The ADO\textit{it} Method Manual

A Practical Guide to Enterprise Architecture with ADO\textit{it}

By BOC Group

FIRST EDITION
The ADOit Method Manual
A Practical Guide to Enterprise Architecture with ADOit
© 2011 BOC Information Technologies Consulting AG

Wipplingerstraße 1
A-1010 Vienna
Tel.: +43-1-513 27 36-1170
Fax: +43-1-513 27 36-1198

Disclaimer
© Copyright BOC AG, Vienna
The BOC Management Office as well as ADOscore, ADONIS, ADOlog and ADOit are registered trademarks of the BOC Group. All of the content is protected. All other named brands are property of the respective companies. All changes can only be made with a written letter of agreement from the BOC Group. Reproductions in any form are only allowed with the Copyright remark. Publications and translations need a written letter of agreement from the BOC Group.

Source of Images
The following images have been altered or combined with other images by the authors of this manual:
a) Figure 8 and 32
Original images available at
http://www.opengroup.org/architecture/togaf9-doc/arch/
(The Open Group:TOGAF Version 9; The Open Group, 2009) [9 March 2011]
b) Figure 33 and 34
Original images available at http://cio-nii.defense.gov/sites/dodaf20/
(The Department of Defense Architecture Framework Version 2.02) [9 March 2011]
c) Figure 35
Original image available at http://www.pragmaticea.com/
(Pragmatic Enterprise Architecture Framework) [9 March 2011]
d) The FIREFOX logo used in figure 115 is a registered trademark of the Mozilla Foundation.

All other images (including the cover image) used in this manual are property of the BOC Group.

Printing History: March 2011 - First Edition
Preface

The BOC Group is a technological leader in IT-based management tools and offers consulting services in the fields of Strategy and Performance Management, Business Process Management, and IT Management. The BOC Information Technologies Consulting GmbH was established in 1995 in Vienna as a spin-off from the Business Process Management Systems (BPMS) Group from the department of Knowledge Engineering at the University of Vienna. The BOC Group operates internationally with offices in Germany, Austria, Ireland, Poland, Spain, Greece, and Switzerland.

ADOi it is a BOC Management Office tool for IT Architecture and Service Management. It is based on a dynamic meta-modelling concept. Utilising this concept customer may extend the out-of-the-box scenarios of ADOit or build new ones, without any programming efforts. Furthermore the concept allows for the integration of any framework, such as TOGAF, DoDAF or COBIT.

For further information please visit our website: http://www.boc-group.com
# Table of Contents

1 **Introduction**

1.1 About This Manual
   1.1.1 Content
   1.1.2 How to Read This Manual

1.2 Introduction to EAM
   1.2.1 The Overall Target
   1.2.2 EAM and ITSM – The Two Pillars of Successful (IT) Governance
   1.2.3 Interfacing and Influencing Management Endeavours

1.3 ADOit Product Philosophy

2 **BOC Approach to EAM**

3 **Frameworks Compatible with ADOit**

3.1 TOGAF 9
   3.1.1 EA Processes – Architecture Development Method (ADM)
   3.1.2 Architecture Repository
   3.1.3 Business Architecture – Recommended Viewpoints in TOGAF 9
   3.1.4 Application Architecture – Recommended Viewpoints in TOGAF 9
   3.1.5 Data Architecture – Recommended Viewpoints in TOGAF 9
   3.1.6 Technology Architecture – Recommended Viewpoints in TOGAF 9
   3.1.7 TOGAF Meta-Model in ADOit
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Demand and Project Portfolio Management</td>
<td>105</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Objectives</td>
<td>105</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Complementary Scenarios</td>
<td>106</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Stakeholders and Their Role</td>
<td>107</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Meta-Model</td>
<td>107</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Procedure</td>
<td>108</td>
</tr>
<tr>
<td>4.6</td>
<td>Master Planning</td>
<td>114</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Objectives</td>
<td>114</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Complementary Scenarios</td>
<td>114</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Stakeholders and Their Role</td>
<td>115</td>
</tr>
<tr>
<td>4.6.4</td>
<td>Meta-Model</td>
<td>116</td>
</tr>
<tr>
<td>4.6.5</td>
<td>Procedure</td>
<td>117</td>
</tr>
<tr>
<td>4.7</td>
<td>Integrating EAM and BPM</td>
<td>127</td>
</tr>
<tr>
<td>4.7.1</td>
<td>Objectives</td>
<td>127</td>
</tr>
<tr>
<td>4.7.2</td>
<td>Complementary Scenarios</td>
<td>128</td>
</tr>
<tr>
<td>4.7.3</td>
<td>Stakeholders and Their Role</td>
<td>128</td>
</tr>
<tr>
<td>4.7.4</td>
<td>Meta-Model</td>
<td>129</td>
</tr>
<tr>
<td>4.7.5</td>
<td>Procedure</td>
<td>130</td>
</tr>
<tr>
<td>5</td>
<td>ADO\textit{it} EAM Meta-Model</td>
<td>140</td>
</tr>
<tr>
<td>5.1</td>
<td>Business Architecture and Organisation Concepts</td>
<td>143</td>
</tr>
<tr>
<td>5.2</td>
<td>Information Systems Architecture Concepts</td>
<td>144</td>
</tr>
<tr>
<td>5.3</td>
<td>User</td>
<td>144</td>
</tr>
<tr>
<td>5.4</td>
<td>Technology Architecture Concepts</td>
<td>145</td>
</tr>
<tr>
<td>5.5</td>
<td>Project Portfolio Concepts</td>
<td>145</td>
</tr>
<tr>
<td>6</td>
<td>ADO\textit{it} Architecture</td>
<td>146</td>
</tr>
<tr>
<td>6.1</td>
<td>Overall Architecture</td>
<td>147</td>
</tr>
<tr>
<td>6.2</td>
<td>Interfaces</td>
<td>148</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Common Interfaces</td>
<td>148</td>
</tr>
<tr>
<td>6.2.2</td>
<td>The BOC Management Office - Integration Concepts</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td><strong>List of Abbreviations</strong></td>
<td>151</td>
</tr>
</tbody>
</table>
## Introduction

1.1 **About This Manual**
- 1.1.1 Content 11
- 1.1.2 How to Read This Manual 12

1.2 **Introduction to EAM**
- 1.2.1 The Overall Target 13
- 1.2.2 EAM and ITSM – The Two Pillars of Successful (IT) Governance 14
- 1.2.3 Interfacing and Influencing Management Endeavours 15

1.3 **ADOit Product Philosophy** 17
1.1 About This Manual

This manual deals with how ADOit assists you in your endeavour to implement Enterprise Architecture Management (EAM). It provides you with guidance on how to successfully set up, develop, manage and maintain the Enterprise Architecture (EA) of your organisation in an easy and sustainable way. For this purpose, this manual offers a wide range of best practices in the field of Enterprise Architecture Management, which have proven their benefits to our customers. It provides you with an end-to-end process enabling you to start off with the EA programme that best suits your needs. Required EA processes, roles and responsibilities, suitable meta-models and reports are comprehensively described in this manual.

As for the audience of this manual, it is intended for both beginners and experienced users of ADOit. However, we recommend participating in ADOit training prior to dwelling on this manual.

1.1.1 Content

This manual supports a collective and quality-assured use of ADOit that can be adapted to suit the needs of your organisation. It is considered as a working document for ADOit users. Starting with chapter two, this manual deals with the positioning of ADOit including the tool’s philosophy. Chapter three deals with a thorough presentation of ADOit-compatible frameworks, such as TOGAF, PEAF, DoDAF, among others. Subsequently, chapter four provides you with in-depth information on ADOit best practice scenarios including guidelines and techniques. A detailed description of the ADOit EAM Meta-Model including concepts regarding business architecture, information systems architecture, technology architecture, and project portfolio is presented in chapter five. Finally, you will be provided with insight into the architecture of ADOit including the tool’s interface and integration concepts in chapter six.
1.1.2 How to Read This Manual

If you are new to Enterprise Architecture Management, it is recommended that you read this manual from front to back at least once. However, it is left to you to choose whether to read it in chronological or alternate order. It is essential for any new user to read the introductory chapter in order to understand the conventions used in this manual.

1.2 Introduction to EAM

In times of turbulent markets almost every organisation is confronted with fast-changing business requirements. More importantly, increasing cost pressure is a critical factor that affects the alignment of an organisation’s information technology with its business needs. The timely and efficient implementation of various business requirements is consistently facing the increasing complexity and heterogeneity of the IT landscape.

Established IT landscapes with hundreds or thousands of applications, interfaces and functions develop into large networks where all applications are closely linked together. All too frequently, IT architects are confronted with applications which have been in place for more than 20 years. Moreover, the increasing number of inadaptable and technologically out-of-date legacy applications makes it extremely difficult to integrate new or modified systems.
1.2.1 The Overall Target

Under these circumstances, it is quite a challenge to shape the IT strategy for two simple reasons. Firstly, it is necessary to develop the IT landscape in such a way as to meet the requirements of all departments of an organisation in the shortest time possible. And secondly, it is crucial to keep a close eye on the overall structure to prevent weakening of the architecture, while at the same time technologies and standards need to be kept up to date through periodic technology refresh.

This being the case, it seems to be a rigorous challenge to maintain the balance between an organisation’s business and IT requirements, and at the same time gain control of the skyrocketing IT costs.

Enterprise Architecture Management provides the techniques for effective planning and further development of not only your IT but your entire organisation. Focusing on strategic goals and requirements, EAM provides you with ways to propel your organisation forward.

EAM maximises cost savings while striving to minimise all associated risks. You will be able to assess your organisational and technical architecture artefacts on different levels. From an IT point of view, you can gather information on applications including their software architecture as well as their technical platforms. On the other hand, you will also be able to assess your organisation
from a business point of view, which means that you gain understanding regarding your products and services, business processes and organisational units. As a result, you will have an integrated view of your Enterprise Architecture which can be used to have valuable discussions with the business side, allowing you to align business and IT of your organisation. ADOit facilitates target-oriented development of the Enterprise Architecture. The strategic, tactical and operational planning and maintenance of the organisation’s IT is guaranteed, aiming at achieving business goals efficiently and sustainably.

Thus, ADOit not only provides a blueprint of your organisation’s current architecture, but also a permanent inventory of the Enterprise Architecture. This inventory builds the basis for analysing and planning the architecture. It enables setting up the projects required to develop the organisation based on business and IT strategy. Plenty of measures can be implemented to uncover and release idle potentials of your organisation. In order to implement EAM, all typical scenarios are consistently supported by ADOit. Examples are Technology and Application Portfolio Management and (IT) Demand Management.

1.2.2 EAM and ITSM – The Two Pillars of Successful (IT) Governance

In the EAM context, the term architecture refers to the description of all relevant parts or components of an organisation including their relations and interdependencies. Enterprise Architecture represents a structured and harmonised collection of construction plans. It helps you

- Plan and evaluate the business and IT architecture of your organisation by providing a variety of integrated views on different abstraction levels considering current, transition, and target state of your architecture.

- Migrate from the current state architecture to the future state architecture by providing a detailed migration plan.

Moreover, EA enables the IT managers of your organisation to manage the overall IT investment in a way that meets your business needs compliant with defined architecture principles.

Following the above definition of the EA, EAM is defined as principles and methods for designing and implementing the EA. IT Service Management (ITSM) generally deals with the operational concerns of Information Technology Management and focuses on running the IT. Together both management disciplines form the main pillars of IT Governance.
1.2.3 Interfacing and Influencing Management Endeavours

EA is not an ivory-tower practice. It should not be implemented in isolation from other initiatives as it is most effective when integrated into the various management domains and processes of your organisation.

For example, EAM should be closely aligned with Business Process Management (see chapter 4.7 ‘Integrating EAM and BPM’ for details). This allows your enterprise architects to have a closer coupling between IT and business processes, leading to effective business and IT alignment.

Furthermore, a reusable business architecture can be established, which represents an ideal basis for any form of architectural work.
**Strategy Management** defines the strategic objectives that drive the organisation forward. Business strategies are used to derive IT strategies which guide the structure of the EA, for example through architecture principles. The strategic objectives are typically decomposed into various tactical objectives to achieve traceability. Given these objectives, a strategic plan providing the milestones to achieve the objectives is developed. In case of a tight coupling between both management domains, EAM will provide blueprints of the current architecture supporting situational analysis to derive appropriate objectives and measures. The strategic plan and EAM’s target architecture will be seamlessly aligned.

**Risk and Compliance Management** is another example of an interfacing management discipline. An accurate and up-to-date EA repository helps to manage regulatory and compliance requirements set forth by internal or external governing bodies. The EA inventory provides the organisation with the means to clearly define and document all risk/compliance-relevant Enterprise Architecture artefacts enabling analysis of the EA on all levels in order to identify and mitigate risks.

**Software Development** should be driven by architecture principles and technology standards defined in Enterprise Architecture Management. The EA repository represents a sound basis for project initiation and scoping. Projects that do not comply with architecture principles and standards should only be implemented if the EA team validates the projects in spite of the deviations from the defined architecture principles and standards. Typically, measures for re-alignment of these architectural solutions need to be defined and governed by the EA team in order to prevent weakening of the EA and misalignment of architectural solutions with the organisation’s strategy.

**Service Management**: EAM focuses mainly on the design of architectures and the planning of the migration of these architectures, whereas Service Management focuses mostly on operational issues. Configuration and Change Management processes are ideally harmonised with Enterprise Architecture practices. Advantages of tight integration include joint decision-making by bringing together the change advisory board with the EAM architecture board, or enhancing co-operation between technical architects and operational staff. Using the EA repository as a knowledge base for Service Management processes is another tremendous advantage. The integration of the EA repository and the Service Management’s Configuration Management System will allow for cross-domain dependency analyses.

**Project Portfolio Management** is a management discipline implemented to help an organisation acquire and view information about all of its planned and ongoing projects. Projects are sorted and prioritised according to certain measures or indicators such as strategic value, required resources and costs. Enterprise
Architecture Management usually defines blueprints of the current and target architectures. In order to migrate from the current to the target architectures EAM proposes to run a set of projects. Furthermore, projects might emerge from concrete demands requested by the various business departments. These demands are typically affecting parts of the EA. Project Portfolio Management is used to analyse costs, time to market, business impacts, dependencies and priorities regarding the proposed and ongoing projects. It helps to define and select those projects that need to be implemented. Based on decisions regarding projects, EAM refines the architecture blueprints and road maps.

An important link between Business Process Management (BPM) and EAM becomes immediately apparent: The process architecture usually depicted in a BPM tool like ADONIS is a sound basis for describing the business architecture of the EA. Sharing a common process architecture in both management domains will drive business/IT alignment.

Other interfacing disciplines like Business Continuity Planning and Disaster Recovery Planning profit from the EA repository or even contribute to it by keeping parts of the EA data up-to-date and accurate.

1.3 ADOit Product Philosophy

EAM is often regarded as a large, all-or-nothing approach seeking to cover the needs of an entire organisation on all architecture levels at once. This usually requires a large staff, a large budget and a long time frame.

ADOit does not strive to implement EAM in a ‘big bang’ approach but rather in a step-by-step approach. Therefore, your organisation can grow into EAM as it is needed. You can decide how detailed you would like to work on your architecture. ADOit users typically apply a step-by-step approach, e.g. starting with gathering information about their applications, and continuing with collecting information on technical capabilities and their relations to business processes and organisational units.
Based on ADOit’s project-driven approach, you can break up the EA programme into different, easy-to-manage projects. As soon as the procedures, which you have implemented and evaluated within the EA projects, have been put into daily operation, you can go on with the next step. Expand the initiative to other business domains or departments of your organisation. Further EA scenarios, such as Application Portfolio Management, Business Demand Management and Project Portfolio Management can profit from the available EA data. These scenarios will be aligned with architecture principles, which are fundamental rules and guidelines applying to a standardised and target-oriented development of the Enterprise Architecture.

**Master Planning** closes the gap between strategic and operational Enterprise Architecture Management. You will be able to design blueprints of your entire organisation on a regular basis, but most importantly, EAM guidelines will be applied to all projects. This means that the developed and agreed target architecture will be the benchmark for the implementation projects in conformity with your business and IT strategies.

Through ADOit’s meta-modelling capabilities almost all EA frameworks, such as TOGAF, Archimate and DoDAF to name a number of the popular ones, can be supported. However, an EA framework is not a silver bullet, but it can assist with your EA programme providing that it is implemented correctly. By utilising the meta-modelling capabilities of ADOit, your favoured framework can be tailored and enhanced to fit your organisation’s needs in the best way possible. Take a pragmatic and benefit-based approach!
Introduction
BOC Approach to EAM
BOC aims to support EAM not only through consulting expertise but also by providing the optimal tool support for your given requirements. To achieve this goal, it is first important to understand what issues need to be addressed during the implementation of EAM in an organisation. Figure 7 shows the core issues that have been identified in past ADOit and EAM implementation projects.

First, you have to be clear about what you want to achieve with EAM in your organisation. EAM is certainly not an end in itself. The implementation of EAM and corresponding tools should always be geared to predefined goals and derived requirements. Usually, the driver of an EAM implementation is a programme or several IT projects that require a well documented IT architecture in order to be able to plan the architecture change that comes as the resulting output of the projects (e.g. the introduction of an ERP system, other core applications, or a Service-Oriented Architecture (SOA) transformation).
Apart from this, typical goals or expected benefits of EAM may be the following:

- **Document the Enterprise Architecture to provide transparency in order to meet compliance regulations:**
  - Quality Management (e.g. ISO9000ff),
  - Service Management (e.g. ISO20000),
  - Risk Management (e.g. ISO27005),
  - Sarbanes Oxley Act,
  - 8th EU Directive,
  - Basel II, III,
  - Solvency II,
  - Other.

- **Provide a basis for all other EAM activities.**

- **Support measures that aim at cost consolidation:**
  - Optimising the application portfolio,
  - Fostering standardisation and homogenisation of the architecture,
  - Planning outsourcing activities,
  - Analysing weak spots and cost intensive areas of the architecture,
  - Other.

- **General increase of Enterprise Architecture quality to maintain momentum:**
  - Managing technologies and technology road maps,
  - Planning future states of the Enterprise Architecture,
  - Constant alignment of IT Strategy and Enterprise Architecture,
  - Tracking the quality by measuring the degree of implementation of architecture principles,
  - Other.

The above list is far from complete, but it gives an impression of the various approaches that EAM might follow. Once you have identified the most important goals and expected results, and have shaped requirements regarding an EAM implementation, you have to derive the following: The EAM processes as a subset of IT Governance processes, the meta-model that gears to the structuring of EA information, and the necessary views and reports.

Part of the governance process definition activities is to identify EA roles and management processes. Typical EA roles are enterprise architect, domain architect, application owner, business analyst etc. Every role owner performs specific
processes. An architect plans and analyses architecture zones. An application owner maintains application data and the integration status of his applications in the overall architecture (interfaces, services, business support). Business analysts define IT requirements coming from the business managers, and communicate business/IT capabilities to the business (business/IT alignment). Various other roles may exist that perform additional processes. These processes and their corresponding roles have to be defined and documented.

