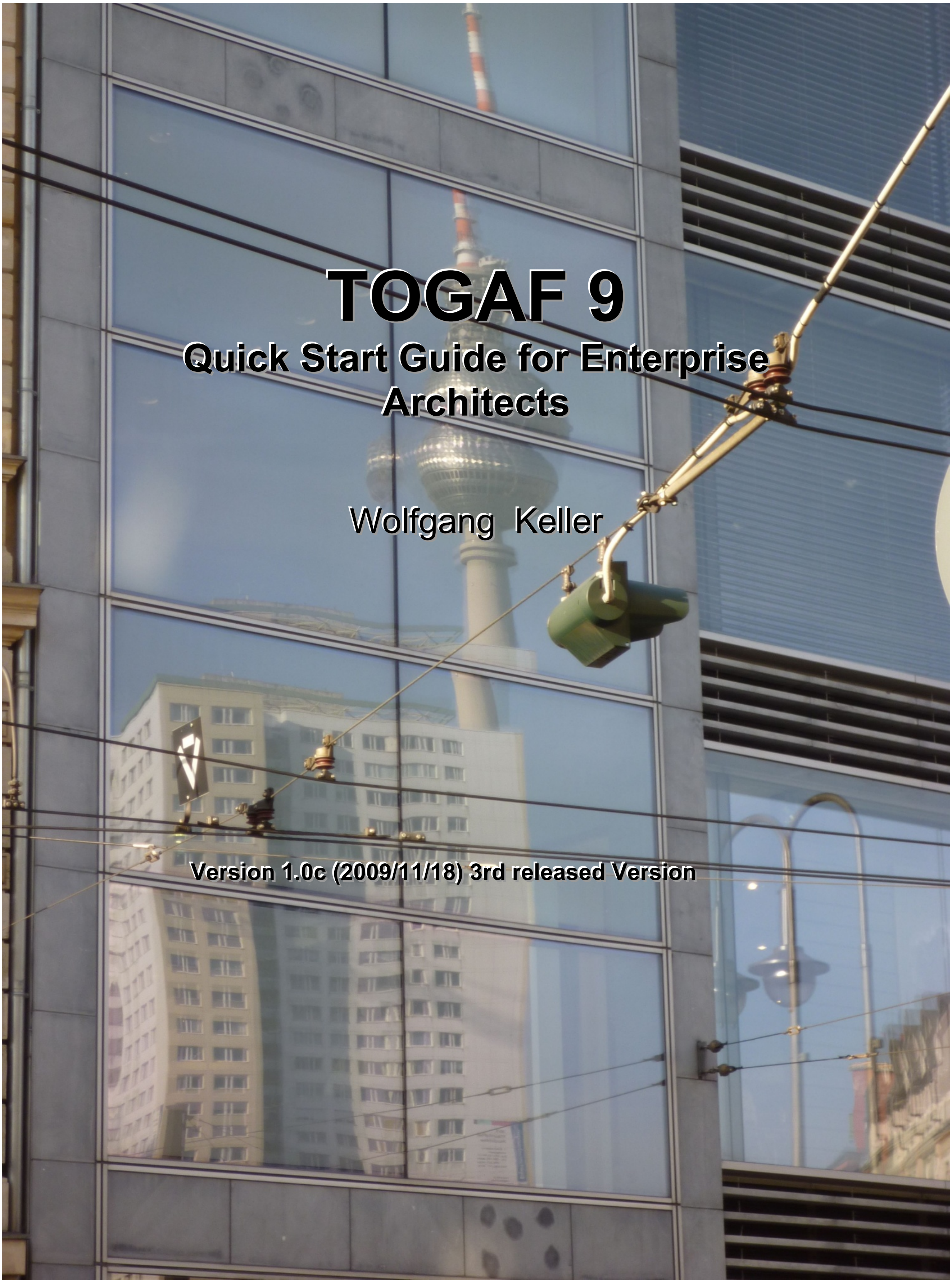


TOGAF 9

Quick Start Guide for Enterprise Architects

Wolfgang Keller

Version 1.0c (2009/11/18) 3rd released Version

Copyright and License

Disclaimer

While the author used his best efforts in preparing this book, he makes no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. The advice or strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. The author shall not be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential or other damages.

Cover Picture

By Wolfgang Keller © 2009 – all rights reserved. Location: Berlin; Hacke'scher Markt; Reflection of Berlin TV Tower.

Version History

Date	State and Purpose
2009-11-18 Version 1.0c	A few remaining typos removed
2009-10-24 Version 1.0b	Language improvements by Andrew Black
2009/07/27 Version 1.0a	Includes Feedback from various colleagues. Special thanks to Gernot Starke. Have the permission to use original figures from TOGAF in the meantime but will stay with lookalikes for the moment as this results in a more consistent layout
2009/03/28 Version 09a	First Version released. Any figures critical with respect to copyrights have been removed. Some might be replaced by TOGAF figures as soon as permission is granted. Feedback and corrections welcome. Don't ciriticize it – help improve it! ☺

Preface

Why would anybody need a 60 pages short book on TOGAF 9 if TOGAF itself is a 780 pages architecture framework, written by renowned experts from 300 top IT companies who sure know their stuff and will provide you with anything you need as an enterprise architect?

The answer is: YES – TOGAF 9 provides you with very helpful, very sound and extensive lists of WHAT to do in Enterprise Architecture. The advice from various people working in Enterprise Architecture is: Use TOGAF! There's no reason not to use it.

But TOGAF does not yet provide you with a QuickStart and Accelerators that give an Expert a fast to read ballpark view of which items of an enterprise architect's task list are covered by TOGAF and which are not. And as you will also see, there are areas of an Enterprise Architect's task list which are not covered by TOGAF at the moment. This makes it a rewarding task to give people interested in TOGAF an idea of what they can expect and what they cannot expect.

It was the initial plan of the author of this book to write just a second edition of his own book on IT Enterprise Architecture which has a substantial share on the German market. But having seen the new TOGAF 9 document and taking into account that there is an audience of about 10.000 certified TOGAF practitioners out there and maybe far more than 100.000 IT professionals who deal with TOGAF world wide it seemed an intersting task to offer a Quick Start for TOGAF 9 for such a growing audience, interested in Enterprise Architecture and interested in how TOGAF can help them do a good job at "Architecting the Enterprise"

About the Author

Wolfgang Keller is a freelance consultant who has Enterprise Architecture as his professional hobby. Until 2006 he used to work for one of the top 5 insurers world wide as a Chief Architect first for their Austrian and CEE business and later for their German business generating around 13 Billion Euro in insurance premiums.

Wolfgang has done work in the field of software architecture since the term was coined by the CMU SEI in the early 1990ties. Besides software architecture he was always in large scale software project management. At the moment he works as a project manager doing "honest work" ;-) which has nothing to do with methods but everything with delivery.

Wolfgang holds a Diplom (M.S.) in computer science from Technical University Munich (Germany). He is the author of numerous articles in English and German language trade journals and has authored two German books: One on Enterprise Application Integration and another one on Enterprise Architecture Management.

Website: http://www.objectarchitects.biz/
E-Mail: wk@objectarchitects.de

Table of Contents

1 Introduction

1.1 What is TOGAF 9?

At the present time TOGAF, the Open Group Architecture Framework is a very popular framework for Enterprise Architecture (EA) worldwide.

Version 9 – the Version released in February 2009 – is the result of a major rework of Version 8.1.1. TOGAF Version 8.1.1 was the latest update of the version 8 family which has been around since 2002, when the extensions for enterprise architecture were added to version 7 in the form of the so called enterprise continuum.

Like many other frameworks in the IT architecture and management arena TOGAF 9 is not a "use it right out of the box item". It has some 780 pages and is first of all a major challenge to read.

Also, like many other IT frameworks, TOGAF will tell you more about WHAT to do than HOW to do it. Hence, when you start e.g. a new job as an Enterprise Architect and somebody passes you a copy of TOGAF 9 you might not really know what to do, unless you have had some other work experience besides TOGAF and have read a few other books.

There is also the criticism that TOGAF does not provide you with working examples, document templates and other quickstart tools. Providing such material would indeed be very tough, as processes, documents and other material will differ depending on the kind of enterprise for which you have to develop an architecture. The good news is that TOGAF provides you with very useful checklists. The bad news is, that TOGAF will not provide you with an idiot's guide to enterprise architecture. Even with TOGAF you are still allowed and required to think. Such an idiot's guide does not yet exist and it is very unlikely that it will ever exist. Any approximation to it would consist of thousands of pages, would not be manageable and would be outdated by the time it was published.

1.2 This Book is most useful for two Groups of Readers

Experienced EA Professionals who are not yet "deep into TOGAF": Will get a quick overview on where to find what in TOGAF 9 and where TOGAF 9 has still gaps if compared to a task list of daily enterprise architecture work. People from this group should be able to read the book in a cursory style in something like an hour. Wherever you need details, you can dive into them and if this book is not enough you will get a pointer into the respective TOGAF content. To make a long story short for such readers: TOGAF is sold to you as a framework for Enterprise Architecture. It is very **strong on** developing **architectures** but it is still somewhat **weak on** Enterprise Architecture **Management**. If you are an EAM expert and you know e.g. TOGAF 8.1.1 in detail, you are done here. TOGAF 9 is not much better at EA Managament than TOGAF 8.1.1 used to be – but it is stronger on EA Development than 8.1.1 used to be. You can go on flipping through the book in order to draw some hints from the short recipes and further reading sections that you might not have come across before.

People new to Enterprise Architecture and interested in TOGAF: Will get a ballpark view of what to expect from TOGAF and what not to expect from TOGAF. You will be guided through a process framework for day to day architectural work providing you with an overview of architectural work as a whole and you will get information on which of these processes are supported by TOGAF 9 and the level of quality you can expect.

The benefit for both of the above groups of readers is that they should get an **idea** of **how** they can **profit from TOGAF** way **faster** than by reading the full original text. But they will still not get the worked examples and all the supporting material that would pile up to another thousand or more pages. Also because such material is of limited value once you change the industry or move on to an enterprise with totally different IT strategies.

If you are an **experienced TOGAF expert** then you will find that **this book is not written with you as the target audience** in mind. You are better off with one of the migration guides for transition e.g. from Version 8.1.1 to Version 9.

A short note on TOGAF certification: Having worked through this short book you will have learned that TOGAF certification can be useful to prove that you did work through TOGAF and that you understood the documents. TOGAF certification does not certify that you are a professional Enterprise Architect. Even though some HR departments may be made to believe this and even though there may be vendors for TOGAF courses out there who tell you this. Hence you can also use this book to find out quickly what TOGAF could give you and whether or not you think that it worth taking the exam.

1.3 Your Feedback Welcome

The author acknowledges that this book might attract some critique from experienced TOGAF acolytes and also from people who have a marketing pitch that sells TOGAF as the "one size fits all" solution for Enterprise Architecture.

Also some less experienced Enterprise Architecture professionals might be looking for the "thousand page start-up material".

Your feedback to improve this book is very welcome. The version you are currently reading has been written with the intention of saving a lot of people a lot of reading time and also to confirm for them that they still have an appropriate notion of Enterprise Architecture Management even though they have the feeling that there is something missing from TOGAF.

This book will be distributed free of charge over the Internet while it is planned that the book will later be used as part of a regular book. Feel free to contact me with your suggestions for improvement.

2 What is Enterprise Architecture?

As with building architecture there are many many defintions of Enterprise Architecture. In this chapter you will learn to recognise a few access paths to Enterprise Architecture and we will give a deeper explanation of a pragmatic access path that sees an Enterprise Architect as a very important aide to the CIO who is in charge, amongst other things, of strategic planning for the enterprise's IT assets. We will call this the **Pragmatic Business Approach**. It will become clearer, once we outline the tasks of an Enterprise Architect as top aide to the CIO.

The other two approaches are:

The IT-Architecture Approach: This approach mainly has its roots in IT systems architecture. The systems architects often get to be in charge of a system cluster or even of a whole IT landscape and have tried to evolve their systems architects' methods for use at the enterprise level. You will learn that both the Zachman Framework and TOGAF have been greatly influenced by this approach.

The Academic Approach: This approach first asks "what is architecture of an enterprise" and then looks for methods for modeling an enterprise, constructing meta-models for an arbitrary enterprise, evolution of meta-models and similar questions. Such questions focus on how to model an enterprise from top to bottom.

