’s theory is his claim of objectivity when it comes to the right result.

- A holistic process is what matters most to Alexander. ‘Guided growth’ is the method of choice. Architect, user and builder should preferably all blend into this process. This part of his theory can seemingly be translated to IT with ease, since within IT itself many have come to embrace iterative and incremental development methodologies. There is, however, an issue here: many IT-people would not consider such development an ‘act of architecture’.

- Alexander sees a strong moral engagement as a necessary condition for good architecture. There are two main concerns. One is that the architect must aim high: a superb result, as a matter of professional obligation. The other is that the architect must take into account the impact of his designs on the ‘users’, taking a broad scope. The commissioner and the government are rather invisible with Alexander or are even considered hazardous to good architecture.

---

89 Polanyi, ibid.

90 Late in his career Alexander has been invested in trying to determine what exactly made people like details. He has established a limited number of principles that determine this. He sees rather fundamental aspects of human vision as the explanation for human appreciation, and implicitly sees the ‘grand designs’ of modern architecture as artificial nonsense. For IT this part of his theory is of lesser importance, with the exception of user interface design.

91 Kent Beck developed the concept ‘Xtreme Programming’ (XP) and his approach is motivated by Alexandrian principles: “...XP is an experiment in answer to the question, “How would you program if you had enough time?” Now, you can't have extra time, because this is business after all, and we are certainly playing to win. But if you had enough time, you would write tests; you would restructure the system when you learned something; you would talk a lot with fellow programmers and with the customer. Such a “mentality of sufficiency” is human, unlike the relentless drudgery of impossible, imposed deadlines that drives so much talent out of the business of programming. The mentality of sufficiency is also good business. It creates its own efficiencies, just as the mentality of scarcity creates its own waste”. From: Beck, K., “Extreme programming explained”, Addison Wesley, US, 1999.
Alexander hardly ever mentions the commissioner and sees his role as limited: it is the person who initiates the process, but is otherwise just another interested party in the process who deserves to be treated respectfully.

Interestingly, Alexander prefers the ‘design and build’ process in order to be able to take responsibility as an architect for the ‘growth’ aspect, that is, if the architect is fully in charge.

Even if the IT community does not use the ‘Alexandrian’ patterns as intended, he clearly sees IT architects to a certain extent as ‘colleagues in architecture’. This proves that Alexander considers architectural quality as something that is not predominantly involving the esthetical.

The famous ‘pattern’ is not of great interest for this study. The typical software engineering interpretation of the Alexandrian pattern is a standard solution for an engineering problem, while for Alexander the value for users was essential. The Alexandrian pattern in IT would probably better translate to successful reusable concepts in human computer interaction.

**Conclusion Alexander**

**What is architecture?**
Alexander’s answer to the question ‘What is essential about architecture?’ is remarkably similar to that of Vitruvius: process is dominant. Both have pointed to the importance of the application of scientific knowledge to improving the practice of architecting. Both have also pointed to the importance of tacit knowledge and personal qualities.

The main contribution of Alexander’s theory to the object of this study is that the explicit deduction of architecture on the basis of a list of requirements is an indication that engineering methodology has been applied to a problem that possibly requires an architectural approach. We can say that Alexander made explicit what in the case of Vitruvius remained implicit, however clear Vitruvius’ message may be for those willing to read it.

**What is good architecture?**
Good quality as a result of architecting is based on a strong moral commitment of the architect to map the commissioner's needs and intentions, find a solution that improves the environment from a humanistic point of view, and involves this environment in the design process.

The solution must preferably mature slowly and adapt to the context.

**How do Alexandrian criteria translate to IT?**
Alexander’s preference with respect to slow maturation can more easily be realized in IT than in the building sector, since in IT iterative and incremental development is an established good practice.\(^{92}\)

---

\(^{92}\) Obviously, the flexible nature of IT is also a risk that must be managed: iterations are necessary to find out what the commissioner really needs and must be done as economically as possible as part of the architecting process. Engineering must be done as engineering should be done: methodically and with predictable results.
The Alexandrian pattern—as he intended it—has up till now not really been used in IT (architecture). Alexandrian quality standards with regard to the architecting process can in fact be readily applied in systems architecting, but the acceptance of induction and the necessity to ‘grow’ solutions will meet with severe resistance from those who prefer to see the operation of the state as strictly rational and planned top-down. This will be addressed in more detail with the next author, Scott.

**Summary of the results**

In the following table we summarize the results for Alexander. We use an article\(^93\) dating from 1991 by Alexander that we have not treated before, but which (in part literally) sums up his own conclusions. It is difficult, however, to appreciate these conclusions without following the developments leading up to these conclusions as they can be found in Grabow, dating from almost twenty years earlier.

<table>
<thead>
<tr>
<th>Actor/object</th>
<th>Quality aspect or related role</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commissioner</td>
<td>Commissioner hardly mentioned, or only as a problem</td>
<td>Grabow, p. 144, 177.</td>
</tr>
<tr>
<td></td>
<td>Combining role of architect and builder preferably</td>
<td>Grabow, ibid, p. 172, 175, 222. Alexander, ibid, p. 111.</td>
</tr>
<tr>
<td></td>
<td>Hands-on experience with building</td>
<td>Grabow, ibid, p. 88, 111, 112, points 1, 8, 14.</td>
</tr>
<tr>
<td></td>
<td>Is responsible for the prevention of cost overrun</td>
<td>Alexander, ibid, p. 111.</td>
</tr>
<tr>
<td></td>
<td>Is responsible legally</td>
<td>Alexander, ibid, p. 111.</td>
</tr>
<tr>
<td></td>
<td>Requirements must be met, not optimized</td>
<td>Alexander, “Notes”, p. 99.</td>
</tr>
<tr>
<td></td>
<td>Must be the result of a ‘generative’ process, a process of growth</td>
<td>Grabow, ibid, p. 138-39.</td>
</tr>
<tr>
<td></td>
<td>Must have the quality without a name</td>
<td>Grabow, ibid, 21, 68. Alexander, ibid, p. 111.</td>
</tr>
<tr>
<td></td>
<td>Guided by users and their interests</td>
<td>Grabow, ibid, p. 153, 154, 162, 169. Alexander, ibid, p. 112.</td>
</tr>
</tbody>
</table>

3.1.3 James Scott

James Scott is a professor of Political Sciences and Anthropology at Yale University who has studied, broadly speaking, the use of land and related (government) policies. In the case of Scott we cannot expect to find answers about the concept of architecture, but rather about an important category of risks facing larger public sector ambitions changing the physical world. His views can help though to find out about the specifics of large scale policy development and implementation that good architecture must address adequately.

In his book “Seeing Like a State”\(^\text{94}\) he shows how policy making often fails to take into account the complexity and diversity of real world situations. This results in both ineffective policy and missed opportunities. The relation with IT architecture, especially in the public sector, is that often IT projects are supposed to implement policy.

The urge to simplify comes from ‘Authoritarian High Modernism’, an early 20\(^\text{th}\) century phenomenon Scott describes as depending on three factors. Firstly, there was the belief in linear progress, absolute truths, and rational planning of ideal social orders under standardized conditions of knowledge and production. The inventors and main proponents of this kind of thinking were the avant-garde among engineers, planners, technocrats, high-level administrators, architects, scientists, and visionaries\(^\text{95}\). Secondly, the modern state provided the power and apparatus to enforce the necessary sophisticated control. Thirdly, Scott notices that a weakened civil society leaves room for visionaries to develop grandiose and utopian plans. On the extreme political right, Nazism is the famous example. On the left wing, progressive, often revolutionary, elites have wrought havoc using ‘the armoury of holistic social engineering’. Even if problematic IT projects are not in this general category, there are parallels: these projects are also ‘well intentioned’, large scale, visionary, and often constrain citizens. Another factor that makes the theory interesting for our purposes is the fact that high-level administrators, engineers and architects were among those susceptible to the temptations of High Modernism.

Seeing Like a State

A fundamental problem of policy on the level of the state is that reality needs to be simplified to a large extent in order to be able to design a ‘fitting’ policy. This simplification means that phenomena in the field of interest are categorized, and subsequently treated as if they are all equal to the mean of their category. It may also mean that reality is adapted to fit that mean. German forestry policy invented the ideal ‘standard tree’ (Normalbaum), which, when implemented too literally, lead to large scale ecological problems.

Scott has identified five ways\(^\text{96}\) in which a state reduces reality: only those aspects of social life that are of official interest are taken into account; only documentary information counts (numbers, description); only relatively static facts count; often the relevant facts are aggregate facts; and finally, standardization and categorization often serve to classify humans.


\(^{95}\) Ibid, p. 88.

\(^{96}\) Ibid, p. 80.
A reality that is standardized accordingly can be reasoned about more easily. Scott gives two examples of radical large-scale policies that, on the one hand, show obvious advantages but, on the other hand, show that the necessary conditions to make this kind of ‘rigorous and consistent’ application of policy feasible, are hardly desirable.

- The first example Scott presents\(^97\) is the amazingly efficient cooperation between Germany and AEG during the First World War. Scott sees this simplification of reality as typical not only of states but also of large organizations. Large organizations do it in an even more radical form because organizations are in fact more homogeneous, being goal oriented and under strict hierarchical command. This means that states and large organisations have a natural basis for cooperation. The engineer Walther Rathenau from AEG, an interested private party, was asked to design a national coordination scheme for raw materials that were of vital importance to the war industry. In only a couple of months he managed to not only design it, but also put it into practice effectively, ignoring all considerations of private property and market relations. Rathenau was a broadly educated intellectual, and he drew the conclusion that his success proved that after the war the global economy should be turned into an ‘uninterrupted community of production and harmony’. He had followers from all political creeds. Yet, this success was never repeated since only under a condition of widely available technical resources and totalitarian leadership can this approach be successful. Much policymaking, however, assumes the cooperation of all involved, underestimating the disruptions that are caused by the implementation of the policy.

- In a similar vein, the French architect Le Corbusier drew the conclusion that having a ‘Plan dictateur’\(^98\) was the only way to realize serious improvements in city environments. The monstrous plans—visions may be a better word—Le Corbusier developed for many large cities show no consideration for anything that may have been there before. Yet, his analysis of the existing situations was in part correct: many older parts of these cities were prone to serious problems of public health and public order that could not easily—that is, on the basis of a relatively simple plan—be overcome in the existing context. The plan of Le Corbusier was simple: ‘tear it all down and build something straight’. Especially streets, so often the scene of problematic city phenomena as a result of their multifunctional and open nature, were to be abolished altogether: the principle of ‘death of the street’. None of Le Corbusier’s large scale reconstruction plans was carried out. Scott’s conclusion\(^99\) is that the simplifying models underlying these visionary approaches are an important cause for their failure. Planned economies fail to follow the countless and complicated changes in the actual processes of production, their relative efficiencies and related social contexts. Likewise, ignoring the concrete interests of individual house owners and tenants in large cities on the grounds that sometime in the future their lot will (on average) be improved is bound to mobilize effective opposition.

This effective resistance could be seen as an obstacle to necessary innovation. The fact that Le Corbusier’s ‘Ville radieuse’ was not realised did not make it could be seen as a dramatic case of missed opportunity. Old buildings and crooked streets remain where much more adequate and efficient new construction should have taken its place, to the benefit of all. Then, how can it be that these old cities are today in fact considered to be attractive environments? Scott

\(^97\) Ibid, p. 98.  
\(^98\) Ibid, p. 111.  
\(^99\) Ibid, p. 342.
refers to the analysis of Jane Jacobs\textsuperscript{100} who gave voice to the countermovement in the 1960s by explaining how it was exactly the multifunctional street and diversity generally speaking that made cities not only survive but in fact flourish. Alexander’s previously mentioned article ‘A city is not a tree’\textsuperscript{101}, which was published a few years later, confirmed her analysis: the multitude of relations between very different functions cannot only be accommodated by a concept as simple as the street, but the complex maze structure in fact determines its contribution to attractive city life. The tree design where every entity (house, shop) is a leaf on the end of a branch (residential branch, shopping branch), actually has a deadening effect.

Jacobs describes the mentality of politicians and the public service, intent on large scale reconstruction of old neighbourhoods, as follows: “An all too familiar kind of mind is obviously at work here: a mind seeing only disorder where a most intricate and unique order exists, the same kind of mind that sees only disorder in the life of city streets, and itches to erase it, standardize it, suburbanize it”\textsuperscript{102}.

Scott uses the Greek concept ‘metis’\textsuperscript{103} to identify the typically human capabilities that Jane Jacobs sees as responsible for the survival of old cities and which are neither seen nor respected by the typical bureaucrat. ‘Metis’ stands for practical knowledge, to be applied in dynamic circumstances: roadside repair of vehicles, bicycling, sailing a boat. Things you cannot learn to do effectively from a book even if there is lots of theory. The capability is also characterized by local knowledge: the farmer, who knows his soil, the pilot steering boats through dangerous waters. Language is a typical example: the knowledge is local and mastering the language to perfection is possible without understanding a single rule of grammar. This typically human capability is at work – given favourable circumstances – in neighbourhoods. This capability is not seen by official policy, and therefore it is ignored and replaced by one-dimensional approaches to problems.

Scott’s conclusion\textsuperscript{104} is that there is a strong tendency to overestimate the effectiveness of central planning in issues where the interaction of people and their environment is at stake, and to underestimate the problem-solving capabilities of existing structures of social interaction.

His advice\textsuperscript{105} is to avoid this pitfall by adopting an approach that is characterized by a policy that is modest and interactive: take small steps, favour reversibility, plan on surprises and human inventiveness.

\textit{Analysis}

On the basis of Scott’s theory we can look at the relationship between policy and architecture. Policy and architecture are related concepts. Policy is about the choice of means used to reach certain ends, as is architecture.

Ex ante policy evaluation has much in common with validation of architecture, in that one must establish if a design, be it a policy or an IT architecture, is adequate. In public

\begin{footnotesize}
\begin{enumerate}
\item Ibid, p. 132.
\item See footnote 77.
\item Scott, ibid, p. 311.
\item Ibid, p. 344.
\item Ibid, p. 345.
\end{enumerate}
\end{footnotesize}
administration in democratically governed states the procedure for implementing this check is known as the democratic process: policy proposals are sent to the representative body. It is interesting to note that that apparently this check is not a technical matter, to be decided, ultimately, by competent professionals. This is consistent with the similarity between policy and architecture: both deliver solutions but one cannot deductively prove if they are ‘the best’ solutions. The democratic process – more especially open debate – seems to be the best solution the world has found so far for establishing the quality of a policy ex ante and making it acceptable. However, it may not be enough for the following reasons.

- Firstly, interactivity requires preferably the involvement of those directly concerned, rather than just a ‘representative body’. In practice it is difficult to organize a useful public debate about ambitious undertakings. The value of Internet- and Twitter-based public debate is yet to be established. It is necessary, therefore, that the public service takes a more ‘Popperian’ attitude towards its own plans by trying to find fault with them through discussions with those directly affected by their proposals. Face to face (rather than face to Facebook) feedback from a sufficient number of citizens may help to get a balanced opinion.

- Secondly, if IT is involved in large-scale policy issues, public debate may be hampered by a strong tendency to overlook the policy and concentrate on implementation details. Large scale solutions generate a lot of press coverage and the combination of these effects may lead to a public debate that fails to address the real impact.

