Jack van 't Wout
Maarten Waage
Herman Hartman
Max Stahlecker
Aaldert Hofman

The Integrated Architecture Framework Explained

Why, What, How

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Springer

The Integrated Architecture Framework Explained

Jack van 't Wout · Maarten Waage ·
Herman Hartman · Max Stahlecker ·
Aaldert Hofman

The Integrated Architecture Framework Explained

Why, What, How

Springer

Jack van 't Wout
Capgemini Nederland BV
Papendorpseweg 100
3528 BJ Utrecht
Netherlands
jack.vant.wout@capgemini.com

Herman Hartman
Capgemini Nederland BV
Papendorpseweg 100
3528 BJ Utrecht
Netherlands
herman.hartman@capgemini.com

Aaldert Hofman
Capgemini Nederland BV
Papendorpseweg 100
3528 BJ Utrecht
Netherlands
aaldert.hofman@capgemini.com

Maarten Waage
Capgemini Nederland BV
Papendorpseweg 100
3528 BJ Utrecht
Netherlands
maarten.waage@capgemini.com

Max Stahlecker
Capgemini Nederland BV
Papendorpseweg 100
3528 BJ Utrecht
Netherlands
max.stahlecker@capgemini.com

ISBN 978-3-642-11517-2 e-ISBN 978-3-642-11518-9
DOI 10.1007/978-3-642-11518-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010924818

ACM Computing Classification (1998): J.1, I.6, H.1, H.4

Cover design: KünkelLopka, Heidelberg

Printed on acid-free paper

Springer is part of Springer Science + Business Media (www.springer.com)

Foreword

When I joined Capgemini back in 1996 I was amazed by investment that had been made in developing Enterprise Architecture, and at the root of this, the IAF methodology. Back in the mid 1990s the importance of architecture was dimly recognised but certainly it was not widely understood as a crucial element of successful enterprise wide IT implementation. A decade later with the huge growth in the role, the sophistication, and importance of Information Technology it has become recognized, and established for the value it brings.

With this recognition has come various forms of 'standardization' ranging from the work of the Open Group and its moves to establish TOGAF as a common framework, together with ITAC to certify architects, through to a wide variety of product vendor architects, even to some industry sectors establishing their own architectures. Has this diminished, or even may be removed the need for IAF?

Well it might have done if the world had stood still, but it hasn't. Simultaneously the range and complexity of technology has increased, the functionality has been extended to embrace new front office capabilities and most of all the externalization and globalization of business has added a whole new extra dimension. Standardization might have improved connections and interfaces, and in so doing produced 'systems' of apparently limitless extendibility, but it has done little to improve the necessary 'understanding'.

So here we are 13 years later, 13 years of consistent development of IAF, and yet looking ahead the requirement for even more development is clear. But looking back it's equally clear to see the value delivered and the foundation it has built in Enterprises in so many countries. The future will rest of the past, the legacy of IAF is one of enabling for the future and there are not many areas of technology where that remark can truthfully be made.

This book records an impressive journey, and points to an important future, and is written by those for whom Enterprise Architecture and IAF is a passion. Reading the book will help develop that same passion, as well as

increasing the understanding of one of the most important aspects of any enterprises success in business today. I thank my colleagues for introducing the topic to me, and for helping me to appreciate its value to my everyday work.

Woking, UK Andy Mulholland

Preface

IAF Is Here to Stay!

IAF is here to stay! Even though I am very much in favor of open standards such as TOGAF and ArchiMate, I still say IAF is here to stay! No; the authors didn't pay me off to say this, although I wouldn't say no to a nice bottle of wine. However, I make this statement out of my own conviction. I'll explain why in a moment, but let's first start with some history.

When I got involved in the field of architecture in 1997, Capgemini was already working toward the creation of the Integrated Architecture Framework (IAF). Since then, the IAF has been evolving continuously. Fueled by daily experiences, discussions among architects, ample feedback from client engagements, the original framework has evolved into its current version.