In this chapter we will explain the three approaches and will position TOGAF 9 with respect to their viewpoints. In the section on the Pragmatic Business Approach you will become familiar with a list of top level tasks which will be used later to give you an idea of which tasks are supported by TOGAF 9 – and which are NOT supported by TOGAF 9.

2.1 The Pragmatic Business Approach to Enterprise Architecture

The Pragmatic Business Approach to Enterprise Architecture starts by asking "what needs to be done to make the most of the enterprise resource IT". In more educated circles this is called "IT / Business Alignment". IT / Business alignment is not a fully deterministic task. The way you try to pursue it depends on the maturity level of your organization.

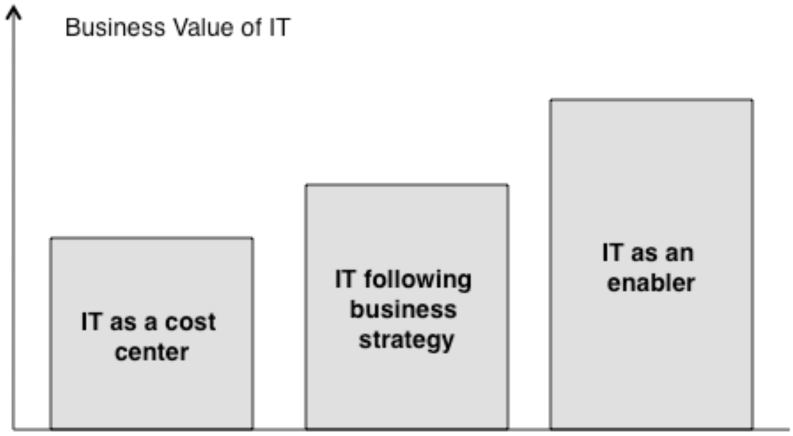


Figure 1: Levels of IT / Business Alignment.

There is a predictable set of tasks in a CIO office that serve to govern the expensive resource IT in an enterprise. The degree to which such a CIO office has implemented these depends on the importance of the production factor IT for the enterprise

- An enterprise which deals with commodity goods and does not see IT as some kind of factor relevant for its strategic positioning in competition will seldom ever have a complex CIO office. If it even has a CIO.
- A multi-billion business running an IT budget of several hundred million Euros a year will have all the functions that will be outlined here – maybe even more.

In a typical CIO office you will find more or less the blocks of tasks outlined in *Figure 2*. Only two of these have something to do with enterprise architecture.

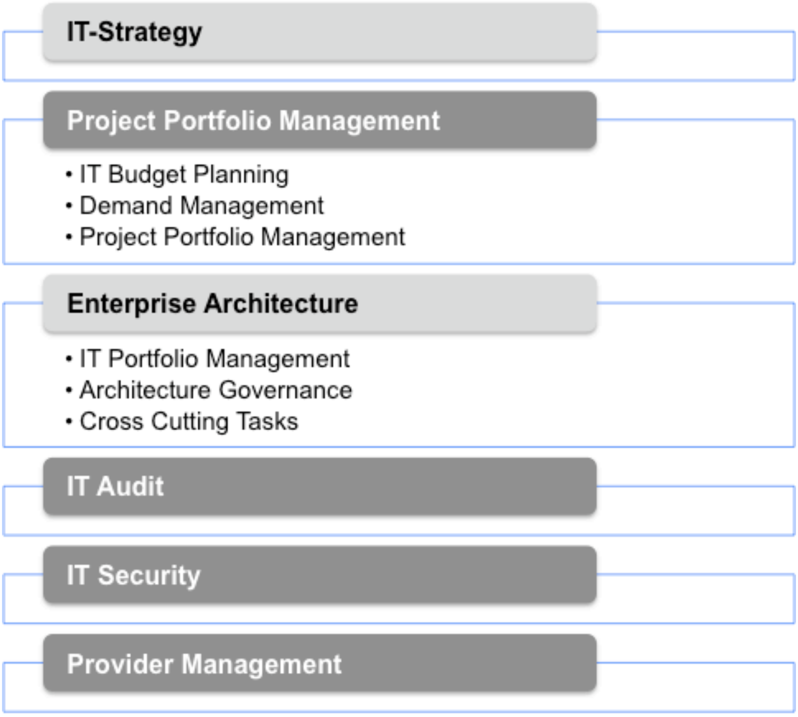


Figure 2: Task blocks for a CIO Office. Dark shaded topics are not subject of EAM and will hence not be discussed in this book

- **IT Strategy:** Somebody has to define a strategy for the management of the enterprise's resource IT.
- **Project Portfolio Management:** Budgets for IT have been around much longer than IT project architecture or even Enterprise Architecture. That's why a manager like a CIO used to have somebody in charge of his IT budget and the balancing of budget and demands that results in the list of demands which are to be implemented each year.
- **Enterprise Architecture:** We will explain the tasks an Enterprise Architect has to perform in section 2.2.
- **IT Audit:** As the IT function is crucial to the success and sheer existence of most enterprises today, IT is subject to audits in most organizations. Hence there is a task to audit IT functions. Today's de facto standard for auditing IT is COBIT. We will not drill into this any deeper as it will not help you too much in understanding TOGAF.

- **IT Security:** is crucial today for the reputation, integrity and security of your enterprise. Hence a CIO will have people in charge of IT security. There's also a broad array of frameworks that will help you manage IT security. We will also skip this aspect for the time being
- **Provider Management:** Most IT organizations today outsource or outtalk a lot of their work. This can apply to almost all tasks an IT organization performs, except the core management tasks. These core tasks are, by the way, described here for a CIO office.

In the remainder of this book we will be dealing with the IT Strategy and the Enterprise Architecture blocks. Very often the Enterprise Architect, as a CIO's top aide, will help him define the enterprise's IT Strategy. In any case, Enterprise Strategy and hence IT Strategy are the most important influences for you as an Enterprise Architect, even if your CIO does not consult you about formulating them.

This and the elements of Enterprise Architectural work will be described in section 2.2. The structure explained here will be used later in chapters 3 thru 7 to explain what is in TOGAF to help you accomplish your tasks as an Enterprise Architect.

2.2 Work Breakdown Structure for Enterprise Architecture

The tasks of Enterprise Architecture can be split into three main blocks as outlined in Figure *Figure 3*.

- **Strategic Tasks:** Often the Enterprise Architects are the people who help a CIO ***develop his IT Strategy***. But besides this there are more strategic tasks –meaning tasks that have a planning horizon of more than 3-5 years. ***IT Portfolio Management*** will deliver the basic data needed for ***Strategic Planning,*** which brings together the goals from strategy and the as-is situation from portfolio management in order to develop a to-be situation. This will be underpinned by a strategic roadmap that is a coarse program plan for a major part of the project portfolio.

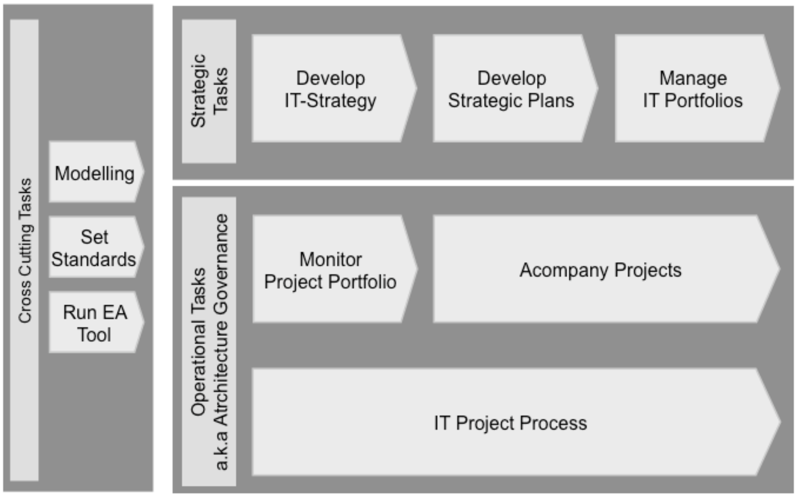


Figure 3: Enterprise Architects Process Map

- **Operational Tasks:** form the day to day work of Enterprise architects. Strategies are nice – but you need to communicate them and you also have to make sure that they are applied and implemented. This is the field of ***Architecture Governance***. First of all you will need to find critical projects – the ones that have the potential to change your architecture. This is ***done by Monitoring the Project Portfolio***. Once you have identified the 10% or so interesting strategic projects, you will set up an architects team to ***accompany them***. As they run through the enterprises normal ***IT Project Process***.
- **Basic Tasks:** In order to get Enterprise Architecture up and running you will need to create a few foundations. In many cases it will be useful to ***run an EA tool*** in order to have a chance to track what you have in your application and infrastructure portfolios. In order to do this you will need to find or develop the right ***meta model*** **f**or the purposes and in order to reduce complexity by standardization you will first have to ***develop standards*** that will be valid in the scope of your enterprise. There are more basic tasks – but these are the most prominent ones.

The following will explain each of the tasks listed above and in chapters 3 thru 7 you can find what material is available in TOGAF to help you perform them.

2.3 The Roots of the IT-Architecture Approach

In this section we will give a short history of software and also of software architecture. The history of software development has produced a set of abstractions and metaphors that over time gradually became stronger. This applies to all software related disciplines, including design of programming languages, platforms, processes, organizations and tools used to develop software. This is due to the fact that we are confronted with a quantity of software that is growing exponentially since the term software itself appeared in the mid 1950s.

2.3.1 Software Architecture as a Profession

The job title software architect is very "en vogue" today. It has only been around for maybe 20 years. Nowadays it is tough to find programmers. Most people in the trade will try to call themselves software architects, also because the pay is much better and because the title is more appealing than just "programmer".

The term architecture has been around for at least 20 years in the world of software. Architecture as the art of building design has been around for thousands of years. There was architecture even before people built the pyramids. Compared to that software itself is very young. At the beginning of the 1950s software was not even an independent item of trade. Software as we know it today, an artifact independent of the hardware, began to develop in the ninteenfifties. The GUIDE / SHARE organization that dealt with exchanging software for IBM computers, was founded in 1955. Following that, software projects often developed operating systems. The system OS/360 was a typical representative of such projects. Fred Brooks the chief designer of this project published his experiences in the well-known book "The Mythical Man month" [Brooks75]

The leading figures at the time performed Programming in the Large[1]. They met in 1968 at the famous NATO conference in Garmisch in order to think about how to cure the "software crisis" by applying the new metaphor of software engineering. Software architects were not to be seen at that conference or at that time.

At that time the foundations were laid for techniques such as data abstraction and modularization. The research was mainly focused on programming. Instead of the then common machine languages they created higher level abstractions for programming in order to deal

[1] Programming large systems comprised of many software components

with the exponentially growing quantity of software. Programming languages that contained less powerful abstractions, such as COBOL appeared in the early nineteen-sixties, while more powerful languages like ADA appeared around 1980. And still there were no software architects to be found in the programming crowd. A programmer was still the thing to be.

It was not until the mid nineteen-nineties that the term "Software Architecture" was documented in a systematic fashion [GarlanShaw96, Bass+98]. Software architecture like software engineering is a metaphor that tries to explain processes around the creation of software using analogies to other disciplines of engineering – in this case building architecture. People tried to transfer methods and processes from building architecture to the creation of software. At about the same time the Gang of Four [Gamma+95] created the foundation to describe recurring solutions in software using design patterns. It has to be noted that Gamma were not the first ones who transferred patterns from Christopher Alexander's [Alexander79a, Alexander79b] works to software design. You can find so called architectural styles that form the repertoire of many software architects nowadays either as formal architectural styles as described by Garlan and Shaw [GarlanShaw96] or as architectural patterns as described by Buschmann et al. [Buschmann+96]. Software professionals, who were using these abstractions tended to call themselves software architects later on. From that point in time the profession of software architect, as we know it today started to evolve. This profession can be taught in seminars and university courses and there are also certifications for software architects. There is nowadays a reasonable measure of agreement on the curriculum for software architects. And as this group receives far better pay than "just plain programmers" you can observe a tendency of people to migrate into software architecture instead of just plain programming.

2.3.2 Enterprise Architecture

The growth of software did not stop at the border of single (even large) software systems. The metaphor of Software Architecture is not good at explaining how to deal with complex landscapes that are composed from hundreds or even thousands of major software systems. Hence we can observe the evolution of a new metaphor, called the City Planning Metaphor.

