

BEST PRACTICE

ASL[®] 2

A Pocket Guide



APPLICATION SERVICES LIBRARY



ASL® 2 – A Pocket Guide

Other publications by Van Haren Publishing

Van Haren Publishing (VHP) specializes in titles on Best Practices, methods and standards within four domains:

- IT and IT Management
- Architecture (Enterprise and IT)
- Business Management and
- Project Management

Van Haren Publishing offers a wide collection of whitepapers, templates, free e-books, trainer materials etc. in the **Van Haren Publishing Knowledge Base**: www.vanharen.net for more details.

Van Haren Publishing is also publishing on behalf of leading organizations and companies: ASLBiSL Foundation, CA, Centre Henri Tudor, Gaming Works, IACCM, IAOP, IPMA-NL, ITSq, NAF, NgI, PMI-NL, PON, The Open Group, The SOX Institute.

Topics are (per domain):

IT and IT Management

ABC of ICT
ASL®
CATS CM®
CMMI®
CoBIT
e-CF
Frameworkx
ISO 17799
ISO 27001/27002
ISO 27002
ISO/IEC 20000
ISPL
IT Service CMM
ITIL®
MOF
MSF
SABSA

Architecture (Enterprise and IT)

ArchiMate®
GEA®
Novius Architectuur Methode
TOGAF®

Business Management

BiSL®
EFQM
eSCM
IACCM
ISA-95
ISO 9000/9001
OPBOK
SAP
SixSigma
SOX
SqEME®

Project, Program and Risk Management

A4-Projectmanagement
ICB / NCB
ISO 21500
MINCE®
M_o_R®
MSP™
P3O®
PMBOK® Guide
PRINCE2®

For the latest information on VHP publications, visit our website: www.vanharen.net.

ASL[®] 2

A Pocket Guide

Yvette Backer, Remko van der Pols



Colophon

Title:	ASL® 2 – A Pocket Guide
Authors:	Yvette Backer and Remko van der Pols †
English translation:	Mark Smalley, ASL BiSL Foundation
Reviewers of the	Greet Mattheus (Ordina)
Dutch version:	Machteld Meijer (Maise) Frank van Outvorst (The Lifecycle Company) Lex Scholten André Smulders (Ordina)
Text editor:	Steve Newton
Publisher:	Van Haren Publishing, Zaltbommel – NL www.vanharen.net
ISBN Hard copy	978 90 8753 643 5
ISBN eBook	978 90 8753 803 3
Edition:	Second edition, first impression, February 2014
Layout and design:	CO2 Premedia, Amersfoort – NL
Cover design:	CO2 Premedia, Amersfoort – NL
Copyright:	© Van Haren Publishing, 2013, 2014

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, neither the author nor editor can accept any liability for damage caused by possible errors and/or incompleteness in this publication.

TRADEMARK NOTICES

ASL® and BiSL® are registered trademarks of the ASL BiSL Foundation.
ITIL® is a registered trademark of AXELOS Limited.

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.

Copyright protected. Use is for Single Users only via a VHP Approved License.
For information and printed versions please see www.vanharen.net

Foreword

This Pocket Guide has been updated with a summary of the latest insights in the new version of the Application Services Library, ASL 2.

Application management is currently undergoing very interesting changes. The professionalization, the componentizing of services, the integration of services, and the degrees of freedom have all grown enormously, making the profession of application management not only more complex, but also more challenging.

Despite these developments, lessons learned in the past are still valid. The core of application management hasn't changed much at all. The combination of wisdom and past experience, and the ability to adopt, and adapt to, modern developments, determines whether an application management organization will survive.

You will discover this in ASL 2. The structure of this publication is unchanged, while the changes in the content address today's challenges.

You will see this progression in the case contained in this Pocket Guide. We have taken a historical perspective by transforming the organization in the previous Pocket Guide. Six years have passed. Some best practices have been lost but are rediscovered. And new best practices are developed.

Our intention is that you will not only learn from this Pocket Guide, but also enjoy reading it.

Many thanks to the reviewers Greet Mattheus, Machteld Meijer, Frank van Outvorst, Lex Scholten and André Smulders for their contributions.

Finally, a special mention for Eilko Bronsema and Otto Strijker who both work for Wegener. They kindly provided us with insight into IT developments at a modern newspaper publisher.

Yvette Backer

Remko van der Pols †

Contents

1	INTRODUCTION	11
1.1	Background	11
1.2	Target audience	11
1.3	VGK and the case	12
1.4	Structure of the book	12
2	APPLICATION MANAGEMENT AND ASL 2	13
2.1	What is application management?	13
2.2	How can application management manifest itself?	15
2.3	Levels of management	17
2.4	ASL and other frameworks	18
2.5	Best practices and implementation of ASL 2	19
3	ASL 2 FRAMEWORK	21
3.1	ASL 2	21
3.2	Core issues and challenges for application management	26
3.3	Explicit messages and starting points	29

4	APPLICATION SUPPORT CLUSTER	33
4.1	Introduction	33
4.2	Use support	36
4.3	Configuration management	39
4.4	IT operation management	41
4.5	Continuity management	46
5	APPLICATION MAINTENANCE AND RENEWAL CLUSTER	51
5.1	Impact analysis	53
5.2	Design	57
5.3	Realization	60
5.4	Testing	63
5.5	Implementation	68
6	CONNECTING PROCESSES	73
6.1	Introduction	73
6.2	Change management	73
6.3	Software control and distribution	77
7	MANAGEMENT PROCESSES	81
7.1	Introduction	81
7.2	Contract management	84
7.3	Planning and control	89
7.4	Quality management	93
7.5	Financial management	96
7.6	Supplier management	99
8	APPLICATION STRATEGY	103
8.1	Introduction	103
8.2	IT developments strategy	108
8.3	Customer organizations strategy	109

8.4	Customer environment strategy	109
8.5	Application lifecycle management	111
8.6	Application portfolio management.....	111

9 APPLICATION MANAGEMENT ORGANIZATION STRATEGY .. 115

9.1	Account & market definition	117
9.2	Capabilities definition	118
9.3	Technology definition	119
9.4	Supplier definition	119
9.5	Service delivery definition	120

10 IMPLEMENTATION AND USE 125

10.1	Implementation and configuration.....	125
10.2	The framework and reality	126
10.3	The secret of best practices	126
10.4	Scenario's and implementation	128
10.5	Starting with ASL.....	130

Appendix 1	Case study VGK/ISPM	133
Appendix 2	ASL BiSL Foundation	139
Appendix 3	Sources and more information	143
Appendix 4	The ASL 2 process model	145
Index	147

1

Introduction

■ 1.1 BACKGROUND

ASL 2 was introduced in 2009 as an evolutionary update of ASL, the process framework for application management. The Pocket Guide also had to be updated and this is the result. It is intended to be an accessible and simple introduction that provides insight into the theory and practice of application management and the ASL 2 framework. It is an easy-to-read yet thorough introduction to what application management comprises and what the ASL 2 framework looks like.

■ 1.2 TARGET AUDIENCE

This Pocket Guide is intended for managers, boards of directors, students, staff members and, of course, everybody who is involved with application management. Some affinity with information provisioning or IT would be handy, but this is by no means essential. We have limited our use of technical terms, and have explained those that we have used. Please let us know if we have missed anything. The most important terms are explained at the end of the book.

■ 1.3 VGK AND THE CASE

In order to make the content light and illustrative, we have used a case throughout the whole book. The case is about the IT organization ISPM, the new name for the VGK organization. This organization features in the case in the previous version of the ASL Pocket Guide. We liked the idea of introducing an historic perspective. The organization is fictitious, but of course based on real life. In appendix 1 there is a description of the case.

■ 1.4 STRUCTURE OF THE BOOK

We start, after the introduction, with a chapter explaining application management. What are the main parts of application management and in which permutations can they occur? ASL 2, Application Services Library, is a process model for application management and so we also introduce ASL 2.

In chapter 3 we discuss the ASL 2 framework in detail and describe the content of the framework. In this chapter we sketch current developments that are relevant for application management and describe the challenges for the coming decades. We also suggest how you could deal with these developments.

Chapters 4-9 describe the various process clusters of ASL 2.

The book also includes some appendices. Appendix 1 contains a description of the case and appendix 2 gives some background information about the ASL BiSL Foundation. Appendix 3 provides an overview of the references and points you to additional sources. And finally, the complete framework is described in appendix 4.

2

Application management and ASL 2

■ 2.1 WHAT IS APPLICATION MANAGEMENT?

The first topic in a book about ASL 2 has got to address application management. We use Looijen's model in order to clarify the term application management. Looijen discerns three forms of IT management, three forms of tasks related to the support, maintenance and renewal of information provisioning in the broadest sense (figure 2.1). These three forms are described below:

- Business information management: realization of adequate use and renewal of the information provisioning from a business perspective;
- Application management: support, maintenance and renewal of applications (information systems);
- Infrastructure management: support, maintenance and renewal of the infrastructure.

These three forms are described further in the following paragraphs.

Infrastructure management supports, maintains and renews the infrastructure. The infrastructure comprises all of the hardware (computers, disks, displays), the communication structure (networks) and system software (such as operating systems) that are needed to

run information systems (applications). PC's, computers (servers), networks, storage etc., but also database management systems, compilers and drivers can be part of the infrastructure.

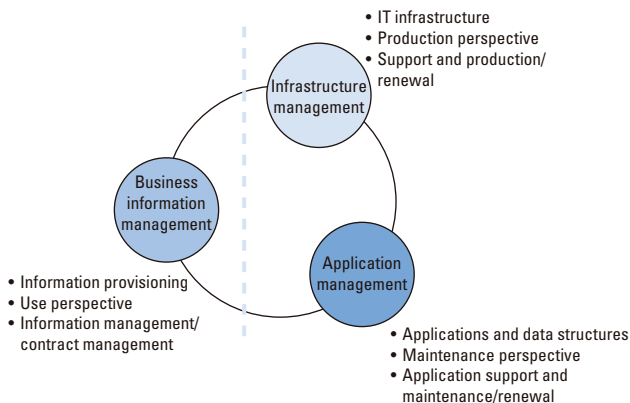


Figure 2.1 Triple management model (Looijen and Delen)

Application management concerns itself with the support and change (maintenance or integration) of applications (information systems) and data structures. This includes activities such as design of applications, coding, and testing. So this book addresses this profession and its framework: ASL 2.

Business information management covers the demand side of information technology (IT) / information provisioning. Business information management describes the user organization's responsibilities to support the use of information provisioning, and to maintain and manage information provisioning. BiSL is the framework that describes this domain.

■ 2.2 HOW CAN APPLICATION MANAGEMENT MANIFEST ITSELF?

In normal language, application management is the support and change of application software and data structures. ASL is a process model for application management and provides a framework for application management activities.

But despite these tasks being defined and described by such a model, there are various ways in which these tasks actually manifest themselves in practice. This is due to the significant impact of the underlying technology, the way the software is provided, the kind of services provided and the way it is charged and managed.

Software comes in various forms, such as a standard package that is ready for use, a configured package, a custom-made system, or an application that is built from various software components or separate components. These factors usually influence how application management is managed.

The way a supplier of packaged software manages applications differs from that of a supplier of custom-made services. A supplier of point solutions needs other competences than an application integrator (this is an organization that aligns and connects components of the information provisioning). The nature of the service is therefore an important variable for organizing application management.

In addition to the nature of the service, there are other influential variables (see figure 2.2). Cost charging, for instance, determines control and reporting to a strong degree.

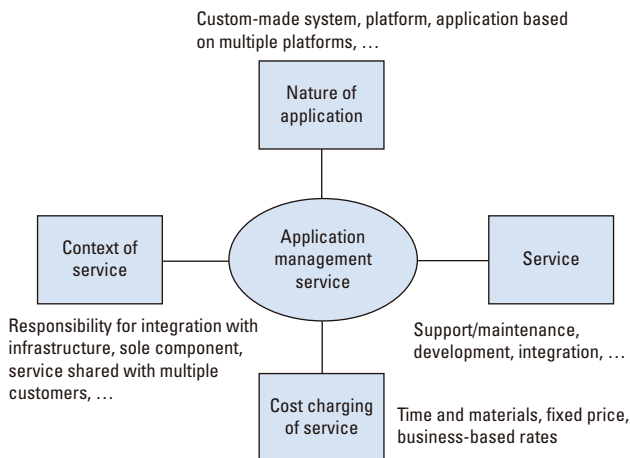


Figure 2.2 Variables for organizing application management

The technology also has an impact, including the impact on the content of processes and the tools that are used. The waterfall approach uses forms of design that do not occur with the Agile approach, where user stories are used.

Approaches change. Today's approach is modern but in a few years' time it will be called old-fashioned and ready for replacement by a new approach.

We design ASL 2 to accommodate the use of a wide range of current and future approaches. This means that the technology, development approach and methods have less impact than the previous topics.

But there are still many ways to organize application management. ASL 2 supports these variations with the freedom to choose various options.

The model itself has been kept simple. Best practices can be used to achieve the actual organization of application management, for instance the best practices that are freely available at the ASL BiSL Foundation.

■ 2.3 LEVELS OF MANAGEMENT

Application management comprises more than operational activities: ASL 2 also discerns between application management activities at the managing and strategic levels.

The tasks at the operational level deal with creating and changing the applications according to the users' needs, and with supporting the day-to-day operation and use of the applications.

Activities at the managing level control the operational service in order to comply with customer agreements and organizational preconditions regarding topics such as capacity, finance, agreements with suppliers and internal quality.

The strategic activities focus on the application(s) and the service organization that are needed for the next three to five years, and the development of scenarios to realize these changes.

The importance of ensuring alignment of the tasks at the various levels of management is more important now than in the past. Due to the often critical nature of the information provisioning and how it is integrated with business processes, the possibilities of the application

and the degree and speed of its change also determine the level to which business processes can be changed.

Insight into the strategy of the user organization is needed in order to ensure that the applications will support the business processes in three years' time and that the correct route is followed when changing and improving the applications. The investment possibilities are usually limited, particularly for dealing with shortcomings in applications.

Service organizations that take these aspects into consideration when supporting and maintaining applications are of added value to the customer. It is self-evident that the service is provided according to the contract, but the service organization that takes account of the future has a stronger competitive position.

■ 2.4 ASL AND OTHER FRAMEWORKS

ASL is a framework that describes tasks that are needed (and to some degree explicitly executed) for the support, maintenance and renewal of applications. The number 2 after ASL indicates that this is the second version of the framework.

ASL 2 is not the only process framework: there are many others. For example ITIL, COBIT or BiSL. ASL 2 differs from these frameworks; ASL 2 focuses on application management. This focus on support, maintenance and renewal of information systems and applications differentiates it from other frameworks.

In addition, ASL 2 takes into consideration the various manifestations of application management and related requirements.

■ 2.5 BEST PRACTICES AND IMPLEMENTATION OF ASL 2

ASL 2 is a framework as well as a public domain library for application management. This library comprises hundreds of best practices: process descriptions, procedures, templates, checklists and other documents that other organizations have donated. The best practices are freely available from the ASL BiSL Foundation¹, and can be adapted to individual situations.

This means that ASL 2 can be used to implement a working and tailored quality system quickly and simply, because the components are already available and only need to be tailored to the specific organization.

¹ See appendix 2 for more information about this organization.

3

ASL 2 framework

■ 3.1 ASL 2

ASL 2 comprises three levels and, in total, six process clusters. There are process clusters at the operational, managing and strategic levels. Each process cluster comprises a group of clearly interrelated processes with a collective goal. The six process clusters are (see figure 3.1):

- Application support;
- Application maintenance and renewal;
- Connecting processes;
- Management processes;
- Application strategy;
- Application management organization strategy.

Application support

Applications are built and maintained in order to be used. ‘Used’ meaning: installed on one or more computer systems and started and operated for years, so that users in the organization can (continue) to use the applications. The execution of these activities and ensuring their continuity in the future is part of the responsibilities of the application management organization. In ASL terminology, these activities are called the Application support activities within

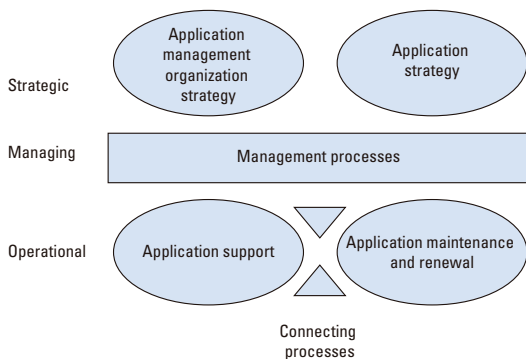


Figure 3.1 The ASL process clusters

application management. These activities are often underestimated, but are very important: if an application doesn't work, the business process that the application supports, also won't work. This is why the processes *Continuity management*, *Use support*, *IT operation management* and *Configuration management* (figure 3.2) are very important for the quality experience.

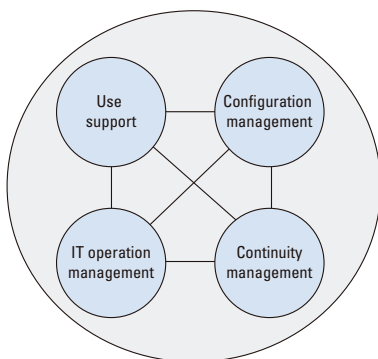


Figure 3.2 The process cluster Application support

Application maintenance and renewal

The organizations that the applications support, and these organizations' environments, both change. This means that business processes change and that the applications that support these processes also have to change. This is why application maintenance is needed. This maintenance can vary in size and nature. Sometimes it involves small changes to the application, such as making a change to a screen or a report. But sometimes the changes are so fundamental that a substantial part of the application has to be modified. The size of the change can vary from a one or two percent to a substantial percentage of the initial value of the application. This is why ASL 2 contains the process cluster Application maintenance and renewal with the processes *Impact analysis*, *Design*, *Realization*, *Testing* and *Implementation* (figure 3.3).