In the past, Capgemini has been rather shy in communicating about the IAF. Rather than showing this diamond in the making to the outside world, they kept it quiet and continued polishing it. In my own past as a full-time Professor at the University of Nijmegen, I was involved in several architecture related research activities, such as the ArchiMate project. When I became aware of IAF's existence, I immediately challenged Capgemini to more widely publish their IAF. Meanwhile my curiosity was mounting. To me, IAF provided a welcome, practice based, complementary view to the results of the ArchiMate research project. So, when I joined Capgemini at the start of 2008, I was more than happy to attend a course on IAF. This strengthened my conviction that it was time for Capgemini to finally produce a book on IAF. There was more than enough reason to be proud about IAF as a tried and tested architecture framework, and make it available to a wider audience. Therefore, early 2008 I started a lobby to produce an IAF book, and late 2008 we were finally able to give the go ahead to the team of authors to do the really hard work and produce the book. I am really grateful to the author team for their commitment in finishing this book. It shall be no secret that 2009 has been a difficult year for our industry. Despite commercial pressures, the authors spent numerous hours in their spare-time to continue working on this great book on architecture.

So how about open standards? Large parts of TOGAF 9's content framework have already been based on IAF. Currently, The Open Group has two complementary standards for architecture: TOGAF, the method for doing architecture, and ArchiMate, the language for representing architectures. So is there a future for IAF amidst all of these standards? Well, in my opinion there certainly is. Standards evolve based on consensus. As a result, standards can only (and must) evolve slowly. At the same time, consultancy firms such as Capgemini will continue to gather their own experiences. This is where company specific frameworks such as the IAF can play a crucial role. They will allow for faster innovations, leading the direction in which future versions of the standards can evolve.

Therefore, I expect IAF in its next evolution step to become fully compliant to the TOGAF/ArchiMate tandem, while at the same time going beyond these standards based on the fast amount of – ever growing – experience embodied in the IAF. In the first decennium of this century the 'I' in IAF stood for integrated to signify the need to integrate different views and aspects when developing an architecture. In the next decennium I expect this first 'I' to stand for innovating, to signify the fact that the Innovating Architecture Framework will lead the way for future versions of the open standards. Therefore, I can say with conviction: IAF is here to stay!

I sincerely hope you enjoy reading this book, as much as I have enjoyed seeing the authors write it!

Nijmegen, Netherlands — Prof. Dr. H.A. (Erik) Proper

Preface

This book captures and communicates the wealth of architecture experience Capgemini has gathered in developing, deploying, and using IAF since its documentation in 1993. It intends to guide the reader through the corners and crevasses of IAF. We aim to help the reader understand why we have done the things we did to develop IAF specifically, and the IT architecture profession in the IT industry more in general. We hope we have achieved our objectives and readers are welcome to provide us with feedback. This is because we are sure we are not there yet. The architecture profession in the IT industry still needs a significant amount of time to mature. Just imagine how long it took architects in the building industry to come to the point where they are now.

Utrecht, The Netherlands

Jack van 't Wout
Aaldert Hofman
Max Stahlecker
Herman Hartman
Maarten Waage

Objectives of the Book

This book has two main objectives. The first is to explain the background and mindset behind IAF. As IAF usage becomes more widespread over the globe, more and more people need to understand its background and mindset to fully benefit from it. The second objective is to capture the body of knowledge we have been assembling since 1993. It is not uncommon to see the same question pop up on newsgroups and forums several times throughout the years. This book not only intends to provide answers to the most common ones, but should also provide answers to all those questions regarding IAF you have had through time, but never dared to ask.

Intended Readers

We have written this book with the following readers in mind:

(Potential) Architects who want to thoroughly understand IAF so they can use it in their environment. This does not mean that one can read this book and benefit fully from IAF without additional training. Architecture is a trade one has to learn together with colleagues, in real life.

Others who want to understand architects that use IAF. IAF architects have one thing in common. They apply IAF in their work. This means they use specific terminology, especially amongst themselves. People who want to know more about IAF's terminology and mindset are also invited to read at least a few parts of this book (e.g. Chap. 2 or Chap. 4).

Engineers who work with IAF deliverables. Engineers are working together with architects to create effective solutions that meet certain business objectives. Thus they might be interested in getting a better understanding of the thinking behind these IAF deliverables. We recommend taking a look at the physical artifacts & views (covered in Chap. 3) and IAF's interplay with solution development (covered in Chap. 4).