The nineteen-eighties and nineties saw a lot of so called stovepipe architectures. Systems were very well integrated vertically from the user interface to the database layer. They were not so good (and often still are not so good) at horizontally integrating business processes. Horizontal integration often happened (and still happens) via the likes of data integration and batch data exchange.

You might say: Long ago – but remember that workflow systems have been around for roughly 15 years - in the perspective of building architecture this is just a blink of an eye. The people who pioneered these technologies in the nineteen-nineties did not even call themselves “software architects” – and no way “enterprise architects”.

Nevertheless there were people practicing enterprise architecture at that time – even though they did not know it. Terms like “application portfolio” that will be discussed in section 4.2 in this book did not exist in those days and are still not present in TOGAF (see section 4.4).

If you are a fan of John Zachman’s work you might state that he created the foundation for Enterprise Architecture in his famous paper [Zachman87, Sowa+92]. A closer look at that paper plus a look at the tasks enterprise architects perform today (see e.g. section 2.1) will show you that Zachman’s work is project architecture for very large projects. If you search this work (and also TOGAF) for terms like Application Portfolio, Zoning Maps, and also Governance you get few to zero results.

The City Planning Metaphor appears some 10-15 years later than John Zachman’s work e.g. in a book by Longépé [Longepe03] which is far less well known than the famous articles by Zachman.

2.3.3 Enterprise Architecture using the City Planning Metaphor

If a Software Architect deals with a single system or maybe a cluster of 7 or 10 systems then an Enterprise Architect has to deal with the set of all applications of an enterprise. For a small enterprise this set may have 50 applications, for a major insurance company the figure might be 300 systems and a global car manufacturer typically has a portfolio of more than 3000 – 4000 IT applications – individual applications for employees workstations not counted. An Enterprise Architect as the CIO’s city planner has to deal with almost any group of stakeholders in a company, be it senior management, be it groups of users, or be it public agencies like the SEC (Securities Exchange Commission) or agencies that deal with data protection, and last but not least those groups who want a new application. As these tasks have some similarities to city planning, the profession of Enterprise Architects has sometimes adopted that term for their occupation. If you look at the zoning maps, city planners are using, then the analogies go even farther.

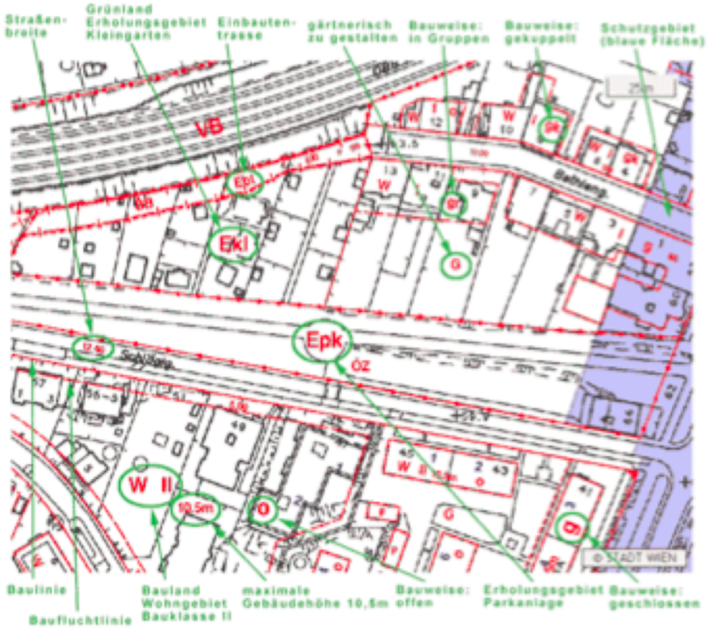


Figure 4: Zoning Map; Part of the City of Vienna, Austria

2.3.4 Preview: What has this got to do with TOGAF

City Planners do not provide detailed plans for a single building. They will provide zoning maps that define which kinds of buildings will be arranged in which zone. They care for the set of all buildings of a city even though they would also have the qualification to plan a single one.

In section 4.3 you will also get a glimpse of zoning maps for Enterprise Architecture and also other artifacts, which are frequently used by Enterprise Architects. In sections 4.1 thru 4.6 you will get a rather clear demonstration that TOGAF supports you with methods mostly derived from the Building Architecture Metaphor, but not with the later artifacts needed to help you when looking at your applications as a portfolio of assets that needs to be managed. Given the fact that TOGAF has it's roots in the DoD's TAFIM framework, which was handed over to the OpenGroup as early as 1995, this is somewhat natural just looking at the period in which TOGAF and TAFIM were created.

2.4 A Few Notes on the Academic Approach

As one sample for the academic approach to enterprise architecture, we will use the approach pursued by the (let's call it) St. Gallen School of enterprise architecture. St. Gallen University is a very renowned school for business administration and also business oriented computer science. This "school" uses a definition of Enterprise Architecture, which is also promoted by Open Group.

ENTERPRISE ARCHITECTURE: DEFINITION

According to ANSI/IEEE Std 1471-2000, architecture is defined as the "fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" (IEEE 2000). Enterprise architecture (EA) therefore is understood as (1) the fundamental organization of a government agency or a corporation, either as a whole, or together with partners, suppliers and / or customers ("extended enterprise"), or in part (e.g. a division, a department, etc.) as well as (2) the principles governing its design and evolution (OpenGroup 2003) .

While an EA model is a representation of as-is or to-be architecture of an actual corporation or government agency, an EA framework provides (OpenGroup 2003)

- One or more meta-model(s) for EA description,
- One or more method(s) for EA design and evolution,
- A common vocabulary for EA, and maybe even
- Reference models that can be used as templates or blueprints for EA design and evolution.

The components of an EA framework should be applicable for a broad range of corporations and government agencies.

Source: Robert Winter, Ronny Fischer: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture; [WinFis06]

If you see this definition you might think that the definition comes right from OpenGroup of TOGAF. This is not exactly the case. TOGAF and also the academic approach both like to cite IEEE 1471-2000 for the basics of Software Architecture. But the above definition is a compilation of various spots from TOGAF and way more rigorous than the sources it is drawn from. While TOGAF, as we will see in chapter 5 is interested in how you develop a concrete architecture, many academics are more interested in laying the foundations by e.g. dealing with questions of how to model an enterprise the right way. Some sample questions of current interest to the St. Gallen School include:

- How do you model an enterprise as a whole from strategy via capabilities and business processes down to the level of infrastructure while being able to deduce properties or make statements about what is “good or bad” with respect to the strategic goals of an enterprise
- How do you find a proper meta-model for enterprise architecture in your enterprise
- How to you evolve or federate meta-models.

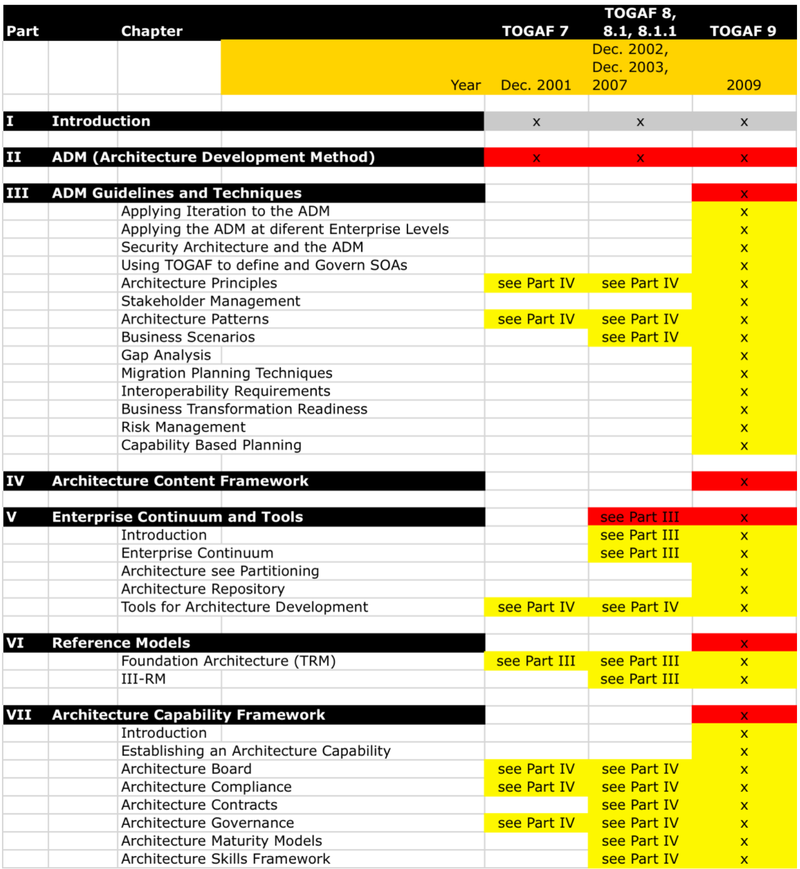
From a research perspective all these questions are highly interesting and need to be asked with the long term perspective in mind that you want to do “real Enterprise Architecture” – which means you do not only architect the IT part but the whole enterprise and you model Business and IT as a whole.

2.5 The TOGAF 9 Approach

After you have been tortured with three approaches to Enterprise Architecture and since this is a book on TOGAF it is somewhat overdue now that you get an idea of how TOGAF sees its approach to enterprise architecture. Best let TOGAF speak for itself:

> **TOGAF as an EAM Framework (From TOGAF 8.1. (Enterprise Edition)**
> TOGAF in its Enterprise Edition remains what it has always been, namely an architecture framework - a set of methods and tools for developing a broad range of different IT architectures. It enables IT users to design, evaluate, and build the right architecture for their organization, and reduces the costs of planning, designing, and implementing architectures based on open systems solutions.
>
> The key to TOGAF remains a reliable, practical method - the TOGAF Architecture Development Method (ADM) - for defining business needs and developing an architecture that meets those needs, utilizing the elements of TOGAF and other architectural assets available to the organization.
>
> Source TOGAF 8.1. [TOGAF8.1.1] (Also valid for TOGAF 9)

So what does this tell us with regards to Enterprise Architecture tasks as described in section 2.2? **The emphasis of TOGAF is on developing concrete architectures**: Be it for a system, be it for a cluster of systems and maybe also be it for a to-be architecture of an enterprise as a whole. Hence the emphasis is on the concrete task of developing architecture.



Part	Chapter	TOGAF 7	TOGAF 8, 8.1, 8.1.1	TOGAF 9
	Year	Dec. 2001	Dec. 2002, Dec. 2003, 2007	2009
I	Introduction	x	x	x
II	ADM (Architecture Development Method)	x	x	x
III	ADM Guidelines and Techniques			x
	Applying Iteration to the ADM			x
	Applying the ADM at diferent Enterprise Levels			x
	Security Architecture and the ADM			x
	Using TOGAF to define and Govern SOAs			x
	Architecture Principles	see Part IV	see Part IV	x
	Stakeholder Management			x
	Architecture Patterns	see Part IV	see Part IV	x
	Business Scenarios		see Part IV	x
	Gap Analysis			x
	Migration Planning Techniques			x
	Interoperability Requirements			x
	Business Transformation Readiness			x
	Risk Management			x
	Capability Based Planning			x
IV	Architecture Content Framework			x
V	Enterprise Continuum and Tools		see Part III	x
	Introduction		see Part III	x
	Enterprise Continuum		see Part III	x
	Architecture see Partitioning			x
	Architecture Repository			x
	Tools for Architecture Development	see Part IV	see Part IV	x
VI	Reference Models			x
	Foundation Architecture (TRM)	see Part III	see Part III	x
	III-RM		see Part III	x
VII	Architecture Capability Framework			x
	Introduction			x
	Establishing an Architecture Capability			x
	Architecture Board	see Part IV	see Part IV	x
	Architecture Compliance	see Part IV	see Part IV	x
	Architecture Contracts		see Part IV	x
	Architecture Governance	see Part IV	see Part IV	x
	Architecture Maturity Models		see Part IV	x
	Architecture Skills Framework		see Part IV	x

Figure 5: Evolution of TOGAF from Version 7 to Version 9. Source: Own Research.