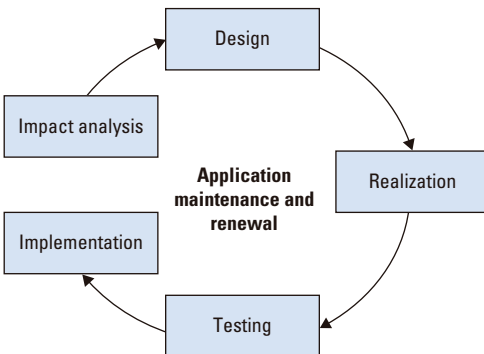


Figure 3.3 The process cluster Application maintenance and renewal

The Connecting processes

The third process cluster at the operational level is the process cluster Connecting processes. There are several relationships between the process clusters Application maintenance and renewal, and

Application support, and these relationships often interfere with each other. This is why processes are needed to organize this alignment: namely the Connecting processes. There are two connecting processes: *Change management* and *Software control and distribution*.

Management processes

The process cluster Management processes is positioned between the strategic and operational process clusters. Customers expect controlled and managed execution of the application management processes. Unfortunately it is not unusual to be confronted with higher costs than agreed, delayed delivery and other application management agreements not being honored, resulting in dissatisfied application management customers.

In order to prevent these mishaps, ASL has defined the Management processes. The processes are *Planning and control*, *Financial management*, *Contract management*, *Supplier management* and *Quality management* (figure 3.4). These processes manage the Application support processes, the Connecting processes, and the Application maintenance and renewal processes. This is important because it prevents the products of Application maintenance and renewal being ‘thrown over the wall’ to Application support. It also helps to prepare Application support during the maintenance activities.

Application strategy

One of the big problems in application management is lack of longer term planning with a horizon of approximately five years. There is hardly any innovation. There are often thoughts about wonderful new application landscapes, but the existing applications are usually left

out of the picture. This limits the possibilities to invest in existing applications, despite the fact that more than eighty percent of the applications will still be in use in five years' time.

Application strategy is the cluster of processes that creates vision and policy for the future of the applications and the information provisioning. This is executed in close collaboration with business information management and infrastructure management.

The goal is to ensure that the applications provide adequate support for the business processes during the next three to five years. To achieve this goal, the business' demands are translated into pragmatic and executable improvement or renewal of the applications.

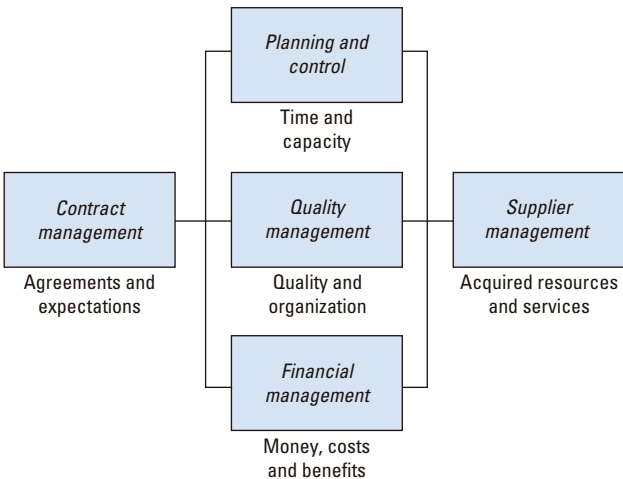


Figure 3.4 The Management processes

Application management organization strategy

But it's not only the applications that have to be kept up to date. The application management organization itself has got to have a vision of where it wants to be in three years' time. What kind of services does it want to provide, using which technology, how does it want to operate, and within which market segment? Of course customers have to pay for the IT organization's services, but it's no longer acceptable to have the customer pay for the IT organization's mistakes.

This is why it is important that an application management organization decides in good time which direction it wants to follow.

Application management organization strategy is the process cluster that deals with these topics.

The process clusters are detailed in the following chapters.

■ 3.2 CORE ISSUES AND CHALLENGES FOR APPLICATION MANAGEMENT

Various developments during the past decade have changed the starting points and issues for application management to a large degree. Some of these developments are described below (see figure 3.5).

A demand organization has arisen

A demand organization for the information provisioning has arisen within the customer organization. In many organizations the demand organization has become explicit. The demand organization acts as the customer for the IT supplier and is usually mandated to take decisions. Increasingly, this organization takes business-based decisions instead

of following the way IT thinks. The policy and the decisions are, therefore, based less on IT principles and IT thinking than was the case in the past.

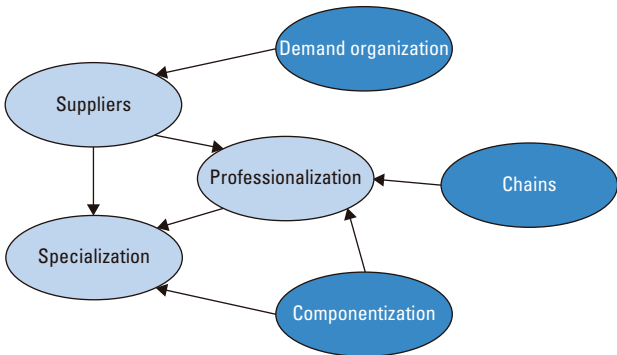


Figure 3.5 Developments that influence application management

IT departments have become suppliers

Application management and infrastructure management have become forms of service, and their services are defined in a customer-supplier relationship. These are tasks that are no longer by definition executed by the organization itself; they are outsource-able services. Where IT once used to be a staff department with the authority to take autonomous decisions and set policies for information provisioning, now it has become a (replaceable) supplier of IT services. Outsourcing, offshoring, professional service – these have become common terms within IT services. This has changed the role of the internal IT organization fundamentally. Organizations now hardly ever accept that the internal IT organization can determine the policy for information provisioning or IT.

Componentization and standard solutions

During the last decade, the development and use of standard solutions have become common practice. Even new applications are no longer built from scratch but comprise standard solutions or platforms with specific components, augmented by custom software. Integration has, therefore, become a very important topic for application management. Another consequence is supplier specialization: both technologies and techniques have become so complex that specialization in multiple technologies is very difficult for most people and IT organizations.

IT organizations continue to specialize

The aforementioned developments have led to increased specialization within IT organizations. Customers expect that suppliers will be experienced in their services. The combination of the adoption of commercial practices and the ever growing complexity has resulted in IT organizations/departments specializing in parts of the services, technologies and/or market segments. Use of (sub)contractors has become normal practice.

Commercialization and professionalization

These developments have made commercialization and professionalization a necessity, a requirement, but also normal practice. Because IT organizations have become suppliers, they need to be conscious of running their operation as a business. They have to decide which technology to use and the kind of service they want to offer. Clearly defined borders with customers go hand in hand with a commercial, business-like attitude. And professionalization is also required in order to survive the threats posed by the competition.

Chain computerization

A final and very visible development is chain computerization, in other words the end-to-end connection of the information provisioning

of different organizations. An important consequence of this development is that control is more ‘vague’; the principle ‘you never decide alone’ has arisen. This environment does not automatically respond to the internal release policy of one of the organizations in the chain. Sometimes components in the environment have releases that conflict with the interests of other organizations. In a world of chains, the environment is a stabilizing and yet sometimes inhibiting factor.

■ 3.3 EXPLICIT MESSAGES AND STARTING POINTS

The first version of ASL anticipated many of these developments. Its messages were:

- A single point of contact: service team;
- Clear agreements: Service Level Agreement;
- Future perspective: proactive IT support.

These have not changed.

But the new version of ASL also pays special attention to:

- Flexibility in process implementation;
- Acting in a service chain;
- Proactiveness.

Flexibility in process implementation

Standardization and componentization lead to specialization. Not only in terms of technology, but also in terms of services. Control mechanisms can demonstrate strong differences.

In practice, unfortunately, frameworks are often applied in a dogmatic and theoretical way, and frequently implemented dictatorially and according to formalistic principles.

Processes are implemented in an instrumental and almost standard way, and the most important and often implicit driver in the actual implementation is to prevent people from making errors.

The customer or manager is seldom asked about their personal opinion as to what is essential; employees are seldom regarded as intelligent and capable people (who can learn – and sometimes have to learn – from their errors) and seldom are other goals defined aside from preventing process errors.

ASL 2 recognizes flexibility in the way processes are designed and implemented. In addition to explicitly supporting this with parameters for implementation, the core values that play a role during delivery are essential. Sometimes the service has to be cheap, sometimes flexible, and sometimes reliable. This leads to different implementations. This is why there are best practices, which can – and should be – adapted to the specific situation.

ASL 2 also comprises the Application management organization strategy processes, in which the core values that an organization has or strives to achieve can be considered, together with which part of the market the organization has the capabilities and ambition to service, and how any necessary change can be achieved.

Acting in a service chain

Another consequence of specialization is that an organization (or department) has got to collaborate with other parties, usually with other core values and other processes. Fragmentation of services is standard. Each organization is part of a constellation of suppliers and customers. The position within the constellation can also differ. This is even the case when the specific application management organization is part of a large IT organization. This is due to differing core values

causing people to take their own department's perspective. Integration of services is the big issue.

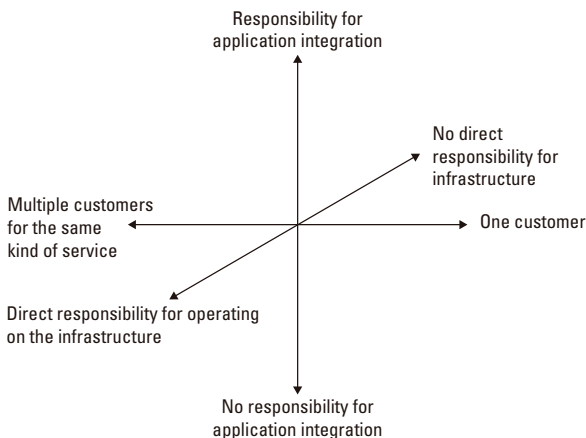


Figure 3.6 Design/implementation factors

Proactivity

Proactivity was an important message in the first version of ASL. Now it is even more important. Proactivity manifests itself in many ways:

- The service has to be known and deliverable before the customer asks for it;
- It's not important whether the SLA is achieved (the customer expects it anyway), but rather whether the expectations and needs of the customer are fulfilled. These can, and in practice often do, deviate from formal agreements;
- Problems should be solved preferably before they occur. So problems should not only be defined based on incidents that have occurred, but by proactively looking for problems that *could* occur.

Proactivity in ASL 2 also means that the attention is shifting to the outside. The process is becoming an internal issue. Of course, the process has got to work, but the design and implementation, the tooling, and the use of process models are internal issues for the IT organization. This means that the interface with the customer is becoming more important. Customer satisfaction regarding the IT organization is determined by the output that is delivered, but to a greater degree by the actual way in which the output is delivered and the interaction with the customer.

Six years have passed and VGK² is now called ISPM. In the previous couple of years there had been a new director who focused more on achieving high profit margins rather than delivering quality services and paying attention to user satisfaction and employee satisfaction.

Now there is a new director who has a good eye for quality and the future. In order to improve the quality, they have engaged John Hollander to lead an improvement program. John used to be a manager at VGK and he is now self-employed as an interim manager. John readily accepted the challenge.

² See appendix 1 for a description of the case.

4

Application support cluster

■ 4.1 INTRODUCTION

The goal of the Application support cluster is to ensure that applications can be used. In other words that applications are operable and available.

This is, therefore, an important cluster within application management for the following reasons:

- Applications (information systems) are built and maintained in order to be used. This is the reason why so much money is spent on applications. Managing this usage is directly related to this goal, making this cluster extremely important;
- In addition, much of this cluster is frequently overlooked. Infrastructure management is often tasked with these processes. The importance of application support as part of application management is poorly recognized and the (potentially) available knowledge and experience is not utilized for improvement.

The Application support cluster

The Application support cluster comprises four processes. These are:

- *Use support.* This process achieves optimal support in the use of applications by ensuring the best possible communication with

customers and the most effective handling of calls about the use of the applications and related services;

- *Configuration management.* This process encompasses the activities for the registering and updating of information about the use of (versions of) objects belonging to an information system/ application and its corresponding services;
- *IT operation management.* This process ensures, monitors and guarantees that applications (or application components) display the correct and agreed-upon behavior in operational situations;
- *Continuity management.* This process concerns the range of measures that must be taken in order to ensure the continuity of execution and support of the information provisioning through the use of information systems in the longer term.

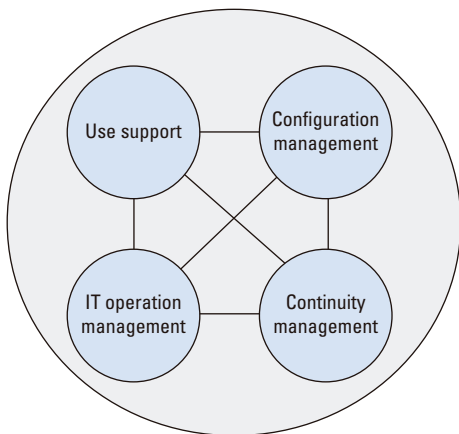


Figure 4.1 Cluster Application support

Application support variables

Various variables have a significant impact on the way in which Application support processes are executed. These variables therefore also impact on how the processes should be implemented in an organization. Examples of these variables are:

- *Multi-infra or single-infra. Is there just one production environment for the application or more?*

Whereas with custom software, the application usually runs in one production environment and/or for one organization, a supplier of software packages has to consider multiple production environments and organizations where the application (hopefully) runs.

When an application runs in multiple locations, it is often the case that several versions are supported, usually tailored to the various platforms (a version for Windows, for Linux etc.). Requirements and agreements can also vary per organization.

- *Responsibility for the underlying infrastructure.*

A second variable is whether application management is responsible for the underlying infrastructure. Of course there is always a sense of responsibility ('it has to run'), but this is sometimes formalized. For instance if a supplier provides an ASP solution, then the supplier will also be responsible for the infrastructure. The supplier is able and obliged to manage the infrastructure. This is also often the case with custom software. However, some software packages aren't provided in an ASP solution but run on the customer's infrastructure. In this case, the supplier has no responsibility for the infrastructure. Because the infrastructure supplier cannot be controlled directly, a different approach is needed.

- *Responsibility for application integration.*

New applications are frequently built using many standard components and additional components and these components

are often maintained and/or run by another party. Whether the application management organization is responsible for all or just some of these components can make a significant difference. Software packages, SaaS (Software as a Service, an online service), custom software and web services are all different forms of applications and application management. The differences between these forms create differences in implementation and responsibility in the organization.

■ 4.2 USE SUPPORT

The first and most visible process is *Use support*. Applications exist to be used. Effective and efficient usage by users and user organizations doesn't just happen. Almost always some kind of support is needed. *Use support* organizes the necessary communication.

The goal of *Use support* is to achieve optimal support in the use of applications through the best possible communication with customers and the best possible handling of calls about the use of – and possible deviations in – services, according to the agreements.

In contrast to other forms of IT services, the target audience of *Use support* (in other frameworks often referred to as 'incident management') is usually not end users but those tasked with business information management. Infrastructure management or another application management organization can also be part of the target audience.

The target audience of *Use support* usually consists of fewer people than the group of end users, and the target audience is usually more experienced and qualified. This is why this process for dealing with queries and incidents is generally less bulky and intensive than the corresponding process for infrastructure management.

But this doesn't mean that the process is not important. The 'customer' experience is influenced to a strong degree by the quality of this process, particularly for suppliers of standard packages, and also for custom software.

Proactive communication and reactive communication

Many process frameworks have an 'incident management' process, with the goal to deal with failures efficiently. Incident management is therefore a reactive process, it reacts to events.

ASL's Use support has both reactive and proactive components. It is important to deal with request, complaints, failures and other calls according the agreements and in an agreeable manner.

However the supplier's objective is to be proactive and to communicate as efficiently as possible in order to prevent incidents and requests.

ISPM's customers have noticed that it's getting more difficult to get requests and complaints dealt with.

The Customer Care Center was replaced by a helpdesk that was run by a company in India. The standard service was in English. For the Dutch customers there was the option of support in Dutch, for a fee.

Not only all change requests and incidents, but also all questions had to be addressed to the helpdesk.

The benefits of this change, apart from superficial savings, were zero. The helpdesk could not respond to most calls and only fulfilled an administrative and routing function. The customers became increasingly dissatisfied.

Long-term customers, who knew the ISPM employees, tried to contact them directly to get their calls dealt with. New customers sometimes just gave up.

In addition, ISPM employees weren't happy. Whenever they spoke to customers, they were confronted with complaints, questions and general dissatisfaction. More often than not, they couldn't do anything to help.

Frederic, who used to manage incidents, together with Henry and Tim, was appointed as team leader of the support cluster for PARIS just before John arrived on the scene. He started immediately to re-introduce old best practices.

"Well", he said to Henry, who thought it was ridiculous that they had to start from scratch again, "it helps to pass the time."

With Henry's help and John's approval, they introduce the procedures and agreements quickly. It even looks like a Customer Care Center is being created, although Frederic and Henry think that they better not use that term. Some people in certain parts of the organization believe that this kind of work should be outsourced to low-cost countries.