Participants of the Capgemini's Architecture Learning Program. Together with the training experience and this book, architects should not only be able to get up to speed even quicker but also better prepared than in the past. They will have a document that they can fall back on when they enter an area of IAF they have not been in a while. Especially Chap. 3 helps to get a better understanding of IAF's artifacts and views.

Experienced IAF architects. Of course experience IAF architects interested in learning more about the history of IAF or in specific IAF artifacts and their usage are invited to use the book as an extended glossary.

Structure of the Book

Throughout this book 'we' stands for 'we, as IAF architects' and thus might not be limited to just the authors – we hope we captured the thoughts and opinions of many people who use IAF in their daily work. That's why the book was extensively reviewed with a focus on those ideas that make IAF what it is now.

Chapter 1: *IAF background, value and strategy* explains why IAF was initially developed, what it has delivered, its added value, and how Capgemini intends to go on with the framework.

Chapter 2: *IAF's architecture* provides insight into the mindset and mechanisms that are part of IAF.

Chapter 3: IAF's *aspect areas explained* gives an overview of the most common architecture artifacts and views of IAF and shows what the content of the main elements of the framework are.

Chapter 4: *IAF in perspective with other frameworks and methods* elaborates on the different ways IAF can be used in projects, in combination with other tools, methods and frameworks.

Chapter 5: *Applying IAF and using its outcomes* shows the best ways IAF can be used to professionalize the architecture function in an organization.

Chapter 6: *Real life case studies* exemplifies the use of IAF through the description of a number of real life case studies.

Chapter 7: *The making of IAF* explains the history of IAF.

Acknowledgements

The Authors are pleased to pass on the following acknowledgements:

IAF would not have existed without the ongoing support of the *Capgemini University*. Through the years they have been sponsor for the development of IAF and the deployment material related to IAF. They have enabled internal business models that provided the architecture community with the means to develop their trade through the IAF. Key players from the university have been Stephen Smith and Régis Chassé.

Another group of people in Capgemini that have sponsored IAF through time is *Global delivery*. Especially Mark Standeaven and Bart Groenewoud have helped us come to where we are today, by sponsoring IAF and the architecture community.

Of course we also need to acknowledge all of our colleagues that have helped develop IAF since 1993. We apologize up-front for the names we have

forgotten. No offence is intended, it just is a fact that so many people have shared their experience through time. With this proviso, key contributors include:

- *From the USA*: Doug Houseman, Anne Lapkin, Meir Shargal, Hervé Bertacchi, Leigh McMullen, Mike Cantor, Bob Reinhold, Aaron Rorstrom and others;
- *From the UK*: Geoff Pickering, Mike Paulson, Andrew Macaulay, Kirk Downey, Stuart Curley, Taj Letocha, Colin Metcalfe, Gunnar Menzel, Ian Suttle, Andy Mulholland, Peter Truman, Stuart Crawford, Una Du Noyer and others;
- *From the Nordic countries*: Ivar Laberg, Mats Gejnevall, Stefan Olsson, Joakim Lindbom, Paul Carr and others;
- *From the Netherlands*: Stefan van der Zijden, Hans Baten, Dave van Gelder, Annet Harmsen, Mark Hoogenboom, Petra van Krugten, Jeroen Jedema, Arend Saaltink, Jaap Schekkerman, Raymond Slot, Sander Zwiebel, Bas van der Raadt, Daniel Eleniak, Peter Barbier, Frank van Ierland, Peter Kuppen, Erik Onderdelinden, Marco Muishout, Barry de Vries, Han van Loenhout, Paul Hillman, Arnold van Overeem, Pieter Hörchner, Ron Tolido, Jeroen Bartelse, Lucas Osse and others;
- *From France and Iberia*: Bernard Huc, Hyacinthe Choury, Jean-Christophe Salome, Jacques Richer, Philippe Andre, Patrice Duboe, Bjorn Gronquist, Jorge Villaverde Illana, Luis Gonzales, and others;
- *From Australia*: Peter Haviland, Mick Adams and others;
- *From India*: Nitin Kadam, Shrinath Rao, Inderjit Kalsi.

Finally we also want to thank the people that have directly contributed to the creation of this book. Erik Proper was our inspirator, without him this would not have become real life. Dave van Gelder, Colin Metcalfe, and Doug Houseman provided us with additional inspiration, especially in the areas around governance. Barry de Vries helped us out with ArchiMate specific topics. Dave van Gelder and Eric Onderdelinden provided specifics on TOGAF 8 and 9.