Taking Scott’s concerns seriously is not a matter of taking ‘small is beautiful’ as a universal point of departure. Advantages of scale and standardization do play a large role when IT is involved. Implementation of local public transport chip cards would almost certainly have had advantages, but the disadvantages for the traveller are obvious.

**Conclusion Scott**

We can learn from Scott that large scale indiscriminate designs as a solution for a multitude of supposedly similar problems with a social component are rarely successful. The reason is that the problems have only become ‘similar’ after a process of unwarranted reduction of reality. The efficiency of ‘once and for all’ solutions must not be overestimated. Scott’s analysis highlights an important category of problems that may exist with requirements that are in fact the amateurish architectural ideas of the commissioner. Such a misguided vision may not be corrected because large scale IT tends to be much more ‘media genic’ than the policy issue itself. IT-related implementation issues such as security or possibilities to manipulate, sabotage or at least undermine the system may dominate the press by the time the proposal reaches parliament.

The worst case scenario, according to Scott, is a design that consists in part of social engineering in order to adapt the social reality to the scheme the solution itself is based on. The approach may intellectually be consistent, but is in practice, apart from ethical considerations, rarely effective.

---

106 A security problem of the public transport chip card that is implemented in the Netherlands is a typical example of an implementation detail that drew an enormous amount of attention.
Scott does not advise against using epistemic knowledge as an important factor in a specific design. When it comes to the necessary changes he advises avoiding the ‘grand design’ and opting for a more dynamic approach to large scale developments: take small steps, favour reversibility, plan on surprises and plan on human inventiveness.

**Summarizing the results**

In the following table we have again summarized the findings and provided sources. However, the mapping of criteria to the actor/object in the first column is, this time, largely our own interpretation, where we attributed criteria requiring ‘strategic wisdom’ to the commissioner and the ‘good practice’ criteria to architecture. The architect, as such, is, in the case of Scott, the public service. The role/quality column this time contains measures/advice to avoid the risks of ambitious projects based on unwarranted simplifications.

<table>
<thead>
<tr>
<th>Actor/object</th>
<th>Quality aspect or related role</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commissioner</td>
<td>Must resist the temptation of simple design solving problems involving complex social behaviour</td>
<td>Scott, p. 343.</td>
</tr>
<tr>
<td></td>
<td>Should not believe that the future can be predicted because some impact on it is being designed</td>
<td>Scott, p. 345.</td>
</tr>
<tr>
<td>Architecture</td>
<td>Take small steps</td>
<td>Scott, p. 345.</td>
</tr>
<tr>
<td></td>
<td>Favour reversibility</td>
<td>Scott, p. 345.</td>
</tr>
<tr>
<td></td>
<td>Plan on surprises</td>
<td>Scott, p. 345.</td>
</tr>
<tr>
<td></td>
<td>Plan on human inventiveness</td>
<td>Scott, p. 345.</td>
</tr>
<tr>
<td></td>
<td>Must respect local knowledge and associated skills dealing with changing circumstances</td>
<td>Scott, p. 351-52.</td>
</tr>
<tr>
<td></td>
<td>Must not depend on social engineering</td>
<td>Scott, p. 348.</td>
</tr>
<tr>
<td></td>
<td>Must enhance knowledge, skills, initiative, morale and experience of users</td>
<td>Scott, p. 349, 356.</td>
</tr>
<tr>
<td></td>
<td>Must be attractive for users</td>
<td>Scott, p. 356.</td>
</tr>
</tbody>
</table>

### 3.1.4 Conclusion: architecture and its key quality aspects

We have now come to the point where we have to try to draw the overall conclusions about what is architecture and when is it good architecture. We will identify the essential characteristics of architecture, select key quality aspects, discuss the relevance for 21st century IT in the public sector, and then summarize.

As we did in the last paragraphs, we will present a combination of tables, diagrams and text. This time we will present the tables first since we use them to extract conclusions in a systematic and complete manner. The texts explain the significance of the findings for the research questions.

We will start by compounding the tables for the three authors into one. In a first table we will simply list all the criteria for commissioner, architect and architecture, thus providing the

---

‘sum total’ of what we have found so far. In a second table we will select criteria for architecture as such. In the last table we will identify a number of key quality aspects of good architecture and link criteria to them. Finally, we will summarize the findings in text and explain the relevance for our study regarding 20th century IT.

Table 4. The ‘sum total’ of criteria derived from Vitruvius, Alexander and Scott: criteria relating to commissioner, architect and architecture

<table>
<thead>
<tr>
<th>Role</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>commissioner</td>
<td>ambition, intuitive idea, hiring the right architect, determines how luxurious, hardly mentioned, problem, avoid simple designs solving problems involving complex social behaviour, should not believe that the future can be predicted</td>
</tr>
<tr>
<td>architect</td>
<td>good reputation, proven quality, high integrity, independence and modesty, is asked for a commission and does not ask for one, appreciates advice from builders and laymen, strong moral ambition to deliver a superb result</td>
</tr>
<tr>
<td>architect</td>
<td>intellectual training broad in scope: writing of commentaries, philosophy, law, geometry, physics, astronomy, medicine, trained in theory and practice, combining role of architect and builder preferably, hands-on experience with building</td>
</tr>
<tr>
<td>architect</td>
<td>responsible for the contract with the builder, responsible for the prevention of cost overrun, legal responsibility</td>
</tr>
<tr>
<td>architecture</td>
<td>strong and durable, functional, attractive to users, not be detrimental to the health of users, fits the economic status and possibilities of the commissioner in compliance with building code, relies on induction, not deduction, ‘quality without a name’, requirements must be met and not optimized, guided by users and their interests, must respect local knowledge and associated skills which deal with changing circumstances, enhance knowledge, skills, initiative, morale and experience of users, small steps, reversibility, plan on surprises and human inventiveness, refrain from social engineering</td>
</tr>
</tbody>
</table>

What is architecture?

In the following table we have selected the criteria which we think are the best candidates for a set of ‘necessary and sufficient criteria’ for ‘architecture as such’. We have selected these criteria by removing from the above list all criteria that are not essential for a technically sound product, that is, a product that is both functional and durable. It seems essential that the architect is trained as such and has suitable experience. However, we have left out the personal qualities, essential for ‘good architecture’. The architecture must be based on a process that is inductive; thus, the architect cannot prove that it is the best solution. The result should be functional and durable, but it will not necessarily be attractive. We have chosen the engineering quality standard as the minimum result the inductive process must deliver in order to deserve the label ‘architecture’.

Table 5. Necessary and sufficient criteria for architecture

<table>
<thead>
<tr>
<th>Role</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architect</td>
<td>trained in theory and practice</td>
</tr>
<tr>
<td>Architecture</td>
<td>relies on induction instead of deduction, functional, durable</td>
</tr>
</tbody>
</table>
In the following diagram we have shown the result. It resembles the diagram in figure 2 where we also considered the domain of designs. However, there we focussed on attractive designs and this time we do not. Furthermore, there we were dealing with a Vitruvian architect, this time the architect is lacking those specific personal qualities. In this model we have dubbed the architect an ‘opportunistic’ architect.

![Diagram](image)

**Figure 6. Necessary and sufficient criteria for architecture?**

The question is now: is architecture ‘as such’ a useful concept, does it help us to formulate a sort of minimum standard for when something can be considered architecture even if it may not be good architecture? It is obvious that the concept does not match Vitruvian or Alexandrian standards. Both consider the personal qualities of the architect to be essential. Both also require an attractive result. Scott’s more risk-oriented analysis requires architecture to have positive meaning for the environment created in the process of change.

Our conclusion is that criteria for architecture as such are hardly useful. Architecture is essentially a qualitative concept: it is architecture because it is attractive. The ‘induction’ criterion provides a negative litmus test: no induction means no architecture. But even if there is induction, if the architect is opportunistic it is doubtful if the resulting work deserves the title ‘architecture’.

Both the inductive nature of the design process and the personal qualities of the architect are necessary criteria. They are not sufficient criteria, however, as we have seen in an earlier diagram. If the result proves unattractive, it must be labelled ‘failed architecture’.
**What is good architecture?**

In the following table we have tried to group the quality aspects we found under a number of more general headers. Note that the list is influenced by the finding that process is essential: the key quality aspects apply to an architecting process that can produce good architecture. In a few cases the writers show different opinions or approaches. In these cases we have been motivated our own preferences.

<table>
<thead>
<tr>
<th>Commissioner, role and responsibilities:</th>
<th>ambition, intuitive idea, determines how luxurious, avoid simple designs solving problems involving complex social behaviour, should not believe that the future can be predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different: Alexander pays relatively little attention to the commissioner and sometimes sees him as a problem. Our position is that firstly that the role of the commissioner is in the scope of this research and the framework should take his role into account. Secondly, the integrity of the architect requires him not to accept a commission that does not grant him sufficient professional independence.</td>
<td></td>
</tr>
<tr>
<td>Architect, education and experience:</td>
<td>intellectual training broad in scope: writing of commentaries, philosophy, law, geometry, physics, astronomy, medicine, trained in theory and practice, hands-on experience with building</td>
</tr>
<tr>
<td>Architect: responsibilities:</td>
<td>responsible for the contract with the builder, responsible for the prevention of cost overrun, legal responsibility, in compliance with building code</td>
</tr>
<tr>
<td>Different: Alexander is in favour of combining the role of architect and builder in order to be able to really make the architecting process a ‘growth’ process[^108]. Our position is that it is not a good idea for the architect to be an interested party in the building process, which is economically of a different order of magnitude than the architecting process.</td>
<td></td>
</tr>
<tr>
<td>Architect, personal qualities:</td>
<td>high integrity, modesty, strong moral ambition to deliver a superb result</td>
</tr>
<tr>
<td>Assignment:</td>
<td>Commissioner: hiring the right architect, of good reputation and proven quality Architect: is asked for a commission and does not ask for one, independence</td>
</tr>
<tr>
<td>Architecting: the design of the architecture by the architect relies on induction, not deduction, guided by users and their interests, appreciates advice from builders and laymen, respects local knowledge and associated skills dealing with changing circumstances, enhances knowledge, skills, initiative, morale and experience of users, refrains from social engineering</td>
<td></td>
</tr>
</tbody>
</table>

[^108]: This means that Alexander is not looking for a ‘design and build’ model which is used, for instance, in housing and industry where the work to be built is chosen from the catalogue, and implemented with small adaptations at most.
In the following text we provide a more ‘readable’ version of the findings, highlighting a number of important elements.

The role and responsibilities of the commissioner

The role of the commissioner is to choose the architect with great care, so that he can and will trust the architect to act relatively independently. Vitruvius was particularly keen on a clear mandate: the role of the commissioner is to make a positive and motivated choice for a specific architect. This is rather different from a practice where architects are competing for work trying to improve their position by applauding any preconceptions the commissioner or his entourage may have about the solution. Architecture competitions may seem a good sort of ‘in between’, where the architect is stimulated to come up with a personal view and the commissioner has a choice, but the great disadvantage is that there is no possibility for the architect to really delve into the context and let the solution grow and mature. Secondly, the commissioner’s choice is not necessarily the best option. The most ‘fashionable’ name or style may win in the end.

The role, qualities and responsibilities of the architect

The role of the architect is to propose a concrete design. Both Vitruvius and Alexander have stressed the importance of quality and the fact that this quality should be evident to users. The solution must have personality, an attractive personality. This matches Scott’s concern about human inventiveness: if it’s not attractive, people will find ways and means to avoid using the system (as intended). Requirements must either be met (preferably exceeded, not compromised) or shown to be irrelevant. Meeting the requirements, however, does not in itself make a system attractive. ‘Negotiable’ requirements (such as performance between good and very good) give commissioners the feeling that they are ‘making the decisions’ and therefore get lots of attention, but these options are side effects of the main characteristics of a solution.

Architecture at the same time must improve the quality of the context. Alexander’s analysis in ‘A city is not a tree’ describes best what is meant: a newly constructed element in an existing environment will automatically create a multitude of relations, intended and not intended, on very different levels. For it to have real added value, a design must fit the logic of the environment as humans experience it. It is important to realize though that there is a limit to ‘zooming in’; things can easily get more complex instead of simpler.

An architect who just fits a solution to a list of requirements the commissioner hands him has little added value, apart from the engineering aspect. It is the role of the architect to take a critical attitude towards the commissioner’s ambitions and do his own research into the ambition the commissioner wants to invest in. In order to be able to do this the architect must take a broad view that encompasses the social, economical, biological, psychological, and technical aspects. Architecting relies on the application of tacit knowledge to the observations made in the domain by the architect, taking his commission as the point of departure. Once the initial requirements are sufficiently clear in the mind of the architect, that is, his commission is understood in the light of his observations, the actual design phase starts.
The architecting process

Alexander made clear that any claim that an architecture is the result of deduction from requirements is false. Requirements are not objective. Secondly, there are too many requirements making the calculation of a design impracticable (‘combinatorial explosion’).

Using aggregate solutions (the ‘patterns’ that made Alexander famous in IT) solves neither problem. Thirdly, Scott has made a similar point about the objectivity of requirements in the case of large-scale public policy problems. He warned of the simplifications that are the analytical basis of ‘objective’ requirements, and even more of social engineering based on this kind of analysis, manipulating people in order to make them and their behaviour fit the models.

All this is not to say that one should not take requirements seriously, but rather that one should not present a design as the outcome of an engineering process that merely translates requirements into a construction plan. The process must generate the plan.

The early phase is most typical of architecting: once the architect feels he has understood the context, a path is chosen, tentatively, where maybe countless other options would have been possible. Where engineering is application of proven method, architecting is a ‘hands on’ process, determined by context. An architectural composition is based on analysis, experience and induction. Yet, the outcome of this first phase must be recognizable and concrete from the commissioner’s point of view: Alexander considered the ‘constructive diagram’, showing the core characteristics of both problem and solution, the ideal presentation but this may not always be feasible.

The essence of architectural design can be further clarified by looking at the relation between city planning and the design of an individual building. A city plan does not provide the rationale for the existence of the city as such. It describes expected developments in the future as a backdrop for the solution to existing problems. A city plan is there mainly to curb unwanted developments and stimulate wanted developments, given the business agenda. In that sense, it is very similar to an IT enterprise architecture. However, city plans that prescribe the details of a future state (a total redesign, either by first largely demolishing what was there, or by virtue of the fact that there was nothing there in the first place) rarely bring good results. Too little potential is left for diversity, growth and natural development. One of the reasons for this is explained in Alexander’s ‘A city is not a tree’ in conjunction with Scott and Jacobs: large scale planning tends to see a city as a hierarchy of functions which it is not; everything is intertwined. It is obviously a good idea to have a look at the sewage system and other utilities while making a city plan. But if the city plan becomes too dominant, users end up with houses and neighbourhoods with great plumbing but that are otherwise impractical and unpleasant to live in.

As Scott has explained, organizations—hierarchically organized as they are—have more possibilities to enforce a vision than a democratic state. In his view, states are trying to copy the success of Taylorism in the public sphere. This is rarely successful since the fundamental liberties of citizens and the mechanisms of the democratic state prevent the enforcement needed.