Next, the meta-model has to be defined and implemented. The meta-model defines the structure of the EA information, which is collected and entered into the architecture repository. The architecture repository ideally serves as the single source of truth for EA data. Certain analyses and reports can be created to finally address the questions that have to be answered to achieve the initially defined goals. These analyses and reports are based on the definition of architecture concepts in the meta-model (i.e. what concepts and properties exist, how are the properties linked to each other; see chapter 5 ‘ADOit EAM Meta-Model’).

Take for instance the goal ‘SOA transformation of core applications before the year 2020’ derived from the IT Strategy ‘Enhance reusability and flexibility of the IT architecture’. To meet this goal, you have to document the currently available core applications including the provided functionality, current support of services, technical interfaces, required technologies etc.

The meta-model should provide concepts like application, service, business function, interface, and technology with the respective attributes storing additional data in order to capture all necessary information.

Views and reports finally should show what applications support what business functions revealing potential redundancies; what kind of technology is required to run the various applications, if SOA compliance already exists; are the services reusable; how tight the application integration is implemented etc.

The EA tool acts as the provider of the meta-model definition, with a data container in the form of a repository, and the means of entering and retrieving architecture data. Basic EAM processes are supported by workflow mechanisms reacting to events, e.g. by sending emails, providing role specific GUIs etc.

Since every organisation has different primary goals and requirements, it is important that an EA tool provides means to implement the user specific functionality with low effort, and thus, the EA tool should be meta-modelling enabled and easily customisable. ADOit is designed to provide out-of-the-box functionality and a meta-model that can be used as the case arises, but is easily modifiable to any specific customer requirements. ADOit enables you to achieve the optimal support for EAM activities in your organisation.
Frameworks Compatible with ADO\textit{it}

3.1 TOGAF 9

3.1.1 EA Processes – Architecture Development Method (ADM) 27
3.1.2 Architecture Repository 29
3.1.3 Business Architecture – Recommended Viewpoints in TOGAF 9 31
3.1.4 Application Architecture – Recommended Viewpoints in TOGAF 9 34
3.1.5 Data Architecture – Recommended Viewpoints in TOGAF 9 41
3.1.6 Technology Architecture – Recommended Viewpoints in TOGAF 9 44
3.1.7 TOGAF Meta-Model in ADO\textit{it} 46

3.2 DoDAF

3.2.1 Viewpoints and Models 49
3.2.2 DoDAF Meta-Model 50
3.2.3 Architecture Development Process 50
3.2.4 DoDAF Support by ADO\textit{it} 51

3.3 PEAF

3.3.1 Procedure – PEAF Processes 52
3.3.2 The PEAF Meta-Model in ADO\textit{it} 55
3.3.3 Viewpoints - PEAF Viewpoints in ADO\textit{it} 56

3.4 Other Frameworks and Standards 56
There are many existing frameworks and approaches for developing EAs. Architects have been successfully using EA frameworks since 1987 when John Zachman published his Zachman Framework. While it can be useful to start EA initiatives using EA frameworks, they are not always applied in a consistent and correct way that would benefit decision makers from the business.

All EA frameworks share a similar structure as well as core components, yet provide a varying degree of detail in certain areas. The core components of an EA framework are:

1. The governance processes,
2. The meta-model,
3. Reports and views.

The process model describes how EAM is conducted in an organisation. It typically covers the following aspects:

- Goal definition,
- Requirements analysis,
- Documentation of the current state,
- Planning steps,
- Implementation guidelines,
- Optimisation and controlling activities.

Some frameworks really shine here (e.g. TOGAF); some are rather minimalistic (e.g. Zachman). You can also differentiate between frameworks that aim at supporting EAM processes which focus on daily operations in contrast to others which focus on EAM projects.

The meta-model for defining the architecture data structure is the second key element of an EA framework. Again certain frameworks are quite detailed in this respect, while others do not bother providing more than a few hints on what basic concepts may be necessary to depict a reasonable Enterprise Architecture.

TOGAF has not been provided a meta-model until version 9, which now includes at least a set of classes and relations, and thus defines a core meta-model with several extensions for additional use cases.

DoDAF in contrast has been providing a very detailed meta-model and became more focused in version 2.0 by narrowing down the number of artefacts. What most frameworks have in common (with some exceptions) is the lack of adequate attributes and properties of the classes in the meta-model.
In the course of adapting an EA framework to the needs of an organisation the necessary attributes of the classes have to be identified and implemented. Tools like ADOit provide a meta-model that contains a set of attributes in the form of a best practice approach, derived from experiences from past EAM projects.

Reports and views are the means to retrieve the information from the repository and present it in a descriptive and comprehensible way. EA frameworks often define so-called views, viewpoints, perspectives, or models, to name just few possible notions in that context.

They basically serve as filters that provide an optimised view on the architecture data geared to a specific set of questions that are to be answered by EAM. For instance, the question of what functional support does IT provide to the business might be best answered by a matrix showing business processes, applications and the business functions or capabilities, which are provided by applications to support the business processes. Every view and report should help answer a specific set of questions. Nearly every EA framework defines such views. EA tools should be able to present the most important views in order to be able to support most of a concrete EA framework.

Some EA frameworks also provide a set of sample or reference models, but most of them do not. Those providing such content are still very basic with respect to the amount of data and its applicability.

It is recommended to implement an EA framework which is pragmatic, benefit-driven, and realistic, focusing only on those topics that add real value to your organisation. In the following section, a number of popular EA frameworks - namely TOGAF, DoDAF and PEAF - and the way they can be applied in ADOit are discussed. There certainly is a plethora of additional frameworks. They mainly share the same pedigree. In this manual, we mainly focus on the frameworks listed above. We also state that the mere act of selecting a framework is not the primary key to success. It is important to select an EA framework that is well understood and can be applied to a concrete organisation with the least need for changes. But in general, nearly every framework can be used with nearly every organisation – when configured properly. We recommend not putting too much energy in the EA framework selection, but instead focusing more on organisational specifics and concrete tailoring of the chosen framework. ADOit supports this approach with proper configuration and customising capabilities.
3.1 TOGAF 9

As The Open Group Architecture Framework (TOGAF) is one of the most accepted and widespread EA frameworks, and the ADOit standard configuration is based on TOGAF 9, this chapter explains how to approach and perform EAM considering TOGAF 9 as a basis. In the following, the term TOGAF refers to TOGAF 9.

The core deliverables of TOGAF 9 are:

- TOGAF Meta-Model which provides a set of architectural content and its relations,
- TOGAF Architecture Development Method (ADM) which provides a method for enterprises to develop their EA following a phased approach, including guidelines and techniques to create the required work products of each phase,
- TOGAF Enterprise Continuum and Reference Content which provides organisations with a large amount of architectural content and reference models to develop their own EA.

In the next sections, built-in ADOit support for TOGAF is discussed in detail. It includes:

- ADOit’s ability to support the Architecture Development Method (ADM) enriched with best practice approaches derived from many years of experience of working on numerous architecture projects.
- A discussion of views, perspectives and deliverables TOGAF recommends, and their mapping to ADOit artefacts.
- A mapping of the ADOit Meta-Model considering TOGAF’s recommendations.

3.1.1 EA Processes – Architecture Development Method (ADM)

When you launch an EA programme in your organisation, the responsible architect usually searches for standards or best practices available in this field. The architecture cycle (TOGAF’s ADM) depicts the framework for the architectural work. In many cases the organisation will certainly not run through this cycle in a straight sequential way.
• **Phase A - Architecture Vision:** Define your scope and vision, and map your overall strategy. On the organisational side, you should establish an organisational model for Enterprise Architecture. Furthermore, the architecture framework is tailored according to the needs and objectives of your organisation. This includes adapting the content meta-model (deliverables and artefacts) to the organisational needs and processes which sometimes requires to customise ADOit.

• **Phase B, C, D - AS-IS and TO-BE Architectures:** Describe your as-is and to-be architectures on business, information systems and technology levels, and perform gap analyses. Often organisations begin with capturing the as-is state using ADOit’s inventorying capabilities, and subsequently, they develop to-be architectures on a high level. Then, analyses and reports are conducted which help organisations identify gaps and possible weaknesses. Finally, they need to track the road map from as-is to to-be.

• **Phase E - Opportunities and Solutions:** This phase deals with identifying the gaps between the baseline and the target state of the previously mapped architectures. Develop the overall strategy, determine what you want to buy, build or reuse, and how you would like to implement the target architectures described in phases B to D. As a result of this gap analysis, you will typically come up with different solution alternatives. Compare and evaluate the different solutions, and select the most appropriate ones.
• **Phase F - Migration Planning:** This phase focuses on developing a migration road map. Projects will change the as-is architecture in a stepwise manner. It is necessary to define and prioritise projects and develop the migration plan for the transition from as-is to to-be.

• **Phase G - Implementation Governance:** Determine how you want to govern the implementation. Recommendations from the responsible architects need to be included early in the project life cycle. In the implementation phase, architectural advice will guarantee projects stick to architecture principles where time and budget shortcomings often put pressure on the people involved in the projects.

• **Phase H - Architecture Change Management:** Monitor the running EA programme and determine whether to start a new cycle, looping back to the preliminary phase. Continuous monitoring and documentation are necessary to ensure that the projects finally produce the expected business value.

The ADM is an iterative process. In every phase you can select appropriate reference models, viewpoints and tools, develop baseline and target architecture description, perform gap analyses, and define the road map.

While the basic approach does not change, the development of the baseline and target architecture will change according to the information needs of the stakeholders in different phases. A major task in the ADM is to develop a complete view of your organisation’s as-is and to-be state. Hence, in the following we will focus on the ADOit support for depicting and defining business architecture, application and data architecture, as well as technology architecture (ADM phases B to D) based on the chosen viewpoints.

How to create a complete picture of the organisation’s as-is and to-be states is discussed. The required means for gap analyses (Phase E – Opportunities and Solutions) and developing migration plans (Phase F - Migration Planning) are introduced briefly. However, the main focus is on phases B to D. All other phases are discussed in chapter 4 ‘ADOit Best Practices’.

### 3.1.2 Architecture Repository

ADOit offers a comprehensive architecture repository, which allows you to bring together information on various architecture artefacts (e.g. processes, functions, applications, technologies, domains) and relate these architecture artefacts to each other. Based on the repository, different views and reports can be created to support the different stakeholders in your enterprise performing their tasks. See chapter 4.2 ‘Populate the EA Repository’ for further information on this topic.
In terms of ADOit you need to:

- Utilise the object catalogue to capture lists and catalogues of required architecture elements,
- Make use of cross-references between the architecture artefacts captured in the various object folders within the object catalogue,
- Make use of ADOit’s timefilter concept to depict current and target states of architecture artefacts (production and decommission dates),
- Generate useful views of the architecture artefacts and their relations in order to support informed decision making, and
- Print or export the available information (or deploy the ADOit Web Client) to share this information with all stakeholders.

TOGAF 9 distinguishes three major types of viewpoints all of which are supported by ADOit:

- Catalogues represent specific, foundational viewpoints including lists of architecture artefacts. The ADOit object catalogue is used for listing all types of architecture artefacts.
- Matrices represent specific viewpoints that show the relationship between building blocks of specific types. ADOit’s matrix view can be used to depict the recommended matrices.
- Diagrams represent graphical viewpoints that display building blocks in a rich and visual way, and more suitable for stakeholder communication. In ADOit you can use the graphical model editor for displaying the required diagrams. Diagrams can be either created manually or generated automatically. An example of an automatically generated diagram is the so-called Business Impact Analysis Diagram (dependency analysis). It presents relations of architecture artefacts on all levels of the EA.

ADOit supports further viewpoints such as portfolio view, GANTT chart and tabular view. Examples can be found in chapter 4 ‘ADOit Best Practices’. 
3.1.3 Business Architecture – Recommended Viewpoints in TOGAF 9

ADOit provides many features and viewpoints for defining and depicting business architectures. All aspects are captured either by ADOit itself or by tight integration with ADONIS, the BOC tool for Business Process Management. In the following, important viewpoints recommended in TOGAF 9 for depicting the business architecture are briefly discussed.

3.1.3.1 Required Catalogues

Based on ADOit’s Meta-Model (see chapter 5 ‘ADOit EAM Meta-Model’), the following catalogues – recommended by TOGAF – can be established:

- **Organisation/Actor Catalogue**: ADOit provides means for cataloguing organisational units, actors and locations. This catalogue is intended to list all participants that interact with IT, including users and owners of IT systems (such as applications and business services).

- **Role Catalogue**: Roles can be catalogued in the object catalogue. The purpose of the role catalogue is to provide a listing of all specific skill sets, responsibilities, or positions in the organisation. Roles allow for defining criteria required for performing processes, rather than the specific actors who will perform the activity.

- **Business Service/Function Catalogue**: The standard meta-model of ADOit allows to populate the EA repository with domains, business capabilities, and business functions.

- **Location Catalogue**: According to TOGAF 9 the location catalogue contains all locations where the organisation carries out business operations or has architecture-relevant assets. Examples are computer centres or end-user computing equipment.

In ADOit, you can populate your EA repository either by creating new objects (e.g. applications) in the Rich Client, creating new objects in the Web Client, importing information via the ADOit Excel Interface or importing information (e.g. processes) from ADONIS to ADOit via the BOC Management Office Interface. For further information, please refer to chapter 4.2 ‘Populate the EA Repository’.
3.1.3.2 Recommended Matrices and Diagrams

Business Interaction Matrix:
The Business Interaction Matrix shows dependencies and communication between organisations and business functions/business services within the enterprise. A two-dimensional matrix with organisational units (x-axis) and functions or services (y-axis) can be used.

If necessary, you can break services down to smaller units. You can create a service tree by using the ADOit Dependency Analysis as shown in figure 10.

Figure 9
Business Interaction Matrix in ADOit

Figure 10
Service Composition Tree in ADOit (Dependency Analysis)
Business Service/Information Diagram:
The Business Service/Information Diagram shows the information necessary to support one or more business services. It shows which data is consumed or produced by a business service and can also show the source of information. In ADOit you can use dependency analyses to depict the relations between services and data. In this case, the dependency analysis (viewpoint ‘Business Impact Analysis’) brings together data elements (so-called business objects), business services and applications.

Functional Decomposition Diagram:
The purpose of the Functional Decomposition Diagram is to show on a single page the capabilities of an organisation that are relevant to the architecture. By examining the capabilities of an organisation from a functional perspective, you can quickly develop models which show what the organisation does, without being dragged into extended debates on how the organisation does it.

ADOit supports the creation of Business Capability Maps. Heat mapping capabilities can be provided optionally. You can use the model type ‘Business Architecture’ to map diagrams of this type. Figure 12 gives an example.
Extension Diagrams:
Furthermore, TOGAF considers the following diagrams as extensions:

- Product Life Cycle Diagram,
- Goal/Objective/Service Diagram,
- Business Use Case Diagram,
- Process Flow Diagram.

In order to support these extension diagrams ADOit can be extended via customising or integrated perfectly with other modelling tools, e.g. process modelling or UML tools. For example ADONIS supports these views out-of-the-box.

3.1.4 Application Architecture – Recommended Viewpoints in TOGAF 9
The objective here is to define the major types of application systems necessary to process data and support business. According to TOGAF 9, central concepts of the application architecture are application systems and their interfaces.

3.1.4.1 Required Catalogues
The following catalogues recommended for capturing the application architecture are provided out-of-the-box by ADOit:

- **Application Portfolio Catalogue:** Identify and maintain a list of applications in ADOit using the object catalogue.
- **Interface Catalogue:** Maintain a list of interfaces in ADOit. Interfaces are typically provided by one application and used by other applications.
3.1.4.2 Recommended Matrices and Diagrams

The following matrices and diagrams for depicting the application architecture recommended by TOGAF 9 are provided out-of-the-box by ADOit:

**System/Organisation Matrix:**

The purpose of this matrix is to show the relation between systems (i.e. applications) and organisational units within the enterprise. A two-dimensional matrix with organisational units (x-axis) and applications (y-axis) is used.

![System/Organisation Matrix](image1)

Organisational units perform business functions of which some are supported by IT. To gain an overview of the dependencies, typically a matrix with applications and functions is used. You can generate a three-dimensional matrix with organisational units (x-axis), business functions (cells) and applications (y-axis).

![System/Organisation Matrix with Functions](image2)
Role/System Matrix:
The purpose of the Role/System Matrix is to depict the relationship between systems (i.e. applications) and the business roles that use the systems within the enterprise. A three-dimensional matrix in ADOit containing roles (x-axis), business functions (cells) and applications (y-axis) is used to gain an overview of the dependencies of these architecture artefacts.

System/Function Matrix:
The purpose of the System/Function Matrix is to show the relationship between systems (i.e. applications) and business functions within the enterprise. The System/Function Matrix is a two-dimensional matrix with applications on one axis and business functions on the other.
Alternatively, you may want to use a matrix that includes applications, business services and business functions. For this purpose you can use a three-dimensional matrix with applications (x-axis), business services (cells) and business functions (y-axis).

**Application Interaction Matrix:**

The purpose of the Application Interaction Matrix is to depict the relations between systems (i.e. applications). An ADOit matrix with applications (on x-axis and y-axis), and interfaces (cells) is usually used here.
Application Communication Diagram:
The purpose of the Application Communication Diagram is to depict the communication between applications. It shows applications and interfaces between the applications. Interfaces may be associated with data entities where appropriate. The built-in ADOit interface report can be used to generate application interaction relations automatically. It includes applications, interfaces, services and the transferred business objects.

Application and User Location Diagram:
The Application and User Location Diagram depicts the geographical distribution of applications. For this purpose, you might define a matrix with applications (y-axis) and locations (x-axis). The analysis of this matrix might help to identify opportunities for rationalisation as well as identify duplication or gaps.
Application Migration Diagram:
The Application Migration Diagram identifies application migration from the baseline architecture to the target application architecture. This diagram is intended to support a more accurate estimation of migration costs by showing precisely which applications need to be brought into production or are being decommissioned between migration stages. The ADOit GANTT chart supports timeline views to depict application migration stages and parallel running applications. It shows when one application is phased out, and when the new application is introduced.

Alternatively, you can use the model type ‘Analysis Model’ to depict the migration diagram. Use the modelling class ‘Block’ for depicting the timeline. Put applications including their ‘replaces application’ relations on top. Figure 22 shows an example.
Software Distribution Diagram:

The Software Distribution Diagram shows how application software is structured and distributed across the organisation. This diagram shows how physical applications or technologies are distributed across locations. The ADOit dependency analysis as well as a matrix diagram with applications or technologies (x-axis) and locations (y-axis) can be used to create a Software Distribution Diagram.
3.1.5 Data Architecture – Recommended Viewpoints in TOGAF 9

The goal of this step is to define the major types and sources of data necessary to support the business. Please note that this effort is not concerned with database design. In this step, you define the data entities relevant to the organisation from a business view. Designing logical or physical storage systems is not discussed in this step.

