

Global Standards and Publications

EDITION 2012/2013



IT Service Management
IT Governance & Security
Application Management
Information Management
Enterprise Architecture
Project Management
Program Management
Risk Management
Outsourcing
Business Management
Quality Management
Business Process Management

Global Standards and Publications
Edition 2012 / 2013

Global Standards and Publications

EDITION 2012/2013



Colophon

Title:	Global Standards and Publications Edition 2012 / 2013
Publication of:	Van Haren Publishing, www.vanharen.net
Editors:	Jane Chittenden René Visser Johannes W. van den Bent
ISBN Hard copy:	978 90 8753 703 6
ISBN eBook:	978 90 8753 866 8
Print:	Second edition, first impression, May 2012
Layout and design:	CO2 Premedia, Amersfoort - NL
Copyright:	© Van Haren Publishing 2012



TRADEMARK NOTICES

ArchiMate® and TOGAF® are registered trademarks of The Open Group.
ASL® and BiSL® **are registered trademarks of ASL BiSL Foundation.**
COBIT® is a registered trademark of the Information Systems Audit and Control Association (ISACA) / IT Governance Institute (ITGI).
ITIL® is a registered trademark of the Cabinet Office.
M_o_R® is a registered trademark of the Cabinet Office.
MoP™ is a registered trademark of the Cabinet Office.
MSP® is a registered trademark of the Cabinet Office.
P3O® is a registered trademark of the Cabinet Office.
PRINCE2® is a registered trademark of the Cabinet Office.
PMBOK® Guide is a registered trademark of the Project Management Institute (PMI).
SqEME® **is a registered trademark of Stichting SqEME.**
TMap® is a registered trademark of Sogeti Nederland B.V.

For any further enquiries about Van Haren Publishing, please send an e-mail to: info@vanharen.net

Although this publication has been composed with most care, author nor editor can accept any liability for damage caused by possible errors and/or incompleteness in this publication.

No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by the publisher.

Dear readers,

In this rapidly changing IT and business environment most things should and could be more easy. It is no wonder that methods like Agile and Scrum are gaining popularity. New developments offer great opportunities for those willing to make the most out of it but it can be difficult not to get overwhelmed.

In the current environment with constant changes and almost infinite ways accessing information and communicating it is essential to make communication as clear as possible and ensure the quality of information. Van Haren Publishing makes general Best Practices available to provide quality, practically validated information worldwide. The use of standards and frameworks gives everyone the same language thus minimalizing the chance of errors due to unclear communication. Best Practices regarding these standards and frameworks provides you with information summarizing years of experience by the best in the industry.

Not only do we publish books on Best Practices, we also actively and independently promote the standards and frameworks via our freely accessible eKnowledge. To make communication on standards everywhere a little easier, we provide you with a basic summary of 36 relevant standards in our catalog. It is an illusion those standards will lead to better results. More important is the people factor, as without people all these things don't evolve at all. But that is beyond the service we provide, all we can do is give a start in sharing best practice and share generic solutions. The rest should come from you.

Kind regards,

Ivo van Haren, CEO Van Haren Publishing

Contents

1. Agile	9
2. Amsterdam Information Management Model (AIM)	13
3. ArchiMate®	17
4. ASL®	20
5. Balanced Scorecard	23
6. BiSL®	26
7. CATS CM®	29
8. CMMI®	32
9. COBIT®	35
10. EFQM	40
11. eSCM-CL	44
12. eSCM-SP	47
13. Frameworx	50
14. ICB®	53
15. ISO 9001	56
16. ISO 14000	59
17. ISO/IEC 15504	62
18. ISO/IEC 27000 series	66
19. ISO 31000	69
20. ISO 38500	72
21. ISO/IEC 20000	75
22. ITIL® 2011	78
23. Lean management	82
24. M_o_R®	85
25. MoP™	88
26. MSP®	91
27. OPBOK	94
28. P3O®	97
29. PMBOK® Guide	100
30. PRINCE2®	103

Contents

31. SABSA®	106
32. Scrum	113
33. Six Sigma	117
34. SqEME®	120
35. TMap® NEXT	123
36. TOGAF®	126

1. Agile

1.1 Title/definition

Agile software development approach

1.2 The basics

Agile software development is set of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.