The emphasis of TOGAF is not on tasks such as

- Developing an IT Strategy based on a Business Strategy
- Application Portfolio Management: Dealing with thousands of existing applications and judging which have a future, which need change and which need to be replaced
- Architecture Governance: This is mentioned but not treated as a main item

The ADM (Architecture Development Method) is still the kernel of TOGAF and is TOGAF's biggest asset. You can also see this concentration on the ADM yourself if you study the evolution of TOGAF from e.g. Version 7 (published in Dec. 2001) thru various versions of TOGAF 8 (which was introduced as the "Enterprise Version of TOGAF 7" and first featured the "Enterprise Continuum") to TOGAF 9 which was published in February 2009.

Figure 5 can be read so that Part II – the ADM – has been rather stable throughout the evolution shown in *Figure 5*. The major new features of Version 9 over Version 8.x are the Architecture Content Framework and the material added in Part III: The ADM Guidelines and Techniques.

So it can clearly be seen from this, that TOGAF worked its way up from a framework for project architecture and major development projects to the enterprise level. TOGAF will continue to grow towards the enterprise level but it has not been designed from the beginning with the Enterprise Level or a Pragmatic Business Oriented Approach in mind. But let TOGAF (again 8.1.) speak for itself again:

> A number of enterprise architecture frameworks already exist and are widely recognized, each of which has its particular advantages and disadvantages - and relevance - for enterprise architecture. They are discussed in Part IV (of TOGAF 8): Resource Base, Other Architectures and Frameworks.
>
> Although a number of enterprise frameworks exist, there is no accepted industry standard method for developing enterprise architecture. The goal of The Open Group with TOGAF is to work towards making the TOGAF ADM just such an industry standard method, which is neutral towards tools and technologies, and can be used for developing the products associated with any recognized enterprise framework - such as the Zachman Framework, Federal Enterprise Architecture Framework (FEAF), Treasury Enterprise Architecture Framework (TEAF), and C4ISR/DoD Framework - that the architect feels is appropriate for a particular architecture.
>
> ….
>
> With the migration of TOGAF to an enterprise architecture framework, this flexibility becomes even more important. TOGAF is not intended to compete with these other frameworks; rather, it is intended to perform a unique role, in distilling what these other frameworks have to offer, and providing a generic ADM that can be adapted for use with any of these other frameworks.
>
> The Open Group's vision for TOGAF is as a vehicle and repository for practical, experience-based information on how to go about the process of enterprise architecture, providing a generic method with which specific sets of deliverables, specific reference models, and other relevant architectural assets, can be integrated.
>
> Source TOGAF 8.1. [TOGAF8.1.1]

2.6 Summary

So far we have seen that there are various approaches to Enterprise Architecture depending on ones background and objectives. The rest of this book will use the Enterprise Architects task list and will go into each major block of tasks, demonstrating what TOGAF has for you if you want to perform that kind of task and giving you pointers to additional material that you might want to use, when you are confronted with that kind of task.

3 TOGAF and IT Strategy

In the early reactions to TOGAF 9 bloggers commenting on the new release have criticized TOGAF for falling short of providing help for the strategic planning tasks in IT management. The three main strategic tasks have already been outlined in section 2.1 and IT strategy is the first one of them.

A first check is quite simple: Just take the TOGAF pdf file, which is available at a modest charge from the OpenGroup and have the occurrences of the term “IT strategy” counted in it. You will find the term three times in the text and this makes it clear that TOGAF does not provide you with a method, to develop one.

If you just wanted to know, whether TOGAF provides you with a closed set of tools and methods to develop an IT strategy, you’re done for the moment. Skip this chapter and GOTO chapter 4 on page 26 in order to see what TOGAF has to offer you for portfolio management.

If you want to know where to find material on how to write an IT strategy and what TOGAF still has to offer besides obvious spots marked with IT strategy, hang on.

3.1 What is an IT Strategy?

In order to talk about IT strategy it is first necessary to talk about strategy in general. In order to avoid copyright problems we use a Wikipedia definition

> A strategy is a plan of action designed to achieve a particular goal.

You could also use the more militaristic version by Clausewitz [Clausewitz98] but there’s no real difference.

First of all you need a goal. Then you need more than one possible way to get there and formulating your strategy is about choosing one of the possible ways to go and turning it into a plan.

You might ask, whether one way (or option for attaining the objective) isn’t enough? Many strategies do not pass the triviality test where you ask whether the opposite of the way you choose is still something that makes sense. E.g. if you do not hold a monopoly (you are in a competitive market) and you state “Customer Orientation” as your strategy, then the contrary “being non-customer oriented” would lead straight into bankruptcy. Hence for this example enterprise “Customer Orientation” as a strategy would not pass the triviality test because there’s no alternative to it.

You should find some indications of the strategy of your enterprise in its business strategy. We will also deal with the case where your enterprise doesn’t have one. But let’s postpone that one for a moment. That case will be covered in section 3.3. You also need to know something about the maturity of your IT organization. If IT is treated as a pure non-strategic cost-center in your organization (remember *Figure 1* for the different steps of IT / Business alignment of an IT organization) then you will have a tougher time influencing your enterprise’s strategy. When IT is seen as an enabler in competition, you should have less trouble entering into a true strategic dialog.

3.2 What needs to be described

No matter at what level your IT organization finds itself – the basics that need to be described (strategy elements) are always the same.

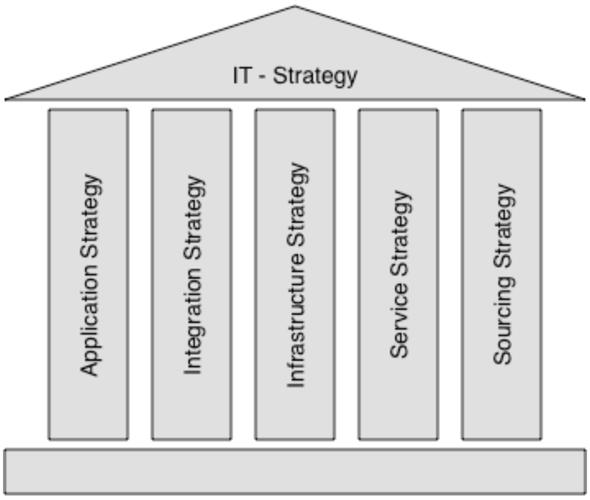
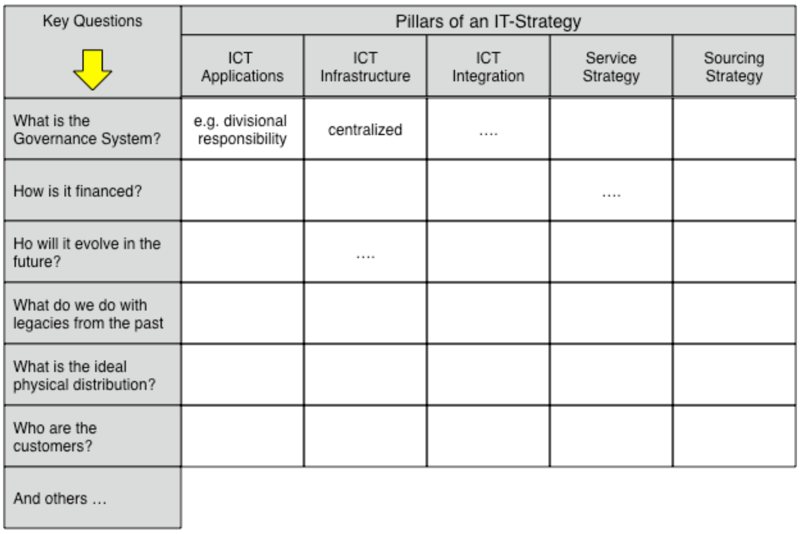


Figure 6: Main Pillars of an IT Strategy

The above figure can use a bit of explanation. It is also reflected in the strategy matrix (see Figure 7) to follow. No matter what else you have in an IT strategy. You will always need to make decisions about the following:

- **Application Strategy:** No matter what else you plan in your IT strategy, you need an application strategy – a strategy that contains guidelines on how you want to deal with the IT applications in your company and how you want to support your business strategies using IT applications.
- **Integration Strategy:** In most cases you don't have one perfectly integrated application but you have a major set of applications that need to be integrated with each other in order to be able to perform business processes. These apps also might need to be connected to the outside world. SOA or EAI strategies are forms of integration strategies. But there are also more primitive ones.
- **Infrastructure Strategy:** You also cannot evade having an infrastructure strategy. Even if you outsource your complete infrastructure, you still have a strategy how to deal with it: in this case "Outsourcing". Other factors that determine your infrastructure strategy is your global scope, your needs for security and special solutions and many other factors that you would check if you use a matrix like the one in Figure 7.
- **Service Strategy:** If you want to be perceived as a provider of services, you will end up having a service strategy, that determines how your customers will get which services at which service levels plus some more decisions concerning
- **Sourcing Strategy:** Even if you should decide that you will produce all your IT assets vertically integrated starting with developing your own operating system: you still have a sourcing strategy. Normally you will make decisions here on what to outsource, and what to produce yourself. You will also decide whether you want to work together with a single provider, multiple providers, whether you want to source in an opportunistic fashion or relying on a few strategic partners to name a few of the decisions that need to be made here.

If you define what you want to do and achieve in those 5 fields and if you can explain why this enables your business to pursue its Business Strategy, you are well off. In order to help you think about your decisions systematically and even deeper than that, you can apply key questions to each pillar the likes of "how do I govern that specific area?"; "how do I finance it?", and a few more. As an example see Figure 7 below. This will force you to reflect on what you will do about certain general concerns, such as the regional distribution of your business.



Key Questions	Pillars of an IT-Strategy				
↓	ICT Applications	ICT Infrastructure	ICT Integration	Service Strategy	Sourcing Strategy
What is the Governance System?	e.g. divisional responsibility	centralized	….		
How is it financed?				….	
Ho will it evolve in the future?		….			
What do we do with legacies from the past					
What is the ideal physical distribution?					
Who are the customers?					
And others …					

Figure 7: Sample IT Strategy Matrix

This matrix is like a checklist. If you are able to answer the questions in the matrix then you can be sure to a certain degree that you did not miss a major issue that you should have taken into consideration.

You can then write down your five strategy elements. If you're good at it, you should have a short document of maybe less than 20 pages that informs your IT people about the strategic directions in each column and field.

3.3 How do you get there?

The next question is: What is the process in which you need to involve Business that leads you to such a strategy document?

3.3.1 Strategic Dialog

The best way would be if your CIO and a few key IT people meet with key Business people in order to discuss and decide on a Business Strategy that also includes the IT Strategy.

This state of affairs is somewhat rare. Analysts state that about 90% of enterprises do not have a written business strategy. Often CxOs do not spent the time for a proper strategic dialog or find it annoying to talk to IT underlings about business strategies. In such cases you will need to find other ways to extract the information you need in order to formulate a proper IT strategy, which will lead you to the Maxim Process.

3.3.2 IT Maxims from Business Maxims

The Maxim Process is described by Broadbent and Kitzis in [Broadbent+05] as a pragmatic way to extract enough information for a good enough IT strategy while not investing more than a day's workshop with senior management. The CIO will organize a workshop with CxOs, which will lead to documenting 2 kinds of so-called Maxims:

- Business Maxims,
- And as a result IT Maxims

Maxims are a few concise principles that are used to document the strategic direction of an enterprise. A Maxim workshop will usually not produce more than around 5 business maxims. For each of those, management will derive 4-5 maxims for the IT function that will help to support them.

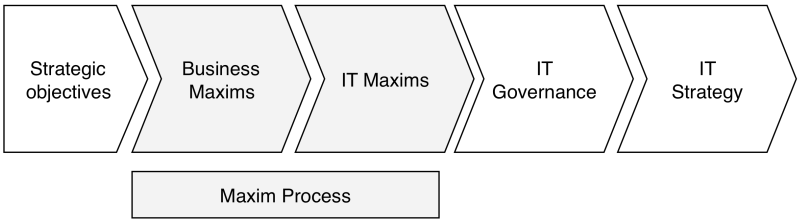


*Figure 8: Maxim Process by Broadbent [*Broadbent+05*]*

A typical Maxim Workshop will be split up into two parts:

- Part 1: Finding the Business Maxims,
- Part 2: Deriving the IT Maxims

An external facilitator should moderate the workshop day and process.

To give examples imagine an old economy financial service provider like a big insurance company that runs more than one brand name on the market. For such an enterprise you could find the following business maxim:

Create synergies in back office and service functions wherever brand identity is not compromised

IT maxims that could be deducted from such a business strategy could be:

(1) Define standard architectures and platforms used by all of the group's companies in order to leverage synergies and to reduce IT cost

(2) Harmonize the IT application systems for the group's companies wherever there is a business case for this.