If a city plan is architecture, then its calling is specific: leave as much room as possible and curb as many unwanted developments as possible with as little intervention as possible.

\[\text{\textsuperscript{109}}\] The exception is that when the state implements systems used by public servants that have no direct relevance for the public at large there is more room for enforcement.
All writers stress the importance of an architect’s sensitivity to the user’s identity and needs. Vitruvius in particular emphasizes the importance of health and attractiveness. Alexander and Scott see an even more prominent role for users; architecture is supposed to support their natural capabilities. Thus, users are co-authoring the final outcome consisting of the symbiosis of built artefact and human activity. They may also influence the very design by pointing to specific needs a design does not cover, or to disadvantages. This does not diminish the role of the architect for it is he who mediates what may very well be the most pervasive channel of user influence: the architect’s observation of users and generally of the social context in which the construction to be designed will yield its influence.

**Application in 20th century public sector IT architecture**

Now we come to the translation of these findings to the 20th century (where Vitruvius’ contribution is concerned), IT, and the public sector.

**Twentieth century**

Vitruvius’ description of the problems of architecture is, twenty centuries after its writing, still highly relevant for IT architecture, for two reasons. First, Vitruvius has painted a very vivid and recognizable (from a twentieth century IT perspective) picture of the problems of an immature architecture discipline: an unclear profile of required professional training and education and commissions given to people with commercial motivations rather than architectural talent and skills. Two thousand years later we can say that these problems—given time—can be overcome: building architecture now has a clear professional profile that is accepted worldwide.

Secondly, with his concern for user well-being, Vitruvius sets a high and modern standard. He was aware of the fact that architecture is about administering forceful technology to human existence. In order to fathom its full impact, one must be broadly educated. But, more generally, understanding the need of the client depends on broad insight into the client’s situation. In this regard modern architecting practice can still improve considerably. The ‘Architecture crisis’ has much to do with the fact that architects for quite a while seem to have lost the ability to ‘naturally’ design buildings that are ‘venustas’, that is, attractive, even if users may appreciate a design in the same way they would an abstract sculpture.

**IT**

The translation of the findings to IT is unproblematic. The phenomena and roles in the architecture of the building sector that have been discussed have clearly recognizable IT counterparts. Alexander’s historic OOPSLA speech was highly symbolic in this respect. Even if Alexander had strong doubts about the ‘application’ of his pattern theory in IT, he clearly saw IT professionals as people who have an important role in constructing the modern world. The essence of architecture is the need to apply technical means in order to improve a context, in an inductive design process, which is what happens in IT too.

The fact that it is hard to see IT architecture as aesthetic is no hindrance for seeing important parallels between building and IT. Building architecture is about much more than aesthetics. An important effect though of the esthetical aspect in the design of buildings is that it gives the architect a different status, and greater claim to independence: he is considered an artist. The IT architect struggles with his reputation of senior engineer.

‘City planning’ has a comparatively higher status in IT than in building. Enterprise Architecture is a first class citizen in IT policy both in the public sector and elsewhere. Yet,
experienced IT architects recognize that an EA is not a plan that is ‘realized’. However, the architecture of an individual IT system is often seen as a derivat of the Enterprise Architecture, which, in the case of a well-developed EA, may mean that the technical infrastructure standards are largely taken care of. This may include what is often called ‘software architecture’, which is predominantly an engineering artefact in spite of its name. The architectural aspect of the design of an individual IT system is mostly ignored, or treated as marginal, often with disastrous consequences. As in city planning, too great a dominance of enterprise architecture leads to systems based on general concerns rather than the concerns of commissioner and users. This results in systems that, if they get built at all, are suboptimal and unattractive from a user’s point of view.

**Public sector**

The findings can be readily applied to the public sector. Building has an inherently larger public sector involvement than IT. A city plan is clearly a public affair. Building codes and building authorities are standard examples of public involvement in construction in all developed economies. The more content-oriented concerns, such as the meaning of publicly commissioned works to citizens, were central in the work of Scott and, in fact, also in the case of Vitruvius as we have seen. Alexander has worked almost exclusively for public commissioners as an architect, yet his essential theory is about quality of design and the nature of the design process. It is applicable wherever architectural design takes place, whether it is in a public or a private context.

**3.1.5 Conclusion with regard to the first and second research questions**

**First research question**

“How can we define ‘IT architecture’, conceptually?”

The analysis of the concept of architecture has been presented in this chapter so far. We have concluded that it is basically pointless to separate ‘architecture’ from ‘good architecture’ since architecture is essentially a qualitative concept. To summarize, we present the following description of the concept of architecture, which we use as a basis for this study.

*Architecture is the set of functional properties of a purpose-oriented construction and the technical principles supporting it, determining its attractiveness for the context in which it is intended to function, as conceived, at least in part by inductive knowledge application with regard to the main underlying choices, by a professional architect who has been selected for the assignment by a commissioner.*

This description reflects the implicit finding that architecture is neither mysterious nor magical as long as one understands and accepts, firstly, the nature and role of induction and, secondly, the inspired, intersubjective aspect of the division of roles between architect, commissioner, and context. Acceptance may be hindered mostly by the first element, the fact that the underpinning of an important decision relies on induction. Although the vast majority of human decision-making is probably inductive in nature, when it comes to important decisions, we like them to be ‘well grounded’ in some inescapable logic derived from hard facts.
Smolander’s metaphors of architecture as either ‘blueprint’, ‘literature’, ‘language’ or ‘decision’ are in our opinion astute observations about different aspects of architecture, but do not as such reflect the essence of architecture. The richness of architecture stems from the interplay of all these elements. Corneliussen’s mapping of the metaphors to software architecture and business architecture doesn’t help either, but he may have a point where he considers ‘sensemaking’ (Weick) an important element of the architecting process. Sensemaking, according to Weick, is “less about discovery than it is about invention. To engage in sensemaking is to construct, filter, frame, create facticity ... and renders the subjective into something more tangible....sensemaking suggests the construction of that which then becomes sensible”[110].

**Second research question**

“What are the key quality aspects of an IT project architecture that should play a role in validation?”

The answer to the first research question shows why it is difficult to answer the second research question, that is, to clarify the criteria we need to verify the quality of an architecture. Once formal and straightforward requirements and assumptions about facts are verified we come to the intractable part: the induction. Finding ‘flaws’ in the architecture then becomes a matter of identifying an alternative architecture that is considered more attractive. However, this kind of critique comes too close to ‘re-architecting’. The latter approach is pointless; no doubt other approaches are possible, but the question is if the presented architecture is of good quality. An architecture involves application of tacit knowledge, which means that the architect cannot produce a full and explicit motivation for his choice.

This outcome is not unexpected, since we have already established that the authors we studied also relied on indirect, process-oriented criteria for determining the quality of architecture. These criteria concerned the architect, the conditions of his commission and the concern he has shown in the process for the interests of the context, objective motivation and careful implementation. Both Vitruvius and especially Alexander seem to consider the right process and conditions a necessary and sufficient condition for good architecture, be it that it is very difficult to get the process right. We can therefore conclude that the criteria found as summarized in Table 6 are the best indicators with a general scope that we have found. We rephrase the seven elements of Table 6 to the following seven criteria with a view to practical application: criteria 1 through 4 apply in the early phases when the architect is selected and the assignment formulated. Criteria 5 through 7 apply when the project is underway. The answer to the second research question therefore is found in the next table.

<table>
<thead>
<tr>
<th>Table 7. Key quality aspects for an architecting process resulting in good architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
</tr>
<tr>
<td><strong>2</strong></td>
</tr>
<tr>
<td><strong>3</strong></td>
</tr>
</tbody>
</table>

![Weick, ibid p. 13, 14.](image)
The architect’s commission grants him professional independence in the execution of his mission and ample possibility to acquaint himself with the context relevant to the commissioner’s ambition.

A balanced approach is chosen towards the interests of the context and local differentiation therein, applying high ethical standards.\(^{111}\)

An economic and, more generally, quantitative and qualitative analysis, for which the architect takes responsibility, underpins the architecture in order to be able to establish the value and opportunity cost of the architecture as objectively as possible.

Transition of the existing situation to the new architecture must be planned in stages that are each recognizable improvements, while at the same time creating an opportunity for adjustments to dynamics and more effective risk management unless a different approach is presented with a very strong motivation.

## 3.2 Operationalising the quality of IT architecture

Taking the concept of architecture and the key quality aspects as developed in the last paragraph as our point of departure, we will now turn to modern literature on IT and systems architecting. By critically comparing the findings from the last paragraph with writings on the quality of IT architecture we hope to be able to operationalise both architecture and key quality aspects using IT terminology. In doing so, we will try to use mainstream concepts and terminology from the IT domain as much as possible. This will make it easier to use the Architecture Validation Framework in practice. This paragraph will provide the answers to research questions three and four. In the next chapter we shall use the findings to construct the Architecture Validation Framework.

**Third research question:**
“How can we operationalise IT architecture, taking into account different levels of architecture and different approaches in current literature?”

**Fourth research question:**
“What are possible criteria per key quality aspect?”

### 3.2.1 Quality in modern literature on IT and systems architecting

The operationalisations we are looking for must be consistent with our analysis of the essence of architecture and the key quality aspects identified. For each book we first describe the major tenets. Next, we establish to what extent it matches (parts of) the concept of architecture as we have analysed it, then we look for operationalisation of the concept and key quality aspects.

\(^{111}\) This requirement may sound idealistic to some people, but for example European rules have already implemented mandatory surveys of environmental impact for major installations or preparatory planning.
We have not tried to pre-select books that match our analysis, but rather we have selected a number of books that examine IT Architecture from different angles. Two books on Enterprise Architecture are written with the Dutch context in mind. Our aim has been to establish where our analysis can be placed in relation to the mainstream of architecture generally and in the Netherlands more specifically.

We see a clear distinction in available literature between the rather different worlds of Enterprise Architecture\textsuperscript{112} and Software Architecture\textsuperscript{113}. Both have similar claims about their significance for the end result. Textbooks on Software Architecture\textsuperscript{114} take for granted that software architects are fully in control of the entire architecting process relevant for the client. Their activities are an implicitly sufficient condition for delivering first class IT to the business. Cross cutting investments by the larger organization in infrastructure, standards and innovation are considered incidental\textsuperscript{115}. The opposite view is taken in textbooks on Enterprise Architecture\textsuperscript{116}, where software is mostly treated as the engine in the perspective of a ‘car architecture’: an important element of the construction, but rather technical, an engineering artefact that should conform to the decisions made. Both traditions, using their own families of models, coexist peacefully, but nevertheless add to the confusion a commissioner may face. In this study we will consider literature from both creeds.

Enterprise Architecture (EA) is relevant because it focuses on high-level governance, which is also the case in the public sector, as investments in these architectures\textsuperscript{117} over the last decade have proven. However, our focus is on individual systems and, in that respect, Software Architecture is more in line with our analysis. An Enterprise Architecture is concerned with change: explicitly or implicitly there is a vision of the future state of the organization as a whole. EA is also concerned with standards in the organization: guidelines and solution patterns for the type of system or service that the organization typically needs. Although the EA process can be extremely useful, there is the danger that the specific characteristics of individual systems do not get sufficient attention, which poses a great risk for the implementation. Even if a full set of underlying architectures has been developed on a business, information, application and technical level, an Enterprise Architecture is still not a cookbook containing recipes for all thinkable systems.

From an academic point of view, Enterprise Architecture is not well covered if one takes the Computer Science angle. Books about Enterprise Architecture and the related topic of the (role of the) CIO are abundant, but academic backing comes mainly from the Business

\textsuperscript{112} The Gartner definition for Enterprise Architecture reads: “Enterprise Architecture is the process of translating business vision and strategy into effective enterprise change by creating, communicating and improving the key requirements, principles and models that describe the enterprise's future state and enable its evolution. The scope of the Enterprise architecture includes the people, processes, information and technology of the enterprise, and their relationships to one another and to the external environment. Enterprise architects compose holistic solutions that address the business challenges of the enterprise and support the governance needed to implement them.” Gartner Research, “Gartner Clarifies the Definition of the Term ‘Enterprise Architecture’”, ID Number: G00156559, 12 August 2008.

\textsuperscript{113} Definition: The software architecture of a program or computer system is the structure or structures of the system which comprise software elements, the externally visible properties of those elements, and the relationship among them. From: Bass, L., Clements, P., Kazman, R., “Software Architecture in Practice”, Addison-Wesley, second edition, 2003, p. 21.

\textsuperscript{114} The book we will use is an example: Bass, L., Clements, P., Kazman, R., “Software Architecture in Practice”.

\textsuperscript{115} Ibid p. 8-9.

\textsuperscript{116} See the next paragraph for examples.

\textsuperscript{117} For example, the ’Dutch Government Reference Architecture’ (NORA).
Administration discipline. The “world’s leading information technology research and advisory company” Gartner is an important source of publications\textsuperscript{118}.

We will start our analysis with two books representing the mainstream of Enterprise Architecture in the Netherlands. Next, we will take a broader view on the basis of a textbook on ‘systems architecting’, which addresses systems architecting generally and not specifically for IT. This broader view is rare and interesting with regard to our analysis, which has taken the history of building architecture as its point of departure. The book on Software Architecture is chosen because it presents explicit methods for evaluating architectures.

\textit{Van der Sanden and Sturm, the infrastructure approach}

The book\textsuperscript{119} is written from the perspective of a large scale administrative organisation. Its foreword is written by the general manager of the IT department of the Dutch tax authorities, indicating that the public sector belongs to the intended audience.

The book takes the lessons learned from the early nineties as the point of departure: interconnections between systems that used to be ‘islands’ are unavoidable, but result in unmanageable complexity. The answer is to establish an integrated infrastructure avoiding duplication of data and functionality by ensuring a strict separation of the two. The idea is that As an important advantage is seen that this infrastructure can be built and maintained without knowing the exact informational needs of the various user groups. All that one needs to know is “the average information need and variations therein”\textsuperscript{120}. The idea is that such an approach gives business optimal freedom to build applications without having to be bothered by technical detail. This is the typical ‘city planning’ approach to architecture, reflected in many Enterprise Architectures of large (public sector) administrations. ‘Data’ is the overwhelmingly dominant perspective of the book: once a coherent unified infrastructure for managing data is in place, building good application systems should be easy.

The five defining characteristics of architecture presented are: it contains a description of structure in relation to rationale, it is an instrument to manage quality, it is based on a vision, it is the result of a negotiation/optimization, and finally, it makes the link between objectives and means.

It is clear that the approach to architecture is, in many respects, the opposite of our position. The individual system is considered ‘an application’ of the infrastructure, an ‘area of change’. Context, apart from the highest level of abstraction (‘the vision’), is irrelevant; the new infrastructure is considered to be independent of business functionality. Different architects are involved in a project but there is no need for a lead architect who proposes an architecture as systems are engineered. Although the introduction mentions the necessity to complement analysis with creativity in order to create a ‘qualitative future’\textsuperscript{121}, the remainder of the book is silent on the subject. Data modelling techniques fill a large part of the book.

\textsuperscript{118} Gartner has developed a type of research that is not academic in the traditional sense, consisting mainly in collecting experience from their clients, analyzing and aggregating it and distributing the result back to their clients again. However, they employ high level expertise to do it, and they have a unique insight into what is happening on the ground worldwide. Their publications are always succinct, readable and to the point.

\textsuperscript{119} Sanden, W. van der, Sturm, B., “Informatiearchitectuur - de infrastructurele benadering” (Information architecture - the infrastructure approach), Panfox, Rosmalen, 1997.

\textsuperscript{120} Ibid p.49.