Now there is also more proactive communication, partially initiated by the new marketing manager who considers this important. The team e-mails a frequent PARIS-newsletter to the customers. The newsletter is also published on their website. Consultants in particular provide content for the newsletter. Their assignments for customers give them plenty of ideas. During one of the user meetings, Frederic thought about sharing customers' tips. The marketing manager is very enthusiastic about this. Customers can log in to a blog on the website and not only read but also contribute their own tips. The tag cloud on the blog page makes it easy for customers to find useful tips. And they can use Tweets to inform everyone about a new tip. Progress is rapid: within a month they have fifty tips on the blog.

It's a win-win situation: the tips also give the product consultants a better perspective on the kind of functionality the customers need.

■ 4.3 CONFIGURATION MANAGEMENT

Applications are increasingly built using components. But various versions of those applications or application components are used by different customers, or on different sites. *Configuration management* is responsible for the information about what is in use where and which agreements apply. *Configuration management's* goal is to keep a record of all application objects/configurations and services for which the application management organization has a responsibility, and to provide accurate information about this to support other application management processes.

Records are therefore kept about which versions, releases and patches (an 'interim' new version of a program to solve an issue quickly) that customers have and (in some cases) which support is applicable.

Good configuration management prevents elementary errors in the support service because there is up to date and correct information about what the customer uses and which services are applicable.

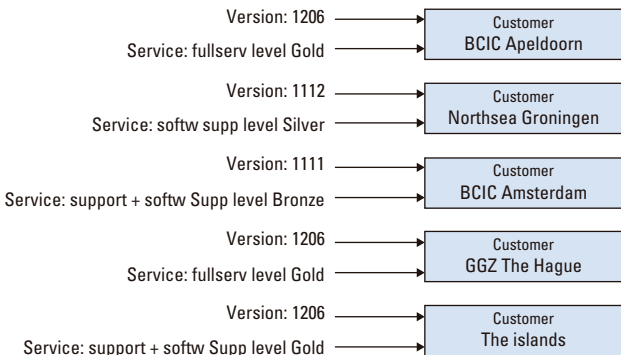


Figure 4.2 Configurations and services

Configuration management is relatively easy to handle with custom software, where there is one customer and one production environment. It is much more complex for large and extensively used systems with many versions, patches and customers.

In addition, customers don't always have the same services and service levels. This is important information for application support.

Configuration management is not a process where a customer is actively involved. It provides information and is a prerequisite for the other application support processes.

Configuration management of PARIS is pretty well organized. The organization is disciplined and used to a high level of capability in Configuration management. There is up to date information about which versions are used by which customers under which conditions. There is also up to date information about the agreements with the customers.

ISPM hasn't suffered from any deterioration of quality of service. This isn't strange: it has never cost that much and having this information is beneficial. It's also handy that the employees are aware of the agreements with the customers.

This is different at the new 'AM Factory' division, where it's not well organized. They have some Excel sheets, with information about the versions of the various applications and their breakdown in components, but this information is not up-to-date and complete.

Configuration management is stable at the PARIS group, where Harvey Bennet, process owner, has been in charge for years.

The previous director proposed to integrate Configuration management of PARIS with the similarly named process in infrastructure management. It took a fair bit of effort from Harvey to convince the director that it was better to retain a separate Configuration management process. His determined attitude ("Over my dead body!") convinced the director that he was serious.

Harvey was well-prepared. He'd seen the colleagues from infrastructure management come to inquire how the CMDB was organized. They didn't understand at all why different versions of the same software package were recorded. Let alone why 'service items' were also recorded. As far as they were concerned this could be deleted. They were of the opinion that this should be written into the contract. If things went that way, they wouldn't be able to provide the right information at the right moment.

John is delighted that Configuration management is still well-organized. He is reassured that the previous years' improvement initiatives have delivered benefits. And it gives him the opportunity to work on more pressing problems.

But not before asking Harvey to get Configuration management at the AM Factory up to speed, something that Harvey takes a month to do.

■ 4.4 IT OPERATION MANAGEMENT

The process *IT operation management* controls aspects of the production environment and the supplier. Computer infrastructures are needed to run applications.

Managing the infrastructure, including starting and running applications, is part of infrastructure management. However, in order to run and use larger applications, knowledge of the application is needed, in particular when the application doesn't behave as expected.

Increasingly, applications based on Cloud or ASP are being used. This means that the supplier is responsible for all of the related IT services. From this perspective, the application and infrastructure are inextricably linked. Control of infrastructure management is an important part of application management.

But also when application management doesn't have this broader responsibility, infrastructure management must be provided with the appropriate instructions for usage, and the application must be kept aligned with the possibilities and limitations of the platform. And there must be a point of contact for infrastructure management. The process *IT operation management* tackles these points.

The goal of the *IT operation management* process is to ensure, monitor and guarantee that applications (or application components) display the correct and agreed-upon behavior in operational situations, and that the services concerned also run as agreed.

Quality aspects

IT operation management manages three quality aspects: dependability, manageability and efficiency.

Dependability is broken down into availability and reliability:

- Availability is the degree to which an application object (configuration item) is able to provide the desired functionality at a specific moment and/or for a determined period. This concerns the start and finish (execution) of the application, processing at the correct times and in the correct order, the execution of incidental processing, opening times of online processing, storage period of files.

- Reliability is about the application's behavior. Relevant criteria are the number of failures during use, the frequency of errors and the duration of error-free operation.

Manageability concerns the degree to which infrastructure management can bring and keep an application in its operational state. This involves the transparency and manageability of applications from an infrastructure point of view. Production documentation (including the requirements, control and adjustment abilities of the application in processing, and various other aspects) provides much of the information here.

Efficiency is the degree to which an information system efficiently uses the technical infrastructure and thus becomes useable for the customer.

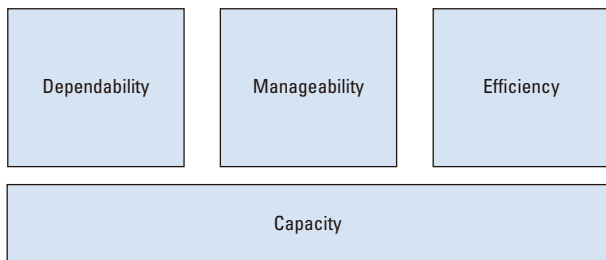


Figure 4.3 IT operation management topics

Capacity provides the foundation to these quality aspects. The production environment needs a certain amount of capacity in order to run properly. Capacity management concerns itself with the allocation of infrastructural resources in relation to the demand. The

goal of the sub-process capacity management is the allocation of the correct resources at the right time with the correct capacity to the services concerning the application support, use and production of the system.

Relationship with infrastructure management

Whether and how infrastructure management has to be controlled (or just supported or advised) depends strongly on the characteristics of the application management organization in the service provision constellation. There are various possibilities:

- A software package supplier who has no direct influence or responsibility for the infrastructure, should be both proactive and reactive. Proactive in the sense that the supplier should design and test that the application will run on all of the platforms that are supported. Reactive, because it's not possible to control the infrastructure and infrastructure management directly. So it will be necessary to react when questions, failures etc. arise. And in these situations, a quick response will be needed.
- In the case of a Cloud/ASP-situation or custom software, the application management organization will be expected to act more decisively. Of course this does not mean that application management will control or execute infrastructure management. But application management will have a coordinating responsibility for all of the services and the customer will expect not to deal with infrastructure management as a separate party.
- In some cases application management will fulfill the role of system integrator and will be responsible for all of the application services. The organization will be deemed accountable for all of the underlying application management organizations. Sometimes the system integrator is also responsible for infrastructure management, but this is not always the case.

- Sometimes other parties provide software packages. In these situations, where the infrastructure supplier or another party has the overarching responsibility, application management will have a role as facilitator.

So the way the process is implemented depends on the situation. But the core of the process is always comparable:

- Communicating expectations, developments and requirements;
- Routing and monitoring events, questions and complaints;
- Monitoring and adjusting where needed;
- Determining the impact of changes, and the measures required regarding the efficiency, reliability, availability and manageability of the application.

The core of PARIS, which runs mainly on a mainframe, is very robust and stable. Of course there are some incidents, but they are never severe. And they are usually solved quickly. That isn't so strange. The system is quite old and the teething troubles have been dealt with years ago.

That is in contrast with the parts that run on smaller platforms and in particular for the parts that have a direct link with the internet. The interfaces with the customers' systems, such as financial systems, are also often problematic. Because this is worrisome and because of frequent complaints, John decides to do something about this. Advised by Harvey, he engages one of the consultants at the data center. Although he doesn't know anything about PARIS, he has plenty of knowledge about platforms, networks and interfaces and is also fairly proactive and assertive.

Eric Dekkers, the consultant, establishes that the tooling on the mainframe works well and provides the right information. Because hardly any tooling is in use for the servers, he selects something suitable. After installation the

tooling gives him information about usage etc. Not yet very structured, but it's a start and provides him with some insight.

Some customers have also shown interest.

When he and John take a look at the agreements and procedures, they notice that several support tasks aren't always executed. The employees don't know how to justify their time and have the impression that it necessitates a lot of time without much in the way of benefits. Eric writes specifications and gets a PARIS developer to code a program. This provides them with a plan for the various support tasks. In turn, this makes it much clearer for both employees and management as to what needs to be done and how much time it requires. Very quickly it is evident that many of the support tasks don't take much time at all, sometimes just a quarter of an hour, but they prevent many problems. The support employees like having this work that they can do in between more important tasks. They even start a little competition with each other to see who can complete the most tasks in a week.

John is pleased with this progress. The number of incidents caused by problems with interfaces or capacity diminishes visibly. He thinks it's a good idea to share this experience with the data center management. Maybe Eric can achieve similar results there as well.

■ 4.5 CONTINUITY MANAGEMENT

The fourth process in the application support cluster is the process *Continuity management*. *Continuity management's* goal is to achieve continuity of the business processes by taking measures regarding the longer-term continuity of the applications and the data processing.

Information systems often store data that is critical for an organization. Without this data, the organization can't function,

and sometimes even can't survive. Take the example of a bank that accidentally loses the account numbers and the transactions of its customers and can't restore or reproduce them. Or a hospital that loses patients' medical records.

The process *Continuity management* ensures the longer-term continuity. This incorporates measures that address extraordinary circumstances.

Continuity management topics

Various situations threaten continuity, for instance improper external use (hackers) or unauthorized internal use (fraud, internal employees doing unlawful things). Security should prevent this unlawful use. Examples of security measures are physical security, passwords, access protection (for both the infrastructure and the application), applications that use and enforce segregation of duties in the functionality, various profiles in the applications.

The resources can also be threatened. Fire can destroy a computer center, communication cables can be – accidentally or deliberately – severed, computers or their parts can crash. Such calamities are very risky for an organization. Fallback (the possibility to continue processing at another location), back-up (data archiving), mirroring and double processing (simultaneous processing in two or more locations and/or data storage) are measures that can reduce these continuity risks for the organization.

There are also risks to the continuity of the resources that are in use. For instance, platforms that are no longer supported by the supplier or that lag too far behind new developments. The effect is less of an acute problem, but can have dramatic effects if neglected. The continuity of

	Use	Resources
External	Security	Calamity
Internal	Fraud prevention/ protection	Continuity of resources

Figure 4.4 Continuity management topics

the resources (platform, database technology, programming languages etc.) is therefore also an important topic in this process.

John is surprised that continuity management is under control at ISPM. But this is isn't that strange. PARIS and other systems transport financial data, data about internal and external staff, customer data. Customer files in particular contain increasingly confidential information such as account numbers.

A couple of the larger stock exchange listed customers have had some problems in the past. In order to prevent that from happening again, they stipulated security and continuity requirements during contract renewal.

So there has been justifiable attention placed on continuity management in the recent past. Measures have been taken such as an extensive contingency facility that includes the development environment, the sources and the documentation. And of course there are also agreements and procedures for ensuring availability of the application code for the customers in case of bankruptcy.

The contracts now contain a clause that allows continuity measures to be audited annually by an external party. ISPM has the opportunity to solve any problems within a month.

Given the fact that the previous director was focused on cost cutting, it seems strange to invest so much in continuity. But the customers were more than willing to pay for this. So ISPM could justify their investment.

These good continuity measures have also improved ISPM's image. Several potential customers have indicated that this is a reason to migrate to ISPM and PARIS.

Now John only has to find a new process owner for continuity management.

5

Application maintenance and renewal cluster

The second process cluster at the operational level is Application maintenance and renewal. Application maintenance and renewal comprises the tasks that address development and change of the application and the functionality of the application.

This cluster usually takes up most of the human resource capacity within application management. This is logical: the core of application management is about changing applications to meet changing needs.

Processes in the cluster

The process cluster Application maintenance and renewal comprises five processes (see figure 5.1):

- *Impact analysis;*
- *Design;*
- *Realization;*
- *Testing;*
- *Implementation.*

Forms of Application maintenance and renewal

There are many variations in which application management occurs, as previously mentioned in the introduction of this book. Examples are:

- Maintaining a custom solution: an application that was built specially for one customer;
- Developing a standard component for use by other parties;
- Building a management information system (or another application) that uses data from other systems;
- Renewing an application landscape: renewing a conglomerate of applications that are related in some way;
- Building and maintaining a software package;
- Executing system integration or application integration for an organization.

This means that the implementation of Application maintenance and renewal can change according to the situation.

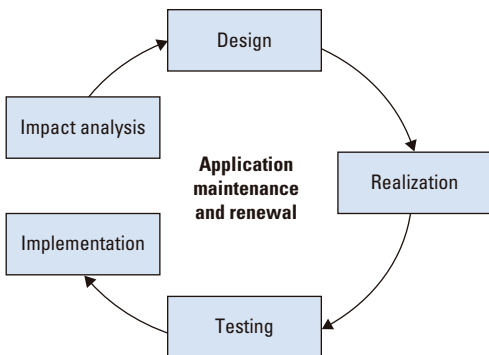


Figure 5.1 The process cluster Application maintenance and renewal

Tasks

Despite these differences, the core of the application management tasks remains reasonably comparable and is also based on a more or less comparable theoretical background. Design, realization, testing and implementation are a fixed pattern in each approach. The content, control and structure of the activities can vary.

To be more specific: the core of design, coding and testing is fairly comparable. But there can be significant differences in how the tasks are embedded in the environment, the way they are structured, the method, the number of steps that are taken, the phasing and the way the tasks are managed.

In a waterfall approach each phase is completed before moving on to the next phase. With Agile there is more interaction with the customer during shorter cycles. Implementing a standard package entails different activities than pure coding. There are plenty of degrees of freedom, but also enough similarities.

■ 5.1 IMPACT ANALYSIS

The first process in this process cluster is *Impact analysis*. The goal of the process *Impact analysis* is to enable controlled execution of the realization of a new release or solution. This is why the following questions are addressed at the start of a new release or solution:

- What is the content of the change or the functionality?
- Which parts of the existing application or the environment will be affected?
- What are the consequences for the application, the environment and the rest of the maintenance process?

- Which alternatives are available for realization, which solution directions will be proposed?
- What are the main points of how the change or the release will be realized?

The idea behind *Impact analysis* is that thinking about the direction of the solution and the impact on the environment will enable controlled execution of the other tasks in the process cluster. Many of the overruns and problems in releases or changes are due to too little knowledge at the start: during realization it becomes evident that a change is going to have more impact on the rest of the application or the environment of the application (the platform on which it runs, the adjacent applications) than was originally foreseen.

Worries and costs can be reduced, and predictability increased, by thinking upfront about what the question is, how it will be realized, by providing feedback on the solution and the result of this exercise, and by processing its impact.

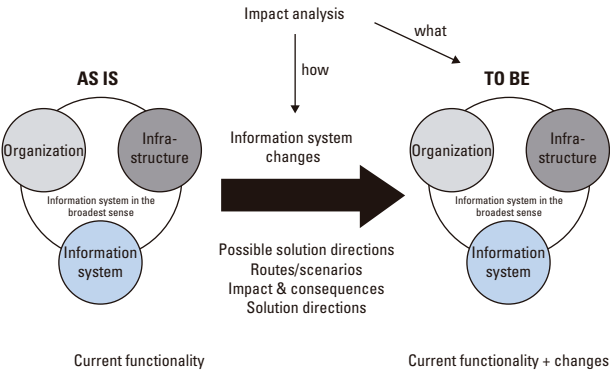


Figure 5.2 The Impact analysis

Concrete and reliable estimates of the necessary capacity, duration, and a breakdown of the execution can be made using the impact analysis. It is important to not only spend time on the technical aspects of the system. The impact on the user organization, the production environment and the longer-term maintainability should also be taken into consideration when evaluating and choosing from the various solution directions.

An impact analysis could lead to a customer choosing a less than perfect solution requiring more effort in the user organization. It's about choosing the most optimal solution.

Shortly after John Hollander left the organization, Tony, the first owner of the Impact analysis process, applied for another job at ISPM. After working on PARIS for many years, he wanted a new challenge and quickly took advantage of the vacancy at the new Consultancy group.

He likes the work. Based on his extensive experience with PARIS, he advises customers about the renewal of their application landscapes. Many customers suffer from bloated application landscapes comprising hundreds of applications, some of which are only used by a few employees. His guidance is highly valued. He has significant insight in the added value of the applications for the business and proposes pragmatic and affordable solutions.

To his surprise, he meets John again at a company reception. John is delighted to see that Tony still works for the company and he asks him for his support and advice.