ADOit is used to:

- Define the relevant data entities in the repository,
- Assign responsibilities to data entities, and
- Analyse data entities via the various viewpoints and reports.

3.1.5.1 Required Catalogues

The Data Entity/Data Component Catalogue recommended by TOGAF is supported by ADOit. The purpose of this catalogue is to identify and maintain the data used across the organisation. This includes data entities, and also the applications and infrastructure artefacts where data entities are stored. According to TOGAF 9 this catalogue serves as a basis for Information Technology Management and Data Governance policies and also encourages effective data sharing and reuse of data.
3.1.5.2 Recommended Matrices and Diagrams

**Data Entity/Business Function Diagram:**

The Data Entity/Business Function Diagram depicts the relationship between the following entities: Data entity (business objects in ADOit terms), business functions and organisational units.

In ADOit, you can use a dependency analysis or a matrix with organisational units (x-axis), business objects (y-axis), and business functions (cells) to depict the relationships between these entities. Figure 25 and 26 depicts examples for both viewpoints.

---

**Figure 25**
Data Entity/Business Function Diagram/Matrix

**Figure 26**
Data Entity/Business Function Diagram/Dependency Analysis
System/Data Matrix:
The System/Data Matrix is used to depict the relationship between systems (i.e. applications) and the data entities (business objects in terms of ADOit) that are accessed and updated by the systems. Here a matrix with business objects (x-axis) and applications (y-axis) is used. The cells represent the relation types (create, read, update, delete).

Class Diagram:
The purpose of the Class Diagram is to depict the relations between the data entities within the enterprise. Use the ADOit model type ‘Application Architecture’ and change the mode to ‘Information Architecture’. ADOit supports class diagram modelling similar to UML class diagrams with all UML relation types. Figure 28 depicts an example.

Data Dissemination Diagram:
The purpose of the Data Dissemination Diagram is to show the relations between data entities, business services, and applications. The diagram depicts how the logical data entities are to be realised by applications and business services. Use the ADOit Dependency Analysis with data (level 1), business services (level 2) and application components (level 3).
Extension Matrices and Diagrams:
If you would like to use Extension Matrices and Diagrams, they can be configured via ADOit customising. The following additional diagrams and matrices described by TOGAF are not discussed in detail here:

- Class Hierarchy Diagram,
- Data Security Diagram,
- Data Migration Diagram (including data flows between applications),
- Data Life Cycle Diagram.

3.1.6 Technology Architecture – Recommended Viewpoints in TOGAF 9

3.1.6.1 Required Catalogues
The following catalogues for technology architecture described by TOGAF are supported in ADOit:

- **Technology Standards Catalogue:** This catalogue documents the agreed standards for technologies across the organisation. It typically contains technologies, versions, the technology life cycles, and the refresh cycles. ADOit offers the concept ‘technology’ including all required attributes in order to represent the required snapshot of the organisation’s standard technologies that are or can be deployed.

- **Technology Portfolio Catalogue:** The purpose of this catalogue is to identify and maintain a list of all the technologies in use across the enterprise, including hardware, infrastructure software, and application software. Tech-
Technology portfolios support life-cycle management of technology products and versions and also form the basis for defining technology standards. The technology portfolio catalogue can be structured according to TOGAF’s Technology Reference Model (TRM). See chapter 4.4 ‘Technology Portfolio Management’ in the ‘ADOit Best Practice’ section of this manual.

3.1.6.2 Recommended Matrices and Diagrams

System/Technology Matrix
The System/Technology Matrix depicts the mapping of applications to the technologies. Use a matrix with applications (x-axis) and technologies (y-axis) for this purpose.

![System/Technology Matrix](Figure 30)

Platform Decomposition Diagram:
The Platform Decomposition Diagram shows the technology package that support the operation of the applications. For this purpose, ADOit can bundle technologies to technology packages. The Platform Decomposition Diagram provides an overview of the organisation’s technology packages and the applications using these technology packages. Use a matrix diagram with applications (x-axis) and technology package (y-axis) for this purpose.

![Platform Decomposition Diagram](Figure 31)

All further diagrams described as TOGAF extension diagrams for technology architecture can be configured via ADOit customising.
3.1.7 TOGAF Meta-Model in ADOit

Starting with Version 9, TOGAF provides a meta-model that describes the main concepts of Enterprise Architecture and its relations. This meta-model is structured into core meta-model and extensions. The TOGAF core meta-model provides a minimum set of architectural content to support traceability across artefacts; the extensions are only necessary for special concerns (e.g. infrastructure consolidation).

The ADOit Meta-Model is based on TOGAF 9 and BOC’s project experiences (best practice approach).

ADOit fully supports the TOGAF 9 core content meta-model. The ADOit Meta-Model is described in the appendix in detail. In the following, a brief mapping is provided to give you an overview of the most important concepts.
### Table 1
Mapping of the TOGAF Meta-Model on the ADOit Standard Configuration

<table>
<thead>
<tr>
<th>TOGAF 9</th>
<th>ADOit Standard Meta-Model</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not defined</td>
<td>Domain</td>
<td>High-level approach to categorise architecture artefacts of any type for the purpose of organisation, standardisation, and definition of clear responsibilities. Typically for each domain, domain managers are defined. Typically architecture artefacts like applications and business objects are assigned to the various domains.</td>
</tr>
<tr>
<td>Business Process</td>
<td>Business Process</td>
<td>In ADOit business processes can be divided into sub-processes.</td>
</tr>
<tr>
<td>Organisation</td>
<td>Organisational Unit</td>
<td>In ADOit organisational units can be assigned to different artefacts.</td>
</tr>
<tr>
<td>Location</td>
<td>Location</td>
<td>Locations can be used to determine the geographical place of different artefacts.</td>
</tr>
<tr>
<td>Actor</td>
<td>Actor, User</td>
<td>In ADOit you can choose to use either the concept ‘user’ or the concept ‘actor’. Both concepts can be assigned to various other artefacts like ‘application’ and ‘business service’. Whereas ‘actor’ is used for documentation purposes only, objects of type ‘user’ are shared from the ADOit User Management Component (Administration Toolkit) and can be used for setting write access to architecture artefacts.</td>
</tr>
<tr>
<td>Role</td>
<td>Role</td>
<td>Users can be assigned to roles. For example, User X can be assigned to role Enterprise Architect. The role concept enables you to define access rights for a group of users.</td>
</tr>
<tr>
<td>Business Capability</td>
<td>Business Capability, Business Function</td>
<td>In ADOit ‘business functions’ can be hierarchically structured. Top-level business functions are called business capabilities.</td>
</tr>
<tr>
<td>TOGAF 9</td>
<td>ADOit Standard Meta-Model</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application Component</td>
<td>Application, Application Module</td>
<td>In the ADOit standard meta-model applications consist of application modules.</td>
</tr>
<tr>
<td>Not defined</td>
<td>Interface</td>
<td>In ADOit interfaces represent the point of interaction between components.</td>
</tr>
<tr>
<td>Business Service</td>
<td>Equally used in both frameworks</td>
<td>In TOGAF a Business Service supports business capabilities through an explicitly defined interface and is explicitly governed by an organisation.</td>
</tr>
<tr>
<td>Not defined</td>
<td>Business Service</td>
<td>ADOit uses the term ‘Business Service’ in terms of SOA services. Business services typically encapsulate functionality (Business Functions) which can be used by various applications or orchestrated in a network of Business Services.</td>
</tr>
<tr>
<td>Data Entity</td>
<td>Business Object</td>
<td>Business objects represent real world objects, such as partner.</td>
</tr>
<tr>
<td>Platform Service</td>
<td>Technology Package</td>
<td>Technology packages consist of different technologies.</td>
</tr>
<tr>
<td>Technology Component</td>
<td>Technology</td>
<td>In ADOit the underlying platform of an application is called technology.</td>
</tr>
</tbody>
</table>

ADOit also offers meta-model extensions to support the additional scenarios described in TOGAF 9.

If required, the complete TOGAF core content meta-model and the recommended extension modules can be provided through customising the meta-model by the ADOit administrator.
3.2 DoDAF

The Department of Defense Architecture Framework (DoDAF) V2.0 is a framework chiefly aimed at providing means for information sharing between various stakeholders in the US Department of Defense. It is based on the DoD’s C4ISR Architecture Framework and focuses on architectural data, defining guidelines for obtaining and visualising EAs as well as requirements on architectures. Driven by the Clinger-Cohen Act, which is designed to improve the way the federal government acquires, uses, and realises IT, it acts chiefly as a tool for the DoD IT Management (e.g. CIO), but today is also adopted by a number of non-military organisations. Since it is not nearly as widely spread as TOGAF, we only give a brief introduction to DoDAF and will not dwell on the details.

DoDAF is basically positioned as a guide for the development of integrated architectures. Other frameworks, such as NAF (NATO Architecture Framework) and MoDAF (Ministry of Defence Architecture Framework), are largely based on DoDAF, and hence not discussed in this manual.

The basic DoDAF concepts are described on the following pages.

3.2.1 Viewpoints and Models

Viewpoints and models are the means to depict EAs.

- **Models:** A model is a template for data collection. Models are used to describe subsets of architecture in viewpoints.

- **Viewpoints:** A viewpoint describes data drawn from one or more perspectives and organised in a particular way useful to management decision-making.

Figure 33 shows the viewpoints that are defined in DoDAF, each of which aiming at describing a specific part of the EA geared to certain stakeholder requirements. For each viewpoint a set of models are assigned that may be used to depict the desired aspects. The models are named according to their assignment to a viewpoint. For example, the Project Viewpoint (PV) is described by using the model types PV-1: Project Portfolio Relationships, PV-2: Project Timelines and PV-3: Project to Capability Mapping.
Models are created by instantiating the DoDAF meta-model concepts, which are introduced in paragraph 3.2.2

### 3.2.2 DoDAF Meta-Model

The DoDAF Meta-Model (DM2) is replacing the Core Architecture Data Model (CADM) and consists of a Conceptual Data Model (CDM), a Logical Data Model (LDM), and Physical Exchange Specification (PES). The CDM defines the high-level data constructs from which architectural descriptions are created in non-technical terms. The LDM adds information, such as attributes and relations, to the CDM and thereby acts as the part of the meta-model that gears to the construction of visual models. PES is based in the LDM with general data types specified and implementation attributes (e.g. source, date) added and is then generated as an XSD, the DM2 PES XML schema (XSD). This exchange format is ideally automatically generated and can be used to exchange DoDAF compliant models between modelling tools.

For a detailed description of the DM2, please refer to the DoDAF V2.0 Volume 2 handbook.

### 3.2.3 Architecture Development Process

DoDAF provides a process model, the high-level 6-Step Architecture Development Process (see figure 34). The process is described as data-centric, thus...
hinting at how to depict architecture in the respective process steps. However, it is very briefly described in the DoDAF handbooks and as such has to be complemented with process models of other approaches, like best practice based frameworks (e.g. TOGAF). It basically gives a rough orientation on what order of activities should be considered when depicting Enterprise Architecture in the context of EAM.

3.2.4  DoDAF Support by ADOit

The ADOit standard EAM meta-model is suited to capture most of the core concepts of DM2, covering more detail on attributes instead of class levels. Thus, the level of detail on meta-model level is kept moderate compared to the complete DM2. However, since ADOit is meta-model enabled, more detail can be added, if desired.

The viewpoints can be expressed by using the ADOit tool concepts Model and Mode, where the first describes the grouping of certain classes in a model (alias diagram). The second is a means to reduce models to subsets of objects and relations in order to reduce further complexity in a graphical view.
Since the process model of DoDAF is ‘data-centric’ it chiefly focuses on the deliverables of each process step. As such, ADOit supports the process model by providing the means to depict all desired artefacts and relate them as prescribed in the process model. Analyses that are defined in step 5 can be performed by using the ADOit mechanism like generated views, ad-hoc and predefined queries, as well as additional configurable mechanisms. Refer to TOGAF (see chapter 3.1 ‘TOGAF 9‘), where these analysis concepts are applied in more detail to an EA framework.

The DM2 PES XML exchange format can be supported by the generic and configurable XML interface of ADOit. This interface can be used to import and export DoDAF-compatible repository content and thereby realising the exchange with other DoDAF-compliant tools.

### 3.3 PEAF

**What is PEAF**

The Pragmatic Enterprise Architecture Framework (PEAF) is a free-to-use architecture framework which provides an elementary set of what is required to launch an EA programme. The PEAF framework is split into two parts: Products and Processes.

Products basically describe the artefacts that are required for an organisation to start an EA programme, whereas Processes describe how to launch an EA programme; in other words, it shows the steps organisations should take if they want to successfully establish and operate Enterprise Architecture Management. However, the PEAF framework does not specify the necessary processes in detail, and therefore, these processes must be developed separately within your organisation.

For further information on PEAF, please visit [http://www.pragmaticea.com](http://www.pragmaticea.com).

#### 3.3.1 Procedure – PEAF Processes

The PEAF framework defines three phases or processes to establish Enterprise Architecture Management: Prepare, Implement, and Operate. However, it is recommended to use PEAF in association with the ADOit Best Practices since the PEAF framework does not include detailed instructions for developing the different processes or advice on implementation steps.
3.3.1.1 Prepare
This phase aims at convincing the business side to launch an EA programme. It is paramount to clarify the reasons for adopting EAM. Moreover, the scope of the initiative should be defined in this process, considering the involved risks and the maturity level of the organisation. Then, it should be defined which maturity level the organisation wants to achieve during what period of time. Finally, it is important to peruse available EA documentations.

3.3.1.2 Implement
According to the PEAF framework, the implementation of EAM takes places in six steps: Prepare EA Education, Change Motivation Model, Define the EA Meta-Model, Setup an EA Modelling Tool, Setup EA Governance, and Mitigate the EA Risks.

Firstly, stakeholders are identified to receive training regarding EAM in general and the chosen framework in particular. This training typically takes place together with ADOit training. It is important to gain credibility and obtain buy-in regarding the initiative. Therefore, the stakeholders have to understand why they need to be involved in EAM and how the organisation and each individual benefit from such an EA programme. Secondly, necessary cultural changes within the organisation need to be considered to be able to successfully pursue the EAM initiative in the long run.

In the next step, the meta-model needs to be defined. ADOit users usually prefer to commence with a small meta-model including the application architecture of the organisation. However, ADOit provides the opportunity to extend the meta-model according to the business needs of your organisation. The BOC Group and its tool ADOit are PEAF certified, and therefore, the PEAF Meta-Model is available in ADOit. For further information on the meta-model, please refer to the upcoming chapter 3.3.2 ‘The PEAF Meta-Model in ADOit‘.
Next, an EA Governance including best practice principles, and strategies for mitigating the risks concerning an EAM initiative should be planned and put into practice in the Operate phase.

3.3.1.3 Operate
Operating means ‘doing’ EAM, which is, integrating EAM and aligning it with the available processes and structures within the organisation. To operate EAM, you need to provide information about EAM by using training materials. Make sure to use the ADO\textit{it} training material for your promotions.

Strategic Planning
The next step is strategic planning, which is conducted usually once a year or when triggered by large changes within the organisation. This step aims at updating the Enterprise Strategy Model, and subsequently, updating the entire Business and IT Strategy of the organisation.

Modelling the Current State Architecture
In this step, the model of the current state of your EA should be populated. For this purpose, various questions have to be asked, such as: Which applications provide certain information? Which technology entities exist? Which applications are based on those technologies?

Once you have defined the necessary questions, you should commence gathering information on different artefacts and populate the EA repository. The repository concept of ADO\textit{it} will help facilitate this task. The ADO\textit{it} repository is split into a model repository, where you store created views of the EA (so-called models), and an object repository, where all the architecture artefacts are saved. Once you have created an architecture artefact, it will be stored in the object repository and you can easily use it in any model. Furthermore, ADO\textit{it} features a powerful Excel interface, with which you can easily populate the EA repository with the information available in your preconfigured Excel sheets. In the Administration Toolkit you can configure the Excel interface.

In addition to the PEAF framework, or as an extension of PEAF, you can create users with different roles in ADO\textit{it} and assign those users to different objects by setting specific rights to each user (see chapter 4.2 ‘Populate the EA Repository’). In order to find the answers to your defined questions, you need to create reports and views. ADO\textit{it} provides you the opportunity to evaluate your current state by generating reports (such as PDF report), analyses (such as dependency analysis) and views (such as portfolio view). Moreover, you can define your very own reports, analyses and views in ADO\textit{it}, and therefore extract information according to your requirements. The decentralised data maintenance is conducted by the specific data owner.
Performing Governance
Perform governance using the best practice and governance principles defined in the Implement phase.

Measure, Analyse and Adjust
Based on your gathered data, you can conduct various types of evaluations. In the Rich Client you can configure views and reports according to your organisation’s goals and based on the organisation-specific extension of the meta-model as well as create users, whereas via the Web Client you can gain decentralised access to your data, reports and views.

In addition to the current state of your architecture, you can also plan and maintain the target state of your organisation. For further information, please refer to chapter 4.6 ‘Master Planning’.

3.3.2 The PEAF Meta-Model in ADOit
The PEAF framework suggests the following models: Strategy Model, Structural Model, Portfolio Model, and Principles Model. ADOit supports the entire PEAF Meta-Model as shown in figure 36. It depicts the recommended model types.
Each model type comprises a variety of concepts. All required model types including the assigned concepts and relations are available in the ADOit PEAF configuration. You can decide on whether to use the ADOit standard configuration, the ADOit PEAF configuration or any other. For a detailed discussion of the PEAF Meta-Model, please refer to http://www.pragmaticea.com

### 3.3.3 Viewpoints - PEAF Viewpoints in ADOit

The PEAF framework is almost exclusively dedicated to defining artefacts and their relations to each other. Viewpoints are not defined in detail. However, the ADOit PEAF configuration is shipped with a variety of predefined viewpoints based on the following generic viewpoint types:

- Catalogues,
- Lists,
- Matrices,
- Diagrams and Dependency Analyses,
- Portfolio Views,
- GANTT Charts.

### 3.4 Other Frameworks and Standards

So far, some of the popular frameworks have been mentioned. However, if you would like to use your own frameworks and standards, the meta-modelling functionality of ADOit fully supports your endeavour.
Frameworks compatible with ADOit
ADOit Best Practices

4.1 Architecture Principles 62
4.2 Populate the EA Repository 72
4.3 Application Portfolio Management 83
4.4 Technology Portfolio Management 95
4.5 Demand and Project Portfolio Management 105
4.6 Master Planning 114
4.7 Integrating EAM and BPM 127
In this section, EAM best practices with ADOit are presented. These best practice scenarios are based on out-of-the-box ADOit configurations and allow you to get started with your EA programme instantly. The presented scenarios are derived from popular EA frameworks – above all TOGAF 9 – and practical experience gained by BOC and ADOit user companies in numerous EAM projects. The discussed best practice scenarios can be adapted to or integrated in your desired EA framework as required. They serve as a ‘modular assembly system’ to build an EA framework according to your needs.