Contents

Chapter 1
IAF Background, Value and Strategy

1.1 What Is IAF? A Short Summary

IAF is Capgemini's architecture framework. It is a toolbox that contains processes, products, tools and techniques to create all types of architectures which are intended to shape businesses and the technology that supports it. Its development started in 1993. It continuously evolves as best practices are added to the framework. IAF has been used in thousands of engagements in most business sectors and many countries.

1.2 Reasons for Having IAF

To answer the question 'Why was IAF developed?' we need to understand why architecture has become such an important topic in the IT industry. In the 1980s life in the IT industry was relatively straightforward. There were centrally managed computer systems with 'dumb terminals' to do the work. Then departmental or mini computers started to show up. Short after that, personal computers and various integration aspects entered into the arena. The IT landscape became more and more complex. Part of the complexity was that we had to design solutions in terms of both software and hardware, whereas until then the mainframe always was implied, and only additional terminals, additional disk space, and additional processing power were to be considered. Thus we needed tools to manage this complexity. Architecture was the term that was coined to describe our activities to successfully manage the complexity. Architecture and Infrastructure seemed interchangeable notions – we still see some residues of that here and there – but the link with other aspects was obvious and needed a broader view.

Within Capgemini work began to formalize what we meant with architecture as part of a transformation program called Snowball that was intended to introduce client-server technology and iterative development into the organization.

Architecture was our approach used to explain how we defined which infrastructure was needed to support the business. The architecture development method

J. van 't Wout et al., *The Integrated Architecture Framework Explained*,
DOI 10.1007/978-3-642-11518-9_1, © Capgemini SA, 2010

(ADM)[1] became part of the training material in the transformation program. The ADM was originally developed in the UK and deployed worldwide. It demonstrated that architecture was a justifiable set of activities in the overall business transformation process. It provided us with standardized mechanisms and templates to communicate our architecture development work. The ADM as an infrastructure architecture method taught us how very beneficial it was to have a common language, approach and set of tools to do our work.

Another reason for having IAF was that projects were getting bigger and more complex. The requirement for transnational staffing of the projects also increased. It was hard to get the right architect in the right place, as they all did their architecture work their own way, except for the infrastructure architecture part, which was better aligned across the architects in this area. Within Capgemini we realized we needed to standardize more of the architecture activities that we performed.

An additional reason for having IAF was that it would provide us with a means to communicate what we did, and align ourselves with other professions that were working on the projects. IAF should not only help us structure our own work but also allow us to innovate the way we do architecture.

We knew we could not do it all at once, so we needed a 'framework' in which we could position the various architectural subjects to cover. We studied the available industry frameworks and architecture approaches and could not find anything that suited our requirements. Thus we decided to develop our own, and the Integrated Architecture Framework (IAF) was born.

1.3 The Value of IAF

For more than 16 years, since 1993, IAF has proven its value. The framework has been used in over 3000 projects worldwide, ranging from chip design to the creation of complete new lines of business for global players. It has been formally adopted by many organizations outside of Capgemini. Implicitly it has been adopted by even more organizations, as they requested us to create an IAF based architecture for them. In doing that we trained their staff to use IAF.

IAF has an elemental line of thought that has proven to be valuable and robust in many different types of projects. The core mechanisms that form IAF's basis are sacrosanct. Traceability of decisions, basing decisions on principles, and reducing complexity by separating concerns are three of these mechanisms that can be applied in any situation.

IAF has also proven to be flexible. It can be adopted by all the different types of projects in which architecture is required. It has been used standalone and in

[1] Despite having the same acronym this ADM is not related to the TOGAF ADM.

combination with other methods and approaches. This works so well because IAF was designed as a flexible toolkit, and not as a rigid cookbook. Of course this flexibility also has its drawbacks. Since IAF is not prescriptive, it is more difficult to initially understand and use because users have to select their own tools. They are not told what to do in a step by step fashion.

One of the core principles of the framework, a traceable link between the business and IT, has proven its value to our customers many times. They could understand and explain why specific investments were being made. It was relatively simple to understand the impact of change in a specific area because of that traceability. One was able to trace back which parts of the business would be impacted by a specific change in IT, for example in case of an upgrade of a server, package or database management system.