Because the *Impact analysis* process has also suffered cost cutting. Tony's successor quickly stopped doing impact analyses. The goal was to be cheaper and more flexible, and the idea was that with new methods and techniques, products like an impact analysis were superfluous. 'Agile' was adopted, but without knowing much about it. It boils down to Agile being used as a synonym for time-boxing. When the time's up, no further effort is spent on the quality of the rest of the functionality. In order to prevent the delivery of products from being delayed, activities have been dropped at the start, for impact analysis, and for testing at the end. Today's motto is: start realizing the solution as soon as possible. At the start the customers were pleased, because new releases were delivered quicker. But this satisfaction has since eroded: the predictability and the quality of new versions have deteriorated and each implementation has to be followed up with a large number of fixes.

Because of his previous experience with Tony, John asked him to take on the improvement of this process.

Tony is pleased to do this. Last time he was pleasantly surprised to see that a systematic approach could reduce both the duration and the risk of incidents after deployment through to production. He still has all of the documents and templates that he developed, so improvement should be even quicker this time. He just needs a little time to reacquaint himself with the renewed PARIS.

The main points of the old fashioned approach have been adopted for large parts of PARIS. A new element is that subcontractors are now also involved in the impact analysis.

This requires some extra coordination at ISPM, which pinpoints the fact that the coordination could be improved.

Everything has been modernized and updated of course. At the start there was some tension within the team: the younger members were very reluctant to adopt it as they found it too bureaucratic and old fashioned. Tony found this amusing. He remembers that he had exactly the same opinion about ASL. He still has the memo that he sent to John, summarizing his objections. After reading it, the younger team members are a little more convinced. They conclude that Tony is not a dusty old system administrator who can only work with rigid processes and that gives them enough confidence.

The goal is to achieve the '10-norm': ten times according to the impact analysis, the eleventh time may deviate. This turns out to be a bit ambitious, but after a few months they are on the right track. The customers are aware of the difference and give the most recent release a significantly higher score.

Tony really wants to go back to his consultancy work, but John has another assignment for him.

■ 5.2 DESIGN

After the *Impact analysis* process comes *Design*. Impact analysis determines the background, the required change and the highlights of the solution direction of the release. The process *Design* describes the required change (or the required functionality) such that the required functionality of the application can be realized and tested unambiguously.

Goals

A design therefore describes, at a detailed functional level, the change to the application. A good design addresses several goals:

- It contains an unambiguous description of what the application has to do, so that the developers know what they have to achieve.

This goal is becoming more important, in particular in the case of offshoring (having the work undertaken in a low cost country): it is essential in this case that the end product is described unambiguously;

- It is a means of communicating with the customer (often business information management), so that the customer can visualize the change or (new) functionality;
- The design can be used as a decision-making document for accepting the release or the application. The sign-off can be based on it. It is therefore also an instrument in the quality system;
- A design also describes what the application does. It is an important and productive instrument for transferring responsibilities, for instance when outsourcing, or when dealing with new employees, because it replaces the need to dive into the code to discover what an application does.

A thorough design addresses three perspectives:

- The data, the limitations (constraints) and the coherence between the data;
- The processing of data by transactions;
- The sequence of processing (the temporal aspect). This is particularly important for workflow systems.

It is easy to work with Tony. He is knowledgeable, not only about PARIS, but also about the organization and the customers. That is not strange, considering that Tony has been working for ISPM and its predecessors for twenty years. Working in various other roles has resulted in a broad range of knowledge.

It was mainly because of his knowledge of PARIS that John asked Tony to also become the owner of the *Design* process.

Quality manager Miriam Hill had previously done some work in this area. She had the challenge of dealing with the lack of documentation. She consequently updated the documentation for each release and after keeping this up for a few years, managed to have documentation in place for three quarters of the system.

This was an opportunity for the previous director to economize in this process. “Everything that has to be changed on a regular basis is now documented, and the rest isn’t important: that won’t need maintenance. It’s a shame to spend time and money on that.” It was unfortunate that the PARIS team members thought that they didn’t have to document anything anymore. The three quarters that had been documented, was now reduced to half.

Tony is now considering whether to restart the process of updating the documentation per release, or to process whatever information is still available about the recent releases. He chooses the second option. A new designer has just joined the team and this is a good job for him, in which he can learn more about the system and the way of working.

Fortunately, the data model is up to date and correct. During the last couple of years they have been confronted with the problem of several people needing to be replaced and having an up to date data model was a great help in getting the new team members up to speed.

The *Design* process has also suffered from the limited interpretation of the Agile approach. An additional difficulty is that – in particular for the newer parts of PARIS – a team in India has been engaged. This means that the designs have to complete and correct in order to be clear and unambiguous for the Indian organization. In the old days the use of in-house developers meant that this level of clarity wasn’t required, but anything less than one hundred percent is now no longer sufficient.

Tony doesn't know whether the Indian offshoring has been profitable, but ISPM now depends on it. This means that some improvements have to be made to the quality of the designs.

In collaboration with the process owner for Realization, Tony sets up a glossary of the frequently used application terminology.

■ 5.3 REALIZATION

The functionality of an information system not only has to be designed, but also realized. This means that people are needed to ensure the correct behavior of the automated system.

Forms of realization

Realization can be achieved by various ways of programming:

- Setting parameters or 'customizing';
- Coding with compilers and interpreters;
- Coding in assembler language.

Nowadays a combination of these forms is used.

Coding is used to modify existing programs, and to create new ones. Coding is done in a specific language, a language that programmers understand and that is translated by a computer program into machine language (a language that controls the computer directly, and is practically illegible for people).

Customizing an application by setting parameters is used to get a standard application (for example SAP) to provide the required functionality.

More and more, standard applications are connected to other pieces of functionality and programs, such as components and web services.

The various parts are joined together during assembly, ensuring that they are connected correctly and provide the required functionality.

It's not only programs that are created and modified; data structures are also often modified. The data structure describes how data is stored in an information system. It is the foundation of a system. Changes to the structure of the data (adding a column to a table, extending the number of positions in a field in a file etc.) are addressed during the *Realization* process.

Difficult

Programming is a profession; principles for design and coding apply, and it requires a very analytic, rectilinear and precise way of working. Because many tools are used, it is a lot more complex than in the past.

A professional and standardized approach to development and technical design is important for knowledge transfer to other programmers. Understanding new programs is easier with smaller programs, while in practice, maintenance tends to increase the size of programs. Maintenance that pays no explicit attention to this aspect, generally leads to poorer maintainability. It is therefore a good idea to restructure a poorly structured program.

John is looking for a new owner for the *Realization* process. Reuben, the previous process owner, gave him a good tip. The most recent addition to the team is an ambitious young woman from India, who ended up in the UK because of a relationship with a British backpacker. Reuben thinks that she will be an excellent successor. She has followed an ASL training course and has passed the exam. A big advantage is, of course, that she speaks Hindi as well as English and can therefore communicate more easily with the team in India.

Sharada is honored to be asked and she accepts the challenge gladly.

She had already started to create a glossary of terms. She had noticed that people used different English terms, resulting in some confusion in India. Some wording resulted in more questions than clarity in India. John has given her extra time to complete this list. Soon there is a list with the frequently used terminology and clear wording on the intranet.

In addition she has made a summary in Hindi of ASL 2. She enjoyed attending the course and wants to share her new knowledge with her Indian colleagues.

Inevitably, the *Realization* process has also suffered from the local interpretation of Agile. In addition, due to limited functional documentation being produced, there was no technical documentation at all for some parts. They agree to adopt the approach that was used a couple of years ago: updating the documentation will precede each change. Fortunately this usually applies to the more recently built parts of PARIS, so the developers can still remember most of the details.

In the recent past there was not much attention paid to application maintenance. Changes and solutions for shortcomings were just realized. Usually without impact analysis and often without considering the consequences of the change. This often led to incidents. Various programs have become very difficult to maintain, with the result that few developers dare to touch them. “In comparison, spaghetti is structured”, according to one of the developers. And this wasn’t said about the oldest programs, but in fact about one of the newer parts.

So Sharada writes a memo with a proposal to improve the structure of these programs and to ensure their maintainability when future changes are applied. She would like to see a more formalized development process, with

more attention paid to quality requirements for the architecture and how to evaluate these, and to the quality requirements for the documentation that is delivered.

She gives her memo to John who proposes that she should take it up with the quality manager.

■ 5.4 TESTING

Application management is a very labor-intensive profession that deals with highly complex solutions and demands extreme care. It is almost impossible to create a program (let alone a whole application) in one go, and the same applies to large changes. There are usually defects in the program, many of which are solved in the *Realization* process. But by no means all of them.

One single defect prevents the application from working as intended. The consequences of this can be extreme. It is therefore essential to test the deliverables of *Realization*. Thorough testing is the reason why applications often work so well when in production.

The process *Testing* ensures that the programs and data structures that *Realization* has produced correspond with the deliverables as specified in the design.

The *Testing* process also has to be designed and implemented in a structured manner. Testing methods, tooling and approaches ensure with reasonable certainty that most of the defects in the programs can be found and corrected.

So it is possible to produce (almost) defect-free programs. Whilst this is the case in many organizations, things can often go wrong due to too little time, too low priority, opportunistic behavior or just plain carelessness.

There is a tendency to compensate for delays during previous development phases and maintenance processes by spending less time on testing. Design can take longer than planned, “we’ll catch up during testing”.

The same applies when using development approaches with short release cycles, such as Agile – all of the programs have to be tested thoroughly.

Types of test

Various types of tests are needed to produce defect-free applications. The most important types are shown in figure 5.3.

The unit test checks the new or changed program and is part of *Realization*. The technical system test or integration test examines the whole application from a technical perspective. The functional system test checks whether the application provides the required functionality. The production test is performed by infrastructure management and tests whether the change complies with production requirements. The acceptance test is performed by business information management (or somebody who is delegated to fulfill that role) and verifies that the deliverables comply with whatever agreements have been made.

These final two types of test (production test and acceptance test) are outside the scope of the ASL framework in the sense that infrastructure management is responsible for the production test and

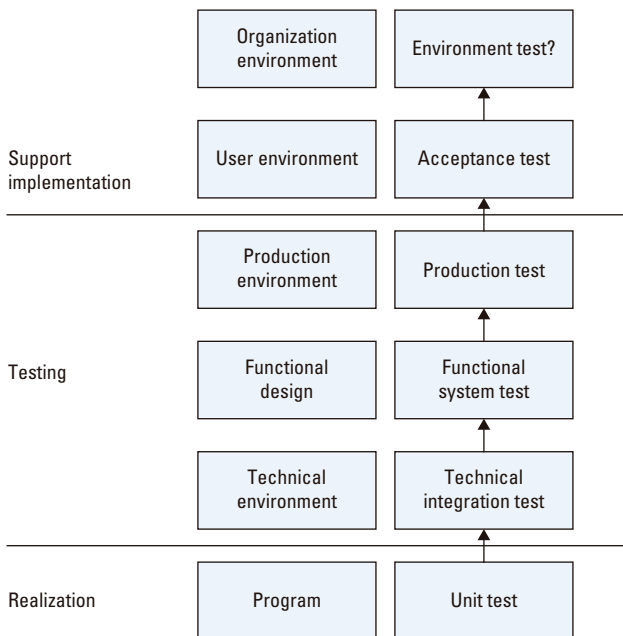


Figure 5.3 Testing in the environment

the customer is responsible for the acceptance test. ASL is a framework for application management but the framework also incorporates support of these other disciplines with their testing activities.

A complicating factor applies to testing in an environment with a chain of information systems owned by various parties. It is usually impossible to involve everybody in the chain in the testing process. An environment test or chain test would be preferable, but it is doubtful whether this is possible.

Previously, John had appointed Paul Green, who at the time was one of the designers of PARIS, as process owner *Testing*. Two years ago Paul has made the move from design to testing and has been the manager of the test consultants since then.

The previous director tried to save costs on testing. He felt that software just had to be built well, so that testing was superfluous. That didn't result in testing being abandoned completely, but it wasn't anything more than a unit test by the developers and a limited functional test of the changed part by the designers.

The technical integration test was skipped because nobody was responsible for the technical architecture. And the director thought that the regression test (part of the functional system test) was nonsensical: why test parts of the application that haven't been changed?

But then things went seriously wrong a couple of times when customers' websites simply 'vanished'. The customers forgave them the first time that this happened. But when it happened again – for one of the larger customer's site – there was less understanding. ISPM got landed with a substantial claim. Fortunately the director got the message and Paul was given the opportunity to give the director a presentation about testing.

Paul told him that testing is something that doesn't just happen at the end, but starts with the conception of the changed software. Errors that can cause problems in production don't only occur during the coding of software, but also in the design phase and when the product consultants write the specifications. And there is now greater risk of errors due to the increased number of handovers to the offshore colleagues in India.

There are no customers using PARIS in standalone mode. There are many interfaces with websites and parts of the PARIS database get input from other systems, sometimes even via other suppliers.

Paul could demonstrate that the earlier in the process that an error was found, the less time and money would be needed to repair it.

This was sufficient for Paul to demonstrate the importance of testing to the director. He got permission to set up a new test group and to train employees. He could even appoint two test consultants with significant experience in testing and test methods.

In consultation with the new consultants, Paul developed a training plan and a new implementation of the test process. They also looked for a tool to help them with automating the regression test. The project managers are now more inclined to schedule a regression test. This doesn't take that much time but it improves the quality considerably.

Testing goes fairly well now. Paul is, however, concerned that it costs him a bit of effort to keep up to date about the projects and releases and to involve his team at an early enough stage.

Something that has always been troublesome is the chain test. It's almost a mission impossible. Each customer has their own environment, with their own infrastructure and other applications that interface with PARIS. And in addition, some of the data in PARIS is provided to other organizations that have their own interfaces. It is impractical to keep track of this and to test the whole chain.

And therefore, from time to time, an error will occur in one of the customers' systems.

But ISPM always ensures that all external interfaces are described in detail and well-tested. So, should something go wrong, the cause will be down to the customer's applications or infrastructure – something that ISPM has neither insight into, nor responsibility for.

■ 5.5 IMPLEMENTATION

Various activities need to be executed before an application can be used. These activities are specified in the *Implementation* process. The objective of *Implementation* is to ensure that all of the preconditions for application management have been fulfilled, so that the new version of the application can be used. This objective merits further consideration.

Implementation

Business information management and infrastructure management perform most of the activities required for the deployment of a new application or placing a new version into production.

The application or the new version is tested and ready and, theoretically speaking, should be correct and ready for use. This is why application management does not have much work to do during the *Implementation* process. Application management only has to finalize the release or project, such as archiving the application documentation and project documentation, so that it is available for future use.

However, infrastructure management has to ensure that the production environment is suitable for the new version or the new application. This can be extensive. Sometimes new hardware is required (though this is usually limited), but almost always the programs have to be transferred and installed in the production environment, production data structures have to be modified, etc.

Business information management also has plenty of things to organize, such as changes to data, communication with users, user training, and the execution of the user acceptance test.

Application management fulfills a supporting role during the execution of these implementation tasks because knowledge of the application is needed.

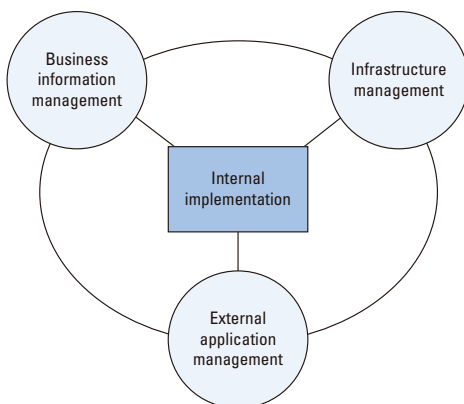


Figure 5.4 Implementation topics

Asked about the status of the implementation process, John got a dossier full of process descriptions, procedures and templates. He frequently comes across the name Jim MacFarlane in the descriptions. He knows Jim well, in fact he previously asked Jim to become the process owner of *Implementation*.

But Jim has left and is now self-employed. Following his departure, his role as new process owner *Implementation* was left vacant. Things went well enough and the director considered process owners to be expensive.

And so nothing has changed in the approach to implementation that Jim had introduced.

Application management still makes the conversion software. The checklist for completeness always gets filled in for every deployment.

Even the Production Change Newsletter is still produced, although this has been condensed into an A4 document that includes only the most important points. And the User Change Newsletter is also still published.

After John's departure, Jim introduced the Management Change Newsletter, something that John had previously held back from. John thought that the customers' management wouldn't be interested in the details of the changes. But now this newsletter is published and included as part of the deliverable for each release.

As a first step, John has decided to do something with the processes himself.

The first thing that he does is to ask the customers what they think about the Management Change Newsletter. Just as he expected, most of the customers consider the newsletter to be superfluous. It almost always gets thrown away unread. So it is not a problem to stop this initiative, with a resultant saving in time and paper.

The User Change Newsletter is still in use and is relevant for the user organizations. There are an increasing number of queries about the structure of the newsletter. Currently, the details about changes to the various parts of PARIS are bundled together. However, the customers would actually like to see separate changes relating to each part, making it easier for them to update their user manuals.

Another change is that it is no longer necessary to provide the newsletter in HTML format. The customers' intranet sites have various ways of distributing information. A text file is sufficient: they can deal with the formatting themselves.

So there are really just minor changes. It seems that the process has been implemented well enough.

But the last couple of deployments have exposed some problems that the process hasn't addressed.

The Production Change Newsletter is put together in English. And because a Dutch customer has complained about the use of English, Google Translate has been used to translate it into Dutch. But this usually results in inexplicable text that is useless.

Luckily, apart from being hilarious, this has not resulted in any real problems. However a more significant problem is that information from ISPM's subcontractors isn't included in the newsletter. This has resulted in problems at customers who run PARIS themselves. These problems are always dealt with, but it's a bit annoying and time-consuming.

Reason enough, according to John, to appoint a process owner for *Implementation* to find a solution to these problems.