\textsuperscript{121} Ibid p. 6
Having thus established that the book is of little use for operationalisation of architecture, we have little hope that the following five general quality criteria for architecture will be of much avail. They are: architecture must be falsifiable (that is: somehow concrete), complete and relevant (but not too detailed), correct (where it models the present or future state), of practical use and finally understandable for those involved also in relation to the rationale underlying the architecture. These criteria are—if not irrelevant—too general to serve a useful purpose and no further operationalisations were found.

The architect is rather invisible in the book. Where he is mentioned his profile is that of an engineer, far removed from what our analysis wants an architect to be.

Having reached this conclusion, an obvious question to ask is to what extent the infrastructural approach has been successful. History has not shown a sharp decline in project failure since it was introduced. Either the method is hard to follow in practice, or the complexity of software engineering is not diminished to the extent intended. That said, the book is an excellent reference for all kinds of information modelling.

\textit{Van den Berg and Steenbergen, DYA}

This book\textsuperscript{122}, dating from 2004, is one in a series initiated in 2001 by Roel Wagter under the ‘brand name’ DYA, short for ‘Dynamic Architecture’ a method aiming to enable organizations to reap the benefits of architecture in terms of agility and coherence without the heaviness that seemed to characterize the ‘infrastructure approach’. The essential problem is still defined using the ‘islands’ metaphor\textsuperscript{123}. This book is written as a reflection on the experience with the DYA method in the first years after its inception. Interestingly, it devotes an entire chapter to the architect and it contains a more or less elaborated test for the quality of the architecting practices in a big organization. Before coming back to these elements, a short explanation of the DYA method is appropriate.

DYA advises organizations to first develop a shared vision on architecture, and then concentrate on the relevant strategic business agenda (“strategic dialogue”). After that, we come to the distinguishing part of the DYA method: architecture is applied in a ‘business case driven’ manner, that is, just enough, just in time. This may even imply that a solution deviates from the general IT policy of the organization: in the case of either a vital window of opportunity or an immediate threat the quickest effective remedy is chosen. DYA calls this ‘developing without architecture’ a definition that suggests that ‘infrastructural’ thinking with a taint of bureaucracy is still alive in DYA. How can a solution that is quickly realized and effective be considered ‘without architecture’? The answer must be that, although action was swift, coherence (and probably structural efficiency) has suffered.

One of the more successful concepts of the DYA method is the Project Start Architecture (PSA). The ‘official’ PSA is of operational scope, concrete and technical. That is, the PSA is supposed to operationalise an Enterprise Architecture and a Domain Architecture in the context of a specific project. We recognize the ‘infrastructure’ approach here. Many organizations, however, not disposing of (widely accepted) Enterprise and Domain Architectures use the PSA to function as a more encompassing architecture, taking into account widely accepted policies and standards of the organization. This makes DYA, in practice, sometimes more ‘business case driven’ than it actually was intended to be.


\textsuperscript{123} Ibid p. 26.
Summarizing, we can say that in this book the concrete business need is what drives IT. There is high context sensitivity. The role and position of the architect is given due attention; he is considered a critical success factor. The concept of architecture in DYA comes much closer to ours than that of Van der Sanden and Sturm, even if the position of the architect is not as central as we would like and in the background there is still a high level of infrastructural thinking. We can conclude that DYA matches our analysis in part.

With regard to operationalisation of the concept of architecture, DYA does not add anything essentially new to Van der Sanden and Sturm. The DYA Architecture Framework\(^{124}\) is a rather basic grid that can be considered a simplified version of Zachman\(^{125}\). Various definitions from the well-known sources are presented.

The book presents a number of quality criteria for the architect. An architect must be customer-oriented, have the right expertise, be empathic and predisposed to cooperation and knowledge sharing\(^{126}\). These general criteria are further elaborated in a ten point ‘code of conduct’\(^{127}\): Make sure you have a commissioner for anything you do; know your stakeholders; make a conscious investment to improve support and acceptance for your architecture; know what the organization wants; go on surveillance trips and use your ears and eyes; it is not the architecture that is sacrosanct but the business objective; share your knowledge with others; dare to present preliminary results; discuss problems with others; and strive for coherence, but do not forget speed.

Appendix A of the book contains an ‘architecture maturity test’. For this research the maturity of the architecture process in the organization may not seem directly relevant since we are interested in measuring the quality of the architecture for an individual project. However, as we have seen, we will probably have to rely on indirect criteria, such as the quality of the architect and the quality of the architecting process. The difference between an individual process and the architecture maturity of the organisation running those processes may not be all that big. Therefore, we will consider if the test can be used for our purposes.

The presented ‘architecture maturity test’ consists of 18 focus points. Per focus point criteria are formulated to distinguish two to four levels of maturity. Assuming a ‘natural’ order and tempo in the development of the various focus points, 13 maturity levels have been defined from initial to perfection. Although the instrument as such is well designed, the criteria reflect a professional profile of the architect that is below any minimum standard that can be derived from our initial analysis. ‘Architect’ is considered a role that can be attributed to IT staff who work ‘a minimum of 16 hours a week’.

The conclusion is that, with regard to the architect, the professional profile is too modest to serve as a model for our purposes, even if a number of criteria match our vision. The maturity test can possibly serve as a source for operationalisation, even if the set of focus points and criteria may need adapting.

**Rechtin, Systems architecting**

“The Art of Systems Architecting”\(^{128}\) has proven to be the richest source aiding us in linking the findings from our conceptual analysis to the practice of modern architecting. The reasons

\(^{124}\) Ibid p. 63.
\(^{126}\) Ibid p. 111.
\(^{127}\) Ibid p. 128.
for this are the following. First, the book is about architecting generally, not in any specific technology domain, thereby matching the level of abstraction that we have applied in our analysis. Secondly, Rechtin is about architecting systems of high technical sophistication, typically outside the building domain, which means that there is a greater chance of finding useful operationalisations relevant for IT. Thirdly, some IT-specific operationalisation is provided. IT is seen as a discipline that has adopted the idea of ‘architecture’ relatively quickly, but needs help in developing ‘the formal partnership with architecting that structural engineering has long enjoyed’\(^\text{129}\). Fourthly, the public sector is well represented, a whole chapter is devoted to the relation between systems architecting and the political process. Fifth, there is a focus on the individual system as is the case in this study. Sixth, useful distinctions are made, such as a chapter on ‘builder architected systems’, which for IT translates to architecting in legacy environments, but also a chapter on ‘social systems’ where client and user are very different entities. Finally, Rechtin sees inductive reasoning as the essence of architecting, thereby matching our analysis.

The explicit focus on induction as the basis of architecture and the perspective of the individual system are indications that the concept of architecture of Rechtin is close to ours, but there is more. The role of the architect and the concern for the context, specifically for public sector systems, are equally close, making it an almost perfect match. Rechtin’s ideal architect in the public service comes close to what Vitruvius was: a public servant serving the state, rather than a department, granting him a certain independence. Obviously, only in rare cases can an architect in our days hope to work for a head of state personally and address him directly as Vitruvius did (or at least intended to do)\(^\text{130}\). In Rechtin’s final conclusion he states that systems architecting ‘could well evolve as a separate business entity’ following ‘comparable developments in the history of classical architecture’\(^\text{131}\).

We will first establish which elements can help us operationalise architecture. We will start with ‘inductive reasoning’, which Rechtin clarifies as follows. He distinguishes four ‘methodologies’ of architecting: normative (following fixed rules), rational (analytical, method- and rule based), participative (stakeholder-based) and heuristic (lessons learned). The first two are considered deductive (engineering, ‘science’), the latter two inductive (‘art’). Rechtin is about the art, seeing the participative approach aiming at consensus as a necessity given the complexities one is dealing with\(^\text{132}\). Heuristics provide guidelines for ‘common sense’. They answer the question: What is sensible in a given context? Rechtin sees the art as the methodologies the architect is involved in personally, and engineering\(^\text{133}\) as a ‘tool’ he uses in the process of architecting, but where the actual work is done by others.

\(^\text{129}\) Ibid p.9.
\(^\text{130}\) There is a famous example in building architecture that has inspired IT architects: the commission given by the French president François Mitterand to the architect I.M. Pei for the refurbishment of the Louvre in Paris. Pei designed the glass pyramid which initially aroused such indignation among the Parisians that the design would not have survived if it were not for the fact that the president took responsibility personally for the architect that he had chosen. The example is used in the book by Marc Sewell and Laura Sewell ‘The Software Architect’s Profession’. The picture of the pyramid can also be found on the front cover of the book on software architecture that we have used, Bass et al.
\(^\text{131}\) Ibid Rechtin, p. 225.
\(^\text{132}\) Ibid p. 4: Rechtin makes very clear that the architect must remain in control of this process in order to reach a result. As the weaknesses of ‘concurrent engineering’ he mentions: ‘undisciplined design by committee’, ‘diversionary brainstorming’, ‘group think’ and ‘members without power to make decisions but with unbridled right to second guess’.
\(^\text{133}\) The preface of the book contains a paragraph on the difference between architecture and engineering ‘Because it is ......most asked by engineers in the new fields......’. This is further elaborated on in the introduction of Part one and on p. 164.
For Rechtin, heuristics is the most objective operationalisation of the wisdom of the architect. The prominence of heuristics throughout the text is a distinguishing characteristic of the book. There are descriptive and prescriptive heuristics. Some are context bound, others apply to all architecting.

Of the books we studied, only Rechtin distinguishes art explicitly from engineering. The other sources also recognize that initial requirements should not be taken at face value, as well as the necessity and influence of social processes. Yet in their case the gap between the initial requirements and the architecture of the finished product is filled with analysis and modelling activity, striving for coherence and optimization, that is, sensible engineering. The role of the architect is, in that case, implicit: either the intuition of the commissioner about the architecture is accepted uncritically, or the architecture is presented as the unequivocal result of an engineering effort and weighted voting among stakeholders. Only the DYA book takes an intermediate position in that it recognizes the importance of ‘momentum’ for architecture, its business- (case) driven nature.

Rechtin presents six ‘fundamentals of architecting’\(^\text{134}\), that can be considered further operationalisations of architecture, the role of the architect and his ‘wisdom’. We can summarize them as follows. The first is ‘a systems approach’: this is at the core and involves envisioning the system in terms of primarily environment, components and interfaces between them, the whole embodying value judgements of what is required and what is feasible. The second is ‘a purpose orientation’: architecting is about a useful purpose, an affordable cost, and an acceptable period of time. The client obviously is the judge here, and must be actively involved. The architect’s role is to concentrate primarily on integrity of purpose. The third is ‘a modelling methodology’: this hardly needs explanation. Rechtin stresses the fact that models are not only a tool for the conceptualization phase of the system. Models evolve and accompany the system during its entire lifespan. Examples are models for engineering, demonstration, subsystems and mental models of users. The fourth, ‘Ultra quality implementation’, is a much less obvious element (and an intriguing one from an IT point of view). This reflects the fact that in highly complex systems there is no tolerance for errors, since the effect of an error being propagated through the system while being compounded with others is soon disastrous. In fact, IT is no exception when it comes to hardware such as a microprocessor: they function flawlessly. However, software engineering is a different case. One trend in software engineering that has similar ambitions, which is highly relevant for the public sector, is rule engines. Although the idea of the rule engine has been around for decades, we now gradually see bigger vendors\(^\text{135}\) and more mature products\(^\text{136}\). The success of the approach, however, remains to be proven. The idea is that instead of trying to make a programmer understand what the regulation is about, which he then translates into code, the rule expert himself uses a tool to feed the rules in a structured way into the system, which then uses them to decide cases. Since the rule engine is a mathematically proven machine, the result is ‘ultra quality’. And if the system cannot figure out what to do, the rules are apparently inconsistent. Advocates of this technology claim that ‘faults’ in legislation can be detected early, which could lead to better quality regulations. This view does not take into account that regulations are political compromises, which are sometimes deliberately vague. Even so, the technology is potentially very powerful. The fifth point, ‘Certification’, refers to the architect certifying to the client that the system meets his acceptance criteria and that the builder can be paid. This role requires that there is no perception of a conflict of interest

\(^{134}\) Ibid. p. 9.
\(^{135}\) For example, the product Oracle Policy Administration.
\(^{136}\) For example, the Dutch product Beinformed.
concerning the architect. That is, he must avoid value judgements, have no commercial interests directly or indirectly in the matter, and keep at an arm’s length of project management responsibilities. The last point is necessary in the first place to avoid being bogged down in operational detail, losing the architectural focus on integrity of purpose, and secondly, again to avoid (perception of) commercial interest, for example, by giving project work to himself. The last point, ‘Insight’, refers to the ability to structure a complex situation in a way that greatly increases the understanding of it. Experience is key here, but also the use of heuristics as a codified form of the experience of the profession.

We now come to quality criteria, starting with the architect. Apart from the heuristics, the architect must dispose of high levels of inspiration and skill\textsuperscript{137}. In that sense, Rechtin considers the best architects to be like ‘real’ artists. Just as Vitruvius before him, Rechtin sees the education of the architect as the best guarantee for quality. The eleventh and final chapter of the book is devoted to it. When it comes to the ‘art’ aspect in the curriculum, his approach is less visionary than that of Vitruvius. It aims at high level abstractions of design phenomena such as ‘complexity’ and ‘problem solving’. Even so, Alexander and his dissertation are mentioned as a landmark in ‘the practice of architecting’\textsuperscript{138}.

Rechtin is very aware of the necessity for the architect to respond to context. Rechtin distinguishes four specific contexts and devotes a chapter to each. The first context is builder architected systems (architected and produced by the same entity that sells them on the market) and the second is manufacturing systems (architecting production systems, the products of which are what reaches the end client). The third is social systems (where large groups of clients depend directly on the system) and in the fourth chapter Rechtin distinguishes software systems but mainly in the sense of ‘software intensive systems’, which are installations that depend heavily on software. This chapter, in spite of its subject, has little to offer us, since it is more about the specifics of producing software than about architecting. When it comes to architecting large scale public sector initiatives, Rechtin clearly sees this as a specialized form of architecting. Part IV of the book ‘The Systems Architecting Profession’ consists of Chapter 10 ‘The Political Process and Systems Architecting’ by Brenda Foreman and the aforementioned Chapter 11 on the education of the architect. The chapter on public sector architecting is written from a purely American perspective, yet the essence can easily be transferred to any other democratic state. The bottom line is expressed by the heuristic: ‘If the politics don’t fly, the system never will’\textsuperscript{139}. The chapter is the perfect complement to Scott’s theory that we presented before, where Scott stressed what can go wrong even if (initially) the politics do fly. One important lesson is shared among both visions (in the words of Foreman): “It helps to have stable and operationally useful interim configurations and fall back positions”. This clearly echoes the principles of James Scott. This chapter helps us develop criteria for the quality of the architect and the process of architecting, specifically when dealing with larger public sector systems.