(3) Support the business by providing harmonized business processes for all brands of the group.

Such a synergistic strategy (or any other strategy) will have implications on one's IT governance principles. Your set of IT maxims defines the IT objectives and the chosen ways for reaching them. (See also *Figure 8*) You will need to install a system of IT governance that matches your IT maxims.

You might say: Easy and straightforward process. There should be no problem to hold a meeting, define a few IT maxims and off you go with a usable IT strategy. Life is not always that easy. There are more than a few things that can go wrong here:

- One we mentioned already above: If the CxOs see IT as a bunch of underlings and not as an important provider of business opportunities, they will not even invest the time in such a workshop.
- Another potential threat is less evident. If the board of CxOs has a culture of hiding conflicts and if they do not agree on a common strategy they will not be too interested that their internal brawl becomes known throughout the enterprise. In such a case you will never get the maxim workshop for whatever reasons.

This will lead you to the next level. You will need to reengineer an IT strategy from the behaviors you observe in daily business. Which will bring us to the next section.

3.3.3 Reengineering the Business Strategy

If your management does not want to spend the time to discuss Business Strategy and IT strategy with you, its time for you to reengineer the strategy from what you experience. There are enterprises around who are very successful and have a strategy or call it a very successful way of operating: But it's neither written down nor could anybody explain it explicitly.

In such a case you need to reengineer or find the business maxims by applying analogies with known strategy patterns. Literature will provide you with an ample body of knowledge on Business Strategy and will enable you to identify the patterns comprising your enterprise's strategy.

Here again a matrix like the one presented in Figure 7 is useful. You can ask questions like "what is the geographic distribution of my enterprise" and "what are best practice patterns to deal with just that kind of geographic distribution when it comes to services, applications, infrastructures or sourcing. That way you can again go through the matrix (Figure 7).

3.4 What support will you find in TOGAF?

As you can see from the TOGAF overview picture[2], TOGAF treats Business Strategy and also IT Strategy as an item outside of the scope of TOGAF.

Phase A (Architecture Vision) of the ADM [TOGAF9, Chapter 7, pages 81ff] tells you that you should get hold of the business drivers before you start an architecture project of some kind. TOGAF also proposes a capability analysis. Still the combination of Business Strategy and IT Strategy seems to be considered as something outside the scope of at least the TOGAF framework.

3.5 Further Reading

Gartner Material on IT Strategies

Gartner usually has some interesting material on IT strategies. We cannot cite the stuff here because their Intellectual Property policy prohibits citations of material older than one year. Have a look at the Gartner Research databases if you happen to have a relation with them.

Marianne Broadbent used to be a Senior Vice President at Gartner, and also served as a professor at Melbourne Business School. She has consulted and interviewed a four-digit number of CIOs for various panels by Gartner. Her book "New CIO Leader: Setting the Agenda and Delivering Results" [Broadbent+05] is a classic on IT strategy;

MIT Sloan School Material on IT Strategies

Gartner is also connected to the MIT's Center for Information Systems Research (CISR) headed by Peter Weill. A very interesting book is "Enterprise Architecture as Strategy: Creating a Foundation for Business Execution" [Ross+06] by Jeanne Ross and Peter Weill. The book will also point you to further MIT resources or check the MIT website at http://mitsloan.mit.edu/cisr/.

[2] For the respective TOGAF picture see http://www.opengroup.org/architecture/togaf9-doc/arch/Figures/01_structure.png The author has to ask for permission before reprinting the material.

Consulting Tools

If you develop strategies it very helpful to have a working knowledge of methods and tools that were developed by the likes of BCG, McKinsey and other well renowned consulting firms. A convenient way to get such information is to get yourself books on consulting toolkits, strategy consulting or just a compact MBA book for starters. There's too much such literature around to be counted.

4 TOGAF and IT Portfolio Management

TOGAF has almost nothing to offer for IT portfolio management. Hence we will give a short summary of Application and Infrastructure Portfolio Management and will set pointers to the few spots in TOGAF, which are of some help for these tasks.

4.1 What is IT Portfolio Management

IT Portfolio Management comes in at least three flavors. Two of them are relevant for Enterprise Architects. And the third flavor is also present in a CIO office and hence Enterprise Architects will at least have an interface to it. The flavors are:

- **Project Portfolio Management:** Of all the flavors of portfolio management, this is the most widespread one. Even if companies don't care about architecture they usually care about, how their money is spent. It is quite a common situation that a company has two or three times more budget proposals for projects than it has money to spend on projects. This explains why a mechanism for project prioritization is installed in almost any enterprise. As stated above, Project Portfolio Management has an interface to Enterprise Architecture but is usually not seen a part of it. This is why this subject is not treated any deeper in this book.
- **Application Portfolio Management:** Most Enterprises have a considerable number of applications that support their business processes. Below you will learn why it is important to have an inventory of applications and what goals "managing" them pursue. Application Portfolio Management is usually seen as a part of strategic Enterprise Architecture Management. Therefore it will be treated below in some more detail.
- **Infrastructure Portfolio Management:** Often Enterprises do not have a proper IT inventory. Nevertheless some of those who do not have a 99% accurate IT inventory still manage their Infra-

structure Portfolio. The reason behind this is in most cases cost savings by standardization. Which can be translated to: The less technologies you own, the lower your maintenance and administration costs. For most companies IT infrastructure is not a matter of competitive advantage. Therefore they can afford to manage it as a commodity where the cheaper solutions (fewer technologies, fewer locations) are often the better solutions.

The rest of this chapter will explain Application Portfolio and Infrastructure Portfolio Management. We will start with Application Portfolios, just because this is handier to explain. Infrastructure Portfolio can often be even more rewarding as Enterprises usually spend much more money on IT infrastructures than they spend on applications.

4.2 What is Application Portfolio Management?

In order to understand why you would like to deal with portfolio management, it is somewhat helpful to explain the terms

Let's start with the term portfolio. A portfolio is a collection of investments held by an institution or a private individual.

This means, in a "normal" enterprise you have an IT infrastructure portfolio and also an IT application portfolio, because you simply possess a set of infrastructure components and also a set of IT applications. And usually you have heavily invested in them.

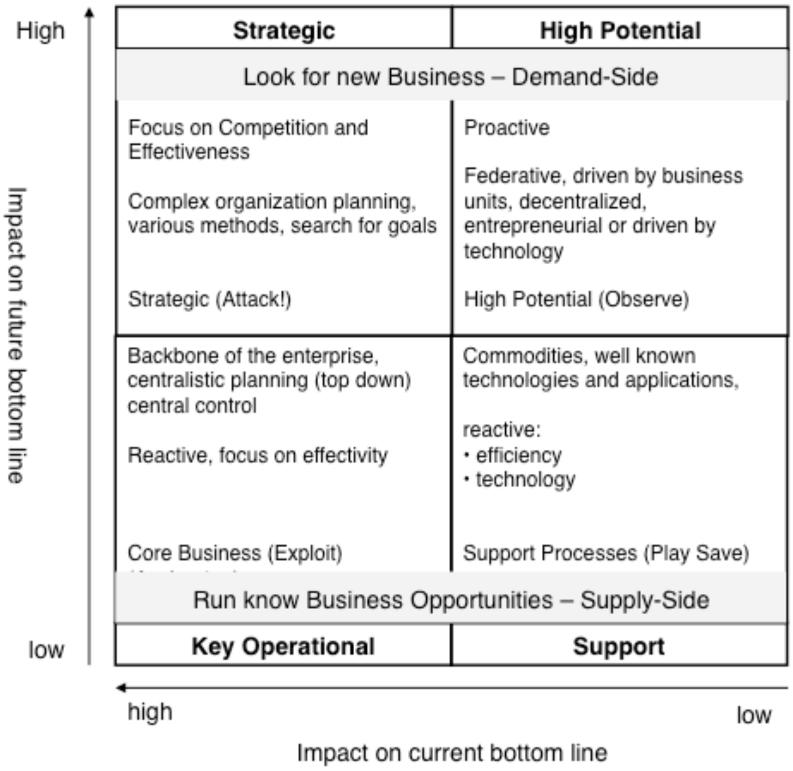
Next let's have a look at portfolio management in finance. A portfolio manager will (in theory) optimize his portfolio so that it yields a maximum return at a given level of risk. Usually: The higher the risk the higher the potential gain. And this is about the end of the analogy for several reasons:

- **Measuring Returns:** For financial assets it is comparatively easy to measure a monetary return. You get dividends, interest or other forms of payments and also the value of your asset may change. For an IT application it is tough to measure the return, because it is tough to determine which part of your company's success can be attributed to the IT application and which to other factors.
- **No Exchange:** Usually there is no stock exchange for IT applications. Investments in a financial portfolio can usually be traded on a stock exchange or some other exchange. There are brokers who deal in used IT infrastructure. But for individual used applications there is nothing similar to a stock exchange or other markets. Individual IT applications, which make up the lion's share of

your IT investment in application systems, tend to have a low (if not zero) fungibility.

- **Covariance:** Have you ever tried to sell the Austrian IT application for the cadastral register to let's say the Chinese government? You will experience problems that stem from use of other applications that your cadastral register application is interfaced with or is based on. For example, maybe, the local authority registers of residents. Often an enterprise application needs a whole ecosystem of other applications it works with. In an investment portfolio you can sell XY stock and replace by YZ stock. In an application portfolio such a switch usually results in expensive and long-term projects.

Then why would people still call managing their applications "Application Portfolio Management" if the analogies are limited? The answer is because a lot of the techniques used to analyze application portfolios use 4-quadrant matrices in the style of the famous BCG matrix.



Strategic	**High Potential**
Look for new Business – Demand-Side	
Focus on Competition and Effectiveness Complex organization planning, various methods, search for goals Strategic (Attack!)	Proactive Federative, driven by business units, decentralized, entrepreneurial or driven by technology High Potential (Observe)
Backbone of the enterprise, centralistic planning (top down) central control Reactive, focus on effectivity Core Business (Exploit)	Commodities, well known technologies and applications, reactive: • efficiency • technology Support Processes (Play Save)
Run know Business Opportunities – Supply-Side	
Key Operational	**Support**

*Figure 9: BCG-style Matrix used to analyze Application Portfolios – Similar matrices can be found in Ward / Peppard [*Ward+02*]*

The categories given by *Figure 9* are just one of many many ways to look at your collection of applications –though it is a particularly common and popular one. The applications you find in each quadrant require a distinct treatment that is common for all the applications in a quadrant.

Dogs (a.k.a. Support Applications): These are those applications that have a comparably low contribution to today's business results and also a low contribution to future business results. E.g. a general ledger system is not a critical success factor for most companies and it will never be one. Hence it is a common strategy to identify such systems and make an analysis whether the company gets a better deal if you outsource such systems completely. Not only development but also operations.

Cows (a.k.a. Key Operational Applications): These are the processes your current bottom line relies upon. In many companies these would be large individual software systems with a heavy investment bound in them and very conservative procedures to change them. Downtime in such systems usually has a direct effect on the bottom line. The company usually could not afford that and hence has to concentrate a lot of attention on keeping these systems up and running. Outsourcing the whole system is seldom an option.

Question Marks (a.k.a. High Potential Applications): Typically you have a research department (or a similar organization) that takes care of you product innovations. If you invent new products, the systems that come along with them need not be extra stable but only need to be demonstrators that allow your management or customers to assess the possibilities in a product or system. Usually you would have many such test systems, as maybe 1 out of 10 innovation really makes it to the market.

Stars (a.k.a. Strategic Application): These are the applications that belong to products that have been selected to be the future cash cows. Such products should be in a heavy growth phase with small absolute numbers sold but high growth rates, often in very competitive and also growing markets. The emphasis here is typically on speed and features and less on an absolutely minimal number of production problems. In many cases you would want to keep the systems that support such products apart from the systems that support the cash cows. But there are also industries like Telecoms where you have to test-drive your innovations with the big iron systems, as there is only one system that allows you to do billing – for all kinds of products.

Using such analyses leads to some interesting other views on a portfolio e.g. which quadrants does your project money go to? The anti-patterns are so instructive that most analysis here is straightforward.

E.g. if you spend 60% of your whole project budget in the “question marks” quadrant on three huge projects, experience should tell you that something is wrong, as spending in the Question Mark quadrant should be relatively moderate. Or if you have a major number of projects in the Dogs quadrant then you should ask yourself, whether it makes sense to nurture the “poor dogs” by additional projects instead of outsourcing them.