Another factor that demonstrates IAF's value is productivity. It is not uncommon to expect an IAF architect becoming productive within hours after onboarding a new project. They understand the 'language' used, and can almost immediately start to contribute to the result. For example, a colleague from the UK once joined a Dutch project and indeed spotted a flaw in the network architecture within half an hour. Second opinions on architectures created in one country are sometimes executed by colleagues in another country to ensure quality. This could not be done effectively without our common IAF language.

IAF is also based on practical experience and knowledge. All Capgemini architects are encouraged to share their best practices. All version upgrades have been based on input from real life practitioners. IAF has always focused on describing what was needed to be done, as well as providing guidance on how it could be done. This makes it one of the most sophisticated, practically proven, and documented architecture frameworks in the IT industry.

Because of its focus on architecture content, IAF has proven to be a valuable tool in combination with TOGAF[2]. TOGAF 8 focuses on process, and advises to incorporate other frameworks for the content. This results in a good fit between TOGAF and IAF which led to the incorporation of IAF elements into the recently released TOGAF 9. How TOGAF (versions 8 and 9) and IAF fit together is explained in more detail in Chap. 4.

1.4 IAF Strategy

Capgemini promotes open standards. We participate in different open standards bodies like The Open Group. We believe that sharing knowledge and experience is the best way to improve the results of the IT industry. Does this mean that we have abandoned IAF, since it's not an open standard? We can

[2] An industry standard architecture framework of The Open Group.

answer this with a clear 'no'. There are a number of reasons why we intend to continue with IAF. The three most important ones are elaborated below.

The most important reason for continuing with IAF is the fact that many of our customers have adopted the framework. They want to protect that investment. They expect support from Capgemini, and we intend to deliver that support.

A second reason for continuing with IAF as a non open standard is the fact that Capgemini wants to continue to play a leading role in increasing the maturity of the architecture profession. To do that Capgemini needs a vehicle that can be used to improve and test architecture tools under their own control. As IAF is in our own control we intend to use it as an innovation platform that continuously provides input to open standards bodies like The Open Group.

Thirdly we want to protect our investment in what matters for us: our architects.

1.5 A Short Recap of IAF Versions

Our architecture innovation investment is demonstrated by its versions. Here they are:

Version 1 was developed in 1994 and expanded until 1998. Its main focus was to build the fundament of the framework. That was done by harvesting the best practices we had and expanding them where needed. We started from the infrastructure side and expanded into creating tools for client – server architectures as well as for security and governance.

Version 2 was developed in 1998 and used until 2000. Its main focus was to provide full support for information systems architecture, and laid foundations in other areas.

Version 3 was developed in 2000. It incorporated best practices from Ernst & Young, which had recently been acquired. Version 3 was used until 2006.

In 2006 we completed the framework by filling all the gaps that were still there. Version 4 was introduced. Because the framework was now complete, we decided to create a new and complete IAF reference manual.

In 2009 we injected some additional best practices into the framework and labeled it version 4.5. This is the current version, which is the basis for this book.

Chapter 2
IAF's Architecture

2.1 Introduction

Explaining IAF must start with the basics, the core elements of the framework. Actually, our aim is that you understand IAF's own architecture. Once you understand IAF's structure and underlying ideas it will be easier to apply it to the specific situation you are in. This chapter is all about helping you understand the architecture behind IAF.

IAF applies the basics of general construction methods. All construction methods have a common approach. They address questions in a specific order. The order is expressed as interrogative pronouns: Why, what, how, with what. There are reasons for the sequence of the questions.

If you don't understand why you need to do something, you can't work out what needs to be done. You have to understand the context before you can start working effectively. Studying the context helps define the scope and objectives, and thus helps with staying focused. So understanding 'why' is the first thing to do.

If you don't understand what needs to be done (the requirements), you will never be able to craft the solution. You have to define the requirements before creating the solution. Once you understand the requirements you are often able to produce a concept of what is needed. So after understanding why you are doing something you need to address the 'what' question by defining the requirements (both functional and non-functional) and getting a concept of the solution.