Rechtin, explicitly seeking the analogy with the Capability Maturity Model of the Software Engineering Institute, proposes a set of five criteria\textsuperscript{140} that can be used to establish the

\textsuperscript{137} Ibid p. 17-18.
\textsuperscript{138} Ibid p. 224.
\textsuperscript{139} It might be remarked here that the chapter could be considered stigmatizing for the public sector. Seemingly ‘irrational’ decision-making is not an exclusively public sector phenomenon, since it is found in all organizations where different rationalities have to compromise. A more theoretical and abstracted approach to the same phenomenon is found in March, J.G., Olsen, J.P., “Ambiguity and Choice in Organizations”, Universitetsforlaget, Bergen, 1976.
\textsuperscript{140} Rechtin, ibid, preface.
maturity and effectiveness of architecting in projects: recognition by clients and others of the need to architect complex systems instead of engineering them; an accepted discipline to perform that function, in particular, the existence of architectural methods, standards, and organizations; a recognized separation of value judgements and technical decisions between client, architect, and builder; a recognition that architecture is an art as well as a science, in particular; the development and use of non-analytic as well as analytic techniques; the effective utilization of an educated professional cadre, that is, of master-level, if not doctorate-level, individuals and teams engaged in the process of systems-level architecting. This can be seen as a less sophisticated version of the architecture maturity test of Steenbergen et al.; yet the criteria are more appealing.

The conclusion is that Rechtin offers a basis for a number of useful operationalisations.

Firstly, the concept of architecture is put into perspective by differentiating between different contexts in which architecture is practiced. The distinction between builder architected systems and independently architected systems is a useful concept. Sociotechnical systems where client and user are the same person are contrasted with manufacturing systems, where the end user of a product has no direct contact with the architected production system. For example in the case of large back office transaction processing systems used for social benefits, we can see that these distinctions are relevant: these systems tend to be builder architected (legacy), manufacturing systems rather than sociotechnical systems. They are operated by professionals, not by the people receiving the benefits.

Secondly, Rechtin provides us with the heuristics that help us to deal with a great number of architectural concerns in different contexts. This provides a foothold, enabling us to establish more objectively if the inductive part of the work of the architect is of sufficient quality.

Thirdly, Rechtin’s descriptions of architecting in different contexts provide us with input for quality criteria for the architecting process.

Fourthly, the quality of the architect can be established along Vitruvian criteria, be it a more down to earth and modern version. His guidelines for education are concrete.

Fifthly, although Rechtin does not very specifically attribute a strong ethical responsibility to the architect, he does recognize the issue as such. He typically looks at legislation to protect end users from the pressure commissioners may put on their architects to take risks with the interests of those who will suffer the consequences of the construction to be architected. Other conflicts of interest should be alleviated by ‘socioeconomic’ bargaining between stakeholders.

_Bass, Clements, Kazman, Software Architecting in Practice_

A software architecture is defined in this book in terms of the structure of the system in relation to its behaviour. The focus on the individual system, its context and the role of the architect all match our concept of architecture. However, induction is implicit and the characteristics of the role of the architect point to engineering more than to architecting.

---

141 Ibid p. 77-79
Requirements are considered to be problematic as the following statement shows: “Almost never are the properties required by the business and organizational goals consciously understood, let alone fully articulated”\(^{143}\). Yet, this highly realistic analysis does not lead to the conclusion that the architect has a role more elevated than extorting ‘requirements’ from ‘identified stakeholders’. The phrase “It should be apparent that architects need more than just technical skills” is another (under)statement indicating that the business context is seen as a mess, but even so, a mess the improvement of which is part of the job of the architect. This is reflected in part in the seven point list of quality requirements for the architecting process. We reproduce the full list here.

“The architecture should be the product of a single architect or a small group of architects with an identified leader. The architect should have the functional requirements for the system and an articulated, prioritized list of quality attributes (such as security or modifiability) that the architecture is expected to satisfy. The architecture should be well documented, with at least one static view and one dynamic view, using an agreed-on notation that all stakeholders can understand with a minimum of effort. The architecture should be circulated to the system’s stakeholders, who should be actively involved in its review. The architecture should be analyzed for applicable quantitative measures (such as maximum throughput) and formally evaluated for quality attributes before it is too late to make changes. The architecture should lend itself to incremental implementation via the creation of a ‘skeletal’ system in which the communication paths are exercised but which at first has minimal functionality. This skeletal system can then be used to “grow” the system incrementally, easing the integration and testing efforts. The architecture should result in a specific (and small) set of resource contention areas, the resolution of which is clearly specified, circulated, and maintained. For example, if network utilization is an area of concern, the architect should produce (and enforce) guidelines for a development team that will result in a minimum of network traffic. If performance is a concern, the architect should produce (and enforce) time budgets for the major threads.”

We recognize three points from our criteria: the architect as an individual with a central and encompassing role, the involvement of stakeholders, and the incremental approach starting with a ‘skeletal’ system (prototype). No further operationalisations are provided however.

When it comes to quality criteria, the book describes the Architecture Tradeoff Analysis Method (ATAM)\(^{144}\). ATAM aims at technical optimization of quality attributes given functionality (for example, balancing performance against security). However, this is done when the architectural choices we are interested in have been made\(^{145}\). A later extension of ATAM, in part correcting its limitations, is the Cost Benefit Analysis Method (CBAM)\(^{146}\). CBAM’s view is economic, quantifying different ways of meeting requirements in terms of cost and benefit, thereby refining the business case. Neither addresses the fundamental choices that we consider the essence of architecture but even so it is clear that the combination of CBAM and ATAM provides a basis for an explicitly documented rationale.

\(^{143}\) Ibid p.9.

\(^{144}\) Ibid Bass, p. 271, or [http://www.sei.cmu.edu/reports/00tr004.pdf](http://www.sei.cmu.edu/reports/00tr004.pdf), for the full report, accessed January 30, 2011.

\(^{145}\) Ibid Bass, p. 72 “Functionality and quality attributes are orthogonal”. Bass was also one of the authors of ATAM.

\(^{146}\) Ibid Bass, p. 307.
The conclusion is that the Software Architecting tradition as expressed in this book has affinity with our concept of architecture, but its operationalisations stem from the engineering tradition and are not suitable for our purposes.

### 3.2.2 Synthesis of the operationalisations found

In this paragraph we make an inventory of the operationalisations that we found in the literature. They must fit the concept of architecture and its quality aspects such as we found them in our earlier analysis of Vitruvius, Alexander and Scott. The final sorting out and filling gaps will take place in the next chapter, where the Architecture Validation Framework will be assembled.

We will present the findings under the headings of the seven key quality aspects that we formulated in Table 6. After having specified the operationalisations, we will try to identify the essence of the key quality aspect in one word, which we can use as a ‘label’.

<table>
<thead>
<tr>
<th>Table 8. Key quality aspects and links to operationalisations in modern literature</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
|     | • The architect must never make the commissioner responsible for choices that in fact reflect the fact that the architect is on some important aspect of the architecture ‘in the blind’.
|     | ‘Confidence’ is the label we choose: the commissioner must be convinced of the utility and feasibility of his mission (be confident), and must have confidence in the architect he chooses to realize it. |
| **2** | Education and practical experience enable the architect to inductively determine the main properties of a work that improves the context on the basis of a deep understanding of this context. |
|     | As we have seen, the professional profile of architects as described in modern IT literature is too modest and too engineering oriented to be a good match with our analysis. Rechtin is concrete when it comes to the education and experience required for an architect in the case of major enterprises. |
Education:
- The architect is of ‘master-level, if not doctorate-level’ after following a specific curriculum.
- Vitruvian has pointed to the importance of breadth in education. We feel the curriculum Rechtin proposes is too one-sided technical. Rechtin rightly claims that the architect has to tackle issues of a serious ethical nature, specifically when dealing with social systems in the public sector, and must be able to cope with the dynamics of national politics.

Experience:
When it comes to experience, the IT literature typically assumes that architects specialize in some facet: business, process, system, infrastructure, or software. We will not follow this approach since it contradicts fundamentally with the holistic responsibility the architect must have in our view. Some form of specialization or experience may be useful though:
- Rechtin distinguishes categories of contexts for architecting. An architect who is to realize an important mission must have experience in the specific context: such as Social Systems, Manufacturing Systems and Systems architected under the auspices of national politics.
- Experience with a specific solution may be a liability rather than an asset for an architect (it is a great asset for an engineer). Even if the advantages of sector knowledge (context) are evident, the risk is that an architect represents outcomes looking for matching problems (typical of builder architected systems), which is the opposite of the kind of inductive reasoning we consider essential for good architecture.
- Relevant experience cannot be operationalised by the number of projects an architect has done. Large projects in the public sector typically take 5 years or more. Rarely will an architect do more than one or two projects of this scale in his career. For now our conclusion is that Rechtin struck a reasonable balance: if a project will draw political attention it may be wise to hire an architect who has demonstrable experience in the public sector, even if not in the role of an architect.
- The track record of the architect as someone who succeeded in architecting systems with acknowledged added value remains an important criterion.

‘Induction’ is a suitable label even if the meaning will not be immediately obvious. Yet it is important that users of the framework understand that education and experience do not serve good architecture as they do good engineering: applying methodology within specific technical fields of expertise.

The professional quality and integrity of the architect must ensure that the architect takes a critical attitude towards the commissioner’s ambition.

All books stress the necessity to be critical of initial ‘requirements’. Even if this is mostly in terms of a warning, we consider this criterion to be relatively easy to test effectively, in two ways:
- Architecture is ‘an early bird’; already fairly early in the execution of the assignment it must be possible to see action taking place in response to the initial

---

147 Ibid, preface p. 4.
commission. This can be in terms of critical investigations going on in the context to verify certain (implicit) assumptions of the commission, or ideas about alternative strategies to the one (implicit) in the commission.

- The interaction between architect and commissioner must be convincing in terms of content, scope and effect; if it is haphazard or superficial it is bound to be unconvincing. Although testing is therefore straightforward, undeniably the judgement required is of high level.

‘Integrity’ is the core concept here, both personally and professionally.

4 The architect’s commission grants him professional independence in the execution of his mission and ample possibility to acquaint himself with the context relevant to the commissioner’s ambition.

Independence is a necessary condition for a number of quality aspects and should be granted explicitly. If not, a necessary condition for other key quality aspects may be missing: inductive reasoning requires that the architect is not bound to any preconceptions that may exist about the solution. The architect must be in a position to consider the interests of the context with an open mind. Business case and transition scenarios must not be under undue influence of the commissioner’s agenda.

Contexts can also be inherently different with regard to the position of the architect, as Rechtin has stressed. This cannot be ignored when judging the independence of the architect. We can summarize the operationalisations found as follows:

- A commission must state a goal, not a solution, leaving room for the ‘art’ aspect of architecting.
- The context of the architect must be aware of the independence of the architect as a natural ingredient of his role.

Possibility of investigating the context

- It is essential that the architect has time to acquaint himself with the context and has the necessary access to any place, person or documentation he may need.
- DYA, however, has rightfully pointed to the fact that there may be little time because of a vital window of opportunity or an imminent threat. We feel that an architect must be able to handle this situation, and he is in fact needed more than ever under those circumstances, although the limitations may have a negative impact. For our operationalisation this means that there are situations where ‘sufficient time’ is not a valid criterion.
- A position of independence is not only something one must aspire to with credibility, but ultimately something that must be granted by the context. It is in this respect that the architecture maturity framework of DYA can play a role. The organization must recognize the position of the architect.

‘Independence’ is the obvious label in this case.

5 A balanced approach is chosen towards the interests of the context and local differentiation therein, applying high ethical standards.

Ethical problems are a severe political liability. It will ultimately not be the architect who decides, but we have identified the responsibility for a design that improves the context (avoids ethical risks) as an important element of the profile of the architect. Of the literature studied, only Rechtin attempts to operationalise this criterion in the chapter
on Social Systems.
Interestingly, Rechtin sees a role for the architect as a sort of mediator (‘socioeconomic approach’), where it is obvious that this has to be a rather informal sort of negotiation. If not, the architect is in the domain of the commissioner or even the political sphere.

- In developed economies dense regulation leads people to think that their conduct is ethical as long as they act within the law. This may not always be the case generally, but it is certainly no guarantee when dealing with the public sphere.
- Potential conflicts can be identified and analysed by using the heuristic ‘the four who’s’: who benefits, who pays, who provides, who loses. Rechtin takes this heuristic as the basis for what he calls the ‘socioeconomic approach’ for handling ethical issues. It aims at reaching consensus among the four who’s by focussing on real value for the parties concerned, not on cost or price.

‘Fairness’ is a label that expresses in every day terms what is meant.

6 An economic and, more generally, quantitative and qualitative analysis for which the architect takes responsibility underpins the architecture in order to be able to establish the value and opportunity cost of the architecture as objectively as possible.

To avoid any misunderstanding, the total cost of ownership of the construction is important, but it’s the qualities and liabilities of the architecture that count primarily here. Although it is possible to reflect critically on presented facts and figures, we ultimately will have to rely on the integrity and modesty of the architect.

- The essential role of the architect here is to identify the information needed in order to present a balanced picture of the quality of the architecture. The commissioner must be able to make decisions knowing what he is doing.
- The context must be informed about the impact the architecture may have on their interests.

‘Business case’ is the obvious label, even if this term is too often understood as meaning guaranteed financial benefit. The essence of a business case is that it should be a basis for a decision under uncertainty, exploring the odds both in financial and non-financial terms. One of Rechtin’s heuristics reads: “When choices must be made with unavoidably inadequate information, choose the best available and then watch to see whether future solutions appear faster that future problems. If so, the choice was at least adequate. If not, go back and choose again”.

7 Transition of the existing situation to the new architecture must be planned in stages that are each recognizable improvements, while at the same time they must create an opportunity for adjustments to dynamics and more effective risk management unless a different approach is presented with a very strong motivation. Scott formulated the operationalisations both succinctly and clearly:

- Take small steps
- Favour reversibility
- Plan on surprises
- Plan on human inventiveness

‘Growth’ is a good label as it expresses the nature of the process. It is important to note though that the message of this quality aspect is not that we are, with the best of
intentions, embarking on missions without destination. Relevant is the example Rechtin uses of the assignment to ‘place a man on the moon’. Rarely has a clearer and more concise requirement been given to systems architects. Yet, defining robust steps with ample room for learning determined the success of the mission.

3.2.3 Conclusion: good architecture operationalised

We can conclude that IT has a long way to go before a mature architecting profession is in place. The fact that the building sector shows a higher maturity in architecting practices on the basis of two millennia of experience is no surprise. The fact that a book on systems architecting proved to be a more useful source for operationalisation than literature devoted to IT architecture is remarkable, especially since the title of ‘architect’ is hardly used in systems architecting and is abundant in IT.

Having come to the end of this chapter without having identified a great number of operationalisations we must ask ourselves if we have enough basis for constructing the Architecture Validation Framework. We think we do for the following reasons.

Firstly, research question six contains the criteria. It reads: “Is the AVF to be constructed on the basis of the aforementioned criteria, on the one hand, sufficiently general (fits different types of projects) and, on the other hand, sufficiently clear and concrete (can be used by any professional meeting certain competence criteria)”. We need fairly general criteria because we are looking for essentials. The number of essentials we found is relatively small which is only helpful because it makes the construction of the framework more realistic.

Secondly, a number of important criteria concerning the architect and his assignment can be applied before the actual work starts. This is a great advantage since early identification of risks means that time and (political) prestige are saved much more effectively than if application of the framework would have to wait until a fully documented architecture would be available for review.

Thirdly, even if we have to do additional operationalising, this concerns a limited number of issues that need further clarification, such as the difference between architects who are employed by the commissioner and architects who are hired for the job.

We must acknowledge though that even if we formally can affirm that ‘any professional meeting certain competence criteria’ can use the framework, the competence required will be of a higher level than we had hoped. Experience and judgement must be of senior level. A typical example is judging beforehand, or at least in an early phase, if the architect is sufficiently critical towards the commission. If the answer is ‘no’, the reviewer runs the risk that both the architect and the commissioner will not be inclined to follow his reasoning and dismiss it as precocious, unless the authority of the reviewer precludes this.