Having a high-level of understanding related to the dominant factors in your organisation, and having identified the need to address current problems and improvement potentials, you can choose between the most relevant best practice scenarios. Keep in mind that by focussing on too many scenarios at once you might encounter difficulties during the process of allocating resources for your EA programme.

In contrast to waterfall-like approaches which focus on detailed designs upfront, the ADOit best practices allow you to perform a milestone-driven approach through proceeding in a step-wise fashion. Start with one or two scenarios and focus on quick wins to gain the commitment of your management and the involved stakeholders, and to obtain buy-in for your agenda. Continuously expand your EAM initiative by adding further scenarios as soon as the ongoing scenarios are implemented in your daily IT operations. The discussed approach allows for implementing a process to capture, monitor, and report on the achieved benefits. By applying revision iterations – as it is recommended with TOGAF’s ADM – your EA programme will continuously be fine-tuned. This constitutes the basis for implementing continuous improvement within your EA programme and your entire architecture.

In order to choose the proper scenarios to begin with, you may ask yourself different questions, such as:

- Who are my EAM stakeholders?
- Which of the best practice scenarios will allow for gaining quick wins for my stakeholders?
- Which deliverables do my stakeholders value most?
- How should I prioritise my EAM activities, given the required benefits and deliverables?

The answer to the above questions will enable you to compose ADOit best practice scenarios according to your own EA framework, exactly tailored to your individual needs.
Do not underestimate the role of stakeholders. Make sure to be aware of each and every stakeholder in your organisation. Know the interest of your stakeholders and stakeholder groups. Classify your stakeholders in terms of influence, power, interest etc. You need to identify individuals who will be involved in your projects if you want your EA programme to be successful. Keep in mind that a stakeholder analysis will help you develop and choose the right scenarios for your organisation. For further information on stakeholder management, please refer to TOGAF 9.

Figure 37 gives an overview of the upcoming best practice scenarios in this section.

- **Populate the EA Repository**: Collecting and organising information about the current architecture of your organisation leads to transparency, which is the first important step towards a resilient business and IT architecture. A comprehensible architecture allows you to predict how changes made to applications, business processes and other artefacts affect your architecture. Also, you can identify associated risks and estimate their impact on your organisation.

- **Architecture Principles**: According to TOGAF, architecture principles define the underlying general rules and guidelines for designing and implementing your EA. They reflect a level of consensus among the various
elements of the enterprise, and form the basis for making future IT decisions. Thus, architecture principles form the foundation of any EA work and should be considered in any of the best practice scenarios.

- **Demand Management/Project Portfolio Management**: Implementing Demand Management helps you assign functional requirements to architecture artefacts. You will see that synergies arise primarily when you minimise cross-departmental redundancies by evaluating and prioritising new demands. Project Portfolio Management provides an understanding of the overall context of both current and proposed (IT) projects. It helps you recognise the impact that projects will have on your business and IT architecture and further, you will see how projects are dependent on each other.

- **Application Portfolio Management**: Classify and assess your applications based on different technical and financial measures in order to define the investment strategy for each application. It provides valuable input for defining your EA road map on an application architecture level.

- **Technology Management**: Prevent weakening of your architecture by implementing Technology Management. This scenario helps you narrow down the number of technologies in use to only the technologies that are necessary. It defines standard technologies and standard technology packages to be used in development projects. As a result, the maintainability of the systems increase and at the same time, you save on expenses and overheads.

- **Master Planning**: Design your organisation’s target architecture. During the designing process, you will typically develop and assess different solution scenarios while considering ongoing projects. The current, target and any transition architecture are comprehensively planned and agreed on by business and IT stakeholders. The master plan will provide the justification, scope and objectives for the required migration projects.

Each chapter on best practice scenarios is structured as follows:

- **Objectives**: This part provides you with the definition and objectives of the particular best practice scenario. It describes why you need to perform this particular scenario and what the outcomes are. It is important to gather this type of information in advance, so that you know for certain where and why to invest time and effort.

- **Complementary Scenarios**: This part depicts how one particular best practice scenario is impacted by or is interfacing with other scenarios. Despite the fact that there are a number of dependencies and overlaps between different scenarios, you can launch each scenario independent of other scenarios. Keep in mind to implement only those scenarios that are useful for your stakeholders.
• **Stakeholders and Their Role:** This part shows which roles are typically assigned to different stakeholders in a particular scenario and what their functions are. Usually, subject-matter experts from different fields and departments should be involved in an EA programme. In an ideal situation members of the EA core team and other stakeholders, such as application owners, operations staff and business analysts, work together. This takes a great amount of commitment. It is important to convince stakeholders of the advantages and positive outcomes of the programme before they become involved in the entire process. Moreover, keep in mind that stakeholders will always have some self-interest; therefore, you have to think about the advantages that individual stakeholders can yield. In other words, think about ways to motivate stakeholders to become involved, commit to the process and stay. Bear in mind how maintaining the EA inventory helps stakeholders meet their own objectives.

• **Meta-Model:** The meta-model can be understood as the construction plan of the EA inventory. Every EAM scenario is based on gathering, interrelating, maintaining and evaluating information on the EA. This part discusses which architecture artefacts and relations are in the limelight when launching a particular scenario. Make sure that the object catalogue is populated with the required information on architecture artefacts, and required relations and attributes in order to perform the EAM scenario of your choice.

• **Procedure:** This part describes step by step what you need to do to achieve the defined objectives. It shows why you should take a particular step and how it affects the entire procedure. Each procedure makes use of the numerous predefined viewpoints in order to support decision making. Most of the procedures are iterative, and need to be performed on an ongoing basis in order to reach your goal.

### 4.1 Architecture Principles

#### 4.1.1 Objectives

In general, principles are rules or guidelines that propel organisations toward reaching their goals.

Architecture principles are high-level statements regarding the use and deployment of (IT) resources and assets. They reflect a level of consensus throughout the organisation and show whether the organisation’s Enterprise Architecture is being developed according to its values, vision and mission. They guide decision-making for different architecture artefacts. According to TOGAF 9, they usually consist of statement, rationale, and implication.

The principle statement describes briefly, concisely but clearly the fundamental rule that should apply to specific architecture artefacts. In other words, the statement describes ‘what’ you have to do.
The rationale highlights the benefits of applying a specific principle, and thus describes ‘why’ you have to do it.

The implication points out the requirements, imperative for both IT and business, for applying a specific principle to the organisation, and thus describes ‘how’ you can put the principle into practice.

4.1.2 Complementary Scenarios

The architecture principles might influence all best practice scenarios. Furthermore, the architecture principles need to be followed in every solution implementation project.

![Diagram of important scenarios interfacing with architecture principles]

4.1.3 Stakeholders and Their Role

Essentially, the enterprise/domain architect is responsible for developing architecture principles which are typically derived from the business and IT strategy. Depending on your architecture principles various people will provide the enterprise/domain architect with appropriate information. For instance, if you are using data principles, the data architect provides the enterprise/domain architect with information regarding the compliance of the data architecture with the agreed data principles.
4.1.4 Meta-Model

The focus on the meta-model varies depending on which architecture principles you have developed. For instance, if you have defined application principles, obviously the focus will be on applications and probably application technologies. If you have developed a set of data principles, the main focus will be on the data architecture and thus, on data and data technology. Figure 40 gives an example, depicting architecture principles, and architecture artefacts they might be assigned to.
4.1.5 Procedure

The following steps assist you in implementing and applying architecture principles in your organisation:

**Step 1 – Develop and agree on architecture principles**

The development of architecture principles may vary from organisation to organisation and depends mainly on the structure, field of activity, and strategy and objectives of your organisation. However, the following general measures can be applied to almost every organisation, and describe how a good set of principles should look like:

1. **Few in number:** Define only principles that are absolutely necessary for your organisation. The less you have, the better the result. Usually a set of 10-20 architecture principles will do. (see table below for examples of architecture principles)

2. **Comprehensible:** Principles should be unambiguous and easy to understand for everyone in your organisation. Clear statements will help avoid or at least minimise violations against principles.

3. **Sound:** Sound principles support decision-making even in complex and controversial situations.
4. **Full coverage:** The defined principles should cover the most important situations.

5. **Consistent:** Principles should be consistent. In other words, adhering to one principle should not lead to the violation of another. Carefully choose the words that describe your statements so as to avoid multiple interpretations.

6. **Long-lasting:** Principles should be long-lasting, and at the same time adaptable to change.

Beside the mentioned measures, you should figure out why you need specific principles and how those principles can support you in decision-making. Principles should serve the entire organisation. A poorly defined set of principles will soon become out of use. Moreover, your entire Enterprise Architecture will lose credibility since your principles will appear as self-serving. Please remember that architecture principles should drive behaviour, and thus, it is crucial for every employee to be committed to each and every architecture principle.

In ADOit, every architecture principle is described via the following attributes: Name, Statement; Rationale, and Implications. See figure 42 for an example.

---

*Figure 42*

Notebook of an Architecture Principle in ADOit
Architecture principles can be subdivided into different groups such as business principles, application principles, data principles, technology principles etc.

Below you will find examples of architecture principles (a number of which were taken from TOGAF 9) that guide the definition process of target architectures and architectural solutions.

| **Business Principles**               | Maximize Benefit to the Enterprise |
|                                      | Business Continuity                |
|                                      | Compliance with Law                |
| **Application Principles**           | Technology Independence            |
|                                      | Ease-of-Use                        |
| **Technology Principles**            | Responsive Change Management       |
|                                      | Control Technical Diversity        |
|                                      | Interoperability                    |
| **Data Principles**                  | Data Trustee                       |
|                                      | Golden Source Principle            |
|                                      | Data Security                      |

For further samples and a detailed discussion on these architecture principles, please refer to TOGAF 9.
The tables below show the data principles ‘Data Trustee’ and ‘Golden Source Principle’ in more detail:

**Example 1 – Clear Responsibility for Business Objects**

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Trustee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement</td>
<td>Each business object has a trustee accountable for data quality.</td>
</tr>
<tr>
<td>Rationale</td>
<td>One of the benefits of an architected environment is the ability to share data (e.g., text, video, sound, etc.) across the enterprise. As the degree of data sharing grows and business units rely upon common information, it becomes essential that only the data trustee is involved in decision-making about the content and structure of the business objects (data entities) he/she is responsible for.</td>
</tr>
<tr>
<td>Implication</td>
<td>The data trustee will be responsible for meeting quality requirements levied upon the business object for which the trustee is accountable. The data trustee is responsible for the structure of the business object (data architecture). Thus, any change or enhancement to the business object is governed by the data trustee.</td>
</tr>
</tbody>
</table>

**Example 2 – Golden Source Principle of Business Objects**

<table>
<thead>
<tr>
<th>Name</th>
<th>Golden Source Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement</td>
<td>Business objects are only allowed to be created/updated/deleted by exactly one application.</td>
</tr>
<tr>
<td>Rationale</td>
<td>It is vital to keep business objects consistent, and thus, it has to be ensured that business objects are manipulated by only one specific application. For instance, if the applications Business Partner (PAS) and Sales Management (SMA) both manipulate the business object Customer Address, inconsistencies might be caused.</td>
</tr>
<tr>
<td>Implication</td>
<td>To avoid this type of data inconsistency, it is required to define the true source of each business object – the so-called Golden Source.</td>
</tr>
</tbody>
</table>
Step 2 – Relate architecture principles to architecture artefacts

In the next step, you should relate the defined architecture principles to the appropriate architecture artefacts. For instance, if your architecture principles concern applications, open the notebook of an application, and add your principles to the chapter Relevant Architecture Principles. The same applies to the given examples on data principles in step 1. Each data principle is assigned to each of the business objects.

![Figure 43 Notebook of Business Object/Chapter Relevant Architecture Principles]

Step 3 – Assess and evaluate architecture principles

Merely the act of defining architecture principles and relating those principles to architecture elements does not mean that your set of principles are efficiently supporting your Enterprise Architecture nor does it mean that your architecture principles promote decision-making in your organisation. Architecture principles will only become valuable when employees are committed to abiding by them, and when you have the opportunity to evaluate your principles. Thus, check your entire Enterprise Architecture against the stated architecture principles.

For this purpose, ADOit provides users with various reports and viewpoints. For each architecture principle you can define queries and viewpoints in the Administration Toolkit or by using the ‘Ad-hoc analysis component’.

Consider the following example: As already mentioned, architecture principles can be linked to different architecture artefacts on any level of the EA. To illustrate this example the discussed sample data principles from step 1 will be used.

The principle ‘Data Trustee’ says that for every data element you have to assign a responsible person. To identify data objects that violate this principle you can perform an ad-hoc analysis which displays all business objects that are not linked to any responsible person. Figure 44 depicts the dialogue for defining the query. The results of the analysis are displayed in figure 45.
Besides ad hoc queries, you can perform predefined queries which you have to configure in the Administration Toolkit first. For example: To find all business objects that are created by more than one application, you can use the predefined query ‘Business Objects written by multiple Applications’. The dialogue for running the predefined query and its results are depicted in figure 46.
Based on the query results, you can generate a matrix with those two principles and all existing business objects. Here you can directly state to which point business objects are covering the defined principles. For example, change the status of all business objects that violate the defined principles. You can choose between the following options: Low, medium or high coverage. In addition, you can mark these objects if there is any action required in the future.

If, for example, the golden source principle is violated, the domain/enterprise architect together with the data trustee needs to define the true source of the respective business object. This has to be considered, when defining the target architecture.
4.2 Populate the EA Repository

4.2.1 Objectives

Many organisations are not aware of their architecture or rather, they do not know how their architecture is structured and how the various architecture artefacts are linked to each other. The employees of such organisations have to maintain applications and technologies which often is a very time-consuming job. To make matters worse, the application and technology infrastructure of many organisations comprise legacy systems that have become embedded over the years, or the infrastructure may have gone through a series of adjustments due to the restructuring of the organisation, splits, mergers and acquisitions. The worst is yet to come: Employees usually memorise architecture-related information. In other words, if these employees would leave the organisation some day, they would probably take along all the information, leaving their successors with barely anything to work with. Unsurprisingly, such a hardly comprehensible situation will inhibit efficient planning and the effective maintenance of the EA.

Considering all the above-mentioned weak spots it is important to understand the current state architecture of your organisation. Otherwise, you will not see what needs to be changed or how changes affect elements across the different layers of the EA. Now, taking inventory of the architecture of your organisation enables a high degree of transparency across the entire EA. The impact of planned changes to applications, business processes, and other elements of the EA becomes predictable. Thus, maintaining an EA inventory instantly leads to the following results:

- **Reduced cost:** The chain of inquiries for data clarification is completely reduced. This minimises the efforts in project planning, and lowers the frustration over repeated data gathering initiatives.

- **Informed decision making:** The EA inventory makes it simpler and easier to understand the complex nature and interdependencies of your organisation.

- **Faster time to market:** A complete overview of the EA allows adapting and extending IT systems to manage new business demands seamlessly.

- **Firm basis for interfacing management tasks:** The EA inventory creates additional value for your organisation by making it reusable for other management tasks in the organisation.

- **Reduced risk:** Risks associated with architecture artefacts can be identified and evaluated with less effort and time commitments.
4.2.2 Complementary Scenarios

As shown in figure 48, the EA inventory builds a valuable basis for almost all other EAM scenarios. Therefore, organisations usually set up and implement an EA inventory at the beginning of their initiative as the first scenario. Architecture principles (see chapter 4.1 ‘Architecture Principles’) provide a robust foundation that can be used for any EA programme. They promote guided EA decisions on both enterprise-level and solution-level and usually should be considered for defining the structure of the EA inventory.

4.2.3 Stakeholders and Their Role

Depending on the focus and the required information provided by the EA inventory, different roles and stakeholders need to be involved. Typical roles are enterprise architects, application owners and technology architects.
4.2.4 Meta-Model

Figure 50 illustrates a typical meta-model serving as the starting point for an EA inventory. It represents an excerpt of ADOit’s standard meta-model.

Keep in mind that the ADOit standard meta-model does not necessarily cover all the required concepts or relations out-of-the-box; it can be enhanced easily in the ADOit Administration Toolkit’s ‘Meta-Modelling Component’.
4.2.5 Procedure

The following steps guide you through creating an architecture inventory, and suggest ways to maintain and assess your data:

Figure 51
EA Inventory Procedure

Step 1 – Define scope and set objectives

Determine the organisational coverage and abstraction level of the EA. While defining the organisational coverage and appropriate level of detail of the EA – which certainly depends on what information you would like to gather - it is imperative to define a consistent and equal level of what you would like to model in each business domain of your organisation. If you miss important information in your documentation, you might not be able to use your EA effectively. On the other hand, if you include unnecessary information, it might not be feasible to effectively maintain and assess your EA inventory due to time and resource restrictions.
Remember, that the continuous maintenance of the EA data might cause considerable efforts. In order to prevent a waste of effort, it is crucial that you look at the inventory and ask yourself a number of questions. Based on the answers, you will be able to tailor the meta-model to your needs. In the following you will find sample questions that might be useful:

• What information is necessary for supporting the business?
• What applications are necessary for providing information?
• What technologies are essential for implementing and operating the applications?
• Which applications/business services are used by which organisational units?
• Which business services are available and by which applications are they provided?
• What business processes are affected if an application is not available?
• Who is responsible for a certain application/business service?
• Which interfaces are provided and used by an application?

Let us assume that the following question is on your list: ‘Which business processes use certain applications?’

In order to answer this question, the required artefacts need to be linked to each other within the meta-model. Since both of the artefacts ‘business process’ and ‘application’ are already part of the ADOit Standard Meta-Model, no further tailoring is required.

It is recommended to write a modelling guideline. Describe which concepts, attributes, and relations you need to populate the EA repository.

The meta-model in focus serves as the basis for estimating the effort, which you have to put into maintaining the EA inventory. It simply means that based on the meta-model (on which you focus) you need to estimate how many objects, how many relations between objects are required, and how long it takes to catalogue and maintain these artefacts on a continuous basis. It is important to communicate this information (effort, time, etc.) to the various stakeholders who are responsible for gathering and maintaining data. Make sure that the required capacities are available.
Usually a pilot project is a good way to get started. It is important to define the scope of the pilot, and usually an incremental approach is appropriate. For example, the EAM team might start the EAM initiative with a limited scope by choosing a certain geographical scope or business domain and later expand the initiative to other areas. This provides an opportunity for the team to gain experience on a more limited scale before implementation is expanded to other areas. The pilot implementation also provides the opportunity to align goals and expectations inside your organisation.