1.3 Summary

Incremental software development methods have been traced back to 1957. "Lightweight" software development methods evolved in the mid-1990s as a reaction against 'heavyweight' methods, which were characterized by their critics as a heavily regulated, regimented, micromanaged, waterfall model of development. Supporters of lightweight methods (and now agile methods) contend that they are a return to earlier practices in software development.

Early implementations of lightweight methods include Scrum (1993), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (1995). These are now typically referred to as Agile methodologies, after the Agile Manifesto.

The Agile Manifesto was written in February 2001, at a summit of independent-minded practitioners of several programming methodologies.

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

- Individuals and interactions** over processes and tools
- Working software** over comprehensive documentation
- Customer collaboration** over contract negotiation
- Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Source: agilemanifesto.org/

The Agile Manifesto has twelve underlying principles:

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Working software is the principal measure of progress
5. Sustainable development, able to maintain a constant pace
6. Close, daily co-operation between business people and developers
7. Face-to-face conversation is the best form of communication (co-location)
8. Projects are built around motivated individuals, who should be trusted
9. Continuous attention to technical excellence and good design
10. Simplicity
11. Self-organizing teams
12. Regular adaptation to changing circumstances

Agile methods break tasks into small increments with minimal planning and do not directly involve long-term planning.

Iterations are short time frames. Team composition in an agile project is usually cross-functional and self-organizing and team size is usually small (5-9 people.) The agile method encourages stakeholders to prioritize “their requirements on the basis of business value.

The Agile approach is supported by the Agile Alliance, a not-for-profit organization that wants to see Agile projects start and help Agile teams perform. It is funded by individual memberships, corporate memberships, and by the proceeds from the Agile 200X series of conferences. It is not a certification body and does not endorse any certification programs.

1.4 Target audience

Anyone involved in an Agile software development project team; including analysts, architects, developers, testers and business customer/users; anyone supporting or managing an Agile project team who requires a detailed understanding of the practices and benefits of Agile software development.

1.5 Scope and constraints

Applicable to software development environments. Improved quality; higher productivity; positive effect on business satisfaction;

Constraints:

- Works less well in distributed development efforts where teams are not located together
- Acceptance: forcing an Agile process on a development team that is unfamiliar with the approach

- Exceptions: mission-critical systems where failure is not an option at any cost (e.g. software for surgical procedures).

1.6 Relevant links (web links)

<http://agilemanifesto.org/> and <http://www.agilealliance.org>

2. Amsterdam Information management Model (AIM)

2.1 Title/definition

The Amsterdam model for information management: A Generic Framework for Information Management

2.2 The basics

The Amsterdam Information management Model (AIM) provides a mapping of the relationships between organization and information.

2.3 Summary

AIM was developed at the University of Amsterdam (paper: Abcouwer, A.W., Maes, R. Truijens, J. (1997), 'Contouren voor een generiek model voor informatie-management', Tijdschrift Informatie en Management.). It can be used as a tool for positioning and interrelating information management functions. It can be applied to the areas of business-IT alignment and sourcing, and can be of use when considering IT governance. It offers a high level view of the entire scope of information management; its main application is in the analysis of organization and responsibilities.

AIM can be used to support strategic discussions in three different ways, as shown in the diagram below (Figure 2.1):

- Descriptive, orientation – the framework offers a map of the

entire information management domain, and can be used for positioning specific information management processes in the organization.

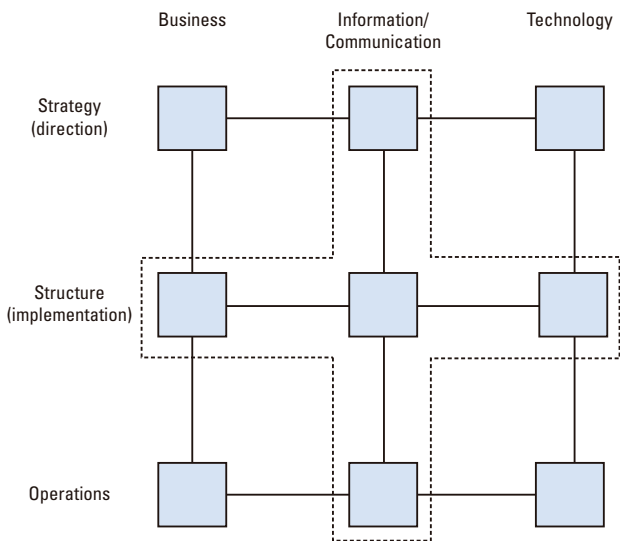
- Specification, design – the framework can be used to re-organize the information management organization, e.g. to specify the role of the Chief Information Officer (CIO) or determine the responsibilities of the retained organization in the case of outsourcing.
- Prescriptive, normative – the framework can be used as a diagnostic instrument to find gaps in an organization's information management, and specifically aimed at identifying missing interrelationships between the various components of the framework.