4.3 Putting it together: From Chaos to a Master Plan

If you get assigned to the new job of “Chief Enterprise Architect”, you will often find no inventory and no analysis. The question in such a situation is: How do I construct a proper To-Be state of my application landscape and how do I produce the master plan that gets me from my As-Is state to a proper To-Be state. *Figure 10* gives you an overview of the analysis elements that are often applied.

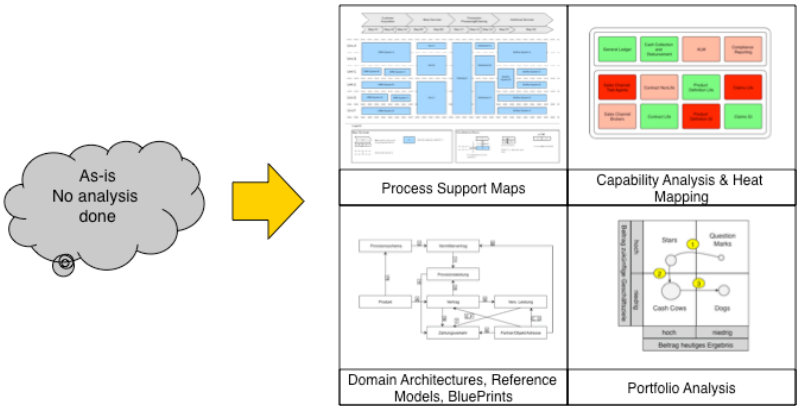


Figure 10: Elements of an Application Portfolio Analysis

- **Application Handbook:** In most cases you start by creating an inventory of all your applications. Often you will also draw **process support maps** that show you how your business processes are supported by applications. This will give you an overview of what you have. But it does not yet really help you to assess what is good and what is bad with respect to your business’ goals.
- **Heat Mapping:** A further step is often a so-called Heat Mapping. This is also available as a service offering called “Motion” from Microsoft for a majority of industries. An expert will bring a list of potential capabilities for your industry and, together with a team of experts, you mirror potential capabilities against your business strategies and you will find out which are the critical capabilities that you need to specially look after in order to implement your

strategy. You can then assess the actual quality of the capabilities as provided by your application portfolio and will get a per capability heat map that is suited to demonstrate the areas for potential action quite intuitively.

- **Reference Models:** Often it is very instructive to map your application landscape against domain specific reference models. Often you will find a bunch of areas for improvement
- **Portfolio Analysis:** And you can also use Portfolio Analysis to cluster your applications into classes that usually deserve some kind of standard treatment like “outsource”, “stabilize”, “renovate” and other treatments.

If you combine the results from those four streams of analysis it is in most cases straightforward to come up with ideas for action, like

- You find that for a critical capability you have 5 different systems in 9 locations that support it. In such a case you might want to consolidate the apps and replace them with the one that best or better supports your critical capability.
- You find that a relatively huge system supports 8 capabilities, 3 of them critical and 5 of them non-critical commodities. You might want to come up with ideas on how to separate the commodity part from the critical part and construct better-suited system support at lower costs.

The advice here is rather generic. The reason for this is that you can describe a generic method like heat mapping or use of process support maps – but it is difficult to foresee the outcome for each possible enterprise and describe a deterministic cookbook that shows how to attack any problem the world of application planning has to offer.

Your “To-Be” architecture will be an array of transformations applied to your “As-Is” architecture. The choice and quantity is determined by your total budget and also priorities. The selection process can be copied from ordinary project portfolio management.

Figure 11 shows representatives for the two additional artifacts that your master plan will contain:

- You will have “To-Be” application maps (e.g. again process support maps) of your target application landscape
- And you will have a master plan showing the “operations” of the application landscape e.g. as project plans or interval maps that show which system is to be replaced by which other system and when.

You could also add the planned state of your heat maps your to-be state or the planned state of your portfolios.

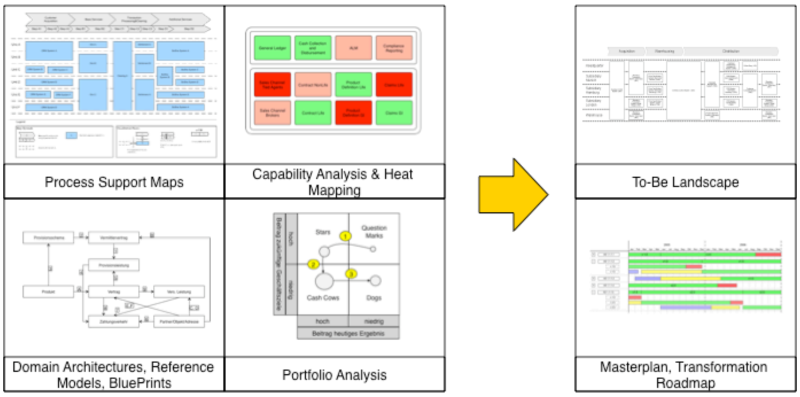


Figure 11: Elements of the Master Plan

4.4 What do you find about Application Portfolio Management in TOGAF?

The answer is that Application Portfolio Management, as a term does not appear at all in TOGAF. As mentioned above, TOGAF is not really intended to help you with strategic IT management and therefore TOGAF is not the prime source to use if you need to learn something about strategic IT management.

4.5 Infrastructure Portfolio Management

Just as you can manage applications you can also manage your infrastructure components, most likely consisting of:

- Your network infrastructure consisting of WANs, LANs and other network components;
- Servers (from PC server clusters to mainframe computers);
- And all kinds of infrastructure services, like operating system services (be they virtual or real) and many other basic services like archiving, or printing

There are at least two differences here to application portfolio management:

- You will most likely manage classes of devices and services instead of single instances (applications). It is not the target of infrastructure portfolio management to replace a CMDB (configuration management database). You may therefore work on a clustered extract of a CMDB and your portfolio will most likely consist of a number of services and heterogeneous implementations thereof. In short, you will find too many implementations for the same service, too many operating systems and variants, too many software products for the same task and your job will most likely be to save money by reducing heterogeneity.

- As most infrastructure components in most enterprises are commodities your main interest in them will be cost reduction while guaranteeing a given level of service. Hence you will in most cases skip a lot of the dimensions that you would deal with in Application Portfolio Management and go primarily for dimensions such as cost, homogeneity and simplicity of a portfolio.

So the most likely form of report or architectural artifact in Infrastructure Portfolio Management is a simple list, which lets you see how many implementations of service X you provide and which allows you to work out how to reduce the complexity of your portfolio.

4.6 What do you find about Infrastructure Portfolio Management in TOGAF?

The straight answer is: If it comes to the management aspect you will find almost nothing. What you can use is TOGAF terminology on infrastructure, as described in the **Technical Reference Model**. This will give you terms and taxonomy of classes of infrastructure and also infrastructure services. For the top level of this see e.g. *Figure 12*. If you say "trivial" then please do not forget the last time consuming discussions you had when trying to agree on such a picture and the terms behind it. So the value of such a de-facto norm should not be underestimated.

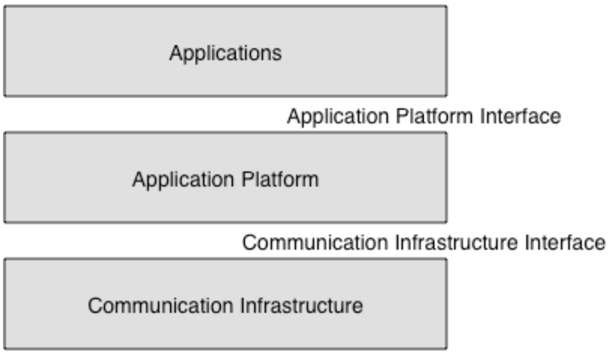

Applications
Application Platform Interface
Application Platform
Communication Infrastructure Interface
Communication Infrastructure

Figure 12: Top Level View of the TOGAF Technical Reference Model. There are deeper levels to this and also taxonomies. Relevant Layers for Infrastructure Portfolio Management are Applications Platforms and also Communications Infrastructure. For the similar figure in TOGAF 9 see figure 43-1. http://www.opengroup.org/architecture/togaf9-doc/arch/Figures/43_trm.png

You can also use the **TOGAF Content Metamodel**[3] in order to model your infrastructure. You will find metamodel snippets for

- Platform Services,
- Logical Technology Components, and also
- Physical Technology Components,

Which will help you with a first cut of a set of entities and attributes you can use for your own infrastructure portfolio management.

4.7 Further Reading

It is somehow tough to find a hands-on guide to Application and Infrastructure Portfolio Management. What you will get is fragments but you will have to integrate them yourself at the moment, starting from the questions you and your management want answers for and then designing the analysis instruments to answer them.

The classic for Application Portfolio Management is the book by Ward and Peppard "**Strategic Planning for Information Systems**" [Ward+02] which has appeared in different editions even with changes in the author's team over the years. The book is not easy reading but a classic and worth owning if you have concrete tasks around Application Portfolio Management.

Kaplan's book on **Strategic IT Portfolio Management** [Kaplan05] has some kind of focus on Project Portfolio Management, but is one of

[3] See e.g. TOGAF 9; Figure 33-3

the few books on the market today that deals with IT Portfolio Management at the title level.

TUM EAM Pattern Catalog: A group at the Technical University of Munich (Germany) collects so-called EAM patterns. Here you find management procedures, viewpoints used for them and also metamodel snippets that support them. You could also start from the questions, look for right patterns and then integrate your personal version of a portfolio management from them. Just have a look at http://wwwmatthes.in.tum.de/wikis/sebis/eampc. Looking at the site yourself is much faster than reading a third party text like this one.

German Books: If you read German you will an array of books that emphasize the management aspect of Enterprise Architecture Management like [Dern09, Keller06]. If you are especially interested in application portfolio management, it is worth having a look at the MS thesis work by Riege [Riege05]. A relatively new book by Inge Hanschke is especially focused on applying Zoning Maps to strategic planning of application landscapes [Hanschke08].

5 TOGAF and Developing Architectures

From time to time even Enterprise Architects need to develop architectures. Those can have such different scopes as

- An architecture for a single system that consumes an effort of maybe only a few person years
- Clusters of systems that support a single business process, spanning maybe 5 – 7 IT systems. Such projects may consume 10 to 50 person years.
- Architecture for the application landscape of a whole company. This will never be implemented in a single project a may take almost a decade to complete.
- Template Architectures and architectures for application frameworks that serve as the template for a family of applications, like e.g. all customer facing web applications of an enterprise. Such BluePrints are typically reused 5 – 10 times.

Such tasks are the core competence of TOGAF – and this is also why this chapter will be short – because we will forward you to TOGAF after a short look at what TOGAF offers you. The TOGAF ADM (Architecture Development Method).

If you know a TOGAF version the likes of 7.0 or 8.x you're done here with this chapter. Have a look at how TOGAF has improved the ADM by factoring out a few things and splitting it into two parts: Part II and Part III.

If you happen to be a TOGAF newbie hang on for 2 pages.

5.1 Part II: TOGAF ADM

TOGAF proposes a cyclic process for architecture development (see *Figure 13*). What you have to do here is quite straightforward for any software architecture professional.

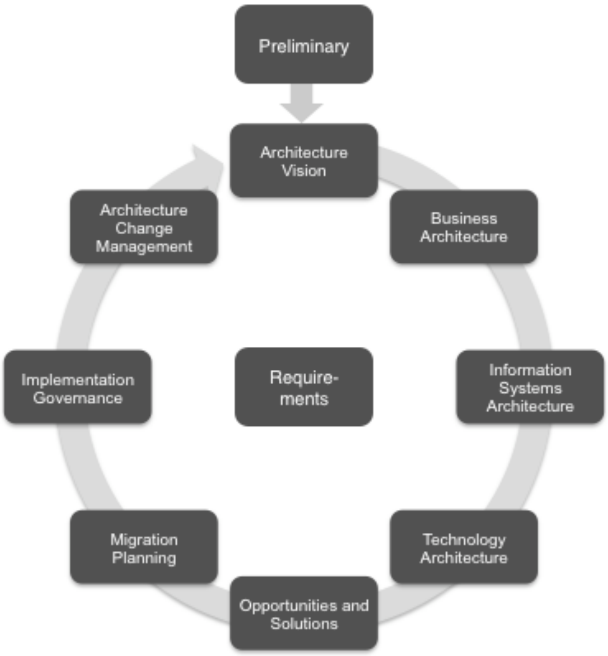

Preliminary
Architecture Vision
Business Architecture
Information Systems Architecture
Technology Architecture
Opportunities and Solutions
Migration Planning
Implementation Governance
Architecture Change Management
Require-ments

Figure 13: Cyclic ADM Process. For the similar figure in TOGAF 9 see figure 5-1.
http://www.opengroup.org/architecture/togaf9-doc/arch/Figures/adm.png

You can see the steps as offerings or as a checklist. If you would e.g. make a minor adjustment to a few systems within a given Business Architecture you what have a look at the check list for Phase B in the cycle and would skip it, if you come to the conclusion that what you would have to do here is already clear or has been done by other colleagues. Each of the Phases at the top level is depicted in *Figure 13*

Each phase is split up in various steps that make a veritable checklist for the phase. TOGAF provides information on documents that should go into a step and should be produced in each step.