He approaches Sophie Graham, a consultant from the Services department. Sophie is often involved with PARIS implementations and also has an extensive network within the subcontractors and customers' IT departments. So she has enough experience to come up with workable agreements and procedures.

6

Connecting processes

■ 6.1 INTRODUCTION

The *Connecting processes* are the bridge between the *Application support processes* and the *Application maintenance and renewal processes*. The most important complicating factor that affects the *Connecting processes* is that there is hardly ever a one-to-one relationship between the clusters.

Several versions of an application can be maintained concurrently. Or support is delivered for an application in multiple locations.

There are two *Connecting processes*:

- Change management, and:
- Software control and distribution.

■ 6.2 CHANGE MANAGEMENT

Change management deals with collecting the changes that need to be applied, making agreements about the execution of these changes (including the associated timelines) and the actual initiation of the changes. *Change management* receives the changes from various sources: from business information management (customer), the

process cluster *Application support*, the *Management processes* and infrastructure management.

Changes and releases

Development and maintenance of software comprises many steps and tasks that support the eventual goal, namely new or changed software that works well.

The efficiency of the execution of changes can be improved significantly by bundling changes in releases. This also gives the opportunity for additional optimization, for instance by applying several changes to one part of an application at the same time.

In application management terminology, a release is a group of changes that are executed and deployed into production at the same time. The term ‘release’ not only applies to the time when the changes are implemented, but also to the prior processes *Design*, *Realization* and *Testing*.

Multiple releases

The result of *Change management* is the start of release activities within the process cluster *Application maintenance and renewal*. A release can be small, for instance the correction of a defect, but can also be large, such as a release with different changes or the complete renewal of a part of the application. There are often several releases at the same time.

Production failures that occur while a release is in execution often have to be dealt with quickly. A ‘fast-track’ release is started, typically called a patch. The patch runs in parallel with the regular release, and often in parallel with other releases (figure 6.1). These parallel releases increase the complexity of *Change management*. This can be even more

complex in the case of software packages and standard components. Multiple versions of the package need to be maintained and therefore defects have to be corrected in all of these multiple versions.

Change management, particularly in larger organizations, can be a complex process in which thorough registration is important.

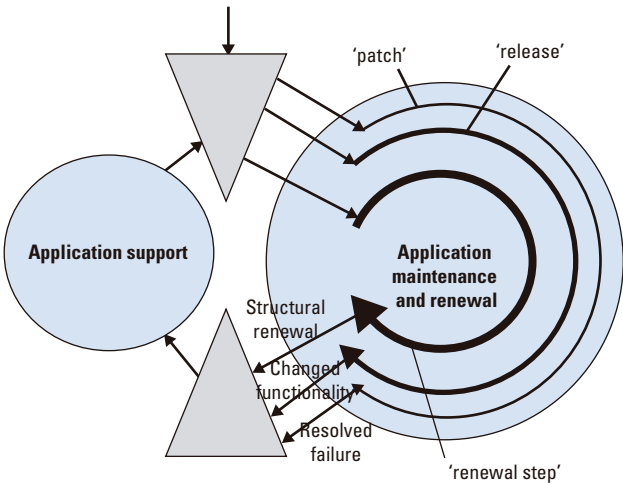


Figure 6.1 Multiple parallel releases

There used to be many problems with changes. There was no overview, the same modules were frequently and inefficiently changed and not bundled into releases. When Mohammed Saleh was appointed as process owner for *Change management* he started off energetically. And when he, before his retirement a couple of years later, transferred his responsibilities to his successor, Pilar Rodriguez, every change request was recorded and clustered in releases.

Nowadays everybody is satisfied with how things are going. Working with releases is the norm. All requests are registered in a tool that is also accessible to customers via internet, enabling them to follow the progress of changes.

There is a permanent user group, the user council, for the software package. The user group used to convene once a year, in October, to agree on the planning for the next calendar year. But in recent years they now convene at least two or three times a year. This enables them to plan for the future but without the need to set the plans in concrete. This has been successful for a couple of years. Releases are no longer filled completely with changes, leaving enough room for ISPM to schedule any fixes or preventive maintenance.

Custom software is a different matter. In this case it's not the user council but instead a single customer that determines the changes for their organization's particular needs. There are usually several bespoke projects for several customers running in parallel. These projects often have different IT suppliers and, at the customer, other contacts. This whole constellation of bespoke projects, standard releases and different suppliers and customers can complicate the work for *Change management*.

There are often problems with the alignment of the various activities. Many meetings and lots of feedback is often needed. "Just implement a change" is no longer possible. The duration of the change is no longer determined by the time needed to realize it but by the meetings and feedback preceding the change and the alignment during the change.

Pilar has thought about this a lot, often together with colleagues or external consultants. But she hasn't found a real solution.

It seems that this is just a troublesome issue and that it will always require

■ 6.3 SOFTWARE CONTROL AND DISTRIBUTION

Software control and distribution deals with the logistics of application management: the distribution of the various application objects (parts and basic elements that play a part in the process of development and maintenance of an application) to the various processes, the delivery of software and objects and the receipt of external application objects.

Application objects

Application objects are objects that play a part in the development and maintenance of an application. The most important application objects are, of course, the programs that an application comprises, but there are other objects, such as:

- Documentation, such as designs and data models. These are documents that describe the construction and working of the system;
- Programs: these are the parts of the application that programmers construct, containing the instructions that the computer has to execute;
- Executables: an executable is the version of the software that the computer runs;
- Scripts: instructions that are used to integrate various parts of a system or to execute other operations;
- Test sets: groups of data that are used to test the system.

These are all of the objects that the application comprises and that are used and modified during maintenance.

Logistics and distribution

Software control and distribution deals with the distribution of the application objects to the various processes, for example *Design* and *Realization*. It is important that the right versions of the right objects are transferred at the right time. Many controls and limitations can

apply. Objects that have been checked out and given to a process, can't be checked out again. In order to prevent inconsistency for example between the software and the implementation of the database, the set of objects that are needed for a change have to be complete. Only authorized persons can have access to the objects.

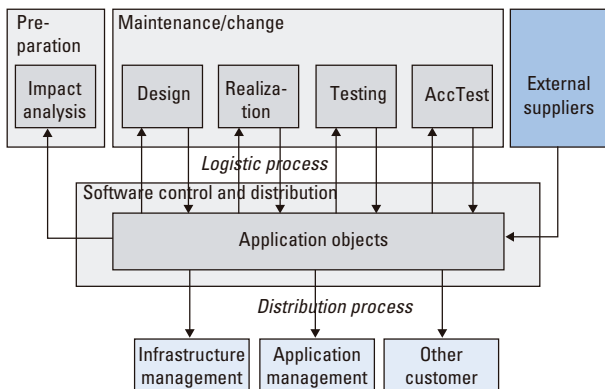


Figure 6.2 The logistics and distribution process

In addition, this process also deals with the distribution (after authorization) of the right programs to the right production platforms at the right customers.

This can be a very extensive process with many mandatory requirements, particularly in the case of large applications with many – and critical – systems.

Complexity

In *Change management* it was mentioned that multiple versions and releases often run in parallel. So sometimes the same application

object will be modified in multiple releases at the same time. In order to prevent the same defect from reoccurring in subsequent versions, it is necessary to be well informed and to tackle this issue adequately. It is evident that this can be a very complex process.

Software control and distribution was implemented thoroughly several years ago by the previous quality manager, Miriam Hill. And because it isn't possible to bypass the process, there haven't been many problems. At least, for the old part of PARIS, being fully integrated in the tool for software management.

The newer parts have never been integrated in the tool and they could, therefore, be worked on in an unstructured way as was the case years ago for the old PARIS. And this is precisely what happened.

When a production problem occurred during a release that was being executed in India, they often forgot to synchronize the sources in India. This resulted in a reoccurrence of the error after deployment of the new release. Or a programmer was given the wrong version of the source code and had to re-do his work a couple of days later.

Not much work was lost because they worked in short sprints. But when taken together, this represented a considerable amount of money. And of course there was the time that irritated customers lost due to fire fighting. John Hollander thinks that it's a bit too much to re-employ Miriam. And in addition, this is a good job for Reuben, the new quality manager.

Reuben has started almost as enthusiastically as Miriam. The first thing that he did was to discover whether the software management tool (still the same one that Miriam introduced in the past) would also work for other environments and methods.

Fortunately the tool supplier has kept updating the tool and the newest version is suitable for newer environments. Reuben and John agree that an upgrade to the newest version of the tool is necessary.

In addition, the old implementation needs to be made suitable for India, with extra workflows for the other parts and some names of flows requiring change.

The flow 'Urgent fix' for changing and deploying software despite any other releases and projects that may have occurred, is now called the 'Hotfix'. 'Urgent fix 3', for a similar deployment of all supporting versions, is now called 'Hotfix overall'. The name 'Urgent fix' now refers to changing and deploying software for one customer only. And so more flows need a new name.

It wasn't easy to find good names for the other workflows and to implement these. But having sorted it out after six months, everybody is very satisfied.

7

Management processes

■ 7.1 INTRODUCTION

The Management processes

Control of the application management organization and processes has become more important than ever. There has also been a shift in the control aspects in recent years. Where several years ago an application management organization used to be an internal IT department that focused primarily on spending the available budget as best as possible, now application management organizations are commercial enterprises with profit targets.

And where there used to be only limited use of other suppliers (often just for staff augmentation), now so much specialist knowledge is needed that it is difficult to acquire this without the use of external parties.

There are five Management processes and each has its own control perspective (see figure 7.1):

- *Contract management*: the agreement and management of customer expectations regarding services and supplied products;

- *Planning and control*: the management and monitoring of time, human resources capacity and delivery dates;
- *Quality management*: monitoring of the power to supply and the quality of the organization, application and services;
- *Financial management*: management of the production factor relating to money (costs, benefits);
- *Supplier management*: management of services and products that are acquired.

Each of these processes is discussed in this chapter.

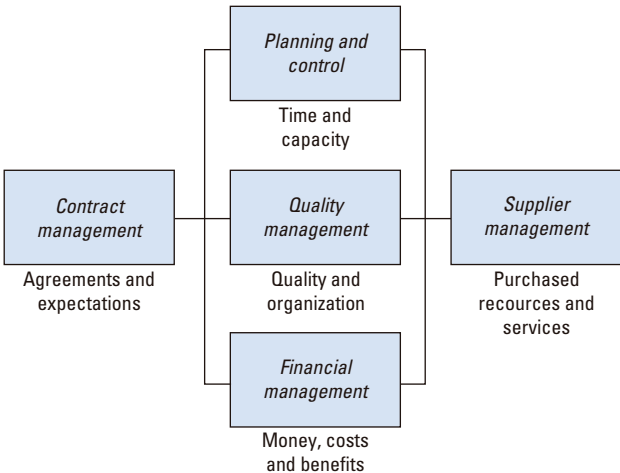


Figure 7.1 The Management processes

Integral control

The Management processes are positioned between the operational and strategic process clusters. This means that the Management processes have to ensure alignment between these different levels. This

should not be translated as needing to ensure that plans at the strategic and managing levels are delegated to the operation level. These processes do more than that: they ensure two-directional alignment:

- They ensure that structural deficiencies at the operational level are solved at the managing level, or at the strategic level if a change of strategy is needed. The first concern of a higher level is to solve bottlenecks and barriers at lower levels;
- They ensure that strategic changes are translated adequately to the operational level. Not all of the strategic plans will always be executed in their entirety. That is not necessary. It is more about strategic *direction*. But they shouldn't be overruled by tactical needs (unless this is agreed upon);
- The Management processes provide the scope and facilities (for instance finances) to which the operational processes have to adhere. This also includes whether activities are organized in the line organization or as a project;
- The Management processes also ensure that tactical developments are translated to the appropriate degree to the operational processes, without becoming too dominant.

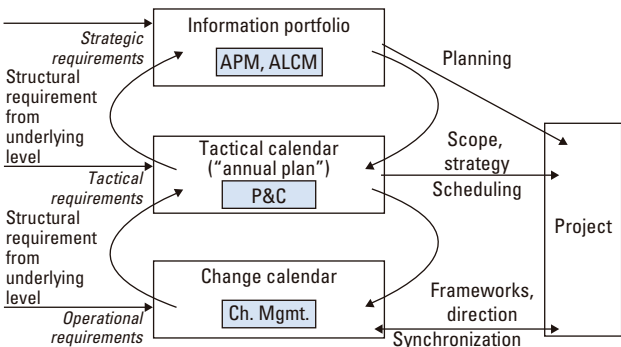


Figure 7.2 Portfolios and their correlation

So the Management processes deal with recognizing and communicating the structural needs and deficiencies to the upper levels and providing the lower levels with scope and direction. The Management processes also need to ensure that sufficient knowledge and facts are available for justifiable decision-making.

The Management processes ensure two-directional traffic so that the levels act as one application management organization.

■ 7.2 CONTRACT MANAGEMENT

Contract management is the process that manages the agreements with the customer. By ‘agreements’ we mean more than the formal rational agreements.

Agreements

There are various forms of agreements. Agreements can be made about the application (that which is bought/sold), and also about the related services. Agreements about the content (the ‘what’) are hardly ever enough: the ‘fit’, and requirements and performance are also relevant. So there are various kind of agreements (see figure 7.3).

The first set of requirements concerns the fit in the environment. This refers to both the application and the services. This results in two kinds of agreements about the environment:

- Interfaces of the application in its environment: agreements about how an application communicates within its environment (and with other applications) and which requirements apply to this communication;
- Rules of engagement: how the customer deals with the application management organization and how in turn, the application

management organization deals with other suppliers in the supplier constellation.

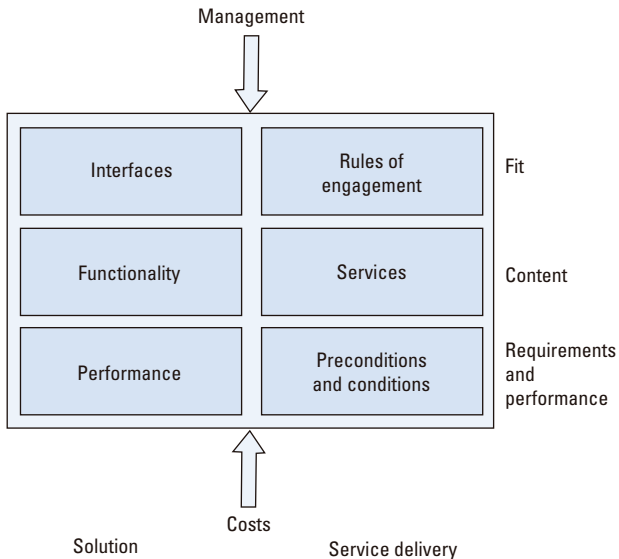


Figure 7.3 The agreements

The core of the requirements is substantive: about the content. Once again, there are two kinds of agreements, about both the application and the environment:

- The functionality of the application (solution), the tasks that the application has to execute and the data that has to be stored;
- The services that the supplier provides. The tasks that are executed (for instance maintenance and renewal) and those that are excluded (for instance support and/or infrastructure management).

Agreements are also needed regarding how the solution behaves and works. Once again there are two kinds of agreements, for both the application (solution) and the services:

- **Performance:** these are the requirements regarding the behavior and working of the application, such as the speed of processing, how many failures allowed, etc.;
- **Preconditions and conditions:** these are the requirements and boundaries that apply to the service. Examples are: a new release may contain no more than five defects; application management is no longer responsible for defects after the acceptance test.

Expectation or agreement?

Expectations are more important than agreements. It is often the case that much energy is expended on contracts in which agreements are described in minute detail in extensive documents. It is hardly sufficient or even meaningful for the supplier to manage these agreements:

- Needs change as time passes and that can happen pretty quickly. Customers often don't understand what they ask for;
- Agreements are determined by people. The experience and personal preference of a particular manager who works for a customer will affect the agreements;
- Customer experience is often a stronger driver than detailed agreements.

It is not unusual that the customer experience factor and a good relationship and understanding with the customer prevail over and above the formal agreements. Knowing the customer's stakeholders, understanding the expectations and their personal preferences for requirements, and ensuring that these are met is often more effective than realizing the formal contractual agreements.

At the time ASL was implemented for the first time, when John and Miriam were there, they started with service level agreements, subdivided into agreements for various kinds of customers that the former VGK organization recognized at that point: internal and external customers.

Tom Board was the service manager and provided the customers with monthly service level reports. And once a month, together with the product manager, he visited the internal customers in order to review the previous period and to discover what short term and longer term plans the customers had.

A half-yearly user meeting was organized for the external customers who just bought the package and ran it themselves.

In the beginning it was fairly easy to compile the report. The data center and the internal application management organization were the only ones who provided the data every month, which Tom transformed into an attractive graph and report, using an Excel spreadsheet that he had developed himself. But it became more difficult when part of the work was outsourced to India and other suppliers were increasingly engaged. In addition, Tom had to spend most of his time in meetings with the suppliers and India just to ensure that the service levels were being met. He could delegate the reporting activity to one of his employees in the support team. But increasingly, he got the product manager to deal with the meetings with the internal customers. This resulted in him being less well-informed about the customers' situations. After six months he was very surprised to hear that the two largest customers, despite the service levels being achieved, were no longer satisfied.

There had been a couple of problems, but Tom thought that these had been dealt with adequately, in any case within the agreed service levels.