On the horizontal axis, the framework distinguishes three domains of governance:

1. Business – This domain comprises all standard business functions such as management, HR, resources and processes.
2. Information and Communication (information domain – This domain describes how information and communication supports the business. In this domain, business requirements are translated into the IT (technology) capabilities that are needed to support the business.
3. Technology (IT domain) – This domain specifically describes the development and management of IT solutions.

The vertical axis describes the three levels of governance:

- Strategy (scope, core competencies and governance);
- Structure (architecture and competencies);
- Operations (processes and skills).



Abcouwer, Maes and Truijens (1997)

Figure 2.1: The Nine Square framework (het negenvlak)

AIM (originally known as the nine square framework) connects the two dimensions of management and information as the central components for Information Management. The dotted line demarks the scope of Business-IT alignment.

2.4 Target audience

The framework was developed for information managers, enterprise architects and IT architects.

2.5 Scope and constraints

The scope of the framework is the information management domain.

This framework enables discussions on the topic of business and IT alignment, but it does not provide information on how organizations can actually achieve better communications between business and IT. The framework is not a method, and cannot be used in a descriptive way; however, it can be a useful addition to enterprise architecture frameworks such as TOGAF™.

2.6 Relevant links (web links)

The framework can be downloaded for free from the website of the University of Amsterdam (Dutch site): <http://primavera.fee.uva.nl>.

Original paper: (Dutch)

- Abcouwer, A.W., Maes, R. Truijens, J. (1997), 'Contouren voor een generiek model voor informatie-management', *Tijdschrift Informatie en Management*.

The model was further enhanced by Maes in the period from 1999 to 2003 in a series of working papers: (English)

- *Maes, R. (1999) A Generic Framework for Information Management, Primavera workingpaper series 1999-03;*

3. ArchiMate®

3.1 Title/definition

ArchiMate®, an Open Group Standard

3.2 The basics

ArchiMate® is an open and independent modelling language for Enterprise Architecture that is supported by different tool vendors and consulting firms. ArchiMate provides instruments to enable Enterprise Architects to describe, analyze, and visualize the relationships among business domains in an unambiguous way.

3.3 Summary

Developed by the members of The Open Group, ArchiMate® 2.0 was released in January 2012 and is now more aligned with TOGAF®, the world's most popular Enterprise Architecture framework. As a result, Enterprise Architects using the language can improve the way key business and IT stakeholders collaborate and adapt to change.

The standard contains the formal definition of ArchiMate as a visual design language, together with concepts for specifying inter-related architectures, and specific viewpoints for typical stakeholders. The standard also includes a chapter addressing considerations regarding language extensions.

The contents of the standard include the following:

- The overall modelling framework that ArchiMate uses
- The structure of the modelling language
- A detailed breakdown of the constituent elements of the modelling framework covering the three layers (Business/

Application/Technology), cross-layer dependencies and alignment, and relationships within the framework

- Architectural viewpoints, including a set of standard viewpoints
- Optional extensions to the framework
- Commentary around future direction of the specification
- Notation overviews and summaries

ArchiMate 2.0 improves collaboration through clearer understanding across multiple functions, including business executives, enterprise architects, systems analysts, software engineers, business process consultants, and infrastructure engineers. The standard enables the creation of fully integrated models of an organization's Enterprise Architecture, the motivation behind it, and the programs, projects, and migration paths to implement it. ArchiMate already follows terms defined in the TOGAF framework, and version 2.0 of the specification enables modelling throughout the TOGAF Architecture Development Method (ADM) cycle.

3.4 Target audience

Enterprise architects, business architects, IT architects, application architects, data architects, software architects, systems architects, solutions architects, infrastructure architects, process architects, domain architects, product managers, operational managers, senior managers, project leaders, and anyone committed to work within the reference framework defined by an Enterprise Architecture.