Step 2 – Choose a maintenance strategy for each type of architecture artefact

It is advisable to maintain central catalogues on architecture artefacts. Central artefacts represent lists of architecture artefacts (such as organisational units, technologies etc.) that are shared throughout your business domains. This applies especially to either architecture artefacts within the organisation that cannot be structured easily without the required know-how or artefacts that need to be governed centrally. Consider the ADOit Standard Meta-Model depicted in chapter 4.2.4. A common example is the catalogue on technologies – the so-called technology portfolio. Technologies are often used cross-domain. Therefore, technologies are usually maintained centrally by a team of technology architects.

An example of decentralised maintenance of architecture artefacts is the maintenance of applications. The reason why applications are typically maintained in a decentralised way is simple. Usually, each application owner has profound knowledge regarding his applications, which makes it easier for him to gather information on his applications. The application owner enters the required information on his applications into the tool and links his applications to other artefacts, such as business processes which are supported by his applications, or technologies that are used by his applications.

Step 3 – Set up the folder structure in the object catalogue

Now, in step 3, it is important to set up an appropriate structure for the object catalogue. The tree structure has to be readily comprehensible, and should consider both centralised and decentralised architecture artefacts. Take your time and do not rush into the next step. Make sure that your object structure is built according to the requirements. Otherwise, you will encounter difficulties in the upcoming step.

Mind the gap!

Take your time to build up a proper folder structure
Figure 52 shows an exemplary object catalogue compatible with the ADO\textit{it} Standard Meta-Model (discussed in chapter 4.2.4.)

The Administration Toolkit provides a function for loading user data (name, etc.) from directory services, such as Active Directory. The loaded users can be shared with the ADO\textit{it} object catalogue, so that you can assign the users to different architecture artefacts.

**Step 4 – Define access rights**

Having a clearly structured and comprehensible folder structure in the object catalogue is of no use if you are not aware of who is permitted to access and modify the different folders and objects. For this purpose, it is important to define access rights.
Within the Administration Toolkit of ADOit, user-based access to the object folders can be granted, as shown in figure 53. For each role you can define User Groups, such as application owner. Inside this group, you will find all stakeholders which have been assigned to the role application owner. You can decide whether application owners are only permitted to read specific objects, have write access to certain architecture artefacts, or have the capability to read and write to all architecture artefacts within a folder. The application owners are typically responsible for the decentralised maintenance of their applications and interfaces.

Furthermore, you might have a user group called ‘Process Architects’ or ‘Process Owner’, who are responsible for the maintenance of process maps. They can use the integrated BOC Management Office Interface in order to replicate process information maintained in ADONIS (see chapter 4.7 ‘Integrating EAM and BPM’).

If required, you can define specific access rights for single architecture artefacts by making use of the following concept: Users can be assigned to various architecture artefacts. The user simply has to open the notebook of an architecture artefact, for example an application, and add a user to the field ‘Responsible Person’. In doing so, the assigned user receives write access to the architecture artefact.
Step 5 – Gather and maintain data

There are four main ways to gather and maintain data: Using the Rich Client for centralised data acquisition and maintenance, using the Web Client for decentralised data acquisition and maintenance, using the ADOit Excel interface, and using the ADONIS interface for exchanging data e.g. process maps with ADOit. In order to understand the difference between using the Rich and Web Client, concentrate your attention on the following example:

As already mentioned, different data is maintained either in a centralised or decentralised way. The same applies to data acquisition. Let us assume that different technologies are used by different applications within your organisation, and there are only few people who have the necessary knowledge to maintain these technologies. In this case, the technology team, typically consisting of technology architects, is responsible for gathering and maintaining data on technologies in a centralised way. They start feeding the ADOit Rich Client with data (see chapter 4.4 ‘Technology Portfolio Management’ for details).

On the other hand, there are stakeholders in your organisation who are responsible for different applications. They should gather the required data on their applications by using the ADOit Web Client which you can easily access via the web. They probably will assign a number of technologies, which have been added earlier via the Rich Client to the object catalogue, to their applications. So, if there is data that is used cross-departmental, it probably would be helpful to gather the data in a centralised way in the Rich Client. Whether to use Rich or Web Client for the data acquisition and maintenance needs to be decided for each case.

Make sure to import your existing data to the tool. Depending on the purpose, you can use either the BOC Management Office interface or the Excel interface. In the Administration Toolkit you can configure both interfaces as needed.

For further information on this subject, please refer to chapter 4.7 ‘Integrating EAM and BPM’.

Step 6 – Govern data quality

Now that you have a repository filled with data, it is important to ensure the completeness and currency of your data on a continuous basis. To do so, ADOit provides you with different options.

On the start page of the Web Client, stakeholders can usually see all architecture artefacts for which they are responsible. Next to the architecture artefacts you can see lights with the same colour code as traffic lights (see figure 54). For example, when the light turns red, it simply means that the information on this specific architecture artefact might not be up to date. You can decide when the light should turn red. Configure the time interval in the Administration Toolkit.
For instance, you could set it to 60 days, if you wish the light to turn red after 60 days. Subsequently, the yellow light will appear two weeks before the red one. You can configure the time intervals according to your needs.

If you want to make sure that the stakeholders notice the red light, you can use the email notification system, which automatically sends emails to the stakeholders as soon as the status of an architecture artefact, and thus the colour of the light changes.

Moreover, the administrator or enterprise architect can see in the Web Client whose objects are not up-to-date. On his start page, all architecture artefacts with red lights are listed (see figure 55). Basically, he could go straight to the stakeholders, with a printed report that shows the currency of their data. Also, you could use the notification system that automatically generates reports on non-current data and send it, for example, to the domain/enterprise architects.

**Step 7 - Generate reports and views on the EA on demand**

Now that you have an inventory filled with current data, you can analyse and assess your EA to find weak spots in your architecture or risks that are seriously affecting your business.

In the modelling toolkit (of the Rich Client), you can create models manually as shown in figure 56. Easily create a new model and visualise dependencies of architecture artefacts. By using the drag-and-drop function, you can fill your model with architecture artefacts listed in the object catalogue. Besides that, you can also create new architecture artefacts directly in the model.
Furthermore, you have the opportunity to generate reports and views in the Rich and Web Client. Depending on the purpose, you can choose between a variety of reports and viewpoints: Dependency Analysis, Matrix View, Interface Report and GANTT Chart (see figure 57).
4.3 Application Portfolio Management

4.3.1 Objectives

Application Portfolio Management allows you to rationalise and justify your application architecture by considering each application as an investment. It provides an understanding of the business value of each application compared to the application’s investment, maintenance and operating costs.

As new applications are added to the application portfolio, organisations often fail to decommission existing, underutilised, and redundant applications. Statistics show that a large majority of the IT budget is spent on operating and supporting existing applications. Application Portfolio Management will enable you to reduce operating costs and set budgets free, allowing investments to be spent on the enhancement of business support and technological quality of the entire application portfolio.

A carefully managed application portfolio will enable you to:

- **Reduce risks** that come along with high maintenance costs by eliminating partially and entirely redundant applications,
- **Avoid the misalignment** of IT with business strategies and business processes by quantifying the condition of applications in terms of stability, quality, and maintainability,
- **Quantify the business value** and the relative importance of each application to the business,
- **Invest directly** in areas that provide the most value to your organisation.

In summary, it can be stated that resource allocation based on informed decision-making will be possible according to the applications’ condition and importance in the context of business priorities. It will become clear which applications require investment as a matter of course, and where there is potential for cost savings.

4.3.2 Complementary Scenarios

In order to make the most of the evaluations conducted with Application Portfolio Management, it is recommended to use the information that has been gathered in advance in various scenarios, such as ‘Populate the EA Repository’. It has to be said that Application Portfolio Management does not necessarily need input from other scenarios. However, it is stringently required to at least gather information on applications.
Once you have implemented the Application Portfolio Management scenario, you can use the evaluated information as an input for other scenarios, such as Project Portfolio Management and Master Planning.

4.3.3 Stakeholders and Their Role

Application Portfolio Management requires the cooperation of different stakeholders in order to be successful. Figure 59 gives an overview of the roles and stakeholders that are typically involved.
Usually the enterprise/domain architect is responsible for the application portfolio. Application Portfolio Management requires different types of information which are provided by subject-matter experts from different organisational units. The application owners, for example, provide the enterprise/domain architect with the inventory of all applications. The technology architect evaluates the technical details of the applications, whereas the business customer supports the scenario by assessing the functional quality of the applications. The enterprise/domain architect is also supported by the business management which provides input to define the strategic value of the applications.

### 4.3.4 Meta-Model

As described earlier, the ADOit Meta-Model comprises various levels of architectures. In Application Portfolio Management the focus shifts mainly to the application architecture. You have to be aware of the technical and functional quality of the applications, and consider the strategic value that an application generates. In order to ensure the proper evaluation and classification of applications the mentioned measures must be collected for each application.

*Figure 60*  
ADOit Meta-Model - Focusing on Application Portfolio Management
4.3.5 Procedure

The following pages guide you through the recommended steps to create an application portfolio and score the applications of your organisation.

**Step 1 - Define scope and set objectives**

Application Portfolio Management is most effective when you take the entire application architecture into account. However, depending on the size of your organisation, you might set the focus to single business domains in a first step. In this case, data collection could be based on the results of Business Capability Management (see whitepaper ‘Business Capability Management’, http://www.boc-group.com) where business domains with weak spots might already be discovered in advance. The applications assigned to these domains represent the portfolio that needs to be analysed.
Step 2 - Build and maintain application inventory

Depending on whether your existing ADOit repository already provides a base on which you can build your Application Portfolio Management initiative, you either need to extensively model your portfolio or you can just build on the existing information. The main architecture artefacts that you should score first are certainly the applications.

Catalogue all applications of the chosen domains in ADOit as described in the following section (for a more detailed discussion on how to gather and maintain architecture artefacts, refer to chapter 4.2 ‘Populate the EA Repository’):

1. The enterprise architect defines the correct structure in the object catalogue. It is recommended to structure the applications according to their business domains, as shown in figure 62.

2. Utilising the ADOit Rich Client, the enterprise architect/domain architect creates applications and assigns an application owner to each application.

3. The application owners are responsible for their applications and should describe their applications’ business functions and further attributes in detail. Depending on the specific objectives defined in step 1 it needs to be agreed which attributes need to be filled with information. For instance, you may want to cut down operating costs as a major objective. This could lead you to assign the technologies the application uses or is composed of to each application. If the evaluation of business support plays a major role, it would be helpful to assign business processes, business functions, or organisational units, in order to support the upcoming evaluations.

![Figure 62: Structure of the Object Catalogue](image-url)
Step 3 – Conduct health-rating on applications

You can score applications in different ways. A critical success factor in scoring your applications is to conduct a ‘normalised’ comparison. It is generally best if a small architecture team (usually led by the enterprise/domain architect) conduct the assessment sessions to ensure a consistent scoring. The scoring results are documented within the respective notebook of each application (chapter ‘Classification’ of applications in the Standard Meta-Model; figure 63 depicts the notebook of an application).

Figure 63
Notebook of an Application

Here we discuss typical measures for scoring your applications according to the ADOit best practices. The measures may be adapted to your individual goals.

Due to the meta-modelling capabilities of ADOit further measures can be defined easily.

1. Define strategic value: Strategic value defines how well an application is aligned with your organisation’s business strategy. Typically you first assess how valuable certain ‘Business Functions’ are to your organisation: Based on this, you derive how critical the supporting applications are in performing those business functions. The ratings (High, Medium, or Low) are assigned
by the enterprise architect. The rating ‘High’ indicates that the application provides your organisation with competitive advantage. Applications of this type are crucial for your business as they usually support your organisation’s strategies and objectives as well as its business-critical processes. ‘Low’ indicates that the application is not critical to the business. Low-rated applications usually support standard business services with no potential to stand out in the market.

In order to classify an application according to its strategic value it is helpful, but not compulsory, to assign the provided business functions to each application. Business functions will not only enable you to derive strategic value, but can also be used to identify applications that provide redundant business functions. Having applications and business functions related within your EA inventory can make running queries easier (e.g. predefined query ‘Business Functions provided by multiple Applications’) and aid with retrieving business functions which might redundantly be provided by or implemented in different applications.

Figure 64 illustrates the notebook of an application depicting its business functions. Analysing the business processes supported by the application also contributes to having more reliable information.

A dependency analysis (viewpoint ‘Business Impact Analysis’) on related business processes and business functions represents a good starting point for defining the strategic value of an application (see figure 65 for an example).
2. Define technical quality: Technical quality indicates how well the application is aligned with the technical road map of your organisation. Determinants to assess an application’s technical quality are performance (the ability to process effectively), reliability (the ability to meet the expectations regarding the application’s stability) and maintainability (the ability to support the application in a cost effective way), plus its ability to interact and be integrated with other technical components. The application is checked against these determinants in order to define its technical quality. This assessment is typically performed in meetings with members of your operations staff and your technical enterprise architect.

In order to classify the application in terms of its technical quality it is helpful, but not compulsory, to assign underlying technologies to the respective application. For example, compliance with your organisation’s technology standards could be one important indicator for defining technical quality. For this purpose, choose all assigned technologies and create a GANTT chart depicting applications, technologies, and their life cycles (see figure 66). Another indicator is whether the application uses technology packages. The prerequisite is to have at least parts of the technology road map defined in advance (see best practice scenario ‘Technology Portfolio Management’ in chapter 4.4).
3. **Define functional quality**: Refers to the level at which an application supports the business objectives in terms of functionality. The following factors are relevant: Functional coverage, usability, and adaptability to business changes. In order to assess the functional quality ideally key users (so-called super users) are consulted.

To classify the application according to its functional quality, it might be helpful to generate reports, as depicted in figure 67. It illustrates the result of the dependency analysis ‘Business Demands – Application – Business Functions’. Open business demands usually indicate functional gaps (see best practice scenario ‘Demand and Project Portfolio Management’ in chapter 4.5).
Step 4 - Define investment strategy of applications

In the previous steps, you identified and catalogued relevant applications. Also, you classified your applications according to strategic value, technical and functional quality. In this step, you should define the investment strategy for each of your applications. The investment strategy determines which applications need further investment, which applications must be replaced, and which new applications must be introduced to the landscape either by developing or buying applications. Based on the assessment results, you can make informed decisions regarding the future use of a particular application:

1. In order to get an overview of the applications, the enterprise/domain architect generates a portfolio view (bubble chart) including the relevant applications. For example, figure 68 depicts technical quality on the x-axis, and functional quality on the y-axis. Strategic value is represented by the size of the bubbles.
2. This combined view on functional quality, technical quality, and strategic value allows the enterprise/domain architect to define the investment strategy for each application. The main objective is to have the applications (at least those of strategic importance) residing in the upper right-hand corner of the portfolio view.

Consider cutting down the number of low-performing applications. Applications, which are displayed in the lower left-hand corner and are of low strategic value and are good bets for being decommissioned. Retiring these applications (and if required, integrating their business functions into other applications) will usually not lead to considerable efforts. Instead, it will set free budget to invest into applications, which are strategically important, enhancing their technical and/or functional quality as required. For example, in figure 68, the application ‘Customer Relationship Management System’ is located in the lower right-hand corner. It is of high strategic importance, but is rated low in functional quality. This implies that your organisation urgently needs to invest in this application to improve its quality. ‘Financing Platform’ is an application located in the lower left-hand corner. Its strategic value is low compared to other applications. Hence, this application is a good candidate for replacement.
Based on the attribute ‘Investment Strategy’, you can classify applications into the following categories:

- **Retire**: The application will be decommissioned. Its business functionality will be migrated or consolidated as required. Retiring the application may imply replacement by another application.

- **Maintain**: Ensure the application is maintained effectively and efficiently in order to guarantee compliance with the technology road map.

- **Invest**: Enhance functionality to hold the balance between technical and functional quality. In case of standard software you should acquire the newest available version of the application.

At this point you have a very good idea of where you are and what you are trying to achieve. In the next step, you should determine how you can get there. The output of the conducted assessment represents a strategic road map for Project Portfolio Management and transformation/migration activities.

**Step 5 – Define road map and optimise application portfolio**

Based on the results of the previous steps, you can define the transition road map of your application architecture. Evaluate costs and benefit from developing and evaluating project proposals and business cases for application renewal, retirements, or replacements. Determine priorities for conducting the transformation of your application portfolio.

For detailed planning of the target architecture and the required transition architectures refer to the best practice scenario ‘Master Planning’ in chapter 4.6.

Moving forward, it is vital that the application portfolio is reassessed on a regular basis. This will ensure continuous support of the changing business priorities. Reassessment must also take place after major business changes such as merger or acquisition.
4.4 Technology Portfolio Management

4.4.1 Objectives

Applications, application modules and devices are composed of technologies. Technologies are architecture artefacts all of which represent the technology architecture of the organisation. The number of used technologies in your organisation often depends on how many systems (e.g. applications) you are currently running. Whether you have developed your applications on your own or bought them from third party suppliers, the number of the used technologies in your organisation will increase. If you do not control the number of technologies in use, you will be instantly surrounded by a vast and fairly vague landscape of technologies, leading to the weakening of your architecture together with high maintenance costs.

Technology Portfolio Management helps to decrease the number of technologies used in your organisation, and thus prevents operating expenses skyrocketing, and adds to the maintainability of your applications. You will be able to recognise weak spots faster, and prevent your technology architecture from falling apart.

The overall goal of Technology Portfolio Management is to define an organisation-wide standard for technologies. Technology standards need to be implemented based on a technology road map, which must be aligned with Migration Planning and Release Management.

In summary it can be said that Technology Portfolio Management enables you to:

- Reduce the number of technologies used in your organisation,
- Provide a comprehensible view on your technology landscape,
- Cut down operating costs, and
- Increase maintainability of your systems (e.g. applications and technology infrastructure).

4.4.2 Complementary Scenarios

Technology Portfolio Management is the bedrock for the IT infrastructure that is available. Almost all EAM scenarios are affected by the results of Technology Portfolio Management. In particular, Application Portfolio Management benefits from this scenario, since applications are built on technologies and technology packages. Besides that, Demand Management also plays an important role in changing the IT architecture, for the simple reason that once you have analysed the technology architecture, you may want to state demands regarding the existing technology portfolio.
The main inputs for Technology Portfolio Management are market trends and strategic decisions, which also have a significant impact on the IT architecture (e.g. decisions regarding preferred suppliers).

### 4.4.3 Stakeholders and Their Role

In the following, you will find recommended roles for the stakeholders involved in Technology Portfolio Management in your organisation. Figure 70 gives an overview.

**Figure 69**

*Important Scenarios Interfacing with Technology Portfolio Management*

**Figure 70**

*Stakeholders and Their Role*

- **Application Owner**
  - Maps applications with technologies and technology packages

- **Infrastructure Architect**
  - Sets up, evaluates and enhances the technology portfolio

- **Security Officer**
  - Supports the infrastructure architect regarding security issues

- **Head of the IT Developers**
  - Ensures that IT developers meet the standard requirements

- **Head of IT Operations**
  - Ensures that IT operators meet the standard requirements
The infrastructure architect sets up, evaluates and enhances the technology portfolio. Within this portfolio, technology standards and perhaps – as a stage of expansion – technology packages (well-defined and approved packages of coherent technologies) are developed based on a defined technology road map.