In order to understand the situation, he decided to accompany the product manager to the next customer meeting. The customer reacted with surprise: "Do you still work at ISPM?" Tom decided to put his cards on the table. It emerged that the reason for the poor customer satisfaction was that he didn't see Tom anymore. That gave him the impression that ISPM wasn't all that interested in the customer. Another issue was that he no longer had the opportunity to discuss the status of the services and to make agreements during the monthly meetings. The customer said that he hardly read the report. "I just assume that everything is OK and that I'll get a call if there's an issue. I can't be bothered to read a report at the end of the month." And he added that it was far too detailed for his needs.

Tom realized that he had spent too much time with the suppliers, and that he had neglected the customers, despite this being his primary interest. Managing the suppliers was completely different from dealing with customers. There was a great deal of administrative work that came with dealing with suppliers, and he had lost touch with the customers.

The director took a bit of convincing that Tom needed somebody else to deal with the suppliers and India. But he was lucky to get support from the product manager. The director recognized the importance of a happy customer. This led to one of the consultants being assigned to implement supplier management and undertake it for the time being. The decision whether to create a vacancy for the role would be taken in due course.

This has lightened Tom's load. Now he only has to deal with one person for service levels agreements instead of many suppliers and India. He can now spend most of his time on his customers.

Based on his finding from the first meeting, he decides to conduct a more thorough evaluation with the other customers. He wants to ask them what they think of the report, what's really on their mind and what their expectations are.

He has only spoken to half of the customers, but it is already clear that the report has become way too long and detailed.

This poses a nice challenge to do something about it.

■ 7.3 PLANNING AND CONTROL

The *Planning and control* process ensures that the agreed services are provided using the agreed human resource capacity, and delivered on time by deploying the right human resources at the right moments. In other words it involves the planning and monitoring of progress, human resources capacity and timelines. Unfortunately, it is still a challenge in many application management organizations to create predictability regarding capacity and delivery.

This challenge has not diminished in recent decades. There is a trend to work in extensive supplier constellations with different parties. Customers also have more stringent requirements regarding speed and flexibility. The complexity has certainly not reduced.

A second challenge in this era of commercialization is to deliver on time (and often as soon as possible).

Limited capacity

For most applications the capacity for application management is limited and there are two reasons for this:

- Application management is labor-intensive and people are expensive. The investments in application management are high, but always limited. So the available capacity is also limited;
- Application management can only be performed effectively (and efficiently) when there is detailed knowledge of the application.

Acquiring this expertise takes a lot of time and the time it takes

to get a new team member up to speed can be up to a couple of years for some applications. Just adding extra capacity is usually inefficient.

This means that it is preferable to keep the capacity as stable as possible and limit capacity fluctuations in order to maintain the efficiency of application management at an acceptable level.

Communicating vessels

A complicating factor with application management is the similarity between the support and maintenance expertise and the expertise needed for the initial development of the application. When organizations build new applications, the development project usually needs the original expertise from the previous application management for a number of reasons:

- The functionality is hardly ever completely new. The current functionality is used in the organization and most of this will be needed in the new situation. The only party who really knows the details is application management. The source code is the best description of how things actually work. And application management maintains the source code;
- Applications always function in the context of existing systems and platforms. New applications have got to be implemented in the same context.

It is therefore inadvisable to isolate projects from application management. Projects and regular maintenance can be regarded as communicating vessels: when extra capacity is needed for a project, there is less capacity for regular maintenance.

The execution of regular, tactical and strategic changes involves at least part of the same capacity. It is therefore essential to consider and prioritize the various demands and plan capacity accordingly.

Measures and activities

There are various tips and measures that can help to design and implement the process *Planning and control*:

- Ensure good estimation, collect norms and ensure good progress monitoring;
- Ensure that there is enough slack in estimates to cope with unplanned work. Organizations are usually confronted with unplanned developments or needs that have to be realized;
- Ensure that there is teamwork and balance between tactical, operational and strategic activities.

Just like before, John has taken on his hobby, *Planning and control*, himself. And he is pleased to see how everybody has dealt with his 'legacy'.

The quality of estimation and planning for PARIS, certainly for its older parts, is good. The deadlines are usually achieved, although they frequently only use 85% of the estimated budget. The latter, John observes, is due to conservative estimating by employees who were reprimanded when they exceeded their budgets. John is going to do something about that because he is in favor of realistic estimates. "Because", he tells the employees at a meeting, "if everyone estimates like that, it's going to be too expensive to sell."

Estimating and planning for the newer parts of PARIS isn't as satisfactory. This is mostly due to lack of routine and experience. A positive point is

that they are filling a database with indicators, as prescribed by the old procedures. They record each estimate and add the actual figures after realization, with an explanation for any deviations.

Although this is not yet optimal and deadlines are often missed, the accuracy of the estimates is improving all the time.

Another challenge is the limited deployability of the team members, despite the team having been enlarged. The old mainframe parts of PARIS are still maintained by the older employees, some of whom are dangerously close to retirement. The younger employees have almost no knowledge at all of these parts and the related development platform. And the older employees have next to no knowledge of the new development platforms. Cost cutting measures in the past have resulted in limited investment in training and knowledge transfer. This means that sometimes a team is up to its neck in work, while employees in another team are twiddling their thumbs. That does not help the planning process, so John makes a mental note. This is an agenda item for the Application management organization strategy workshop.

But all in all, he is not dissatisfied with the planning process. Having the team in India and the extra subcontractors hasn't made it easier. In particular with planning larger projects, it's a headache to align all parties and to limit the idle time due to having to wait for each other.

Seeing as the planning deviations – despite these challenges – are quite acceptable, it's not a bad result.

■ 7.4 QUALITY MANAGEMENT

It will come as no surprise that the *Quality management* process controls the realization of the concept ‘quality’. In ASL, quality addresses four topics, plus a fifth central topic (figure 7.4).

The first topic is the quality of the product. The quality of the product is related to the quality of the application that is delivered and the ‘materials’ that are used in the process, such as the documentation and the test sets.

The second topic is the quality of the (production) process. This addresses the way the processes are implemented and executed, the roles, the responsibilities and the procedures.

The quality of the organization is related to the structure, the management of activities, the characteristics of the organization, and the people and their skills.

The fourth topic is the quality of the quality system. The quality system is the infrastructure (in the broadest sense) with which application management is executed. This is, therefore, related to the tooling, the completeness and alignment thereof, the quality of manuals and process descriptions etc.

The final topic of *Quality management* is the integration of services between suppliers (or subcontractors) and the organization’s own services. The key question is whether the customers’ requirements are adequately covered.

Quality is concrete

Quality is not an abstract concept or a policy topic, it is something concrete and of practical relevance. Quality, or rather lack of quality,

manifests itself in many ways in the workplace. Poor transferability, plans that often change and are not realized, parts of an application that are error prone, designs that are difficult to read or to understand, difficult communication with the customer. All of these signals are a form of insufficient quality.

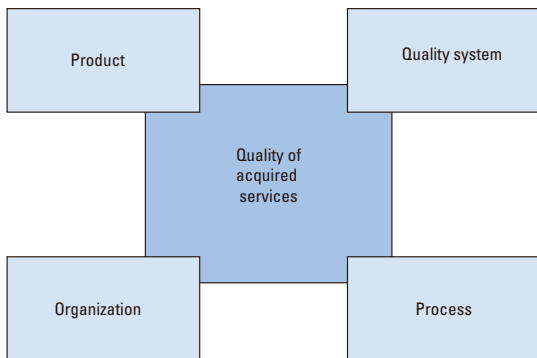


Figure 7.4 Quality management tasks

Many other process frameworks contain a process typically known as problem management, a process that analyses incidents (failures) and identifies underlying problems. ASL does not have a separate process for problem management because problems are one of the most important resources that provide *Quality management* (and therefore no other process) with insight into quality. Furthermore, problems can occur without incidents having occurred.

Evaluations of the other application management processes, recognized bottlenecks or problems are concrete items that are addressed by *Quality management*. This means that it is not an abstract process, but a very concrete process with concrete topics.

Although much has changed, many things have remained the same. Reuben Jones also still works at ISPM and has never even left PARIS. After all these years he is still the process owner of *Realization*. But in a conversation with Reuben, John noticed that Reuben is less passionate than he used to be. This is not surprising. The proposals that Reuben has suggested in recent years were rejected by the director because they were too expensive.

Knowing how important quality is to Reuben, John decides to approach him and offer him the role of quality manager.

One of the first things that Reuben tackles is the development process. The abrupt introduction of an agile method, without much thought and assistance, has been more for the worse rather than the better. After some research and enlightenment, Reuben realizes that his distrust is not based on the method, but on the way it was introduced, and the motive for its introduction.

It was really just a disguised cost cutting measure. And that is a pity, because Reuben now realizes that there are most certainly major advantages in using the method. There is no downside to better collaboration with the customers and reacting more quickly to requests.

He invites an external consultant, an Agile and Scrum expert, to contribute. In a two-day workshop the process owners of the maintenance processes are brought up to date. After which they, together with Reuben and the consultant, consider which parts of PARIS are suited to a more Agile approach. This is not the case for the calculation programs and the parts that deal with the distribution. The customers' needs are clear enough and the programs don't have to be modified that often.

The software needs to be well thought through in order to guarantee good performance. So the waterfall approach will remain in use for those parts. However the process owners, together with Reuben, will revisit this approach and endorse the agreements, modify the templates and make new agreements about collaboration and output.

An Agile approach seems to be well suited to the newer parts of PARIS and for the websites. Customers certainly have many requests to improve the websites, but often don't know exactly what they should look like. They really just want to experiment with the sites and that is tricky if each change has to be planned months in advance.

Reuben gives the consultant the assignment to set up a program to re-introduce the Agile approach properly. The consultant trains all of the employees. But not before he and Reuben give the director a presentation to ensure that he is enthusiastic about Agile and is prepared to invest in it.

Fortunately the new director understands the need for investment and Reuben's plan is approved, with a couple of minor changes.

■ 7.5 FINANCIAL MANAGEMENT

Financial management ensures that the costs of providing an application and related services are planned, and are commensurate with the benefits for application management.

Costs and benefits

In the previous version of ASL *Financial management* was called *Cost management*. This reinforces the statement that the business case for an investment in IT is not the responsibility of the IT-organization, but of the customer, the business (business information management). The IT-organization is not equipped to draw up a business case

because determining the feasibility and the benefits requires detailed knowledge of the business and the users (business information management has this knowledge). But management of the costs of undertaking application management is a responsibility of application management.

This statement is still valid. But we now call the process *Financial management*. There are two reasons for this:

- The costs of application management are always charged in their entirety to the customer (IT as cost center). There are often multiple customers for the same services or there is a different kind of cost charging mechanism. This means that the application management organization now has a cost charging structure, an independent responsibility and therefore an (independent) business case;
- It is also important that the application management organization knows whether there is demand for their services or solutions and whether these can be provided for a competitive price. This is why insight into the customers' business cases is needed.

Financial structure and business case of application management

Financial management at its core is about managing all aspects of the financial structure and the business case of application management, such as:

- Creating a cost charging structure and managing the cost charging;
- Creating and adjusting the cost allocation structure and ensuring that these costs are justified;
- Estimating costs and benefits and monitoring the behavior of costs and benefits (income);
- Optimizing all of this and ensuring that there is also a correct cost structure for external costs.

Financial management has been subject to a lot of attention in recent years. The previous director had demanded a report with clear insight into the return per product and service. This resulted in the creation of extensive spreadsheets to record the costs and benefits of the various products and services. This, in turn, enabled the previous director to manage the finances and reduce the costs for, in his eyes, unnecessary items and in so doing – and this was the intention – to improve returns.

Unfortunately the spreadsheets were too extensive and complicated which meant they gave little insight into the exact costs and revenues. Some employees recently began to doubt whether the calculations were correct.

The contracts remain more or less the same: the internal and external contracts still exist. Some new options have been introduced. Such as the Plus contract where ISPM also executes some of the business information management activities for a customer. And for all contracts it is still the case that customers don't have to buy the whole PARIS suite.

So there are quite a large number of different kinds of contracts, making it difficult to have insight in the financials.

John starts by preparing an overview of all projects and releases in progress and the support tasks. The estimated and actual time spent gives him an overview of the past and future personnel costs. Based on the time spent, he also distributes the costs for tooling, machines etc., per the various projects, releases and support.

It is clear that projects which involve multiple parties are less profitable due to the amount of overhead. Also projects where part of the work is outsourced to India are sometimes less profitable.

For the latter there is a pattern: simple changes and incidents in India are very profitable, due to the team already having acquired the necessary knowledge.

But the more complex projects suffer from the extra costs due to frequent discussions with the customer and any changes of direction during the project – these costs are actually greater than the advantages that accrue from undertaking development in India.

Another thing that has emerged is that the costs for support are only partially charged. There is enough income from the many projects and releases. But John has his doubts whether it will be enough if the large changes dry up.

It is clear that it's not an option to increase the rates. John thinks that this is a good challenge for the financial director.

■ 7.6 SUPPLIER MANAGEMENT

The *Supplier management* process is the counterpart of the *Contract management* process. The process ensures that agreements are made with the suppliers or subcontractors about the services or solutions that they provide.

The topics are the same as in *Contract management*. But where *Contract management* deals with customers, *Supplier management* deals with application management's suppliers.

Explanation of these topics is superfluous: these are the same as in *Contract management*. The challenge here is to understand the need and the demand. It is important to know what is needed, and the expectations, obligations and quality of the supplier.

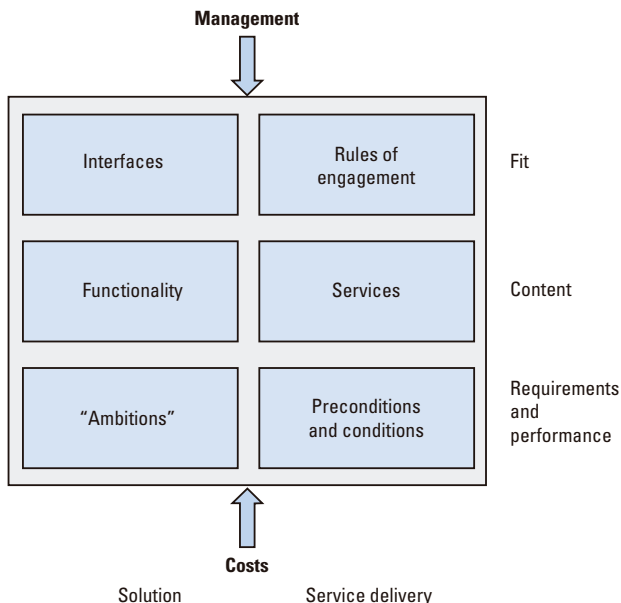


Figure 7.5 The agreements

The big pitfall, particularly with IT organizations containing technically oriented people with a high awareness of quality, is that they interfere too much with the content of the supplier's internal processes or with the construction of the solution. Sometimes this is possible, but it is usually not appropriate. Best practice is to make optimal use of the qualities of the supplier and not to be bogged down by one's own (perceived or real) superiority. The contract should reflect the goals that were aimed for in the selection process. The motto is: treat your supplier as you would like to be treated.

Patrick Johnson joined ISPM a couple of years ago, starting as a PARIS consultant. In recent projects he concerned himself with managing the subcontractors. This entails lots of discussion with the various parties about who does what and how and where and when. So he is well acquainted with various suppliers.

John thought that this was a good reason to ask Patrick to implement supplier management and to take on Tom's current responsibilities.

Tom had already given Patrick lots of documentation. But now he received everything: the requests for proposal, the proposals, the trail of emails that had gone back and forth during the proposal phase: really just a complete dump of Tom's library. Patrick thinks that it would be interesting to read though all of this on a couple of slow Friday afternoons, but at the moment he's got his hands full on other issues.

He has started by making an overview of ISPM's current suppliers. For each supplier, he records the products and services that they provide, which products and services ISPM uses most, which projects they are involved in, and which role – if any – they play in support.

He also compiles all of the suppliers' names and contact data, the contracts, SLAs, other agreements and any procedures that have been formalized.

Many suppliers have no SLAs. In fact India is the only party with a SLA that is reasonably satisfactory. The suppliers' reporting varies a lot, from very extensive and detailed to nothing at all.

When he can take a view on everything, Patrick first explores what the rest of ISPM thinks about the suppliers and which supplies are really needed. It turns out that there are a couple a contracts for components that are no longer used.

In the end there is, apart from the department in India, a clear shortlist with suppliers. This is what Patrick will focus on. He agrees on new service levels with the suppliers. Not too extensive, because that only takes up more paper. And he also schedules frequent meetings with the suppliers.

One of the permanent points on the agenda will be the exploration of the suppliers' possibilities. Patrick believes that they have more to offer.

He will also research whether the assignment of work to India can be improved, because that is also something that goes wrong from time to time.

8

Application strategy

■ 8.1 INTRODUCTION

The Application strategy process cluster takes a long-term perspective and addresses the lifecycle of the objects (applications) in information provisioning. You could say that application policy, application portfolio management and architecture are dealt with in this cluster. So Application strategy covers the substantive strategic part of application management. With limited budgets yet high speed of change, this cluster is becoming increasingly important.

Application strategy is growing

In the past, little attention was paid to application renewal and application policy. There was no thought about using existing systems in the future. Everybody focussed on development of new systems. However when we look at our current application landscape, we see that many applications have reached a considerable age. A twenty-year old (business critical) system is not exceptional. And we have learnt that the majority of plans to create something completely new, particularly when it is large or complex, fails partially or completely. So it is important to think up other scenarios.

The world is no longer designed

Most business processes are, to a greater or lesser degree, supported by applications. Organizations have come to depend on these applications. In some cases, the business process is executed almost completely by applications.

Organizations have got used to the peculiarities of these applications. They have learnt to deal with these peculiarities. Enormous investments have been made in these applications and the same applies to the people who work with them. Organizations are often unable to change these structures quickly and substantially.