5.2 Part III: ADM Guidelines and Techniques

With TOGAF Version 9 a lot of complementing material has been factored out of the core ADM cycle. Therefore Part III contains all the additional material that is there to help you using the ADM such as information on:

- How to use the ADM as a cyclic process (chapter 19)
- Applying the ADM at different levels of granularity (smaller or larger projects) (chapter 20)
- And such aspects as doing security engineering, using TOGAF for a SOA (chapter 22), stakeholder management (chapter 24), architecture patterns (chapter 25), and a few more useful things for architects to know.

Part III is therefore more like a collection of useful stuff than a checklist or a process like Part II – the core ADM

5.3 Further Reading

Even though TOGAF is a very extensive and useful checklist for software architects it is more of a checklist than training material. For example, if you want to apply requirements management (the central item in the middle of the cycle) you need to have knowledge of requirements management and it should be more knowledge than can be found in TOGAF – as the more extensive books on Requirements Managements have their own 500+ pages – compared to the 780 pages TOGAF has altogether.

Or take chapter 25 on architecture patterns. You will get a few references to the likes of the IBM eBusiness Patterns or other pattern sources and will get an idea of what a pattern is and what it is useful for. But you cannot expect a full catalog of all patterns an architect could– or should - know.

Luckily TOGAF has some 3 pages of references and links to literature, so that it does not make sense to cite another page of references for each of the chapters 5 thru 32. Still a few architecture books are quite helpful as the references contain more pointers to other standards and frameworks than to good introductory text.

Which can be translated as: The ADM is not an introductory text to software architecture for beginners but **TOGAF is a condensed checklist for architecture professionals**.

6 TOGAF and Architecture Governance

If you remember *Figure 3*: The Enterprise Architect's Process Map then you will remember that an Enterprise Architecture Group has three big groups of tasks:

The strategic Tasks have been discussed above in chapters 3 and 4. The architecture development tasks have been be discussed in chapter 5. What remains is the question of how to enforce the strategies and architectures that you have developed. This is called Architecture Governance. For the further discussion we will use a part of *Figure 3*

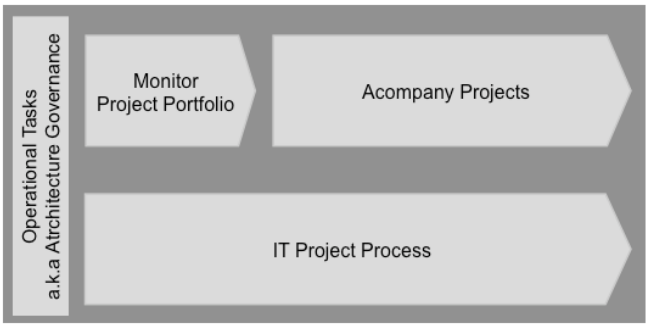


Figure 14: Architecture Governance Overview

So in order to exercise practical Architecture Governance you need to do the following:

- First of all you need an actual overview of the project portfolio of your enterprise. You are not interested in those projects that perform simple maintenance tasks or are the fifth implementation of a standard blueprint. You are interested in those projects that do things that change your overall architecture and that have an impact on your future IT landscape. So your first task is, to find exactly those interesting projects.

- The next important point is to get an idea of whether the team that is assigned to the project that you found interesting, is capable of performing a proper job. If you find that an inexperienced architect was assigned to very difficult job, the project will need far more attention and investment from your side, than it would have needed if your counterpart were a highly professional and well-educated architect.
- After screening the project, and after getting an idea of what you have to expect from the team, you would agree on an audit plan for the project. Depending on the degree of difficulty, you can set up the frequency of meetings necessary to govern the project. In those meetings you will discuss progress with the project's architects and you will also discuss deviations from the company architecture policy.

As you may see, all this has a lot to do with communication and the ability to judge your colleagues' capabilities to come up with the right solutions the first time.

6.1 What do you find about it in TOGAF?

In TOGAF you will find all the technical definitions for IT governance and also for architecture governance. You will not find the tips and tricks you need to deal with your colleagues, the techniques for effective reviews, or what you have to do in cases of deviations.

TOGAF has two spots that deal with architecture governance:

- First, there is a dedicated chapter, chapter 50, on architecture governance. This chapter contains mostly the scope and definitions needed to perform architecture governance beyond what was outlined above. TOGAF also sees enforcing compliance as a task for architecture governance activities.
- Second, phase G of the TOGAF ADM (chapter 15 of TOGAF) covers Implementation Governance. Here you find a more or less detailed recipe on steps needed to perform an architecture audit.

So in addition to a few tricks on how to handle reviews you also find sufficient information.

6.2 Further Reading

Besides the more formal advice on architecture governance, audits, and reviews, there is a lot of useful advice on how to improve practical architecture work in the pattern community. You can, for example, use the so-called writers' workshop in order to improve almost any kind of documentation on architecture work. For a guide on how to conduct the writer's workshop consult work by Jim Coplien [Coplien96, Coplien97].

7 TOGAF and Basic Tasks

TOGAF offers some help, if it comes to other architecture routine tasks. As an architect you will sooner or later want to store the information you acquired in some form of automatic database, instead of in a bunch of Excel sheets and PowerPoint presentations. For this you might need either a software tool or a meta-model. In most cases you would need both. Sections 7.1 and 7.2 will deal with how to design a meta-model for use in enterprise architecture and also what you can expect from TOGAF if it comes to selecting the right tools for Enterprise Architecture Management.

Apart from that we will give you a quick overview in section 7.3 of what you can expect from TOGAF if you want to set up an enterprise architecture function.

7.1 TOGAF and finding the right Meta-Model for your Needs

In Enterprise Architecture Management you sooner or later come to a point where you want to have a database of your IT Portfolio, your IT assets and your application landscape (to name a few items) in order to perform management on the sets of real world items you have modeled in your Enterprise Architecture Database.

You can get ready to use meta-models in sizes between 50 meta-entities up to far more than 500 meta-entities. It should be somewhat straightforward to see that 50 is somewhat limited and more than 500 might be a bit too complex if not covered by a very well integrated user interface of an EAM Tool.

In many cases you will not try to answer all possible questions in EAM at a time. It is more likely that you management has focused its interests on certain points like e.g. cost management for your infrastructure, just to name an arbitrary example.

Start small: In such cases it is important to start with a small solution that is driven by the question you have to answer without having to deal with the full complexity of a 500+ item meta-model.

You can still end up big: TOGAF architecture capability framework offers you a meta-model, which is split or modularized into small areas of interest. If you are interested in a certain area, you need to read the respective sections and can draw from the lists of predefined TOGAF meta-objects. The top-level picture of the TOGAF Architecture Content Framework gives you an overview of the areas for which you can expect support.

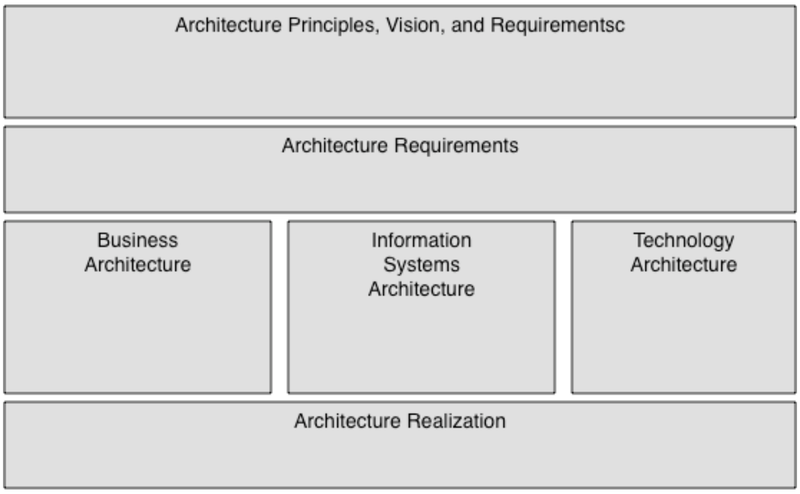

Architecture Principles, Vision, and Requirementsc
Architecture Requirements
Business Architecture
Information Systems Architecture
Technology Architecture
Architecture Realization

Figure 15: Top Level of the TOGAF Content Metamodel. For more details see TOGAF 9, figure 33-3.

http://www.opengroup.org/architecture/togaf9-doc/arch/Figures/34_contentfwk5.png

This is about where we will end with this book and this topic. It is not our aim to duplicate TOGAF texts in other words. Here we can only recommend referring to TOGAF chapters 33 thru 37 in order to get an idea what the Architecture Content Framework can offer you in case you need a Meta model as a basis to start storing information about your architecture.

7.2 TOGAF and finding Tool Support

Most organizations sooner or later arrive at a point where they want to put all relevant information from their enterprise architecture efforts into some form of repository and where they want to edit the viewpoints of their architectures in a single architecture tool. Let's have a look what TOGAF has to offer you, if you want to select a tool. There are two chapters, which are focused on the repository and tool matter:

- **Chapter 41 (Architecture Repository):** Here you get 7 pages (pages 559 thru 565) of information of what TOGAF thinks should be in an Architecture Repository. There are references to the TOGAF ADM (Architecture Development Method) which – as we have already demonstrated – does not really cover strategic IT planning. As a consequence, the recommended content listed in TOGAF Chapter 41 also does not cover much of strategic IT planning.
- **Chapter 42 (Tools for Architecture Development):** gives you a 4-page list of evaluation criteria for architecture tools. If you have ever done a similar tool evaluation you know that criteria catalogs for tool evaluations are way longer in most practical cases. And again – the list does not cover strategic IT planning.

Therefore, if you plan to acquire an enterprise architecture management tool have a look at the evaluation done by Technical University of Munich in 2008 (you find pointer to the material at http://wwwmatthes.in.tum.de/wikis/sebis/eamts2008). The study has a broader approach from the functional side. It also contains aspects of strategic IT planning. And the list of criteria used is by far more extensive than the one provided by TOGAF chapter 42.

7.3 TOGAF and Immersion Paths for Enterprise Architectures

Another thing one would like to find in an Architecture Framework is help with setting up an Enterprise Architecture Management Operation (or call it the EAM department, or processes if you want to be more modern).

TOGAF offers you a so-called “Architecture Capability Framework”. This Framework lists a set of instances you need for a successful EAM operation. And again: Strategic IT planning is not in the center of it – so you will not read anything about IT Strategy or IT portfolio management – but you will get advice on how to use the TOGAF ADM to develop e.g. the systems support for the architecture practice. What you will find is advice on:

- **Establishing an Architecture Capability (Chapter 46):** This tells you how to apply the ADM to an Information Systems and Technology Architecture – not the one for your company but for the architecture practice. It is more or less a list of action items. If you look at this from a management perspective then tool support comes very late in establishing a successful architecture practice. Hence the order of things in TOGAF chapters 45 thru 52 is debatable.

- **Architecture Board (Chapter 47):** Describes what an Architecture Board does and how to set it up (4 pages all in all).
- **Architecture Compliance (Chapter 48):** Contains a checklist for architecture reviews – how to plan them and what to check.
- **Architecture Contracts (Chapter 49):** Gives a few hints on how to make agreements on architecture between the enterprise that sources some piece of software and the contractors.
- **Architecture Governance (Chapter 50):** Gives hints on how to enforce the enacted architecture guidelines. Has been discussed in more extent above in Chapter 6 of this book.
- **Architecture Maturity Models (Chapter 51):** Applies the idea of CMMI and other Capability Maturity Models to TOGAF. Rather brief and already referring to future editions.
- **Architecture Skills Framework (Chapter 52):** Defines a set of roles and skills needed in the opinion of the creators of TOGAF to fill out the roles.