3.2.4 Conclusion with regard to the third and fourth research question

Third research question

Third research question:
“How can we operationalise IT architecture, taking into account different levels of architecture and different approaches in current literature?”
Our conception of IT-architecture proved to be different from what we found in modern IT-literature. Recognition of the ‘art’ aspect of architecture is absent, or marginal. It was therefore only to be expected that we did not find further operationalisations. However, as we demonstrated in the last paragraph, this does not hinder us in constructing the framework.

**Fourth research question**

*Fourth research question:*
“What are possible criteria per key quality aspect?”

Again, IT-literature was not of great use. Rechtin and Scott provided us with a number of important criteria (Table 8) that we can use in constructing the framework.
4. Conceiving the Architecture Validation Framework

The goal of this study is to develop an Architecture Validation Framework (AVF) for the validation of the architecture of larger public sector IT systems. In this chapter we will present the framework, and ‘instructions for use’. We remind the reader, firstly, that the term ‘framework’ is chosen to indicate that within the scope of this research we merely develop a first version, even if it is tested in several concrete cases. Thorough validation of the framework is (far) beyond the scope of this research. Secondly, we focus on the validation of criteria for when an architecture is sufficient, that is, we try to be complete rather than confine the framework to easily measurable criteria.

In the last chapter we have operationalised the concept by formulating criteria in broad terms. Constructing the framework requires that we answer the fifth and sixth research question:

**Fifth research question:**
“How can we operationalise these criteria so that they can be tested?”

**Sixth research question:**
“Is the AVF to be constructed on the basis of the aforementioned criteria, on the one hand, sufficiently general (fits different types of projects) and, on the other hand, sufficiently clear and concrete (can be used by any professional meeting certain competence criteria)?”

A straightforward implementation of the framework would start by listing the essentials of the concept of architecture and the quality aspects as we have operationalised them. Next, we would add the results of research question 5, typically questions concerning the various items on the list. The result would be a questionnaire consisting of two main parts: the first part establishes if what we want to judge can be considered architecture or not. The second part of the questionnaire can be used to evaluate the quality of the architecture.

However, this approach is hard to explain to those responsible for the architecture; for them the issue of whether or not it can be considered ‘true’ architecture is rather academic; they want to hear in what respects the architecture is lacking rather than whether or not it can be considered architecture. Also there is the finding in the last chapter that it is not advisable to wait for validation until the architecture is finished. In the case of the large projects we are looking at, it would be very difficult to change anything at this point.

We feel that we must take a step back and ‘look at the bigger picture’ in order to produce an outline for the framework. We will do this in the next paragraph. After that we will fill in the outline with criteria and operationalisations.

### 4.1 The outline for the framework

We have started our research under the (implicit) assumption that the public service will invest in IT Architecture, following the reports of the Court of Audit. However, we also know
that development is slow as a recent evaluation study shows\textsuperscript{149}. Under those circumstances a norm will be perceived as a (new) instruction, a policy. The perspective changes. A norm formulated in terms of principles, and not as hard and fast rules, will function as a ‘code of good practice’, helping those involved to improve their project. This leads to the following requirements for the framework:

- the framework must be presented as guidance and contain rationale.
- the protocol for using the framework must specify that it must be applied repeatedly and in any case when important decisions\textsuperscript{150} must be made:
  - before starting preparations for decision-making,
  - in the process of decision-making,
  - as soon as a decision has led to results that enable to check either if the concerns of the framework are met or if the risk resulting from not following the framework has materialized.

The picture that unfolds for the framework is strongly reminiscent of the Gateway Reviews\textsuperscript{151}, a standard good practice in the Dutch public sector. We will therefore investigate what we can learn from the Gateway approach.

### 4.1.1 The Gateway review method as an example for the framework

Gateway, as a review method, is close to our goal and has rapidly been instituted as a good practice in the Netherlands. Gateway relies on peer reviews in the public service that are performed in a short period, a week for the actual interviews and drawing up of conclusions. Gateway reviewers are peers of commissioners and therefore of senior level, but they get a ‘crash course’ so as to prepare them. The government disposes of a pool of Gateway reviewers.

There are workbooks to support different types of reviews. Preparations should not take more than three weeks, which means that a report can be ready less than a month after it is ordered. The results are reported confidentially to the Commissioner of a project, in Gateway terms the Senior Responsible Owner (SRO). It should be noted, however, that the confidentiality so far has proven hard to maintain in the Netherlands. Under political pressure SROs (and/or the responsible ministers) have published reviews.

The bottom line of a Gateway Review is a judgement of red, amber or green: green if everything is ok, amber if the project may continue but should solve certain issues, and red if important blocking issues have been found for which the reviewers see no clear solution in the existing context.

**Conclusion Gateway method**

Our conclusion is that the Gateway method can be an example for the framework. It may even be possible to see the framework as an addendum that can be applied if a Gateway review is


\textsuperscript{150} e.g. hiring the architect, important policy decisions regarding the preparation of the project, approval of a project plan, etc.

performed concerning a programme with an important and innovative IT-component. Thus Gateway will not only function as a measure taken in response to the Court of Audit advice to invest in Portfolio- and Project management, but also address architecture, as it is another area of concern of the Court of Audit.

4.1.2 Different contexts for different architects

The next issue we want to consider for the outline of the framework is the matter of the different contexts that Rechtin distinguished. Three of these contexts proved relevant for our work, especially since the position of the architect may be rather different in each of them: systems architected by employees of the builder, systems for the production of end products, and social systems where the user is one of a large number of private citizens. We will have to decide whether this is an issue that must have consequences for the framework or not. Obviously, our choice here is relevant for the balance we have to strike with a view to the concern of research question six: the framework must be sufficiently general but, on the other hand, sufficiently concrete. The limited number of key quality aspects makes for general applicability, but if operationalisation would have to be differentiated strongly for different contexts this would be a threat to general applicability.

Firstly, we have the case of the ‘builder architected’ system where the producer employs architects to ‘invent’ new products that he produces and sells on the market. As a typical risk, Rechtin describes the fact that architects in such a context may cling to their legacy, hampering innovation. Even if the public service will not produce for the market, two other defining elements of this model are found: the architect is employed by the builder and there is a legacy. An example is the IT department of the Dutch tax authorities. It is in this context that the ‘architecture maturity’ of an organization becomes very relevant: in the context of the larger projects only the very mature organizations can rely on their ‘standard procedure’ to obtain good architecture: create room for induction and provide sufficient independence for the architect. We will therefore formulate some operationalisations directed at this issue.

Secondly, we have the manufacturing systems, where the system to be architected routinely produces the end product. This is the dominant model in public sector administrative activities, which are administrations for documents such as passports and driving licences, but also systems administrating permits, benefits or subsidies. The issues the architect is facing are legacy, the necessity to keep production going while making the change, handling the data of the old system, and a multitude of process and people considerations. This type of system does not create such specific conditions for the architect so as to warrant separate treatment in the framework. Furthermore, in the public sector this category probably overlaps largely with the builder architected systems.

Thirdly, Rechtin distinguishes ‘social systems’, where an individual citizen depends on the services. This category overlaps in part with the manufacturing systems, where the citizen has little or no influence on the system, such as in the case of administration of social security benefits. The citizen is hardly a ‘user’. On the other end of the spectrum are systems such as, for example, telephony which, although heavily regulated, are used by the client and depend on the preferences of clients who buy services from commercial providers. Rechtin notes ethical issues that may arise when the commissioner is not the user. We have covered ethical aspects in our criteria and we generally have stressed the importance of these concerns for the architect. Again we see no reason to create a specific category in the framework.
Conclusion with regard to differentiation

Only very limited differentiation is needed. The framework must contain specific criteria for architects employed by the commissioner. In that case, the ‘architectural maturity’ of the organization is an issue.

4.1.3 The outline

In summary, we can specify the following outline of the framework:

- The Gateway review method sets an example for the way the framework will be used.
- The framework can be used both independently but also as part of a Gateway review.
- A set of instructions will be needed explaining the intended use and the rationale of the framework and criteria and the required quality of the reviewers: a succinct manual.
- The question ‘What is architecture?’ will be treated succinctly in the manual but not in the questionnaire.
- We will distinguish between projects where the architect is hired for the job and where the architect is employed by the commissioner. In the latter case an architecture maturity assessment is advised.
- Scoring is relatively simple because we have established “key quality aspects”. There is little room for compromising or compensating between criteria: all criteria must be met. The only ‘defendable risk’ is the one taken if circumstances (can be both threats or opportunities) require immediate action. Under those circumstances it is primarily the commissioner who makes the decisions and it is for the architect to accept the challenge or not.

4.2 Filling in the framework

Filling in the framework is now a matter of reformulating the operationalisations from Table 8 (Key quality aspects and links to operationalisations in modern literature) as criteria. We will differentiate between criteria that are mainly applicable in the start up phase and those that apply throughout the project. Finally, we add criteria specifically addressing the position of the architect who is employed by the commissioner, rather than hired for the job.

The key quality aspects represent important, recognisable aspects of the architecting process, but may be overlapping, that is, not entirely independent as they ideally should be. For example the desired independence of the architect is also a component of what we expect from the commissioner (criterion 1.10 vs. 4.1).

4.3 The manual

The manual must obviously start with a short introduction explaining the rationale for the addendum explaining the need to focus on architecture as a key success factor in IT-projects, and differentiate between the art and the science component. Next, we have to give ‘instructions for use’ explaining how the scorecard is in practice to be used. An annexe will explain how to link Gateway Review 0 to the scorecard.

4.4 The result: the scorecard IT Architecting

Appendix 1 contains the result of our efforts: the Scorecard IT Architecting, being a fairly straightforward implementation of our research so far and the design and final choices as we
have presented in this chapter. Although research questions five and six have in fact been
dealt with in the process described above, we will summarize the answers here.

**Fifth research question:**
“How can we operationalise these criteria so that they can be tested?”

The answer here is that for the first version of the framework we intend to construct we can
use the criteria without further operationalisation (apart from reformulating the criteria from
Table 8 suitably). The professional judgement required to use the scorecard is of a relatively
high level. We assume that no further instructions are needed, given that the judgement
required is only in terms of green, amber or red. The experts will be the first to ‘test’ this
assumption.

**Sixth research question:**
“Is the AVF to be constructed on the basis of the aforementioned criteria, on the one hand,
sufficiently general (fits different types of projects) and, on the other hand, sufficiently clear
and concrete (can be used by any professional meeting certain competence criteria)?”

As we already concluded at the end of the last chapter, the key quality criteria, as presented in
Table 6, match the concerns of research question 6: sufficiently general and concrete. In this
chapter we have only reformulated the findings of Table 8 and determined to what extent we
would need to differentiate for different contexts on the basis of Rechtin's theory. Our
conclusion was that only very limited differentiation is needed (and even then only for
specific criteria) between internal architects and architects hired for the job. Generality is
therefore not an issue. We feel that it is sufficiently clear, but the experts will be the first to
‘test’ this assumption.

4.5 **The framework in the perspective of the literature: a reflection**

Having come to this point it is useful to reflect on the fact that it proved difficult to find
support in modern IT literature for the theory we have developed on the basis of Vitruvius,
Alexander and Scott. The fact that Rechtin perfectly matches the views we developed shows,
firstly, that others have seen that there is more to constructing complicated systems than
rigorous engineering discipline. Secondly, it shows that this approach to architecting is not
just needed for systems where the human factor is important: Rechtin developed his theory in
the context of the American space programme. You apparently need a seemingly soft
approach to successfully develop very hard technology.

Another fact is that there is at least one well-known Dutch IT architecture expert who has
developed views that resemble the ones in this thesis: Dr. Daan Rijsenbrij. However,
Rijsenbrij has never written a book enabling us to really stack up our theory against his.

Also remarkable is the fact that, insofar as there is recognition in IT architecting literature that
there is more than engineering, you will find it sooner in books on software architecture than
in books on Enterprise Architecture 152.

---

152 A typical example is the hardly scientific, but very readable Sewell, M., Sewell, L., “The Software Architect’s
What is typical about our approach to architecting is that it combines the focus on an individual system (which is natural to software engineering) and combines it with the strategic business orientation that we associate with enterprise architecture. Books on enterprise architecture concentrate on how a particular system or demand for functionality fit into the ‘bigger picture’. Our theory prescribes a radical orientation on the actual context of the system and all particularities that are of interest in that context, where IT is only one aspect but where the official business goal is also only one aspect (be it an important one). Maybe this explains why we find a similar approach with Rechtin; in advanced systems architecting of the type needed for developing spacecraft there is also a natural focus on the system to be constructed and any factor relevant to its success.

Then, could it be the case that our approach ‘neglects’ the important goals of efficiency improvement through reuse and standardization (in one form or another) that are core concerns of enterprise architecture? This need not be the case. The concerns of traditional enterprise architecture should be dealt with on an engineering level, which is part of professional good practice, but they are not what make a system successful. In our theory it is more important that the architect concentrates on a really attractive solution, attractive for both the commissioner and the contexts where the system will be used.
5. Testing the Architecture Validation Framework

_Seventh research question:_
What criteria can be formulated to establish if the AVF is successful in a test or not? We must be able to establish the presence or absence of architecture and a score on the different quality aspects respectively. Since there is no straightforward way to test such a framework, expert opinion will be the main approach.

Fully testing the AVF would have required applying it in several real life cases. Since the projects the AVF is concerned about typically last for many years, this was not feasible. We therefore had to rely on expert opinion. Four experts were asked to participate, two with an IT architecting background and two with an IT audit background. Each had (much) more than 20 years of professional experience.

5.1 The judgement asked of the experts

The experts studied the scorecard document (appendix 1). We then convened for a session where all key quality aspects and operationalisations were discussed one by one.

- In the case of a key quality aspect the experts were asked (after the discussion) to individually give a score between 1 and 10 expressing the relevance of the key quality aspect for the success of the architecting process. The scores were then written on a whiteboard and differences were discussed.
- For the criteria operationalising the key quality aspects the question was whether they contributed significantly to a correct judgement of the associated key quality aspect.
- Finally the judgement of the experts on the scorecard document as a product for practical use was asked.