The security officer supports the infrastructure architect regarding security issues. Due to the fact that the technology architecture is the basis for the entire IT architecture you have to make sure that applications are built on secure technologies. The technology road map is usually validated by the security officer.

The head of IT operations and head of IT development have to ensure that the skill level of the IT operators and developers is adequate to use and operate the technologies. The IT operators and developers should share their experiences and know-how regarding technologies with the infrastructure architect in order to define a proper portfolio.

The application owners have to map their applications with the technologies and technology packages of the technology portfolio. They support the infrastructure architect in setting up a complete, explicit and comprehensible as-is technology portfolio.

4.4.4 Meta-Model
This scenario focuses mainly on the technologies and technology packages in the meta-model. Technologies are combined to create technology packages. Applications and interfaces (but also application modules and business services) use and are deployed on certain technologies or technology packages. Technologies and technology packages need to be evaluated, e.g. evaluating the degree of standardisation. It is also essential to determine the life cycle dates of technologies and technology packages.

Figure 71
ADOit
Meta-Model
– Focusing on Technology Portfolio Management
4.4.5 Procedure

The following steps show you how to conduct Technology Portfolio Management efficiently:

**Figure 72**
Technology Portfolio Management

---

Step 1 – Define scope and set objectives

First, make sure that your inventory, particularly the application architecture inventory, is populated with sufficient and useful data (see chapter 4.2 ‘Populate the EA Repository’). Then prepare the organisational tasks. Define appropriate roles. Assign roles to the stakeholders of your organisation who are involved in Technology Portfolio Management (see chapter 4.4.3). They define and set objectives. Make clear what you want to achieve with Technology Portfolio Management and how you want to proceed in this matter. Communicate your goals to the stakeholders. Explain the entire process and make sure to mention the benefits of this scenario. If the stakeholders do not understand your intention, you probably will not achieve your goals.
It is also crucial to focus on architecture principles (see best practice scenario in chapter 4.1). Check if you already have defined principles regarding your technology architecture in your organisation. If this is not the case, you could either refer to TOGAF 9 for predefined principles or define your own. For example, according to TOGAF’s ‘principle of interoperability’ software and hardware must conform to defined standards that promote interoperability for data, applications, and technology. Another architecture principle that may be useful is ‘optimising the number of suppliers’. On the one hand, if you buy many products or licences from the same vendor, you might strengthen your negotiation power regarding prices and support. On the other hand, in order to reduce risks, avoid becoming dependent on one supplier.

**Step 2 – Set up current technology architecture**

In this step, you start with acquiring the available and used technologies and technology packages of your organisation. Use the results from chapter 4.2 ‘Populate the EA Repository’, if appropriate. As you collected information to populate your application architecture inventory, you might have gathered information on technologies too.

In order to create your technology artefacts in the ADOit object catalogue, first define an appropriate structure. For example, you can structure your technologies utilising recommendations from TOGAF’s Technical Reference Model (TRM) or other TRMs, e.g. the Federal Enterprise Architecture (FEA, see http://www.whitehouse.gov/omb/e-gov/fea/) another prominent EA framework. On the following pages you will find a sample structure for your technologies geared to the mentioned frameworks.

**Basic Software Tools:** Basic software provides general-purpose business functionality. Examples are standard web browser, flash player, and WinZip.

**Data Interchange:** Data interchange services provide specialised support for the interchange of information between applications and the external environment. These services are designed to handle data interchange between applications on the same platform and applications on different (heterogeneous) platforms. Data Interchange is concerned with the sending of data over a communications network and the definition of data communicated from one application to another. Data Interchange provides a common communication denominator between disparate applications/application modules. Examples of data interchange standards are SOAP, XML, ODBC, OLE, DAO, and DB2 connector.

**Data Management** refers to database and storage technologies and so to say, a collection of programmes which enable storage, modification, and extraction of information from a database, and various techniques and devices for storing large amounts of data. Typical database management systems are Oracle, SQL Server and DB2.
**Operating Systems:** Operating system services are responsible for the management of platform resources, including the processor, memory, files, and input and output. They generally shield applications from the implementation details of the machine. A prominent example is Microsoft’s Windows.

**Software Engineering** covers technologies associated with building software systems. Examples are integrated development environments, programming languages, graphical user interface building, scripting languages, software libraries and run-time environments.

**Delivery Servers** are front-end platforms which provide information to a requesting application. Subcategories are web servers, media servers, application servers and portal servers.

**Security Technologies** are required to protect sensitive EA information. Examples are identification and authentication services, system entry control services, access control services, trusted recovery, and encryption services.

**System and Network Services:** Information systems comprise a large variety of diverse resources that must be managed effectively to achieve the goals of an open system environment. Examples are User Management, Print Management, and software installation services.

**(Hardware) Device Types:** Hardware and infrastructure elements, such as (virtual) servers, server cluster, computers, embedded technology devices, and peripherals.

Figure 73 depicts the ADOit object catalogue containing the recommended folder structure for the technology portfolio.
Application owners and IT operators usually deliver the required input for technologies. It is also important to identify and document the life cycle of the technologies including the vendors’ end-of-support date.

**Step 3 – Evaluate the current technology architecture**

Now that you have populated your repository with data on your technologies, and defined useful principles, evaluate your current technology architecture against the defined architecture principles. Furthermore, check your technology portfolio against current market trends to see whether your technologies are up to date. In ADOit, you can use a matrix view for this purpose. An example is shown in figure 74. It depicts to which degree technologies fulfil the defined architecture principles ‘Control Technical Diversity’ and ‘Interoperability’ (both samples are taken from TOGAF). For further information on architecture principles, please refer to chapter 4.1.

![Matrix View with Principles and Technologies](image)

Based on your evaluation and further parameters, such as quality of support, internal know-how etc., define the road map status of your technologies. In other words, decide whether you want to invest or disinvest in certain technologies. Figure 75 depicts the notebook of a technology and the attribute ‘Technology Road Map Status’.
In addition, you can define the required technology packages. Technology packages represent a suite of technologies to provide infrastructure that supports the delivery of applications. Typically the combination of technologies which a technology package consists of has either been proven to be compliant with the trend of the market or with your projects.

For example, take environments for web applications as candidates for technology packages: XAMPP is an open source technology package for web applications. Among other technologies, it is composed of MySQL (a database management system), and Apache Tomcat (a web server). XAMPP might be one of the technology packages in your technology portfolio. Alternatively you might have a commercial solution in your technology portfolio composed of SQL Server as the database management system and Microsoft’s Internet Information Server as the web server. Both technology packages represent a solid basis for implementing and running web applications.

From what has just been said, the following architecture principles could emerge: Ideally, solution engineers only use specific technology packages to prevent using technologies that have not been tested or approved yet. Another architecture principle could be ‘Primacy of open source technology packages’ in order to save costs. In this case solution engineers at least need to justify their decision if they use technology packages based on commercial technologies instead of available open source solutions.
These architecture principles may also affect the scenario Application Portfolio Management (see chapter 4.3 ‘Application Portfolio Management’).

**Step 4 – Define and analyse the target technology architecture**

In this step, you define and evaluate your target state technology architecture. Use the results of the previous steps as input. According to your objectives and the results from analysing your current state architecture, you may want to introduce new technologies to the technology portfolio. It is important to involve the IT operation and development staff to gather information on applications that are being planned or developed.

Also identify redundant or out-of-date technologies which you may want to phase out from your technology portfolio.

In ADOit, you can create a GANTT chart using the application and technology life cycle dates as parameters to identify out-of-date technologies and see how those technologies affect your applications (see figure 76).

![Figure 76: GANTT Chart with Applications and Technologies (Time-Related View)](image)

Subsequently, you need to define or redefine the road map status for each of your technologies. Use the instructions of step 3 for completing step 4.

**Step 5 – Perform gap analysis and evaluate impact on the entire architecture**

Once you have created both the current and target state architecture of your technologies, perform a gap analysis to identify possible issues. During this analysis, you should also score applications and interfaces that will be affected by changes made to your technology architecture. For this purpose, you can conduct an Ad hoc query in ADOit that shows you all technologies with the road map status ‘disinvest’ (see figure 77 and figure 78). Now that you know which technologies you would like to retire in the future, you can generate a dependen-
cy analysis, where you can see which applications and interfaces will be affected when retiring technologies (see figure 79). You can also repeat this instruction for new and modified technologies.

**Figure 77**
Ad Hoc Query on Technologies in a Certain State

**Figure 78**
Query Results on Technologies in a Certain State

**Figure 79**
Dependency Analysis (Applications and Used Technologies)
It is important to discuss the next steps and migration strategies of the applications and interfaces with the respective application owner to agree on a migration plan. For example, if you plan to use a new version of Java Runtime Environment in the future, you might discuss with the application owners responsible for the affected applications whether their applications are compatible with this new version.

**Step 6 – Define road map**

In this step, you define the transition road map that leads the way for the migration from your current to your future state technology architecture. The road map shows you what projects you need to conduct to perform the migration (see the upcoming best practice scenario ‘Demand and Project Portfolio Management’ in chapter 4.5). For further information on how to define and plan transition and target architectures refer to the best practice scenario ‘Master Planning’ in chapter 4.6.

### 4.5 Demand and Project Portfolio Management

#### 4.5.1 Objectives

Demand Management refers not only to IT demands but also to business demands. It covers all demands within the entire EA. It helps you understand the business and IT needs of your organisation, and enables you to plan the capacities within the business and IT department. It also aims at prioritising, evaluating and realising new or changed demands within well-defined projects.

In this context, IT acts as a service provider that focuses on satisfying customer requirements. As requirements arise almost every single day, IT is likely to become just an ‘order taker’ disconnected from the business world and not understanding what lies behind customer demands. To bridge this gap, the IT department of your organisation has to capture demands as early as possible. Ongoing dialogues with customers need to be set up, for IT to become a strategic partner for customers. Consequently, enabling process improvement and innovation on both the IT and business side is a major goal. Based on these organisational circumstances, IT has the opportunity to build up a demand pipeline, evaluate demands regarding different aspects, i.e. the impact on resource allocation, and set up well-defined projects (proposals) which help realise the requested business needs as accurately as possible.
In summary, it can be said that setting up IT Demand and Project Portfolio Management enables you to:

- Identify business needs as early as possible,
- Evaluate the impact of demands on the current EA and project portfolio,
- Identify demands requesting similar or already available functionalities,
- Bundle demands into project proposals,
- Assess possible costs and risks caused by demands,
- Prioritise demands,
- Realise demands by conducting the appropriate projects, and
- Evaluate projects and project proposals based on cost-benefit analyses.

### 4.5.2 Complementary Scenarios

IT Demand and Project Portfolio Management is related to almost every Enterprise Architecture Management scenario. Therefore, the results from each and every scenario can serve as input. Usually, Demand and Project Portfolio Management strongly interfaces with the scenario ‘Master Planning’, as both scenarios have an impact on each other.
4.5.3 Stakeholders and Their Role

In the following, you will find roles to which the stakeholders who are involved in Demand and Project Portfolio Management in your organisation are likely to be assigned:

The demand creator identifies new or changed business or IT needs and defines demands. People on the business side (e.g. key user or business management) as well as on the IT side (e.g. service/account manager, technology architects etc.) could be demand creators.

The demand manager evaluates and prioritises demands, and generates proposals for projects to be conducted in order to realise the raised demands. In many organisations, the domain/enterprise architect performs the function of the demand manager within his domain.

The project manager is responsible for planning and executing the approved projects. In the context of IT projects, project managers, usually together with a team of business analysts and technicians, are responsible for creating a detailed specification based on the stated demands, and for delivering proper IT solutions.

4.5.4 Meta-Model

This scenario focuses mainly on the demand and project part of the meta-model. Demands are made, and projects (or project proposals) emerge from the demands. It is important to set the priority, measure the risk and estimate the cost of demands. The same measures apply to projects. In the ADOit Meta-Model, demands and projects can be assigned to almost every type of architecture artefacts. Figure 82 gives an overview of concepts and relations in the limelight of Demand and Project Portfolio Management.
4.5.5 Procedure

The following steps show you how to manage demands and related projects and project proposals efficiently.
Step 1 - Define scope and set objectives
Before you start applying this scenario to your organisation, focus on a number of prerequisites. First, your ADOit repository must have been fed with current data on the EA in advance. If you missed this step, please refer to chapter 4.2 ‘Populate the EA Repository’. The scenario discusses how to acquire and maintain your EA inventory, which will act as a basis for Demand and Project Portfolio Management. Remember to ensure the completeness and currency of your EA data.

Second, a number of organisational tasks must be accomplished: You need to establish processes and roles for Demand and Project Portfolio Management. Create the predefined roles (see ‘Stakeholders and Their Role’) in the ADOit User Management Component of the Administration Toolkit, and assign the responsible stakeholders to the mentioned roles.

Before proceeding with step 2, please take your time, and give careful thought to how you would like to evaluate and prioritise your future demands. Define a number of measures which clearly state which key aspects you need to consider when assessing a demand. It is important that the business side of your organisation also agrees with the defined measures. You need to achieve mutual understanding regarding the entire process. In the following, you will find recommended measures which you might use to evaluate and prioritise your demands:

- Business priority
- Related risks
- High-level cost assumption
- High-level benefit assumption

In order to achieve the highest benefit possible, set up Demand and Project Portfolio Management for your entire organisation. However, it is also possible to begin within one business domain and extend the scope after gaining first experiences.

Step 2 - Make demands
In this step, an authorised person, e.g. a key user on the business side or an account manager on the IT side, makes a demand. The ADOit Web Client can be used by users who are assigned to the role demand creator. Figure 84 shows the role-specific interface for demand creators in the ADOit Web Client. It contains a list of all demands for which the user is responsible.
Demands are typically assigned to architecture artefacts on any architectural level. Take a look at the examples shown in figure 84:

- The demand ‘Required adaptation to regulations – Fill in and store Know Your Customer (KYC) Form’ might affect business processes and their supporting applications, whereas

- The demand ‘Make sure that orders are generated only using authorised terms and prices’ might only affect a certain application.

- The demand ‘Change the underlying platform of application ‘Partner System (PAS)’ might affect the application and its underlying technology packages.

In any case, in order to evaluate and prioritise demands, the demand manager should be able to answer the following questions:

- Who created the demand?
- What is requested in particular and why is it important for the business?
- What are the business’ priorities?
- When should the demand be realised?
- Who is responsible for the implementation of the demand?

Once you have completed and saved the demand, it is automatically assigned to the defined responsible person for further investigation. Depending on your organisational setup, the responsible domain/enterprise architect, the IT service owner (in terms of ITIL) or any other stakeholder who performs the role ‘demand manager’ might be the receiver.
The integrated email notification system in ADOit can be configured to send an automatically generated notification to the responsible user.

**Step 3 - Evaluate and analyse demands**

In this step, you should evaluate and analyse your demands. It is important to repeat this step on a regular basis. Depending on the size of your organisation, you might need more than one demand manager to conduct the assessments. The demand manager is responsible for the newly created demands (see step 2).

The demand manager conducts the assessments based on the previously defined measures (see step 1). Additionally, affected architecture elements and related demands need to be identified, as these parameters may influence the prioritisation of the demand. The demand manager will use portfolio diagrams (see figure 85) to communicate the outcome of the evaluation procedure, and determine priorities for demands. For instance, figure 85 shows the assessment of demands concerning associated risks (x-axis), business priority (y-axis) and costs (size of the bubbles). Please note that for an appropriate assessment of demands, you often have to provide business cases and detailed measures for each of the demands. Business analysts, technical consultants and application owners will usually support you in this matter.

![Figure 85 Demand Portfolio](image-url)
Step 4 - Define project proposals and assign demands

Based on demand priorities and implementation dates, the enterprise or domain architect integrates the demands into the overall IT road map. They have to check whether the new demands can be realised within existing, already ongoing projects or if new projects have to be set up.

For new projects, project proposals need to be defined. After a project proposal has been tendered, you need to assign your demands to the project (proposal) for your demands to be realised within the project. Based on this, the impact on the EA can be evaluated. A helpful view, supporting the evaluation of a certain demand, is a dependency analysis depicting demands and affected architecture artefacts. Figure 86 shows an example.

![Dependency Analysis: Demands and Affected Architecture Artefacts](image)

Step 5 – Monitor demands and Project Portfolio Management processes

Ongoing monitoring of the status and progress of the demands is required. By using the ADOit Web Client as shown in figure 84, the demand creators as well as the demand managers can track the demands which they have made or for which they are responsible.

As soon as demands are bundled into project proposals (see previous step), the project portfolio can be analysed. Based on similar measures used for evaluating demands, the project portfolio is assessed. A portfolio view like the one presented in figure 85 supports decision making. As soon as the project proposals are prioritised, project start and end dates can be defined. Another helpful view to examine the project portfolio is shown in figure 87. It pictures the projects (project proposals) on a time bar.
Furthermore, different reports or viewpoints, such as a dependency analysis (illustrating the dependencies between projects, demands and architecture artefacts) will support decision making. Figure 88 shows the dependencies between demands, projects and affected applications.

If overlaps, dependencies or shortages (of time or budget) are identified, the project portfolio needs to be reorganised.

Keep in mind that Demand and Project Portfolio Management will be most effective when integrated with the master planning best practice scenario (see chapter 4.6). The master planning scenario focuses on the current, target and transition states of your architecture. Besides different architecture artefacts, the business demands are also part of the target architecture and need to be considered when planning the target state of the EA. For further information, please refer to the upcoming chapter 4.6 ‘Master Planning’.
4.6 Master Planning

4.6.1 Objectives

When it comes to achieving strategic alignment and growth, as well as risk reduction and cost containment, many organisations recognise the value of master planning in supporting strategic IT planning. The most compelling reason, however, for designing a master plan is that other organisations are modernising their EA from day to day and thus are gaining competitive advantage.

Besides that, your organisation will benefit from

- Better alignment between business and IT,
- Increased agility, and
- Improved business continuity.

The resulting EA road map shows where the enterprise should stand in the future, and what changes you need to make to transform your current state architecture. Master planning is usually performed on a continuous basis. However, there are occasions where it is justified to perform master planning as a one-time initiative without considering implementing it on an ongoing basis. This type of master planning emerges, for example, if:

- Weak spots (on any level of the EA) have been identified during continuous analysis of the current state architecture and change would be beneficial,
- The need arises for setting up a programme that will change large parts of the business, application and/or technology architecture, or
- The need arises to integrate existing IT landscapes due to merger and acquisition.

However, these occasions often lead to a complete and proper implementation of master planning.