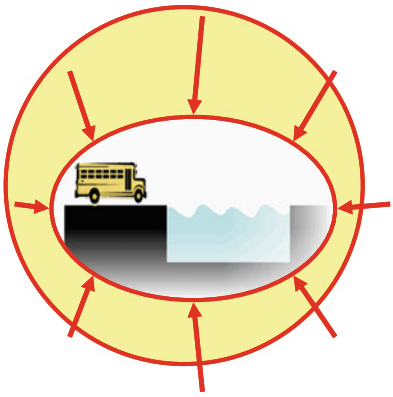
With a good understanding of the requirements you should be able to design a logical solution to the problem, answering the question 'How will the solution look like'. In other words you structure the solution. You create the solution on a logical level to enable flexibility. All architectures will take time to implement. Circumstances as well as real life physical solutions change over time. By describing the architecture on a logical level you will be able to deal better with new insights and adapt it at the moment you physically implement a specific part of the architecture.

J. van 't Wout et al., *The Integrated Architecture Framework Explained*,
DOI 10.1007/978-3-642-11518-9_2, © Capgemini SA, 2010

Finally you can decide with what physical things the solution can be realized. This implies allocation of physical things to the logical solution.

A practical example will demonstrate how this all works.

Imagine you are living in a city that has been built on both banks of a large river. There are a number of bridges crossing the river, but traffic has grown through time and there are constant traffic jams. In two years time the city is planning to host a big cultural event, and it wants to show a modern, young city that is well prepared to handle all the tourists that are visiting the event. So, the scope and context of the problem become clear. A new bridge needs to be built in the city within 2 years and it should reflect the 'young, modern' look that the city wants. The 'why' question should be clear by now.



Now we need to address the 'what' question. We need to find out what the bridge has to do. Does it have to carry pedestrians? Does it have to carry cyclists? Does it have to carry cars and trucks? Does it have to carry trains? How many of each type does it have to be able to carry during average and peak hours of usage? Which types and numbers of boats have to be able to pass under the bridge? What is the river's required water flow capacity and what would happen if that were reduced if the bridge were to have many pillars? What does the patron want it to look like to reflect the young, modern look? Once these types of questions are answered we have a clear understanding of the 'what' question. We know the requirements and are able to craft the solution.

The third main step is creating a logical model of the solution, thus answering the 'how' question. Now there is a slight snag here. Real life has shown us that there never is one solution. There always choices, and all solutions are trade-off of different aspects like cost and performance. So, in effect, if you think there is only one solution, you might have to think a little longer. There are solution alternatives that have to be considered. In this case we can identify different types of bridges that could provide sufficient river-crossing capabilities. Some of them will fit the principles we have defined regarding 'young and modern' better than others. Others might not be able to be built within 2 years. We could even

consider a temporary bridge, which would probably make a financially interested stakeholder happy, could be finished in a short period of time, avoids difficult technological studies on water flows but the patron who dreams of grandeur and eternal fame would be somewhat disappointed. In the end we present the best alternatives to the stakeholders so they can choose.

Now real life physical architecture can start. We choose the materials we need for each part of the bridge. By this time the cooperation with various construction engineering disciplines has begun. First we jointly determine what is needed for the foundation. Then we select the rebar materials and concrete. We define which steel is needed for the deck. We create the specifications the construction teams need to create the design of the bridge. We create other visualizations of what the real life solution will look like. The 'with what' question has been answered. The architecture has been designed. It can be handed over to the engineers who are actually going to build the bridge. The architect will remain involved in order to solve issues that come up during the construction phase and which might affect the starting points or principles that underlie the entire idea behind the bridge.

We are going to explain IAF's own architecture using the 'why-what-how-with what' approach explained above.

2.2 The Context: The "Why" of IAF

2.2.1 Vision

The vision we had when we started to develop IAF consisted of a number of elements. The most important goal was that we wanted to be able to provide world-class services to our clients, and were convinced that architecture was key in this. The ever increasing complexity and risk in the engagements we were working on made this obvious. We also needed a robust and mature toolset to deliver a constant quality of architecture services and a consistent experience where a client is engaged many times over a period of time. The toolset had to successfully address the alignment of business and IT. It had to be independent of a specific architect: the way we work and approach architecture should be common across all architects. In this perspective we now speak of IAF as a