5.2 Key quality aspects and operationalisations judged

The results were as follows:

<table>
<thead>
<tr>
<th>Key Quality Aspect + operationalisations</th>
<th>Expert 1, IT Architect</th>
<th>Expert 2, IT Architect</th>
<th>Expert 3, IT Auditor</th>
<th>Expert 4, IT Auditor</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8,75</td>
</tr>
<tr>
<td>1.1</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>1.2</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1.4</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1.5</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>7,5</td>
</tr>
<tr>
<td>1.6</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>1.7</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>1.8</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>8</td>
<td>6,5</td>
</tr>
<tr>
<td>1.9</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8,5</td>
</tr>
<tr>
<td>1.10</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>8,75</td>
</tr>
<tr>
<td>2.1</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>8,5</td>
</tr>
<tr>
<td>2.2</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8,5</td>
</tr>
<tr>
<td>2.4</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>2.6</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>7,5</td>
</tr>
<tr>
<td>2.7</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>7,5</td>
</tr>
<tr>
<td>2.8</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>6,5</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>8,25</td>
</tr>
<tr>
<td>3.1</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>3.2</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>3.3</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>3.4</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>7,5</td>
</tr>
<tr>
<td>3.5</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3.6</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>7,5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>7,5</td>
<td>10</td>
<td>9</td>
<td>8,625</td>
</tr>
<tr>
<td>4.1</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>4.2</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4.3</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4.4</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>6,5</td>
</tr>
<tr>
<td>4.5</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>5.1</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>8,5</td>
</tr>
<tr>
<td>5.2</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8,75</td>
</tr>
<tr>
<td>6.1</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>6.2</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8,5</td>
</tr>
<tr>
<td>6.3</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8,5</td>
</tr>
<tr>
<td>6.4</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8,5</td>
</tr>
<tr>
<td>6.5</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>7,75</td>
</tr>
<tr>
<td>7.1</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>7.2</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8,5</td>
</tr>
<tr>
<td>7.3</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>7.4</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Mean</td>
<td>8,714</td>
<td>8,15</td>
<td>7,5</td>
<td>7,625</td>
<td>8,571</td>
</tr>
</tbody>
</table>
The table shows high and unanimous appreciations of the experts for both key quality aspects and operationalisations. There are a few outliers where an expert had objections against specific operationalisations but too few to be of material importance.\(^{153}\)

As the next table shows IT auditors showed on average slightly more appreciation for the Key Quality Aspects than did IT architects. The difference however, is entirely explained by the judgement of expert number 2. Conclusion is that there was no marked difference of opinion between experts with a background in IT Architecture and experts with a background in IT Audit.

<table>
<thead>
<tr>
<th>Table 10. Expert judgement by discipline</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Architects</td>
</tr>
<tr>
<td>Mean all Key Quality Aspects</td>
</tr>
<tr>
<td>Mean all operationalisations</td>
</tr>
</tbody>
</table>

5.3 The scorecard judged

The conclusion is that the experts on the whole strongly supported the relevance of the Key Quality Aspects as determining factors for the quality of an architecting process. Support for the operationalisations was only slightly weaker. They tended to agree slightly more on their judgement of the operationalisations than they did for the Key Quality Aspects. Although this test obviously does not permit any conclusions to be made with regard to the completeness of the Key Quality Aspects, the experts were confident that the terrain was mostly covered and especially that important aspects that are often overlooked, such as the vital position of the commissioner, were stated in a convincing manner.

The experts also evaluated a number of other aspects:

- The scorecard (as presented in appendix 1) was considered an adequate product for the use for which it was intended: support a professional judgement ‘Gateway style’. The IT audit experts stressed that the scorecard’s value is to draw explicit attention to a number of important aspects, but cannot be the basis for an audit since it lacks the methodological basis and operationalisation.
- Even if the Gateway reference was considered important because it legitimizes that the Gateway approach is valid and useful, even if it does not qualify as an audit, the experts appreciated the scorecard more as an independent tool than as part of a Gateway Review. The table with references from the Gateway 0 workbook to the scorecard was considered counterproductive in this respect and should be removed.
- The experts considered a good architecting process for the major enterprises of such importance that the scorecard should instead be used as a checklist. Especially the commissioner should read the scorecard and understand its implications.

As suggestions for further research, they mentioned—as apart from further validation—the use that could be made of the scorecard in other contexts, such as smaller projects in the public sector or projects in the private sector.

\(^{153}\) For example expert 4 assigned grade ‘5’ to criterion 4.1 because he considered the overlap with the general concerns of key quality aspect 1 unnecessary. Expert 1 assigned grad ‘5’ to criterion 4.4 since he considered the ‘exception’ that is made in the case of certain circumstances inconsistent with the AVF itself.
6. Conclusion

The research problem underlying this study is that, as the extensive research by the Court of Audit shows, large IT projects in the public sector often fail, even if a considerable amount of time and money has been spent on preparation. The court of audit advises further investment in IT architecture in order to improve the situation but does not indicate how to establish when a project has invested enough in IT architecture. We do not dispose of standards or methods to establish ex ante if a given project architecture is of sufficient quality. Checking compliance with specific and easily measurable standards and criteria is not enough.

The research questions guided an attempt to construct a first version of an Architecture Validation Framework (working title ‘Scorecard IT Architecting’, see annex 1), an instrument to check the quality of architecture. Four experts judged the result, as we saw in the last chapter. Before we come to the final conclusions we summarize where our findings have—in retrospect—shown our research questions to be based on false assumptions. To put it differently, certain insights have provided us with a different view of the problem.

Firstly, the phrase in our research problem “to establish ex ante if a given project architecture is of sufficient quality” assumes that there is an architecture (a set of documents) and a project waiting to get a permission to start if the architecture passes the test. The insight is that such a test comes too late. The investment in time and money makes it very difficult to do more than make marginal corrections, once the architecture is there.

Secondly, the clearest, most powerful and most general criteria have to do with the architecting process, more specifically with the early phases of this process, not with the resulting documents. What we found were criteria for good architecting, not criteria for good architecture.

The good news is that both insights combine very well as a basis for the AVF. Taking the architecting process as the focus of the AVF, we can start literally on day one and suggest important alterations when these can still realistically and without great loss of time, money and political prestige be made.

The question then obviously is: Does a good process guarantee a good architecture? Is it a sufficient condition for good architecture? What we can say is that the literature that proved most useful for our quest assumes this. Further research would be needed to make the claim stronger. Remember though, that our research problem was not about reaching perfection, but about preventing foolish and costly disasters. A good architecting process probably comes very close to a sufficient condition for reaching that goal.

6.1 Can we establish the quality of architecting?

On the basis of the literature studied and the opinion of the experts who tested the AVF, we can answer this question positively. The AVF has proved to be an instrument that is sufficiently general and sufficiently clear in its operationalisations to be handled by expert reviewers of a type and level that are sufficiently available in the public service and consultancies working for the public service.
6.2 Can we improve the quality of architecting?

Again the answer is positive since not only does the AVF enable us to establish if we are on the right track with regard to the architecting process already in the earliest stages, but the AVF also provides guidance. Conditions that are not met are not only litmus tests that may show that something is wrong, but they also indicate what needs to be repaired.

Especially interesting in this respect is the recommendation of the experts about what, in their opinion, could be the most important function of the AVF: providing a document that commissioners should read in order to make them aware of the importance of the architecting process and their specific role in it. Once again in this research we find that Vitruvius has been our prime example: ‘De Architectura’ addresses the emperor in the first chapter as its primary readership, but as Vitruvius notes elsewhere, he considers his book ‘a not unwelcome gift to the whole world’. Apparently the AVF could serve a similar purpose!

6.3 Further research

A number of themes require further clarification:

- We have come across three types of IT architecture with similar ambitions but very different modes of operation: software architecture, enterprise architecture (aimed at ‘city planning’ for the IT landscape) and enterprise architecture in the literal sense where IT is considered a tool helping management to establish an optimal business model. Further research into a more rational division of tasks between these disciplines would help to clarify and improve the role of IT architecture.

- The AVF was written for the largest projects in the public sphere. This helped to get the requirements for good architecting clear. Yet, smaller projects are frequently problematic too. Which elements of the AVF remain valid in the case of smaller projects, which can be removed and what should change otherwise?

- Validation of the AVF is needed, firstly, in order to prove its usefulness above the level of common sense and good practice. Secondly, further improvements may result from such research. Special attention should be given to the question of whether or not the list of Key Quality Aspects is complete.

- Although the research presented here was highly international in its use of literature, we have taken the situation in the Netherlands as the point of departure. A comparative study could reveal to what extent the analysis presented here would also be valid internationally, or could point to instruments similar to the AVF and provide further possibilities for improvement.
Bibliography


Appendix 1. The scorecard IT architecting
Scorecard IT Architecting

Assuring successful IT architecting in major public sector projects
Contents

1 Introduction 1
2 The art and science of architecting 3
3 Scorecard IT architecture 7
3.1 Instructions for use 7
3.2 The scorecard 9
Annex 1. Gateway 15
Annex 2. Linking Gateway Review 0 to the scorecard 17
Introduction

Large IT projects are inherently risky. It is generally accepted that correct use of architecture is indispensable for IT projects, essential both for creating an attractive result, and as the primary means of risk management. Still, there is much confusion and disappointment when it comes to IT architecture. This document helps in two ways. Firstly, it helps by clarifying the concept of architecture in chapter 2, distinguishing between the art and science of architecting. Secondly, chapter 3 contains a scorecard. Successful IT architecture requires very specific conditions that must be rigorously maintained if it is to yield its beneficial and risk-reducing influence. The scorecard helps to establish if these conditions are fulfilled. The theory and the scorecard based on it were developed in the context of a Master’s thesis in Computer Science.

Assurance, as a retrospective judgement on the basis of audits, taking months rather than weeks, may come too late. This scorecard has looked to the example of Gateway™ Review approach. Gateway provides quick, effective and economic assurance for large scale public enterprises with medium or high inherent risk using standard questionnaires (workbooks). The scorecard presented here supports reviews in the ‘Gateway style’ focussing on the key quality aspects of IT architecture. It can be used independently or in addition to Gateway review workbooks. Specifically Gateway Review 0 with a strategic focus matches the scorecard presented here. More details about this are provided in the ‘instructions for use’ paragraph of chapter 3.

---

154 See Annexe 1 for more information on Gateway reviews.
The art and science of architecting

The value an architect can have for any endeavour is critically dependent on his being in control of the conceptualization from the first day, with a clear mandate. The building industry has, literally, thousands of years of experience with the role of the architect. A commissioner with a vision, who feels that by starting a unique construction he can realize this vision, needs a professional—an architect—with a very specific role and position. In the case of buildings, the commissioners and other stakeholders are inclined to appreciate and respect the role of the architect.

Building and IT are very different things though in relation to the organisation and processes they support. IT-systems can be more intertwined with, and vital to, business processes and strategic possibilities than buildings are generally speaking. The technology of IT is probably more dynamic than most building technology. It is therefore maybe only natural that we find different types of architects in IT. However, this means that we must take great care to avoid a situation where we do involve architects in a project, but not ‘the’ architect who is able to take overall responsibility. We look at two types of architects we know in IT as examples of professionals who may be very useful but who are at the same not capable of taking full responsibility. The practice of IT-Architecting unfortunately is not on a level of maturity where an IT-Architect will always refuse a responsibility that is beyond his or her capabilities. The first are the architects with a role of an engineer trying to fit the best possible technical solution to an ‘architecture’ that is mostly implicit and is the outcome of deliberations between commissioner and stakeholders. Thus, professional architecting is absent in the layer between the commissioner’s vision and the technology. Disaster is frequent.

The second architect—often dubbed enterprise architect—is more aware of the greater context but tries to standardize the ambitions of the commissioner to fit a preconceived framework of solutions already available. In large organizations standardization may be an excellent means of controlling complexity, but there are two risks. The first is that the architect is literally far removed from the projects doing the actual work. The second is that, if the project is big or highly innovative and therefore different from the organisation’s standard practice, standardization may not be what is needed and the architect may simply not be up to the job.

If visions require a big and unique project, a more holistic type of architect is needed, who is dedicated to this vision. His scope is not the enterprise architecture or the technology, but the ability to conceptualize the vision into something that fits the context and can be defined and managed as a project. That is what is needed if we want to improve the success rate and manageability of large scale IT-projects.

Such an architect has a supporting role; he works for the commissioner and must come up with an appealing proposition. The difference between this architect and the ones we mentioned is fourfold:

- Firstly, the architect takes a broader responsibility. He designs a coherent architecture having reflected on the commissioner’s vision, what stakeholders have to say, and his own research on the terrain in question.
- Secondly, when he presents an architecture, an abstracted solution, it is obvious that other architectures would have been possible. The architect cannot prove that his choice is the best; the design process is in part inductive.
- Thirdly, during the whole process of design and construction, the architect helps the commissioner to stay in control. He presents the commissioner with decisions of which the latter can oversee the stakes and consequences. The architect avoids
confronting the commissioner with expert opinions as a basis for decisions. The architect also avoids any value judgements when it comes to worth for the client. The client should judge on desirability.

- Lastly, the architect takes responsibility for the feasibility in both the technical and economical sense, oversees construction, and advises the commissioner when the builder has fulfilled his obligations.

In other words, the senior engineer and enterprise architect practice the science of architecting. They work according to proven norms in a strictly rational, method-based, deductive fashion, to design a solution. The architect we are talking about practices the art of architecting. He determines the outline of the solution and is responsible for its feasibility. He deals with stakeholders (the commissioner in the first place, and all those who will experience the workings and effects of the system to be constructed), and works on the basis of experience and his perception of what is essential for an attractive solution. Last but not least, he knows what questions to ask specialists (including the two types of architects we mentioned), and interpret their answers to underpin his architecture.

The good news is that we need relatively few of these highly talented individuals, if compared to the large numbers of other professionals that are involved, let alone the consultants that are often hired to solve the problems of large IT projects once they arise. The deductive reasoning consultants apply to get from objective to ‘proven’ solution is fundamentally unsuitable; the combinatorial explosion of the countless options, possibilities and liabilities that the application cannot be managed in that way. Neither may the well-intended help of commissioners, political leadership and elected representatives be very effective. In a democracy political leadership and democratically elected representatives obviously have every right and responsibility to exercise their influence when it comes to large and costly projects. Yet effective architecture is the only solution and it depends on the wisdom of the commissioner to hire an architect who can understand and handle political processes.

The next table presents the key quality aspects that have been identified for successfully architecting large IT projects in the public sphere. In the next chapter we will present a scorecard based on these key quality aspects.
## Key quality aspects for architecning

<table>
<thead>
<tr>
<th></th>
<th>Aspect</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Confidence</td>
<td>The commissioner has a clear vision but is at the same time aware of the importance of a good architecning process and the importance of selecting a good architect that he trusts.</td>
</tr>
<tr>
<td>2</td>
<td>Induction</td>
<td>Education and practical experience enable the architect to inductively determine the main properties of a work that improves the context on the basis of a deep understanding of this context.</td>
</tr>
<tr>
<td>3</td>
<td>Integrity</td>
<td>The professional quality and integrity of the architect must ensure that the architect takes a critical attitude towards the commissioner's ambition.</td>
</tr>
<tr>
<td>4</td>
<td>Independence</td>
<td>The architect’s commission grants him professional independence in the execution of his mission and ample possibility to acquaint himself with the context relevant to the commissioner’s ambition.</td>
</tr>
<tr>
<td>5</td>
<td>Fairness</td>
<td>A balanced approach is chosen by the architect towards the interests of the context and local differentiation therein, applying high ethical standards.</td>
</tr>
<tr>
<td>6</td>
<td>Business case</td>
<td>An economic and more generally quantitative and qualitative analysis, for which the architect takes responsibility, underpins the architecture in order to be able to establish the value and opportunity cost of the architecture as objectively as possible.</td>
</tr>
<tr>
<td>7</td>
<td>Growth</td>
<td>Transition of the existing situation to the new architecture must be planned in stages that are each recognizable improvements, while at the same time creating an opportunity for adjustments to the dynamics of the context and more effective risk management, unless a more radical approach is presented with a very strong motivation.</td>
</tr>
</tbody>
</table>
Scorecard IT architecture

The scorecard consists of the seven ‘Key quality aspects’ and a number of further operationalisations per quality aspect. In the first column of the scorecard you will find either an ‘I’ or an ‘A’ depending on if the operationalisation is relevant only in the initial stage, or in all stages of the programme.