Organizations will, therefore, have to treat the existing situation as the starting point.

The world is no longer manageable

‘Nobody’ saw the economic crisis coming. Nobody can predict what the organization will look like in a few years time, what the logo will look like, what the name will be and how the rest of the world – for instance customers, the market – sees the organization.

Complete redesign is usually not necessary

The design of completely new systems or architectures is therefore often senseless:

- Organizations don't have the time or money to renew everything;
- Every (technical) architecture is based on principles and visions that will be obsolete in five years' time;
- Organizations have users who are used to a certain way of working and the organization's capacity for change is very limited;
- Organizations can't predict what the world will look like in five years' time (mergers and acquisitions, divest part of the organization or not, crisis or no crisis);

- The business architecture and process architecture will also be obsolete in five years' time;
- 90% of the existing functionality works (and often quite acceptably) and we will also need 90% of the existing functionality in five years' time.

In real life we live in a legacy world. There's nothing wrong with that: our society is also legacy, our ways of working are legacy, our political system is legacy, people are also legacy.

Legacy means that you don't change for change's sake, but that you only change what needs to be changed, in order to change quickly.

Innovation and conservation

Our strength should be our ability to innovate quickly, but only where necessary. To summarize:

- Prevent acute problems and solve structural problems step for step;
- Only do new things if they are important and help the organization to retain or improve its position;
- Support and maintain the rest and leave it alone until something needs to be changed.

A recommendation is to continuously monitor the gap between the business processes' needs and the possibilities of the application (figure 8.1). And take action when the gap is too big.

This 'conservative' approach has become the dominant approach. It can be observed in topics such as portfolio management:

- What am I good at, and what not?
- What should I do to solve or prevent the worst problems?
- How can I benefit from developments that are really important?

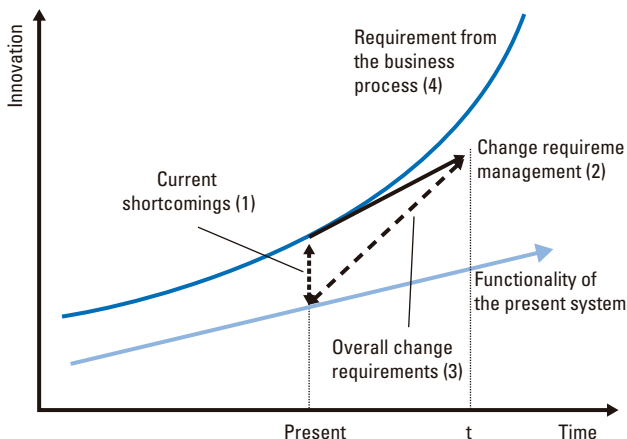


Figure 8.1 Alignment between business processes and information systems

It's all about recognizing the really important changes and recognizing and predicting bottlenecks. That is why it is also important to identify and solve structural bottlenecks in the operational process clusters.

Close collaboration with the other IT management domains

The process cluster Application strategy can't be executed in isolation from business information management and infrastructure management. This can be found in the various frameworks.

Structure

Three external factors impact the information provisioning of an organization.

- Developments in the technology;
- Developments in the environment of the user organization;
- Developments in the user organization itself.

This is why Application strategy contains three explorative processes that aim to obtain insight into the aforementioned developments, in order to determine the impact of these developments on the applications. The developments can impact upon a single application or all applications. The three processes are ‘surrounding’ processes, meaning that they provide input for the two core processes of the Application strategy cluster. These core processes are: *Application lifecycle management* and *Application portfolio management*.

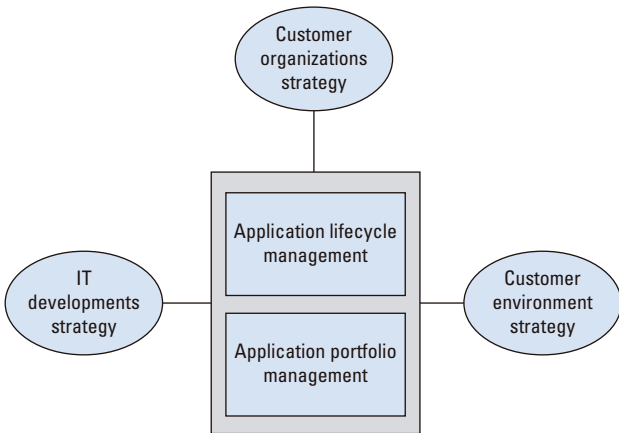


Figure 8.2 Processes in the Application strategy cluster

So the process cluster Application strategy comprises five processes: three explorative processes and two core processes (see figure 8.2). The explorative processes are:

- *IT developments strategy*: the process that monitors and assesses new technological developments;

- *Customer environment strategy*: the process that obtains insight into the developments in the environment of the user organization in relation to the application(s);
- *Customer organizations strategy*: the process that obtains insight into the developments within the user organization(s).

And the two core processes are:

- *Application lifecycle management*: the process that defines a strategy for the future of an application, and translates it into actions;
- *Application portfolio management*: the process that defines a strategy for the application portfolio as a whole.

These five processes are described below, followed by the case for the whole process cluster Application strategy.

■ 8.2 IT DEVELOPMENTS STRATEGY

Developments in IT occur with great speed, particularly in the infrastructure domain, and influence our daily life, the way of working and developments in organizations.

IT developments strategy determines which technological developments in the IT sector are interesting for the information provisioning of the organization and how the applications are impacted.

The technology can refer to tooling for design and development (such as programming languages), functionality (such as the introduction of standard solutions), tooling for support and maintenance, and infrastructure (new platforms, smartphones).

But it is not only about new developments and new possibilities. Sometimes investment in currently used technology is needed, for instance because the supplier will only continue to provide support for newer versions or because the currently used technology is ‘end of life’.

■ 8.3 CUSTOMER ORGANIZATIONS STRATEGY

The information provisioning has always got to be aligned with the business processes of an organization. That is why it is, of course, essential to be aware of the developments within that organization. The *Customer organizations strategy* process identifies the developments in the customer organization(s). The goal of this process is to determine the impact of developments in the customer organization(s) for the application portfolio. For example, application-related barriers can be identified that would affect these developments in a negative way, and appropriate measures can be taken.

Most organizations will be looking two to five years ahead.

■ 8.4 CUSTOMER ENVIRONMENT STRATEGY

Data exchange between organizations is increasing. This exchange is almost always accomplished by means of interfaces between the various applications of the organizations involved.

Customer environment strategy analyzes developments in the exchange of information and data between user organizations (information chains) and translates this into requirements and opportunities for the applications.

The goal of *Customer environment strategy* is to determine the impact of developments in the environment of the customer/user organization for the application portfolio.

The word ‘chain’ or information chain is often used to indicate the information flows across multiple organizations. The term ‘in the chain’ is often used. A complicating factor is that each organization has a different perception of where they are ‘in the chain’, as is illustrated in figure 8.3. Organization C perceives the chain completely differently than does organization B.

Organization C is involved in other organizations and other developments. This means that organizations have little insight into the other organizations’ perspectives, particularly when they are further along the chain, and this can lead to unpredictable reactions from other organizations. This is why it is important to understand developments in the environment of the applications and their impact.

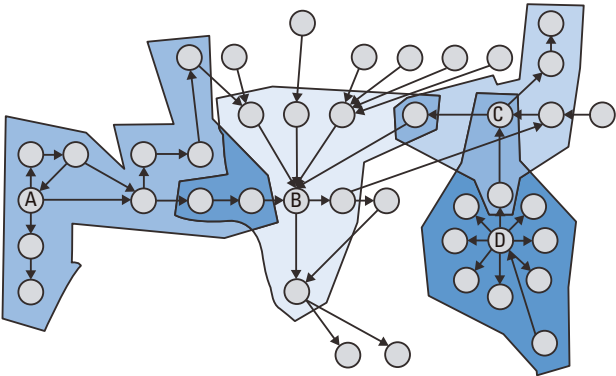


Figure 8.3 Example of connected chains

■ 8.5 APPLICATION LIFECYCLE MANAGEMENT

Application lifecycle management determines a strategy for the future of an application, so that the application can optimally support the business process in the future.

Applications generally age and degrade slowly. *Application lifecycle management* identifies which structural changes and improvements are needed to retain the quality of the application in future years and to realize good alignment with the business process and the desired policy.

Structured investments in the beginning of the lifecycle of an application are often profitable. In any case they are certainly cheaper than having to modify a heavily outdated application.

Application lifecycle management addresses the following tasks:

- Identifying the current strengths and weaknesses of the application;
- Determining the impact of developments in the user organizations and the organizations in the chain;
- Determining the impact of the technology;
- Creating scenarios and sketches in order to achieve a new, or modified, future-proof application.

■ 8.6 APPLICATION PORTFOLIO MANAGEMENT

Whereas *Application lifecycle management* looks at a single application, *Application portfolio management* looks at the whole application landscape. The goal of *Application portfolio management* is to align and coordinate the various parts of the application landscape (or the complete information provisioning) and any associated large and consequential changes. Or in other words, portfolio management keeps an eye on the commonality of the applications.

This is achieved by coordination of three topics:

- The application landscape and its breakdown. The application landscape describes which applications there are, what their boundaries are, and how they interrelate;
- Common resources and standards. Sharing resources and using standards can result in cost savings but there are sometimes disadvantages;
- The change portfolio of the applications. The complete set of consequential changes and renewals of the various applications can result in complications, for instance because of interdependencies or because it is just too much for an organization.

Insight into the whole portfolio and the resultant coherence can lead to an adjustment of the plans for renewal. So *Application portfolio management* organizes decision-making regarding the whole application portfolio and, in so doing, ensures its optimization.

Directly after starting at ISPM, John noticed that Application strategy had become a neglected area.

In the last couple of years it has only had a bit of work done on it, in an ad hoc way and usually in reaction to a customer situation. Not very structured. Each time something was done with the Application strategy, it was undertaken by different ISPM employees. Resulting in the same 'new' functionality being thought up and realized several times. Each time a little bit different, of course.

John thinks that this is a shame. Luckily the new director is interested in the future and not just in cost cutting. He understands that you sometimes need to invest and it doesn't take John long to convince him about the need for Application strategy.

First they make a scan of the current version of PARIS. The conclusion is that although a substantial part of the system is built using old technology, it is solid and flexible enough. The previous renewal, that John managed and that focused on the calculation software, was most beneficial.

There is a need for better and more interfaces. These requests for change are added to the release calendar because they are generally fairly easy to specify.

John and the product manager organize workshops with the various ISPM customers. These workshops elicit many innovative ideas. For instance a self-service portal for subscribers. Subscribers can use this portal to place advertisements in the publications, including their own logos and photos. And they can also set up a blog of their own.

They are considering an app for delivery personnel, available via internet, and also on smartphones and tablets. The delivery personnel can use the app to get an overview of their delivery district, changes in their delivery district and their income. The app also provides an easy way of finding somebody else to take on his or her deliveries, if needed.

Five customers are enthusiastic and sign up immediately for the new facilities. And to John's delight, they included a new customer. They had been negotiating for a while with this customer, but John's innovation workshop made the difference.

The product manager is also enthusiastic. "I've spent too little time on this", he says to John during the evaluation drinks reception. "I thought that it would just cost a lot of money and wouldn't deliver any benefits, but we should do this more often".

John has good reason to be pleased.

9

Application management organization strategy

The last process cluster in ASL is Application management organization strategy. As the name already suggests, it's about the future of the application management organization. The goal of Application management organization strategy is to make choices regarding the services that the organization will provide in the future, to substantiate them, and to ensure that they are translated into concrete services in the future.

There are various reasons why Application management organization strategy is now more important than it used to be:

- Organizations can't learn on the job. It also applies to internal IT organizations: you have to be capable of delivering good services before you start to deliver;
- There are a great many possible forms of application management services. There is specialization according to market, the intrinsic nature of the services (such as the provision of software packages, system integration, support and maintenance), and the cost charging structure of services and technology. There are often multiple specializations along these axes, and consequently specialization in technology and in the form of services and

market. So there is a wide range of potential services to be chosen. However an organization cannot deliver all of these services. Making the right choice is therefore essential for the continuity of the application management organization;

- The fundamental renewal and modification of services is laborious and costly. It not only requires employees who have the new skills in their genes, it also entails having managers who can manage it well. It often also needs another image in order to deliver these services, and frequently another application infrastructure or other (financial) administration. And all of these changes have to occur at roughly the same time, and as quickly as possible. So it is important for the application management organization to continuously monitor the alignment of the policy of the organization with the services.

Topics and processes

The processes in Application management organization strategy are closely related to the topics that this process cluster addresses. The topics and processes are (see figure 9.1):

- Customers and market. The *Account & market definition* process monitors developments and determines the policy with respect to the application management organization's customers and market;
- Suppliers. The *Supplier definition* process monitors developments and determines the policy with respect to the suppliers (or subcontractors) of the application management organization;
- Technology. The *Technology definition* process monitors developments and determines the policy with respect to the technology and the technological developments;
- Capabilities. The central theme of the *Capabilities definition* process is the capabilities of the organization: the core qualities (competences) of the organization, the skills of the employees and the extensive quality system to manage and disseminate the

knowledge. This can be regarded as quality management at the strategic level.

The *Service delivery definition* process is positioned in between these processes.

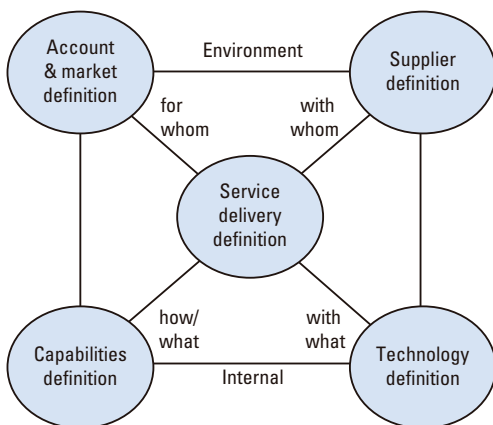


Figure 9.1 The process cluster Application management organization strategy

■ 9.1 ACCOUNT & MARKET DEFINITION

Account & market definition, as the name suggests, concerns itself with customers and the market. In this process the impact of the developments at the customer(s) are considered, as are the developments at other parties related to the customer (competitors or colleagues) and how these can impact the services.

This insight is a good starting point to determine which services, changes to existing services and new services are needed and also to which existing or new customers these services will be offered.

Both external (commercial) and internal IT organizations should, of course, carefully consider which services they are obliged to deliver and are allowed to deliver. It's not only the limited resources for investment but also the perception that the customer has of the application management organization which determine the likelihood of success of the new services and the degree to which the application management organization 'is allowed to' deliver.

So *Account & market definition* not only looks at developments within the customer organization(s), but also at the services that are delivered, the perception of the services and the image that the customers(s) have, and the range of tools (such as relationship management and account management) that can influence these aspects.

■ 9.2 CAPABILITIES DEFINITION

Organizations are good at delivering certain kinds of services. Experience has demonstrated that changing services and changing the core values related to these services is a laborious process. The capabilities, the core competences of organizations, are difficult to change. This is in stark contrast to the frequently changing demands from the customers and the market.

Capabilities definition is the process that equips an organization to fulfill future market needs. This process determines the future requirements for the organization's and employee's skills and knowledge. The process addresses the required breadth and depth of knowledge and skills, and also the instruments (in the broadest sense) that capture and share the knowledge and skills.

It often takes years before a structural change in the organization's way of working has found its way into the genes of the employees. It's even more difficult when the required skills and knowledge differ radically

from the organization's current qualities. Employees, who have been working for twenty years in a structured way to develop applications with a high degree of reliability, can find it very difficult to use an Agile way of developing applications that possesses a throw-away character.

■ 9.3 TECHNOLOGY DEFINITION

Of course technology is another topic that application management has to address at a strategic level. Application management requires hefty investments in technology, not so much to acquire them but more in their use. The employees have got to be able to use them (if you're building systems with Java, you've got to know a lot about Java); they have expectations that the quality system provides tips and tricks for using the technology; they will make extra facilities and standard functionality.

Technology definition is the process that chooses the technical resources which will be used to realize the organization's future services.

Pertinent questions are:

- What are the consequences of the retirement of a particular technology?
- Which new technologies are available and what possibilities do they offer?
- How can a new technology be introduced into the organization?

■ 9.4 SUPPLIER DEFINITION

The introduction in this chapter outlined the fact that there are many manifestations of application management services and that individual application management organizations cannot deliver all of these variations. So it's logical to collaborate.

The *Supplier definition* process looks at suppliers and possible partners for collaboration. The goal of *Supplier definition* is to actively ensure optimal services in the future by determining the role and engagement of external suppliers, and by translation of this policy into a functioning organization and structure.

It is important that there is a clear picture of what the contracting application management organization expects of the supplier or collaboration partner: what contribution will they make? Will they provide a solution or technology, specific skills or cheap capacity, or a favorable image in the eyes of the customer (who appreciates the supplier or partner)?

The nature of the collaboration is also important. There are various kinds: a supplier, a subcontractor, an equal partner, a demand-supply organization, a prime contractor.

■ 9.5 SERVICE DELIVERY DEFINITION

The central process in Application management organization strategy is *Service delivery definition*. This process collects all of the plans and developments of the surrounding processes and forges them into a coherent and aligned strategy.

In the end, it is important to know whether there is demand in the marketplace for the aspired capabilities. Collaboration with another party has got to be organized and assured. The organization must also develop the skills to use the acquired technology and apply it to services for customers. *Service delivery definition* organizes this. Pertinent questions are:

- What are the services that will be provided?
- To whom?
- How will it be realized (make or buy)?