3.5 Scope

The role of the ArchiMate standard is to provide a graphical language for the representation of Enterprise Architectures over time (i.e., including transformation and migration planning), as well as their motivation and rationale. The ArchiMate modelling language provides a uniform representation for diagrams that describe Enterprise Architectures, and offers an integrated approach to describe and visualize the different architecture domains together with their underlying relations and dependencies.

The design of the ArchiMate language started from a set of relatively generic concepts (objects and relations), which have been specialized for application at the different architectural layers for an Enterprise Architecture. The most important design restriction on ArchiMate is that it has been explicitly designed to be as compact as possible, yet still usable for most Enterprise Architecture modelling tasks. In the interest of simplicity of learning and use, ArchiMate has been limited to the concepts that suffice for modelling the proverbial 80% of practical cases.

3.6 Relevant links (web links)

Official websites:

<http://pubs.opengroup.org/architecture/archimate2-doc/>

<http://www.opengroup.org/archimate>

4. ASL®

4.1 Title/definition

ASL® (Application Services Library)

4.2 The basics

ASL (Application Services Library) is a framework and collection of best practices for application management.

4.3 Summary

ASL (Application Services Library) was developed by a Dutch IT service provider, PinkRocade in the 1990s and was made public in 2001. Since 2002 the framework and the accompanying best practices have been maintained by the ASL BiSL Foundation. The current version is Version 2, published in the Netherlands in 2009.

ASL is concerned with managing the support, maintenance, renewal and strategy of applications in an economically sound manner. The library consists of a framework, best practices, standard templates and a self-assessment. The ASL framework provides descriptions of all the processes that are needed for application management.

The framework distinguishes six process clusters, which are viewed at operational, managing and strategic levels (Figure 4.1).

The *application support cluster* at the operational level aims to ensure that the current applications are used in the most effective way to support the business processes, using a minimum of resources and leading to a minimum of operational disruptions. The *application maintenance and renewal cluster* ensures that

the applications are modified in line with changing requirements, usually as a result of changes in the business processes, keeping the applications up-to-date. The connecting processes form the bridge between the service organization cluster and the development and maintenance cluster.

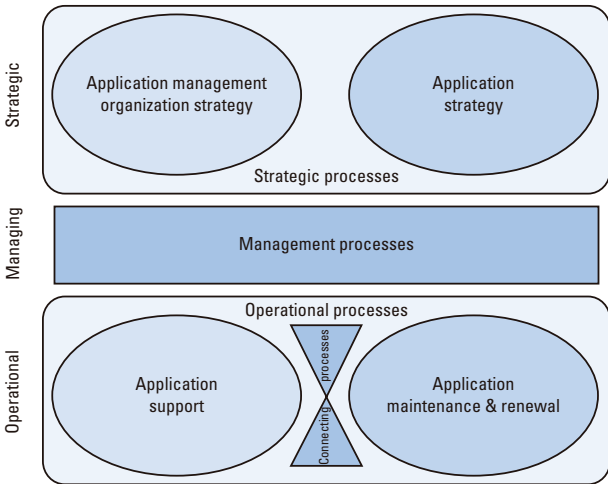


Figure 4.1: The ASL framework

The management processes ensure that the operational clusters are managed in an integrated way.

Finally, there are two clusters at the strategic level. The aim of the *application strategy cluster* is to address the long-term strategy for the application(s). The processes needed for the long-term strategy for the application management organization are described in the *application management organization strategy cluster*.

4.4 Target audience

The target audience for ASL consists of everyone who is involved in the development and management of applications: application support personnel, application architects and designers, programmers, testers, and managers with responsibility for application development or application management.

4.5 Scope and constraints

The scope of ASL is the support, maintenance, renewal, and strategy of applications, and the management of all related activities.

Strengths

- It offers a common language and conceptual framework for application management (maintenance and support)
- It provides an overview of all the activities (from operational to strategic) that are needed to keep applications up-to-date with the changing needs of the organization.
- It is usable in various organizations and It is part of the public domain and owned by one single corporation
- It is supported by a not-for-profit, vendor-independent foundation with participation by a wide range of organizations

Constraints

- ASL overlaps partially with other IT Service Management frameworks.

4.6 Relevant links (web links)

Official website of the ASL BiSL Foundation:
www.aslbislfoundation.org/en