Instructions for use

The most important instruction is: plan a first review using this scorecard as early as possible, that is, when it seems likely that a large project depending on IT will be developed, long before a project brief, let alone a project initiation document are written and project managers are hired. One of the important functions of an architect is to help conceptualize a vision into an architecture that, in turn, will lead to a project. Never inverse this order of events.

The second most important instruction is: don’t use the scorecard unless you and everybody else in the team doing the review have read the previous chapter explaining the ideas behind it.

There are two ways in which one can use this scorecard: independently, or as part of a Gateway review.

In the case of an independent review, it is advisable to follow the general principles of Gateway reviews. The chair of the review team must preferably be a peer, or at least close to a peer, in relation to the (probable) commissioner of the project. Be quick, at most a few weeks of preparation (in the early stages less than a week should do) and no more than a couple of days for the review and confidential report.

Two operationalisations refer specifically to the situation where the architect is employed by the commissioner. Although the head start in context knowledge this provides, if compared to an architect hired for the job, is a great advantage, it may be more difficult for such an architect to be truly independent as an architect should be.

Finally, we want to stress that it is vital to handle architecting carefully for a very simple reason: you cannot rely on ‘guaranteed expertise’ as we like to do. The enterprises we are talking about typically last 3 to 10 years. Nobody has done three such projects in his career as lead architect. Nobody can claim to have done ‘this sort of thing’ before because every case is unique. If the conditions for good architecting are carefully managed, however, there is every chance of success.

The various operationalisations of the scorecard help to see different aspects, but they may overlap within one ‘key quality aspect’ or even between ‘key quality aspects’, in respect to a specific case. This does not hinder its application.

The judgement must be severe. The scorecard is very concise, but this means that all flaws must be repaired, if this can be done convincingly, since architecting is a fairly delicate art. If not: stop and start anew. If it turns out you cannot create the necessary conditions, then limit your ambitions. The public is not served by heroic disasters.

---

155 See annex 2 for more information on how to link Gateway Review 0 to the scorecard.
The scorecard

Scorecard IT architecture in major public sector projects, based on seven key quality criteria

Legenda:
Phase: A = all phases, I = specifically initiation.
Score: Green (condition fulfilled), Amber (shortcomings found that must and can be remedied), Red (shortcomings found that pose a serious threat to the architecting process).

<table>
<thead>
<tr>
<th>Phase</th>
<th>Criterion #</th>
<th>Key quality aspects for an architecting process resulting in good architecture</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1.1</td>
<td>The commissioner’s vision implies progress and has a concrete goal.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1.2</td>
<td>Ministers, senior management and staff who support the initiative understand the crucial function of architecture in the project and the role of the architect.</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>1.3</td>
<td>The commissioner understands his particular responsibility for selecting the right architect.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1.4</td>
<td>The commissioner consults the architect concerning the selection of the project manager and others with a vital role in the project and does not appoint anyone the architect considers unsuitable.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1.5</td>
<td>The position of the architect in relation to the project manager is clearly and explicitly defined.</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1.6</td>
<td>The architect stays at arm's length of project management responsibilities. Project management is about turning the drawings into construction. Architecture is about conceptual integrity and stakeholders.</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>1.7</td>
<td>The architect must be selected for his qualities, not because he offered himself and/or has powerful relations.</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>1.8</td>
<td>The commissioner takes care not to select an architect for his charisma. If charisma is or has been a criterion for selecting the architect that is an indication that the commissioner (or political official) confuses essential roles: the architect helps the commissioner to choose and win his battles, but is not his general in the field. It is</td>
<td></td>
</tr>
</tbody>
</table>
the architecture that must be appealing, not the architect.

| I | 1.9 | The architect was selected early enough to take full responsibility for the architecture. |
| A | 1.10 | The architect is explicitly given full responsibility for the architecting. This is consistently reflected in the roles in the triangle of commissioner, architect and project manager. |

2. Induction

**Education and practical experience enable the architect to inductively determine the main properties of a work that improves the context on the basis of a deep understanding of this context.**

| I | 2.1 | The architect is trained as an engineer on a master, if not doctorate level; high level architecting is an intellectual activity. |
| I | 2.2 | The architect has relevant experience and technical background (not necessarily in the role of architect) from the point of view of the content matter of the project. |
| I | 2.3 | The architect has sufficient academic breadth in order to be able to develop a deep understanding of the interests at stake for various stakeholders; an architect must be more than an engineer. |
| A | 2.4 | The architect feels at home on the terrain and is both able and inclined to communicate with those who will experience the consequences of the system and those who will do the actual building. |
| I | 2.5 | The architect has relevant experience (not necessarily in the role of architect) making it probable that he can manage the scope and scale of the project. |
| I | 2.6 | It is important for the architect to have real experience of the implications of scale and of interference with the dynamics of the political sphere. |
| A | 2.7 | The architect shows strength in maintaining the professional role but has the modesty to assume a serving role generally but especially when it comes to political realities. |
| A | 2.8 | If the architect is internal, his activities must show no undue bias with regard to the ‘installed base’ the architect in question may have had a role in. |

3. Integrity

The professional quality and integrity of the architect must ensure that the architect takes a critical attitude towards the commissioner's vision.

| I | 3.1 | The architect must demonstrate a critical attitude towards all explicit and implicit assumptions underlying his |
### Assignment from the Very Start by Verification Through Research

<table>
<thead>
<tr>
<th>A 3.2</th>
<th>The architect's activities must be guided by the goal of the commissioner but, if underlying assumptions prove false, he will investigate other ways to reach that goal but only as long as he believes substantial progress can be made.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 3.3</td>
<td>If the underlying assumptions of a goal that is of great importance to the commissioner prove unverifiable (but not false) and alternative paths towards that goal show equally high uncertainty, the architect will investigate if he can define a platform that is both meaningful in itself and helps to determine if the goal can be reached.</td>
</tr>
<tr>
<td>A 3.4</td>
<td>The architect is careful to document all assumptions underlying his activities including evidence for their validity or the absence thereof.</td>
</tr>
<tr>
<td>A 3.5</td>
<td>The architect must make the validity of the rationale underlying the architecture an important topic in the communications with the commissioner.</td>
</tr>
<tr>
<td>A 3.6</td>
<td>The architect must not only be constantly critical of the validity of the rationale of the architecture, but also of the feasibility of his assignment, taking into account the interests of the commissioner and the wider context where the architecture has effect.</td>
</tr>
</tbody>
</table>

### Independence

<table>
<thead>
<tr>
<th>I 4.1</th>
<th>The vision of the commissioner must state a clear ambition rather than a problem and does not presume any technological choices.</th>
</tr>
</thead>
<tbody>
<tr>
<td>I 4.2</td>
<td>The architect must be able to rely on a clear mandate from the commissioner assuring cooperation of staff for providing information without the necessity to fully motivate requests for information, let alone bargain for information.</td>
</tr>
<tr>
<td>I 4.3</td>
<td>The architect’s mandate must enable him to discuss the commission with people in the context who may be critical of the commissioner’s intentions.</td>
</tr>
<tr>
<td>A 4.4</td>
<td>If the situation is that an immediate threat or unique window of opportunity requires immediate action, the architect will not refuse to act, but help by defining the shortest path that secures the immediate interests but leaves open as much space as possible for future development.</td>
</tr>
</tbody>
</table>
| I | 4.5 | If an internal architect is considered, either the maturity of the organization with regard to architecture must be on a high level, as demonstrated by measurement\textsuperscript{156} or track record, or the commissioner must very visibly demonstrate a high and frequent involvement in the architecting process effectively providing the architect with the required authority.

<table>
<thead>
<tr>
<th>5.</th>
<th>Fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A balanced approach is chosen towards the interests of the context and local differentiation therein, applying high ethical standards.</td>
</tr>
<tr>
<td>A</td>
<td>Being within the law is certainly no guarantee that an architecture has sufficient ethical legitimacy. Especially in the political sphere, ethical problems are a severe liability. The architecture must defend an explicit position with regard to who benefits, who pays, who provides, and who loses.</td>
</tr>
<tr>
<td>A</td>
<td>There is no fundamental objection to the architect (given his independent position) exploring possible compromises between stakeholders since they may help the commissioner, but the architect should be very careful to avoid a political role.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6.</th>
<th>Business case</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An economic and more generally quantitative and qualitative analysis, for which the architect takes responsibility, underpins the architecture in order to be able to establish the value and opportunity cost of the architecture as objectively as possible.</td>
</tr>
<tr>
<td>A</td>
<td>The architect must identify and supply the information needed in order to present a balanced picture of the quality of the architecture. The commissioner must be able to make decisions knowing what he is doing. This does not exclude high levels of uncertainty with regard to certain parameters. Of special importance is the presentation of the strategic advantages that do not have a short term financial effect.</td>
</tr>
<tr>
<td>A</td>
<td>The information provided by the architect must take into account the interests of all stakeholders.</td>
</tr>
<tr>
<td>A</td>
<td>Issues arising in the course of the execution of projects are the core business of the architect (unless clearly in the domain of the project manager). Both the commissioner and the project manager may expect a clear position from the architect on how to manage them.</td>
</tr>
<tr>
<td>A</td>
<td>In the case of a major issue (or even a small ‘black swan’) affecting the business case, the architect must be</td>
</tr>
</tbody>
</table>

given time to rethink the architecture and possibly come up with new and attractive ways to continue the project.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6.5</td>
</tr>
</tbody>
</table>

7. **Growth**

Transition of the existing situation to the new architecture is planned in stages that are each recognizable improvements, while at the same time creating an opportunity for adjustments where needed and more effective risk management.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7.1</td>
</tr>
<tr>
<td>A</td>
<td>7.2</td>
</tr>
<tr>
<td>A</td>
<td>7.3</td>
</tr>
<tr>
<td>A</td>
<td>7.4</td>
</tr>
</tbody>
</table>
Annex 1. Gateway

The UK Office of Government Commerce\textsuperscript{157} (OGC) developed the ‘Gateway’ review which examines programmes and projects at key decision points in their lifecycle. It looks ahead to provide assurance that they can progress successfully to the next stage; the process is a best practice in central civil government, the health sector, local government and defence. Gateway reviews are ‘peer reviews’ for medium to high risk programmes and projects and they are carried out in a week. They end with a confidential report handed to the senior responsible owner, underpinning a status of either ‘green’ (carry on), ‘amber’ (carry on, but some action is necessary to cover identified risks) or ‘red’ (stop, unless you are able to solve one or more blocking issues).

The six Gateway workbooks provide concrete guidance for the reviews in different stages of the project or programme lifecycle. However, although here and there specific references to IT projects are made, there are no specific reviews for IT-projects, let alone reviews focussing on IT-architecture. Review 0 is specific in that it focuses on the strategic aspects of programmes, thereby matching the scorecard presented in this document.

Gateway is used extensively in the Dutch public service. The Ministry of the Interior and Kingdom Relations has instituted a support unit for Gateway reviews called ‘the Gateway Bureau’, licensed by OGC in January 2010\textsuperscript{158}.

\textsuperscript{157} The OGC is part of the Efficiency and Reform Group of the Cabinet Office. ‘Gateway’ is a trademark of the OGC. http://www.cabinetoffice.gov.uk/content/about-cabinet-office accessed 18 February 2012.

\textsuperscript{158} http://www.rijksoverheid.nl/nieuws/2010/02/02/britse-licentie-voor-bureau-gateway.html accessed 18 February 2012.
Annex 2. Linking Gateway Review 0 to the scorecard

The scorecard can be used as part of any Gateway review, but matches best with a Gateway Review 0, ‘Strategic Assessment’. This review is for programmes and can be applied when initiating a programme, mid-stage and at the end. Basically, it should be applied whenever assurance is needed that a programme still fits the wider context and can be further developed in a meaningful way. Review 0 always asks the same questions, but obviously the focus is different depending on the phase the programme is in. Although it is possible to literally use the scorecard questions as operationalisations of the Gateway questions (using the table below), we advise one to consider the use of the scorecard as something to tailor for each occasion.

The other Gateway reviews target specific project phases and are of a more operational character. This means that there is no direct linkage between those reviews and the scorecard as in the case of Review 0. It may be useful though to use the scorecard prior to a Gateway review 2, Delivery strategy, or Gateway review 3, Investment decision in the case of important and highly IT-dependent projects. If the scorecard shows problems where the Gateway review is more optimistic, a further review 0 is advisable.

Use the table below if you are conducting a Gateway Review 0 and you want to use the scorecard for specific assurance concerning IT architecture. In the leftmost column are all the ‘Areas to probe’ from Review 0 and in the corresponding columns to the right are one or more of the references to the scorecard, being the numbers of the criteria under the respective key quality aspects. An example: when you do a Review 0 and you have reached Area to Probe 1.4, then you can see that criteria 6 to 8 under key quality aspect ‘Confidence’ are relevant for that Area to Probe, as are all criteria under ‘Induction’ and criterion 5 under ‘Independence’.

It is important to remember that certain criteria will only apply in the initial stage of the project or programme. Use of the scorecard presupposes that the user has read chapter 2, which explains the fundamentals of architecting that the scorecard is based on.

Not all the references here lead to criteria that are just a further ‘specialization’ of the Gateway criterion. In some cases the user’s attention is drawn to aspects Gateway does not address.
**Table of correspondence from Gateway Review 0 to the scorecard**

(The numbers in the **cells** refer to criteria under the respective key quality aspects you find in the head of the column. The numbers in the **leftmost column** refer to the ‘areas to probe’ of the respective chapters of Gateway Review 0.)

<table>
<thead>
<tr>
<th>Scorecard Key Quality Aspects</th>
<th>Confidence</th>
<th>Induction</th>
<th>Integrity</th>
<th>Independence</th>
<th>Fairness</th>
<th>Business Case</th>
<th>Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway review 0 chapter 1: Policy and business context</td>
<td>1.1 1, 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.3 2-6, 9, 10</td>
<td>2, 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4 6-8</td>
<td>All</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.6 3, 5, 6</td>
<td>4</td>
<td>3-5</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway review 0 area to probe: Business case and stakeholders</td>
<td>2.1 1</td>
<td></td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.3</td>
<td></td>
<td></td>
<td></td>
<td>1, 2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.4 1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>3</td>
<td>1, 2</td>
<td>2</td>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.7</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8</td>
<td></td>
<td></td>
<td></td>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.9</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway review 0 area to probe: Management of intended outcomes</td>
<td>3.1 1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.2</td>
<td></td>
<td>4-6</td>
<td></td>
<td>1</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.3</td>
<td></td>
<td>3</td>
<td></td>
<td>1, 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td>Gateway review 0 area to probe: Risk management</td>
<td>4.1</td>
<td></td>
<td></td>
<td></td>
<td>2, 4, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.2</td>
<td></td>
<td></td>
<td></td>
<td>3, 4</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.3</td>
<td></td>
<td></td>
<td></td>
<td>4-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.4</td>
<td></td>
<td></td>
<td></td>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway review 0 area to probe: Review of current outcomes</td>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.2</td>
<td></td>
<td></td>
<td></td>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway review 0 area to probe: Readiness for next phase</td>
<td>6.1</td>
<td></td>
<td></td>
<td></td>
<td>1, 2</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.3</td>
<td></td>
<td>5, 6</td>
<td>all</td>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.5 3-6</td>
<td>All</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.6</td>
<td></td>
<td>4-6</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.7 3-6, 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.8</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>