4.6.2 Complementary Scenarios

In order to develop your master plan in an effective way based on profound information, it is recommended to make use of the information that has been gathered in best practice scenarios, such as ‘Populate the EA Repository’ and ‘Demand and Project Portfolio Management’. However, like most of the best practice scenarios presented in this manual, master planning can be implemented independent of other scenarios.

Once you have gone through all necessary steps of the Master Planning scenario, you can use the resulting information for consecutive scenarios, such as ‘Project Portfolio Management’, and the detailed planning of solution architectures.
### 4.6.3 Stakeholders and Their Role

As master planning requires strong business commitment, it is required for all participants to go through a process of bidirectional education (‘learning from each other’) and joint participation (‘working together’) in strategic planning to achieve business and IT alignment. Master planning deals with mediation between business and IT to help match their demands, so that priorities and required initiatives can be mutually agreed on.
Based on identified requirements, the enterprise/domain architect designs possible target states of the Enterprise Architecture. The business management usually states future requirements and signs off the favoured target state of the architecture. No need to say that additional stakeholders, such as technical architect, programme management etc. need to be involved. Technical staff and solution engineers use the master plan to understand on a high-level basis what is planned to be implemented in the next planning phases.

### 4.6.4 Meta-Model

As with most things, there is not a simple answer to what should be part of the master plan. There is no standard template. Every organisation is proceeding differently, and the nature of the master plan changes depending on the organisation’s specific needs.

![Figure 91: ADOit Meta-Model - Focusing on Master Planning](image)

However, typical EA artefacts that are defined and analysed within a master plan are applications, business processes, locations, and organisational units. The meta-model depicted in figure 91 illustrates these architecture artefacts and their dependencies.
4.6.5 Procedure

Follow the steps below to design a master plan and corresponding road maps.

**Figure 92**

Master Planning Procedure

---

**Step 1 - Define scope and set objectives**

Master planning is most effective when the entire EA is considered. However, in large and medium-sized companies, it is often difficult to cover the entire organisation at once. It is simply too complex. In order to reduce complexity, the enterprise architect might, as a first step, set the focus on single business domains (specific areas of the organisation). In this case, the data elicitation of the relevant domains might be based on the results of the Business Capability Management initiative (see ADOit whitepaper ‘Business Capability Management’, www.boc-group.com) where low-performing business domains might have been
identified. Therefore, invest your efforts on areas from which the organisation will benefit most.

For the chosen architectural scope, the master plan typically serves the business by answering the following questions:

- What is possible and which options are available?
- When can it be delivered?
- How much does it cost/save and what are the related business risks?

From the viewpoint of IT Management the following questions are answered:

- Have we done this before?
- How do we get it done and what is generally possible?
- How do we ensure that it is done properly and in time?

Depending on your answer to the above questions the relevant architecture artefacts for master planning are chosen. In the example at hand important architecture artefacts are business processes, applications and organisational units. As has been stated in chapter 4.6.4 ‘Meta-Model’, other architecture artefacts might also be relevant.

Remember that you are planning the necessary transitions for a multi-year effort and not the implementation. The master plan defines key objectives and a high-level, prioritised approach to reach those objectives. Master plans need to contain enough detail to gain buy-in from the business; However, there is no need to develop an in-depth view on required resources or give detailed information about budget matters.

**Step 2 – Create master planning matrix for the current state architecture**

Based on business demands, and outcomes from the previously discussed best practice scenarios, the enterprise architect designs the target architecture – models of the future state outlining the required changes to the current state. In the example at hand (see figure 93) business processes, applications and organisational units are the key determinants of the master plan. The master plan is represented in the form of a master planning matrix. This type of matrix illustration is often recommended in EA frameworks, such as TOGAF, to plan the development of the EA from a strategic point of view. It illustrates the logical relationship between business processes, applications and organisational units and their planned life cycles.
In order to create a master planning matrix you need to perform the following steps:

1. Create a ‘Analysis Model’.

2. Add relevant business processes from the object catalogue to the x-axis of the modelling surface. If the business processes are not catalogued yet (see best practice scenario ‘Populate the EA Repository’ in chapter 4.2 and ‘Integrating EAM and BPM’ in chapter 4.7), you need to define those first. Keep in mind that it is sufficient to define the processes on a relatively abstract and strategic level.

3. Add corresponding organisational units to the y-axis.

4. The intersection of x-axis elements (business processes) and y-axis elements (organisational units) form a so-called business context. A business context is respectively defined by one business process and one organisational unit, and represents, simply spoken, a cell in the master planning matrix. Next, you need to assign the applications to their various business contexts.

5. Save the model to the model catalogue.

Figure 93 illustrates an example of a master planning matrix. On top, business processes are depicted in their logical order in which they are usually processed.
The y-axis shows the organisational units, such as ‘Branch’ and ‘Service Centre’ corresponding to the business processes. Within the intersections of business processes and organisational units (the so-called business context), applications, such as Partner Management (PAS) and Saving Account System (SAS) are placed. As you can see, the business processes ‘Product Opening’ and ‘Administration’ are supported by different applications.

**Step 3 – Create master planning matrix for the target state architecture**

Within this step the future state of the EA is defined. In the upcoming steps you will also need to evaluate the target architecture and define transition architectures to get from the current state to the future state architecture (as shown in figure 94).

The enterprise/domain architect performs the following steps to define the future state architecture:

1. Create a new version (copy) of the as-is architecture model as a starting basis.

2. Create the target state architecture by changing the assignment of the applications to their business contexts. Make sure to consider the input from various preceding EAM scenarios; for instance, the investment status of applications (see scenario ‘Application Portfolio Management’ in chapter 4.3), and business demands related to the applications (see scenario ‘Demand and Project Portfolio Management’ in chapter 4.5) in scope. Check the attribute ‘Action required’ in the notebook of applications to make sure that the previously identified ‘need for action’ is considered in your master plan.
This can easily be done by switching to the tabular view. For example, figure 95 shows applications including the attributes ‘Action Required’, ‘Assigned Business Demands’ and ‘Investment Strategy’.

3. Switch back to the graphical view. Based on this information, you need to change size and/or position of the applications in the graphic in order to change the way they are assigned to business processes and organisational units – respectively to their business context. Furthermore, you may need to create new applications. This can be done in the object catalogue or directly via drag and drop of an application via the modelling bar (see scenario ‘Populate the EA Repository’ in chapter 4.2). To visualise the required type of change the attribute ‘Planning State’ (see figure 96) needs to be set accordingly:

- Applications assigned to business contexts which remain unchanged keep the state ‘No Entry’.
- Applications newly assigned to a business context receive the planning state ‘Planned’. In this case, an already existing application is planned to be used in the particular business context.
- Applications which will be decommissioned (at least from their particular business context) receive the planning state ‘Retired’.

![Figure 95 Master Planning Matrix In Tabular View (Applications)](image1)

![Figure 96 Attribute ‘Planning State’](image2)
Please note that within a particular master planning matrix, the same applications might appear multiple times, but with different planning states. The reason for this is simple: It might be the case that you have been planning to introduce a certain application to a business context in the future while at the present time it is already part of another business context. Figure 97 gives an example: Application ‘Partner Management (PAS)’ has planning state ‘Productive’ in business context ‘Acquisition – Branch’ and at the same time planning state ‘Planned’ in the business context ‘Acquisition – Service Centre’.

Figure 97
A Particular Application with Different Planning States Depending on Business Context

Figure 98 shows the resulting target state architecture including the required changes that need to be made to the current architecture. The applications ‘Saving Account System (SAS)’, ‘Current Account System (CAS)’ and ‘Credit Card Order System (CCO)’ will be decommissioned in the ‘Branch’. They are replaced by the application ‘STANDARD Banking (SBA)’.
Step 4 – Evaluate the target architecture(s)

Sometimes, it is quite helpful to decision-making to define alternatives - so-called planning scenarios - to the proposed target architecture. For this purpose, you can create different models in ADOit, each of which shows a unique picture of your target architecture.

To choose between the alternatives and find an appropriate target state for the architecture, the following measures are commonly used:

- Costs (high-level cost assumptions),
- Benefits (high-level benefit assumptions), and
- Associated risks (business and technical).

The term technical risk refers to the level of risks associated with the development and implementation of a specific solution. Business risk refers to the level of risk of the business line when implementing a specific architecture.

Figure 98
Example of a Master Planning Matrix (To-Be Incl. Required Changes)
Usually there is no need to perform an accurate cost-benefit analysis at this point. The project portfolio scenario (see scenario ‘Demand and Project Portfolio Management’ in chapter 4.5) will go into more detail regarding budget and resources. The costs and benefits estimation during the master planning scenario only goes as far into detail as is necessary to determine the overall multi-year planning.

Use the master planning matrix (including the various alternatives) to discuss the target architecture with the business side and agree on one of the target scenarios. Once a mutual agreement has been reached, the repository needs to be populated with the data of the chosen target architecture. Therefore, the enterprise architect:

- Defines the new repository relations between applications, business processes and organisational units, and
- Sets time of validity (production and decommission dates) of the relations and applications.
Step 5 – Define transition architectures and road map

In order to successfully manage the transformation from the current to the target architecture, you need to introduce a road map. Road maps are usually based on the desired target architecture and the business priorities linked to it. They describe the steps (so-called transition architectures) that lead you to your target architecture. It is important to document the differences between the current and target state of your architecture. It will help you identify details that are required to complete the ongoing transformation and to get the consent of all stakeholders on the transition architectures. To define the transition architectures, simply follow the instructions in step 3.
You will identify gaps between the current and future state architecture. These gaps lead to transformation projects and programmes (see scenario ‘Demand and Project Portfolio Management’ in chapter 4.5 for details). Create, develop, and monitor a detailed implementation and migration plan. It should provide you with the necessary information on how to realise your transformation programme by working your way through the different transition architectures.

**Step 6 – Refine short-term architecture**

In some cases you need to develop an in-depth understanding of the requirements and arrange your master plan in more detail. Therefore, you may want to expand the scope and granularity of the existing models. For instance, you may need to analyse the interfaces between applications in your plans. That means that you need to consider interfaces every time you plan to integrate or retire an application. Figure 101 illustrates an example. The first illustration shows applications and interfaces of the current architecture. The second one shows applications and interfaces of the target architecture.

![Figure 101 Model of Current and Target Architecture (Incl. Interfaces)](image)

You can map the current and target architecture within one model in ADOit. Using the ‘time slider’ functionality current and target architectures and transition architectures can be shown.
4.7 Integrating EAM and BPM

4.7.1 Objectives

This scenario deals with ways to integrate EAM and BPM initiatives. This integration enables you on a tool level to share and reuse catalogues and views of both management domains. In this way, you avoid the repeated gathering of relevant information. More importantly, the integration of EAM and BPM opens the doors to reinforced communication between business and IT. As a result, business and IT departments will be provided the opportunity to share and agree on business process and IT architecture models. It paves the way for the establishment of a sound business/IT alignment and supports business and IT in making joint decisions.

Business process know-how plays an important role within the scope of EAM. Best practice scenarios discussed in this method manual that require having a catalogue of business processes are, for example, ‘Populate the EA Repository’ (see chapter 4.2), scenario ‘Application Portfolio Management’ (see chapter 4.3), and scenario ‘Master Planning’ (see chapter 4.6).

The primary benefits that arise from a formal definition of BPM/EAM integration within your organisation are as follows:

• Enhanced alignment of IT with business processes emerges. Assigning IT artefacts (mainly applications and business services) to business processes (but also to organisational structures, business strategies and other elements of the business architecture) and thus facilitating communication between business executives and IT representatives, helps to achieve business/IT alignment.

• As BPM and EAM endeavours strive for an efficient and economical way of working, the re-use of existing catalogues and diagrams is crucial.

• Mechanisms evolve that help predict impacts on the IT systems before introducing changes to any business process. On the other hand, these mechanisms help discover impacts on business processes that may emerge from changes made to IT.

In the following, you will find several guidelines and conventions that are useful for a pragmatic implementation of BPM/EAM integration. The main scenario discussed on the following pages includes how to set up the exchange of business process models and how to use these models as a solid basis for communication and agreement between business and IT.

Note: Although the best practices discussed on the following pages mainly focus on the integration of BPM and EAM, the main determinants for setting up the in-
tegration scenario and exchange or integration of EA information apply to other integration scenarios as well. Other examples are the integration of business and IT strategies, risk catalogues, and project portfolios. A brief but not exhaustive overview of touch points between the EAM domain and other management disciplines is given in chapter 1.2.3 ‘Interfacing and Influencing Management Endeavours’.

### 4.7.2 Complementary Scenarios

From an EAM point of view the BPM-EAM integration scenario builds a valuable basis for the scenarios ‘Populate the EA Repository’, ‘Application Portfolio Management’ and ‘Master Planning’. All of these scenarios typically involve process architectures for evaluations and planning procedures.

### 4.7.3 Stakeholders and Their Role

In the following, you will find the roles that the stakeholders who are involved in EAM-BPM integration of your organisation are likely to be assigned to. Figure 103 gives an overview of roles that are typically involved in this scenario.
As both business and IT management will profit from sharing the process architectures, typically enterprise/domain architects (often member of the IT department) and business management need to agree on joint initiatives. Business departments (or a staff function) usually own the process models and are responsible for the delivery of process architectures in an adequate quality. A member of the EAM team will integrate the delivered business processes into the EA inventory.

4.7.4 Meta-Model

As the scenario focuses on BPM-EAM integration ‘Business Processes’ play a central role. Typically business processes are structured in a hierarchical order. Architecture artefacts of the IT architecture like applications and business services are assigned to business processes.

Figure 104
ADOit Meta-Model - Focusing on BPM-EAM Integration
4.7.5 Procedure

Follow the steps below to implement the EAM-BPM integration.

**Step 1 - Define scope and set objectives**

Before you begin with the main integration activity, set objectives to avoid long discussions on the deliverables. Make sure that the objectives are clear and comprehensible. Establish a vision regarding the purpose of interchanging the architecture artefacts. Business processes are usually developed and maintained by different departments with different goals within an organisation. Hence, it is essential to adopt several guidelines and conventions regarding how the process of integrating EAM and BPM is performed to provide consistency in the models across the organisation.
Focus on the benefits that add value to both BPM and EAM stakeholders. It is important to start small. Choose an iterative approach for the integration and keep the initial scope comprehensible. Considering the fact that business processes address a vast landscape of the organisation, apply the initial scope only to those business processes that are of high priority to the integration initiative. Once you have completed this task, you can iteratively extend the scope of your initiative so that other systems are also considered. It is important to achieve quick wins. They will have a positive impact on the customer and the business.

Detailed process flows are usually not incorporated into the EA as such, i.e. including all details, activities and descriptions of roles. Usually no more than the process architecture (hierarchical catalogue on business processes) is necessary as a sound basis for any EA work. It is necessary to see the big picture and identify how the entire organisation should work together. The process architectures offer a bird’s-eye view of the activities of your organisation as a whole. By gaining an overall view of your organisation and by integrating processes with architecture artefacts of the IT architecture, it becomes possible for enterprise architects and business analysts to identify the strengths and weaknesses of your architectures. It will enable the architects to identify areas that need improvement, providing the opportunity to develop a strategy to exploit the full potential of the organisation by forming a common platform for business and IT stakeholders.

The process architecture may be refined by modelling process flow charts during implementation projects. An example from an IT point of view is that one could name the detailed analysis of business processes as part of the requirements engineering process. There are certainly many other reasons for a more detailed process mapping on the business side, including process-based work instructions, quality management processes etc.

For a detailed discussion on Process Management refer to the ADONIS Business Process Management Handbook by BOC.
Step 2 - Identify potential sources

Based on the objectives, the EAM team needs to examine the organisation in order to identify existing and suitable data sources and data suppliers. Organisational and/or business departments might be a good contact point for provision of process models. Often business process models are developed and maintained by different departments within the organisation. A high-level evaluation of the existing information needs to be performed to assess the profit potential of the integration. Main factors that should be considered are currency, accuracy, and internal consistency of the existing information.

Step 3 - Analyse the meta-models

In this step, the structure of the information sources – the meta-model – is analysed in detail. The meta-models of EAM (the ADOit Meta-Model) and BPM (the meta-model of ADONIS or any other process modelling tool) are being compared. This is the basis for the identification of shared concepts (concepts represented in both meta-models). If meta-models are not available explicitly or the maturity level regarding the structure of existing documentation is too low, you need to derive the meta-models from data sources in advance.

From an EAM view the business processes represent a central part of the business architecture. Business processes need to be assigned to architecture artefacts of the IT architecture forming a central part of the business architecture. Figure 106 gives an overview. The business process architectures (objects of type ‘business process’ in process/company maps) are usually identified as the architecture artefacts to be shared. However, you might also focus on business functions, or even detailed process flows.
Step 4 - Agree on deliverables

Based on the meta-models, the determinants discussed in this section need to be defined and agreed on for each deliverable and its associated architecture artefacts. As we put a strong focus on the reuse of business processes for further architectural work within the EAM initiative not all of these determinants are of equal importance. However, if you consider working on other integration scenarios, a number of these determinants might be of higher importance.

**Structure:** In the case of hierarchically structured deliverables, the structure of the deliverable plays an important role. For example, in process architectures business processes are usually structured in hierarchical order. Processes are composed of sub-processes. It is important to establish a number of conventions to provide a consistent structure of the business processes (across the organisation), especially if different groups (e.g. different departments of the organisation) are involved in the mapping of the processes. In short, a systematic procedure for the identification and classification of business processes and therefore for the creation of process maps is recommended. Figure 107 ideally illustrates determinants, such as order type and product, determining the structure for the process hierarchy. It is crucial that each level of process architectures is equally structured. For example, if the process map of business area A includes order type on the highest level, it is important that the process map of business area B also has order type on the highest level.
Organisational Coverage: The organisational coverage deals with missing or inconsistent information. As shown in figure 108 process architectures might not be available for all business areas. Therefore, you have to decide how to deal with missing information. In this case either the EAM team defines the missing process architectures (see best practice scenario ‘Populate the EA Repository’ in chapter 4.2) or the responsible business departments commit to provide the required information.
**Abstraction Level:** Depending on your objectives the level of abstraction will vary. The level of granularity of the deliverables needs to be determined according to the objectives. Finer-grained deliverables allow for closer management and measurement but require greater effort to integrate, maintain and govern the data. This of course holds true for the entire EA documentation. The Enterprise Architecture needs to be guided by scope and purpose. It should represent only those parts of the real world which relate to the goals, the scope and the purpose of the architectural work.

Therefore, the business processes typically do not have to be worked out in detail. Integration of high-level process architectures is sufficient for most of the EAM scenarios. More importantly, business and IT departments have to agree on process architectures and ensure the completeness and validity of the process architectures on a regular basis. If done properly, the process architecture forms a solid basis for joint planning.
Change Frequency: Architecture artefacts, such as business processes and applications, are subject to continuous change. However, the change frequency of the deliverables (models), which describe the mentioned artefacts, highly depends on the chosen abstraction level (see above). While artefacts on a low abstraction level are rapidly changing (e.g. activities of a certain process), artefacts on a higher abstraction level (e.g. top-level business processes) tend to be more stable. As a result, creating and exchanging deliverables on high abstraction levels reduces the need for short synchronisation cycles and efforts for integrating the ‘external’ deliverables into the EA repository.