Browsing through this material you might conclude that TOGAF has a somewhat technology- and model-centric view of enterprise architecture. The chapters start with establishing the right tool support for the Enterprise Architecture Practice; you get some information on Governance and Architecture Governance Bodies – but you do not get a business driven immersion path that pushes you towards maximizing business benefit from day zero.

8 What else will you find in TOGAF?

This chapter will explain the parts of TOGAF that have not yet been mentioned. You will be given a short summary for each item.

8.1 Foundation Architecture / Technical Reference Model (TRM)

In this case it is easiest to have TOGAF speak for itself:

> The TOGAF Foundation Architecture is an architecture of generic services and functions that provides a foundation on which more specific architectures and architectural components can be built. This Foundation Architecture is embodied within the Technical Reference Model (TRM), which provides a model and taxonomy of generic platform services.
>
> The TRM is universally applicable and, therefore, can be used to build any system architecture.
>
> Source [TOGAF9, page 576]

Before we waste time explaining what the subject matter of the TRM is about, we better use one picture (*Figure 16*), which explains the subject area much faster than a verbal definition. TRM is about the services any technology stack needs to offer.

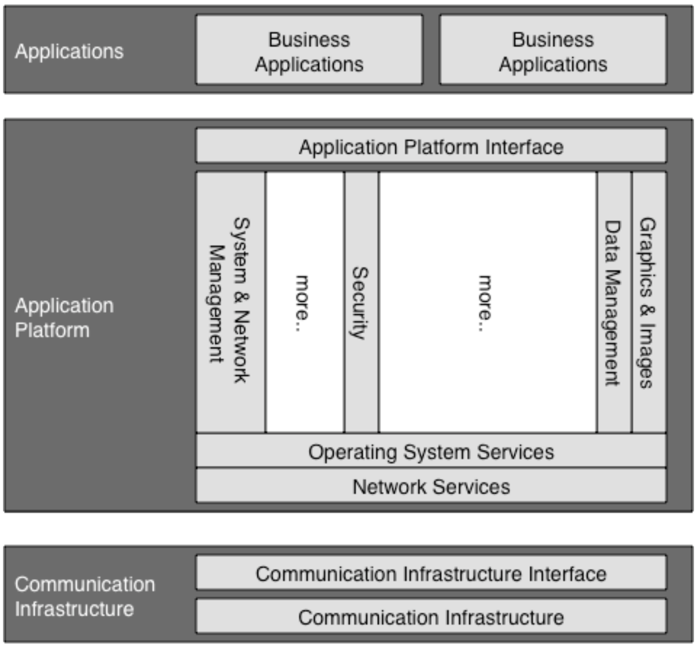

Applications
Business Applications
Business Applications
Application Platform
Application Platform Interface
System & Network Management
more..
Security
more..
Data Management
Graphics & Images
Operating System Services
Network Services
Communication Infrastructure
Communication Infrastructure Interface
Communication Infrastructure

Figure 16: Simplified view upon the TOGAF TRM Overview. For the analogous figure in TOGAF 9 see Figure 43-2.
http://www.opengroup.org/architecture/togaf9-doc/arch/Figures/43_trm_detail.png

As in the III-RM (see next section) the main use of such a reference model is terminology and checklists. You don't argue about the terms and you can check for yourself whether you need the services enumerated in the reference model and whether you have all services you need present in your technology stack.

8.2 The Integrated Information Infrastructure Reference Model (III-RM)

The III-RM gives you a 20 page taxonomy for all kinds of application software in an enterprise that has the goal of so called boundary-less information flow, meaning an enterprise which is integrated with the outside world's logistics chains via buying and selling circles. The benefit of the chapter is that you get a vocabulary for talking about enterprises integrated with the outside world via electronic data exchange.

We will not reprint any Open Group drawings here. Mainly for copyright reasons. Best have a short look at TOGAF Chapter 44

9 Wrap Up: TOGAF for You

9.1 A Collection of Useful Stuff

Throughout this book we have demonstrated that TOGAF is a very useful collection of many methods and tools that an enterprise architect might need for his work.

Nevertheless we could also see, that TOGAF is more a collection than a uniform planned piece of work. The Architecture Development Method is somewhat “closed in itself” – so is the Content Framework. But many other things are collections of tools and additions to the ADM: Which is NOT negative! But as we assumed in the introduction and could now kind of prove: TOGAF is not an “Enterprise Architecture for Dummies” Cookbook. TOGAF has a clear focus on developing architectures: Be it single systems or subsystems, be it clusters of applications systems or be it a blueprint for the top level architecture of an enterprise. The last of these will hardly ever be designed on a drawing board but will be a result of organic growth plus either an explicit or implicit standard for an industry.

9.2 The Two Strongest Points

The two areas in TOGAF that make the kernel of the framework are:

- **ADM:** The Architecture Development Method is a really mature and extensive method, checklist and guide that you can employ when you want to design any part of your IT architecture.
- **Architecture Content Framework:** This Framework lets you bring order to your architectural artifacts even if it does not tell you how to create all of them. The meta-model part gives you an extensive library of meta-objects that you can at least consider using when you have to construct the specific meta-model that is able to answer the questions your management has for their Enterprise Architects in the context of their enterprise.

Besides that we already listed a lot of other useful stuff that you can find in TOGAF.

9.3 TOGAF Certification

If you study job adverts, there's a recent tendency to mention TOGAF certification whenever an Enterprise Architect is wanted. In this book we have demonstrated that TOGAF is not really strong at strategic Enterprise Architecture Management. TOGAF is strong at project architectures of all scales and some of the tasks of strategic Enterprise Architecture Management.

As with any certification, TOGAF certification is for sure beneficial to those who offer the courses and exams. If it comes to HR departments and senior management, the ones who often hire enterprise architects, it is doubtful whether they have deep knowledge of the differences between Strategic IT Management and the kind of Enterprise Architecture that the TOGAF ADM is good at. So if you want to hire somebody who is able to perform Enterprise Architecture Management a TOGAF certificate covers maybe 30% percent of the job – and to the authors regret – the less important 30% because the important part is about business strategies and business management.

So for many Enterprise Architects the TOGAF certificate will be just another drivers license. The license says that you are entitled to drive – it does not say that you are a good driver. Nevertheless you have to take the test if the public (represented by the government) makes it mandatory to have the license. Or if the majority of companies require a TOGAF certificate people will need one as an entry ticket for certain jobs.

9.4 Future

Predictions are hard to make and making predictions by looking at the past often does not make too much sense. The first version of the TOGAF ADM was derived from TAFIM 2.0 around 1995. Today's ADM has a long history and no roots in strategic architecture management. In 2009 the Architecture Content Framework was added from the Capgemini IAF (Integrated Architecture Framework).

This was made possible because Capgemini invested a lot in supporting the OpenGroup and TOGAF.

Whether TOGAF will become a full framework for the more strategic parts of Enterprise Architecture Management is hard to predict – if not unpredictable. In any case it is very useful for developing architectures in concrete projects no matter whether they have a small or a very large scope.

10 References

[Alexander79a] *Christopher Alexander:* The Timeless Way of Building. Oxford University Press, New York, 1979.

[Alexander79b] *Christopher Alexander:* A Pattern Language. Oxford University Press, New York, 1979.

[Bass+98] Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice. Addison-Wesley, 1998.

[Broadbent+05] Marianne Broadbent, Ellen S. Kitzis: The New CIO Leader. Harvard Business School Press, 2005.

[Brooks75] Frederick P. Brooks, Jr.: The Mythical Man-Month. Essays on Software Engineering. Rea-ding, Mass. et al.: Addison-Wesley 1975.

[Buschmann+96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern Oriented Software Architecture, A System of Patterns, Wiley 1996.

[Clausewitz98] Carl von Claussewitz: Vom Kriege. Ullstein Taschenbuch 1998.

[Coplien96] Jim Coplien: Software Patterns. SIGS Publishing, New York et al. 1996.

[Coplien97] Jim Coplien: A Pattern Language for Writers' Workshops; available at http://www.riehle.org/community-service/hillside-group/europlop-1997/p2final.pdf (link checked 2009/03/25)

[Dern09] Gernot Dern: Management von IT-Architekturen. 3rd Edition, Vieweg Verlag, Edition CIO, 2009.

[Gamma+95] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns, Elements of Reusable Object-oriented Software, Addison Wesley 1995.

[GarlanShaw96] David Garlan, Mary Shaw: Software Architecture: Perspectives of an Emerging Discipline; Prentice Hall 1996.

[Hanschke08] Inge Hanschke:: Strategisches Management der IT-Landschaft. Ein praktischer Leitfaden für das Enterprise Architecture Management; Hanser Verlag 2008.

[ITIL02] OGC; Office of Governmernt Commerce: Best Practice for ICT Infrastructure Management. Published by TSO – The Stationery Office, 2002.

[Kaplan05] Jeffrey Kaplan: Strategic IT Portfolio Managment; Governing Enterprise Transformation; Pittiglio Rabin Todd & McGrath (PRTM) Publishing 2005.

[Keller06] Wolfgang Keller: IT-Unternehmensarchitektur, dpunkt Verlag 2006.

[Longepe03] Christophe Longepe: The Enterprise Architecture It Project: The Urbanisation Paradigm; Kogan Page Science; 2003

[Riege05] Christian Riege: Methoden für das Management von Anwendungsportfolios – eine vergleichende Untersuchung. Diplomarbeit, Universität Leipzig, 2005.

[Ross+06] Jeanne Ross, Peter Weil, David C. Robertson: Enterprise Architecture as Strategy: Creating a Foundation for Business Execution. Mcgraw-Hill Professional 2006

[Schekkerman04] Jaap Schekkerman: How to Survive in the Jungle of Enterprise Architecture Frameworks. Verlag Trafford, Victoria, Canada, 2004.

[sebis08] sebis: Enterprise Architecture Management Tool Survey 2008. Available from Technical University Munich, Chair of Florian Matthes; http://wwwmatthes.in.tum.de/wikis/sebis/eampc (link checked 2009/03/25)

[Shaw+96] Mary Shaw, David Garlan: Software Architecture – Perspectives on an Emerging Discipline. Prentice Hall, 1996.

[Sowa+92] J.F. Sowa, J. A. Zachman: Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal, Volume 31, No. 3, 1992. IBM Publication G321-5488.

[Starke05] Gernot Starke: Effektive Softwarearchitekturen. 3. Erweiterte Auflage, Hanser Verlag, 2008.

[TOGAF8.1.1] The Open Group: TOGAF Enterprise Edition. Version 8.1.1: available at http://www.opengroup.org/architecture/togaf8-doc/arch/. The Open Group, 2007.

[TOGAF9] The Open Group: TOGAF Version 9; available at http://www.opengroup.org/architecture/togaf9-doc/arch/ ; The Open Group, 2009.

[Ward+97] John Ward, Pat Griffiths: Strategic Planning for Information Systems. 2. Auflage, Wiley, 1997.

[Ward+02] John Ward, Joe Peppard: Strategic Planning for Information Systems. 3. Auflage, Wiley, 2002.

[Weill03] Peter Weill: Don't Just Lead, Govern! Vortrag DiamondExchange, Juli 2003, http://exchange.diamondcluster.com/archives/200307/weill_0703.pdf (aufgerufen 2.1.2006).

[Weill+04] Peter Weill, Jeanne W. Ross: IT Governance – How Top Performers Manage IT Decision Rights for Superior Results. Harvard Business School Press, 2004.

[WinFis06] Robert Winter, Ronny Fischer: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In. Los Alamitos, CA, USA : IEEE Computer Society, 2006.- EDOC Workshop on Trends in Enterprise Architecture Research (TEAR 2006) within The Tenth IEEE International EDOC Conference (EDOC 2006).- Hong Kong.- ISBN 0-7695-2743-4, page 30.

[Zachman87] John A. Zachman: A Framework for Information Systems Architecture. IBM Systems Journal, Volume 26, No. 3, 1987. IBM Publication G321-5298. Can be obtained via http://www.research.ibm.com/journal/sj/263/ibmsj2603E.pdf (Link checked am 2005-01-27).

[Zachman97] John A. Zachman: Enterprise Architecture: The Issue of the Century. Database Programming and Design, March 1997.