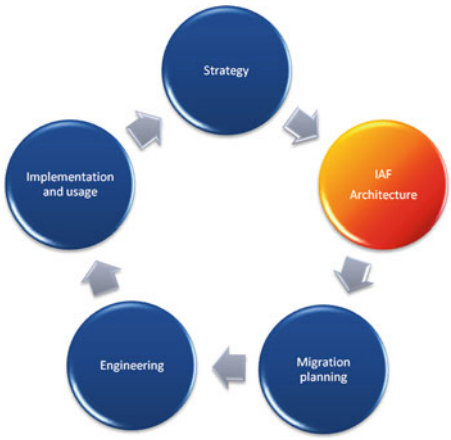
'design school', with very specific style, approach and characteristics that we feel make a difference: for us as Capgemini being a global company delivering services, and for our clients ranging in size from the Fortune-500 to the local medium enterprise. In addition to this it had to provide a platform on which we could expand and improve the architecture profession. We decided up-front that it had to be based on real-life experience, and not on pure theory. We knew that we were embarking on a journey with an uncertain destination, and wanted to base that journey on things that had been proven in the field.

2.2.2 Scope

The original slogan we used in regard to IAF was: 'For a system to work as a whole, it must be architected as a whole'. This slogan nicely depicts many elements that are part of the IAF scope.

The first striking word is 'system'. What do we mean by 'system'? We have small projects that upgrade existing applications to very large programs in which we support complete post merger integrations of global companies or companies constantly reinventing themselves. All these types of projects potentially require some form of architecture. They all create or change a 'system'. Thus IAF's scope needs to cover all types of architecture engagements at all levels in an organization. Enterprise level – spanning business units –, Business unit (business domain) level and Solution level. Enterprise and Business unit level are commonly meant for supporting planning activities. Solution level is aimed at guiding the engineering of the solution.

The second important word in the slogan that is related to IAF's scope is 'whole'. Capgemini always approaches IT from the viewpoint that it has to support the business. This implies that architecture should always be justified in business terms and traceability to business requirements. Even when we are architecting a purely IT system, the business should provide the objectives and drivers for the architecture.



The third word in the slogan that is related with scope is 'architected'. IAF is aimed at the architects profession, and should only describe things for which the architect has the main responsibility. Therefore IAF does not provide support for the creation of an organization's vision, mission and strategy. These are defined as input for IAF. Some basic input that the

architect usually derives from the vision, mission and strategy is contained in IAF to assist architects in collecting input if it is not there. On the other end, migration planning has also been put out of scope of IAF, because that activity is not the architect's main responsibility. In general, migration planning is a joint exercise with the engagement manager as the responsible person. IAF also tries to avoid overlap with other professions like business analysis and engineering. There are many touch points between the architects and these professions, just like there are many touch points between architects and engineers in real life construction architecture. This creates a gray area. Where does the architect stop and the engineer start? The most pragmatic answer to this question is that there is a difference in focus. The architect focuses more on how the 'system' fits into its environment. The engineer focuses more on the internal structure of the 'system', within the boundaries that have been set by the architect. In other words: the architects focuses on the behavior and non functional requirements ('black box') while the engineer on the internal construction ('white box').

2.2.3 Objectives

Through time IAF's objectives have not changed much. The framework has evolved due to increased understanding of architecture in the IT industry as a result of the pursuit of the objectives.

IAF's core objective is to provide a common way of architecting. Originally it was intended to do this within Capgemini. Nowadays more and more organizations also adopt this common way of working.

IAF also must provide a communication framework to achieve the common way of architecting. This 'common language' needs to be adopted throughout our organization and across the regions where we operate, particularly when serving global clients such as General Motors. This objective has proven to be difficult to realize – but we feel we succeeded. One word can have the same definition in multiple countries and still be perceived to be different. An example that is popular in Capgemini is the confusion we had around the term 'Business event'. In the Netherlands this was perceived as 'something that *can* happen in an organization'. In the UK it was perceived as 'something that *has* happened in an organization'. Therefore it was very understandable for the Dutch to propose basing a to-be architecture on business events. The UK architects tended to disagree. Their counterargument was: 'How can you base a to-be architecture on something that has happened in the past?'

Sometimes we have even invented new words to resolve terminology discussions. A nice example is the term 'archifact' that was used in IAF version 3 because we could not come to an agreement at that time on the usage of the term