Time Horizons: Usually the time horizons of the models differ between EAM and BPM (see figure 110). In BPM the ‘as-is’ state (e.g. mapping of work instructions) or a mid-term time horizon (e.g. specification of IT systems) is being taken into account. Strategic aspects in EAM (e.g. planning new products, business models) require the modelling of ‘to-be’ architectures in the long run. The EA is typically described by current and target architectures and several transition architectures that vary in their level of detail (see best practice scenario ‘Master Planning’ in chapter 4.6).
Step 5 - Define integration concept

Depending on the complexity of the integration scenario, it is required to clarify data ownership, adapt existing process architectures, provide adequate tool support, execute cost-benefit analyses, and define service level agreements.

**Define Data Ownership:** The data ownership status of the exchanged artefacts must ensure data integrity and consistency. Responsible roles and accountabilities have to be determined. Business departments usually have the sovereignty over the artefacts of the business architecture (e.g. business processes and organisational structures). The IT department usually owns the architecture artefacts of the IT architecture (such as applications, interfaces and technologies). However, this is not always the case. Defining the target state process maps might be up to the EAM team. Figure 111 depicts stakeholder roles involved in BPM and EAM, and ownership of artefacts and deliverables.
Besides the ownership for architecture artefacts, the ownership of the relations needs to be agreed on.

**Example:** Considering the above figure, we would like to link business processes to applications. In an ideal situation, these links or relations are maintained by business or the EAM team. Let us assume that in this case the EAM team is responsible for linking business processes to applications. Presumably we would like to use the BOC Management Office for this activity. First, we import all available applications to ADONIS. In ADONIS, we link the processes (from the process maps) to the applications. Next, we export these processes from ADONIS to ADOit. In this example, ADONIS is the main tool for defining processes and relations.

**Adapt Existing Procedures:** To keep the common information base up to date, procedures and guidelines for the creation, approval and exchange of deliverables need to be defined and integrated into daily operations. Responsible and accountable roles need to be defined and staffed. If the necessary skills to create the required deliverables are not available among the staff, these skills need to be developed. To find potential roles and stakeholders, please refer to the section ‘Stakeholders and Their Roles’.
Integrate BPM sources/tools with ADOit: ADOit and the BOC Management Office provide a number of interfaces for integrating BPM. Please refer to chapter 6.2.1 ‘Common Interfaces‘ for further details.

ADONIS users usually start by using the Management Office Interface to ADOit. Other users begin by using the Excel Interface to gain a first insight into the interfaces of the BOC Management Office before actually implementing web service interfaces (if needed at all).

**Cost-Benefit Analysis:** The determinants’ abstraction level and change frequency are effort drivers for the integration scenario. Besides the efforts for creating the deliverables on the supplier side, the effort for integration of the deliverables on the receiver side also needs to be considered carefully. Costs in terms of effort for creation and integration of the deliverables need to be balanced with the gained benefits. Appropriate tool support lowers efforts for delivery and integration. Organisational coverage and time relevance mainly influence the benefits of integration.

**Service Level Agreements:** We propose to set up informal maintenance contracts to ensure the quality of the process architectures (or any other deliverables), and meet the maintenance schedule of the architecture. Define the measures discussed in the previous sections in detail to agree on accuracy and quality of the process architectures that should be delivered on a timely basis.

**Step 6 - Feasibility check (pilot) and implementation**

In this step, a pilot of the integration concept is launched. Based on this pilot, the original approach on integration is validated. The solution needs to fit the purpose to support subsequent work in EAM. If the solution does not show the intended effects, the structure and level of granularity of the deliverables and the defined integration concept need to be reconsidered. However, if the pilot is successful, the integration concept can be fully implemented and rolled out across the entire organisation.
ADOit EAM Meta-Model

5.1 Business Architecture and Organisation Concepts  143
5.2 Information Systems Architecture Concepts  144
5.3 User  144
5.4 Technology Architecture Concepts  145
5.5 Project Portfolio Concepts  145
As a starting point, ADOit provides a predefined meta-model based on TOGAF 9 which supports the ADOit best practices out-of-the-box. The ADOit Core Meta-Model can be easily customised and adapted to your organisation’s needs. The reason for this flexibility is ADOit’s meta-modelling concept. When it comes to everyday troublesome challenges, your EA needs to be your gateway to safe and reliable solutions. At the same time the EA has to be flexible so that it can deal with unexpected, future challenges. Utilising the meta-modelling features of ADOit, the meta-model (and your entire EA framework implemented in ADOit) can be adapted to your needs.

The following section discusses the elements of the ADOit Core Meta-Model, which is referred to as ‘modelling concepts’ in the text. Figure 112 gives an overview of the most important concepts and relations of the ADOit Core Meta-Model.

Business Architecture: On the business architecture level you define the process architecture and the organisational structure of your organisation, which basically shapes the organisation’s overall structure. Modelling the business activities, the underlying business functions, and identifying business process improvement opportunities are the main goals of analysing the business architecture.
Information Systems Architecture (Application/Service Architecture and Data Architecture): In accordance with TOGAF 9 the information systems architecture comprises two types of sub architectures, namely the application architecture and the data architecture. The application architecture comprises the major applications imperative to managing business objects and supporting business functions. The data architecture focuses on business objects in terms of information assets. It considers applications, services and functions that are linked to business objects, the way they deal with business objects, and the way business objects can be accessed, manipulated, and transferred.

Technology Architecture: The technology architecture seeks to map applications defined in the application architecture onto a set of technology components, represented by technology packages and technologies. The technology architecture is based on the technology road map which typically contains the technology standards of the organisation.

The infrastructure architecture – as part of the technology architecture – defines the physical hardware and network elements on which applications, application modules and their required systems software (technologies) are deployed.

Demand and Project Portfolio: Project portfolio is the set of all projects (including project proposals) that transform the Enterprise Architecture in accordance with the master plan. Projects are managed and assessed in terms of operative benefits and corporate strategic alignment. Typically projects arise from demands made by stakeholders in business and IT.
## 5.1 Business Architecture and Organisation Concepts

<table>
<thead>
<tr>
<th><strong>Domain</strong></th>
<th>High-level approach to the categorisation of architecture artefacts of any architectural level for purposes of organisation and standardisation. Typical categories for a business level are for example locations (same geographical region), organisations (same departments), and products provided by the organisation.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Process</strong></td>
<td>A business process represents a set of related activities that aim at achieving a specified result. Business processes can be decomposed into sub-processes.</td>
</tr>
<tr>
<td><strong>Business Function</strong></td>
<td>A business operation that is performed by the organisation. Business functions are clustered independent of any system or implementation considerations and organisational constraints. Business functions are not necessarily supported by IT. Business functions can be hierarchically composed if needed. Business functions that are supported by IT are exposed by applications or business services.</td>
</tr>
<tr>
<td><strong>Business Capability</strong></td>
<td>Top-level business functions are called business capabilities.</td>
</tr>
<tr>
<td><strong>Organisational Unit</strong></td>
<td>A self-contained unit of resources with line management responsibility, goals, objectives, and measures. Organisational units may include external parties and business partner organisations.</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>A geographical place where IT artefacts are located or business activity is performed. In ADOit, locations can be implemented as an extension.</td>
</tr>
</tbody>
</table>
5.2 Information Systems Architecture Concepts

<table>
<thead>
<tr>
<th>Table 6</th>
<th>Information Systems Architecture Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td>A collection of software components organised to accomplish a specific business function or set of business functions. Applications are implemented by a suite of technology packages or technologies.</td>
</tr>
<tr>
<td><strong>Application Module</strong></td>
<td>A modular, deployable, and replaceable component of an application. In ADOit, application modules can be implemented as an extension.</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>An interface facilitates interaction between applications or application modules. It is provided by an application (module) and can be used by other applications to access the application’s functionality.</td>
</tr>
<tr>
<td><strong>Business Service</strong></td>
<td>A Web Service that provides business functions through a defined and standardised interface. Business services are explicitly governed by an organisation.</td>
</tr>
<tr>
<td><strong>Business Object</strong></td>
<td>A high-level encapsulation of data that has relevance from business perspective. Business objects are structured in a way to be suitable for communication, interpretation, or processing by manual or automatic means.</td>
</tr>
</tbody>
</table>

5.3 User

<table>
<thead>
<tr>
<th>Table 7</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User</strong></td>
<td>In ADOit a user is a specific type of actor. A user can be assigned (via ownership relation) to each EA artefact. In this case the user is responsible for his assigned enterprise artefacts and has write permission.</td>
</tr>
</tbody>
</table>
5.4 Technology Architecture Concepts

<table>
<thead>
<tr>
<th>Technology Package</th>
<th>A set of technologies required to provide infrastructure that supports the delivery of applications.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>A specific technology product or technology product instance. For example, a particular product version of a Commercial Off-The-Shelf (COTS) solution, or a specific brand and version of server.</td>
</tr>
<tr>
<td>Infrastructure Element</td>
<td>A computational resource upon which applications, application modules or databases may be deployed for execution (usually a server or cluster). In ADOit, infrastructure elements can be implemented as an extension.</td>
</tr>
<tr>
<td>Database</td>
<td>A concrete instance of a database management system, serving as a structured or organised collection of business objects, which can be accessed by an application (module). In ADOit, databases can be implemented as an extension.</td>
</tr>
</tbody>
</table>

5.5 Project Portfolio Concepts

<table>
<thead>
<tr>
<th>Project</th>
<th>Actual transformations of the EA are performed by projects. A project is a group of activities and tasks that change the EA, fulfilling an agreed set of objectives and success measures. There are at least two states for a project: ‘approved’ and ‘proposed’.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand</td>
<td>A quantitative statement of business need that must be met by a particular architecture. Demands typically are implemented via projects. In ADOit, demands can be implemented as an extension.</td>
</tr>
</tbody>
</table>
ADO\textit{it} Architecture

6.1 Overall Architecture 147
6.2 Interfaces 148
   6.2.1 Common Interfaces 148
   6.2.2 The BOC Management Office - Integration Concepts 149
6.1 Overall Architecture

ADOit comprises three main tools, namely the ADOit Web Client, the ADOit Management Toolkit and the ADOit Administration Toolkit.

It is important to ensure that the stakeholders are able to gain access to all EA information without having completed any special training courses. The ADOit Web Client is adaptable to the needs of the stakeholder’s role. It supports personalised views, which means that every involved stakeholder is assigned to a role, such as application owner. This concept helps stakeholders clearly recognise their functions within your EAM processes.

The ADOit Rich Client comprises two tools: The ADOit Management Toolkit and the ADOit Administration Toolkit.
The ADOit Management Toolkit is typically used by stakeholders who are heavily involved in the EA programme. With its components ‘Modelling’, ‘Analysis’, ‘Publishing’, ‘Import/Export’, and ‘Task Management’, it provides advanced means for organising the entire EA initiative – beginning with building up an adequate structure for the EA, followed by extended means for analysis and reporting. Typical supported EAM scenarios are discussed in chapter 4 ‘ADOit Best Practices’.

The ADOit Administration Toolkit supports the user with a variety of components. Significant components are ‘Meta-Model Management’, ‘Library Management’, and ‘User Management’. All of these components support the setup and – in terms of TOGAF the – architecture vision phase of any EA programme.

- **Meta-Model Management**: This component allows for adapting the ADOit Core Meta-Model exactly to your needs. The ADOit Core Meta-Model can be modified and extended in any way depending on the specifics of your organisation. It can be replaced all together if you prefer to define a meta-model of your own.

- **Library Management**: Among other functionalities, this component allows you to define the viewpoints required for your EA programme. Besides the pre-configured viewpoints any other viewpoint can be configured. The following basic types of viewpoints are available: Model types for graphical modelling, dependency analysis, portfolio (bubble) views, bar and GANTT charts, and matrices. Each of these viewpoint types can be configured based on your individual meta-model.

- **User Management**: Utilising this component, roles (so-called user groups) can be defined. Typical roles are enterprise architect, application owner and technology architect. Via the ‘User Management’, access rights, components and single functions can be granted or restricted on role or even user level.

### 6.2 Interfaces

#### 6.2.1 Common Interfaces

ADOit provides several interfaces and can be perfectly integrated into your IT environment. File-based interfaces, like the Excel interface, or web service interfaces can be used. Chapter 4.2 ‘Populate the EA Repository’ discusses some of the typical scenarios in this context.
6.2.2 The BOC Management Office - Integration Concepts

Given the integration concepts of the BOC Management Office, data synchronisation/replication mechanisms can be used. Chapter 4.7 ‘Integrating EAM and BPM’ discusses procedures and guidelines to be considered.
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADM</td>
<td>Architecture Development Method</td>
</tr>
<tr>
<td>APPC</td>
<td>Advanced Program to Program Communications</td>
</tr>
<tr>
<td>BI</td>
<td>Business Intelligence</td>
</tr>
<tr>
<td>BIA</td>
<td>Business Impact Analysis</td>
</tr>
<tr>
<td>BOC</td>
<td>Business Objectives Consulting</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Management</td>
</tr>
<tr>
<td>C4ISR AF</td>
<td>The C4ISR AF (Command and Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance Architecture Framework) is now known as DoDAF.</td>
</tr>
<tr>
<td>CADM</td>
<td>Core Architecture Data Model</td>
</tr>
<tr>
<td>CDM</td>
<td>Conceptual Data Model</td>
</tr>
<tr>
<td>CMDB</td>
<td>Configuration Management Database</td>
</tr>
<tr>
<td>CMS</td>
<td>Configuration Management System</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>CRM</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>DAO</td>
<td>Data Access Object</td>
</tr>
<tr>
<td>DB2</td>
<td>Database Management System</td>
</tr>
<tr>
<td>DDE</td>
<td>Dynamic Data Exchange</td>
</tr>
<tr>
<td>DM2</td>
<td>DoDAF Meta-Model</td>
</tr>
<tr>
<td>DMS</td>
<td>Document Management System</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>DoDAF</td>
<td>Department of Defense Architecture Framework</td>
</tr>
<tr>
<td>EA</td>
<td>Enterprise Architecture</td>
</tr>
<tr>
<td>EAM</td>
<td>Enterprise Architecture Management</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organisation for Standardisation</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>ITSM</td>
<td>Information Technology Service Management</td>
</tr>
<tr>
<td>LDM</td>
<td>Logical Data Model</td>
</tr>
<tr>
<td>MoDAF</td>
<td>Ministry of Defence Architecture Framework</td>
</tr>
<tr>
<td>NAF</td>
<td>NATO (North Atlantic Treaty Organization) Architecture Framework</td>
</tr>
<tr>
<td>ODBC</td>
<td>Open Database Connectivity</td>
</tr>
<tr>
<td>OLE</td>
<td>Object Linking and Embedding</td>
</tr>
<tr>
<td>PEAFL</td>
<td>Pragmatic Enterprise Architecture Framework</td>
</tr>
<tr>
<td>PES</td>
<td>Physical Exchange Specification</td>
</tr>
<tr>
<td>POS</td>
<td>Point Of Sale</td>
</tr>
<tr>
<td>PV</td>
<td>Project Viewpoint</td>
</tr>
<tr>
<td>RPC</td>
<td>Remote Procedure Call</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>TOGAF</td>
<td>The Open Group Architecture Framework</td>
</tr>
<tr>
<td>TRM</td>
<td>Technology Reference Model</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
<tr>
<td>XAMPP</td>
<td>X (meaning cross-platform), Apache HTTP Server, MySQL, PHP, Perl</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XSD</td>
<td>XML Schema Definition</td>
</tr>
</tbody>
</table>
BOC AG
Wipplingerstraße 1
A-1010 Vienna
Phone: +43-1-513 27 36-11 99 | eMail: boc@boc-group.com

BOC Germany
Voßstraße 22
D-10117 Berlin
Phone: +49-30-22 69 25-10 | eMail: boc@boc-de.com

BOC Greece
16, Kaisareias & 5, Avlidos
GR 115 27 Athens
Phone: +30-210-33 90 48-6 | eMail: boc@boc-gr.com

BOC Ibérica
Velázquez, 71
E-28006 Madrid
Phone: +34-91-781 18 80 | eMail: boc@boc-es.com

BOC Ireland
80 Haddington Road
IRL-Dublin 4
Phone: +353-1-63 75-240 | eMail: boc@boc-ie.com

BOC Austria
Rabensteig 2
A-1010 Vienna
Phone: +43-1-513 27 36-0 | eMail: boc@boc-at.com

BOC Poland
Al. Jerozolimskie 109/26
PL-02-011 Warsaw
Phone: +48-22-628 00 15 | eMail: boc@boc-pl.com

BOC Switzerland
Zürcherstrasse 21
CH-8400 Winterthur
Phone: +41-52-26 03 75-0 | eMail: boc@boc-ch.com

BOC Information Systems
Wipplingerstraße 1
A-1010 Vienna
Phone: +43-1-513 27 36-11 70 | eMail: boc@boc-eu.com
‘Yes’ to Enterprise Architecture Management (EAM), and then what?

Growing businesses tend to become more complex from day to day. Regulations, development of new products, business competition, increasingly high business demands, IT maintenance costs and the like, play a tremendous part in contributing to vast application landscapes, skyrocketing costs, compliance issues, and a hardly comprehensible architecture.

While looking for solutions to their problems, organisations usually split in two. The IT is eager to work on the IT structure, and to merely deal with technical matters regardless of the impact they could have on the entire business. On the other hand, the business spends time on the planning of the business architecture focusing mainly on business strategies, processes and products.

But both are missing an important fact: the alignment between business and IT is what makes the enterprise running. Translating the IT language to business and vice versa is considered a challenging task.

EAM practices offer various ways to help organisations align their business to IT. When cultures collide, it is paramount to find common ground. EAM supports business and IT to find a common language and to create shared value.

**EAM certainly sounds great, but:**

- Do you know the stakeholders who need to participate in the EA programme?
- How about gaining credibility from the stakeholders of the organisation?
- How do you suppose to start the EA initiative?
- Which EA framework should you choose?
- What if you want to change your existing EA framework?
- How can you evaluate your architecture?
- How can you get from the current state to the future state architecture?

This manual supports you from the moment you’re having the idea of implementing EAM to the moment you stand in front of the stakeholders trying to gain buy-in from them, and throughout your entire EA programme. It gives you advice on how to choose an EA framework, and how to gather and organise your data in one place, to name but a few. Moreover, it provides you with best practices all of which have been proven to be successful in past EA projects. But most importantly, it shows you how to use tool support for implementing the suggested best practices by the BOC Group. Each best practice scenario is explained at great length, using the EA tool ADOit for putting EA into practice.