- With which *resources/development tooling* or with which suppliers?
- Which (high-level) capabilities are needed?

It will also be necessary to visualize how to get there and how to assure that. An approach that can be used to achieve this is illustrated in figure 9.2.

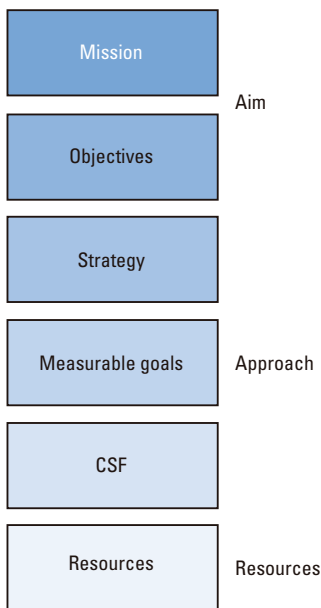


Figure 9.2 Results of Service delivery definition

Possible phases are:

- Formulate a vision. This vision gives a concise view of which capabilities will be needed in two to three years' time to provide which services to which groups of customers;
- Formulate targets in terms of measureable units that make the vision concrete;
- Define one or more strategies to achieve the targets;
- Identify critical success factors (CSF) for the strategy;
- Estimate and allocate the required resources for realization;
- Plan the realization targets.

There hadn't been an Application management organization strategy workshop for ages, in fact not after the first and only workshop that John had organized. He reckons it's a good idea to pick things up where he left off and to base it on ASL 2.

Because the organization has grown, he organizes not one but a whole series of workshops. Each workshop has its own theme: one of the Application management organization strategy processes. The first workshop is about capabilities. It addresses questions such as: how can we develop the capabilities that we need, what are we going to do ourselves, and what will we outsource?

It is evident that they want to get a better grip on quality.

The procedures and processes are well-structured and the costs are predictable, but sometimes it's a bit too formal and customer intimacy suffers.

By managing more based on quality they can achieve an optimum balance. It also improves the quality of work for the employees. But everybody knows: money is money. They haven't forgotten that.

Working with India is one of the hot topics in the 'Suppliers' workshop. This can be problematical. The distance, both physical and cultural, is large and there are often misunderstandings. It always costs a fair bit of effort to straighten things out. They sometimes wonder whether it has saved any money. "Sometimes it costs so much effort to explain what you want, that you might just as well do it yourself".

Somebody suggests exploring opportunities in Eastern European countries. One of the employees has heard positive things about it from his brother-in-law, who also works in IT. In any case, it's closer. And culturally closer as well.

But they can't just drop India as a supplier. Though in the future they are going to take more care in deciding what they are going to have done in India, and what not. The core of PARIS is well constructed, with very good documentation, so that could be done in India. But more discussion with customers is needed for the newer parts, so it would be better to do that locally.

And to do some innovation, they start a small project to explore the opportunities offered by social media.

10 Implementation and use

■ 10.1 IMPLEMENTATION AND CONFIGURATION

In chapter 2 we described application management. The various application management tasks were described in the following chapters.

Organizations often want to improve and professionalize the execution of their work, and sometimes they are forced into this due to circumstances. The question is how? How to start, what are the various scenarios, what is it going to look like and what resources are available?

This chapter deals with these questions. First we will look at the model: what is ASL exactly, what is the relationship with application management in real life, what does 'best practice' mean? After that we will deal with the goals and the way to achieve them.

Finally we will deal with all of the instruments, resources and other forms of support that are available for the implementation and configuration of application management processes and organizations.

■ 10.2 THE FRAMEWORK AND REALITY

ASL is based on day-to-day practice, as we claimed at the start. ASL is a model, created within a real life situation, that after years of application and modification has finally resulted in the current framework.

ASL is also a process model, an abstraction of reality. A model of the tasks, processes and activities that occur within application management. These tasks are structured in a logical way into processes and process clusters.

There is no in-depth breakdown of the processes in process steps. That is a conscious decision. ASL is a universal model, but the implementation of application management differs for each organization. The omission of such an in-depth breakdown helps to prevent over-standardization.

■ 10.3 THE SECRET OF BEST PRACTICES

There are various aspects to the implementation of processes and organizations, not because it is that difficult, but because of the susceptibility to hobbyhorses and 'religion'. Here are some examples of these hobbyhorses.

Processes are organization-independent, so the process model works for each organization.

There are many examples of the actual implementation of process models in a standardized way, independent of the characteristics of the organization where the implementation has taken place. Not only are there differences in the structure of the organization, but also the underlying driver and the professionalization goal, often differ. This means that the organization, driver and goal should be carefully

considered in order to prevent bureaucratic procedures where they are inappropriate.

Everything should be uniform and integral.

Following on from the previous point, there is often a belief that the same process should be used everywhere. This only works when the organization has very comparable processes (both in size and in resources) and also delivers and competes in the same way. The support and maintenance of large-scale bespoke software, with a high reliability level, requires a different configuration than the support and maintenance of a dynamic website.

We are different, that would never work in our organization.

This statement is at the opposite end of the spectrum and is often heard. And of course: organizations are hardly ever exactly the same, with the same values, hardly ever is the driver and goal for implementation of processes the same, leading to processes that differ from organization to organization. On the other hand, there are many similarities between application management in various organizations and most of the 'laws' of application management apply universally.

Reality resides somewhere in the middle, and the golden mean is almost always the right way.

Approximately 20% of application management processes in various randomly chosen organizations differ and these differences usually make sense and are important.

Adjustable best practices: the key to the gap between theory and practice.

This is why ASL has best practices. Best practices are templates, examples, checklists, descriptions, forms, etc., that are necessary or handy for the implementation or execution of the processes. They

can be downloaded from the website of the ASL BISL Foundation, or copied from a colleague. These best practices are adjustable; they can be used as a foundation on which local flavors can be applied. This provides the opportunity to adjust the best practices for the 1 to 20 percent that differ with the specific organization.

The big advantage is that almost 80 percent doesn't need to be thought up. No time is spent on creating a procedure or template from scratch. This results in much speedier implementation of improvements. In addition, best practices help people to visualize a particular process or activity, which steers the discussion in the right direction. Best practices enable re-use of the investment in experience and lessons learned made by other organizations.

Best practices also help in the ways that ASL can be developed in order to respond to new developments without modification of the framework. The fundamentals of the application management processes are relatively stable. The differences are in their application. And the application is captured in the best practices. By collecting as many best practices as possible, that cover as many situations as possible, every application management organization can choose the most applicable best practices. Offshoring and outsourcing, new technologies and new methods: the best practices ensure that ASL can be, and remains, the de facto standard for application management.

■ 10.4 SCENARIO'S AND IMPLEMENTATION

As mentioned in the previous paragraph, the goals and drivers for professionalizing application management and the adoption of ASL can differ significantly.

Outsourcing – which makes lines longer and agreements with the customer more formal – might be the driver for one organization, but for another organization the driver might be cost reduction.

Goals and content can also differ: what is the degree of professionalization required? Is maturity level 2 good enough, or will the organization have to grow to level 4? Is the focus on the operational processes or do the strategic processes need more attention?

How critical is the situation? Is a low degree of professionalization problematic or even terminal for the organization? And how much knowledge and experience does the organization have? Can they execute the implementation themselves or do they need external help?

Because every situation is different, we have described four scenarios covering different situations.

The Quality scenario

If it is urgent, then organizations are often willing to invest time and money in quality improvement. That makes it a lot easier, since if there is not enough knowledge available internally it's possible to hire external knowledge and skills. These hired hands can analyze and describe what work needs to be done.

A risk to consider is that because internal employees haven't had the opportunity to participate, they may not be willing to accept the end products.

The Result scenario

The knowledge is often present in an organization, but it is not available in the capacity that is required. Or multiple initiatives

have been started, but have come to a halt under a mountain of improvement suggestions. Or a perfectionistic attitude was the reason for failure.

In these circumstances the Result scenario can help. Internal employees rapidly set up the organization and create new or improved procedures for the various processes. In this case, best practices are the means to achieve results quickly.

In order to solve any capacity problems, staff can be hired to take on the daily tasks so that the internal employees can focus on the process improvements. This ensures a greater commitment from the employees, which is absolutely essential.

The Growth scenario

If the problems aren't too big, then organizations aren't usually prepared to invest much in improvement activities. The way to achieve steady professionalization despite budgetary restrictions is the Growth scenario. Each year a limited number of bottlenecks are pinpointed and actions are defined to solve them. Best practices are also applicable to this scenario. Self-assessments can help to identify bottlenecks.

The Team scenario

From a certain maturity level onwards, improvement processes are just part of day-to-day activities. Improvement just happens.

■ 10.5 STARTING WITH ASL

If your organization wishes to start with ASL, go to the site of the ASL BiSL Foundation (www.aslbislfoundation.org). Here you will find background information, books, articles and white papers, best practices etc.

One of the best practices is the ASL Starter Kit, an instrument in intended to help you to quickly apply ASL in your organization without the need for consultants.

The site also contains information about thematic sessions that can be attended free of charge, and the available certification and training.

And if you want to know more, then you can always contact the ASL BiSL Foundation. They can direct you to other organizations or consultants.

John has been working as interim manager for a couple of months and it feels as if he never left. During this time he is pleased to notice that several of the improvements that he initiated a while ago, have not gone to waste. So he can focus on the processes that have been neglected in the previous years.

Working with process owners in the past was a success. The processes that still have an owner operate at a higher quality level. So he decides to follow the same approach: designate an owner per process who tackles the biggest bottlenecks on an annual basis, with or without the assistance of an external consultant.

John has to negotiate with the director to appoint new process owners. Although he wants to improve things, and is prepared to invest, the director is concerned about creating overhead and worried that the process owners might lose track of their own activities.

John manages to reassure him that it is not his intention to create extra managers. The process owners regard their extra activities mainly as a means to get a better grip on their own activities and, in so doing, improve their work satisfaction.

His improvement scenario seems to be a mix of the Result scenario and the Growth scenario. This doesn't surprise him. He prefers steady improvement activities; he thinks that they embed themselves better than others. But if there is a big backlog, such as in some processes, some more effort is needed to achieve good results.

Weighing up things, he believes that they are on the right track, something that the employees and the director agree upon.

They will never reach the end, because things can always be improved. But each improvement makes a difference, resulting in better work for the employees and a better result for the customers.

Appendix 1

Case study VGK/ISPM

The case in this book is about ISPM, an organization based in the UK that provides publishers with IT services. The organization is the successor to VGK, the supplier organization that was featured in both the previous edition of this book and the BiSL management guide. This gives an interesting historical perspective.

ISPM emanated from VGK which, in turn, emanated from UPC, the United Publishing Combination. VGK is the privatized UPC IT organization. The most important VGK application was PARIS and the support and maintenance of PARIS were the core activities after the privatization of VGK. Many customers are still using this software package. Five years after the privatization there were 22 customers using PARIS, 14 of which were also supplied with infrastructure management services.

Five years ago the name VGK was changed into ISPM: Information Services for Papers and Magazines. The company wanted to reduce the association with the UPC ‘mother’ organization and was increasingly servicing international markets.

People had forgotten the original meaning of the abbreviation VGK and in any case, the name sounded dated and not aligned with the new activities.

Yet another name change is being considered. The name ISPM doesn't seem quite right. And this abbreviation is confusing too: some people think that the organization does something with project management. Evidence of this can be found in the followers of the Twitter account that the marketing department recently opened. But they haven't yet started looking for a new name.

PARIS

The software package PARIS is still around. It has been renewed fundamentally in the past ten years. The renewal was started when John Hollander was a manager at VGK. It took twice as long as planned and PARIS still isn't state of the art, but it is robust and flexible enough for the customers' needs. PARIS is no longer ISPM's only source of income, but it is still very important for the company.

PARIS' market position has grown strongly during the recent years. Part of its success was due to the decision to keep the software package and renew it where needed. Some of the competitors chose to develop new applications and failed.

There are currently 35 customers for PARIS, and that is a considerably stronger position than before, particularly when considering that many customers have merged. The new customers include other large publishers, resulting in UPC no longer being the largest customer. The position in the market has been strengthened enormously by the acquisition of the large customer GDC, and the first foreign customer: Dutch Magazines.

PARIS is no longer the monolithic software package that it used to be. Some parts of PARIS have been decoupled and some have been transformed into web services. Customers can connect PARIS quite easily to various CRM or financial packages from other suppliers, including recent Cloud solutions.

Additional functionality from other applications has been integrated with PARIS, such as workflow and electronic authorization.

The ISPM organization

ISPM comprises two large divisions and two smaller departments. The largest division is Enterprise Solutions. This division is responsible for the support and continuous development of PARIS and the other large software packages that have been acquired or developed. For example SOFIA, a financial package based on a service oriented architecture, and ACCRA, a system for sales and placement of advertisements.

Then there is the AM Factory. As the name suggests, this is for development and maintenance of smaller applications. The ‘factory’ is flexible and not focused on volume. Alongside various specific applications, there are packages for the administration of subscriptions and advertisements that are much smaller than those provided by Enterprise Solutions. Smaller publishers are customers of this division.

The Consultancy department focuses on supporting customers with the implementation of packages, but recently much support has been provided to the customers’ information management organizations. The ISPM consultants have built up considerable knowledge about the publishers’ business processes and related developments, and have become a valuable sparring partner for their customers.

The Services department provides customers with various all-in services. ISPM sold their data center some years ago but many customers still wanted integrated application and infrastructure services. In order to provide these services, the Services department partners with the new data center owner, and more recently, with a supplier of Cloud services. ISPM has taken on responsibility for all of the services and has made agreements with the infrastructure suppliers.

The Services department also offers new services such as staff augmentation for information management organizations.

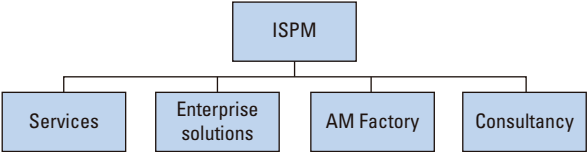


Figure B1.1 Structure of ISPM

History

John Hollander started as manager Application Management twelve years ago at VGK. The organization had been privatized a couple of years previously and as a result of the changed circumstances, such as more formal relationships and more customers, there was a need to professionalize. John did most of this himself, sometimes too much (he always had difficulty with delegating), but he was supported to an important degree by the quality manager Miriam Hill. After leaving VGK, Miriam traveled around the world helping other organizations to adopt ASL. John and Miriam kept in touch and John often received a postcard from India or China.

John left VGK after four years. There were too many disagreements with the new director's commercial policies. The director wanted more profit and this meant that the old shareholders and users, including UPC, became less important.

A year ago, after some changes in the shareholders, a new director was appointed. And this new director is more concerned about the future and has spent the past year formulating a long-term policy. Developments within the publisher's world made this a necessity: a newspaper subscription is no longer such a requirement now that so much news is freely available. This was an opportunity for ISPM to collaborate with customers and explore new scenarios.

There is now financial room for investment. The director encourages all employees to suggest ideas and considers all ideas with care, whoever has provided them. Quality improvement is once again an important topic and the director regards the employees as the most important production resource.

Various employees recommended that John should be asked to rejoin the company and the director has invited him to return as an interim manager.

John has started enthusiastically. He is pleased to see that many of his former colleagues still work for ISPM. Such as Harvey Bennet who still deals with Configuration management, Tony who is still responsible for the Impact analysis process, Paul Green, Reuben Jones and Pilar Rodriguez.

He makes much use of their knowledge and experience, just like he used to do.

Appendix 2

ASL BiSL Foundation

ASL 2 is a public domain library that is managed by a foundation. This foundation, the ASL BiSL Foundation, was founded in 2002 by a number of like-minded organizations that felt strongly about the professionalism of application management and wished to promote it by way of the Foundation. This promotion now takes place via publications, congresses, themed evenings and the collection and publication of best practices in this area.

Objectives

The Foundation's raison d'être is jointly working on improvements and supporting the members in:

- Improving the management processes within the application management domain and the business information (system) management domain and information management;
- Collecting and sharing information on ASL 2 and BiSL;
- Developing and adopting best practices;
- Improving relations between the primary business processes and the IT function.

Activities

The Foundation initiates the following activities:

Best practices

The participating organizations provide best practices. The Foundation reviews the quality of these and makes them available in the public domain.

Development

ASL and BiSL are developing continuously and new contributions are always welcome. The Foundation creates a platform through which new opinions can be bundled and helps to improve the framework by organizing themed meetings, discussion groups, etc.

Training

Of great importance is the training of application managers, business information administrators, information managers and the management and staff who contribute to the support of the business processes by IT. The Foundation stimulates and supports training organizations in offering ASL and BiSL training. The Foundation also facilitates, together with the independent training institute APMG-International, the availability of comprehensive examinations and the accreditation of training organizations. At the same time, educational institutions (intermediate- and higher vocational and universities) are encouraged to include ASL and BiSL in their curricula.

Certification

The Foundation and the Dutch Standards Institute NEN have established a standard (NEN 3434) for the maturity levels of application management organizations. Application management providers can be certified by an independent institute. The Foundation contributes to this by increasing the quality offered by providers.

NEN 3434 has also been used by the International Standards Organization (ISO) as input for the development of an international standard for Application Management: ISO/IEC 16350.

Publicity

The Foundation provides articles, presentations, books, and other media and runs congresses in order to bring ASL and BiSL to the attention of the respective target audiences. All publications can be requested or ordered via the website and/or e-mail.

Membership

There are various ways of participating in the Foundation's activities, depending on the size and goals of the organization. For information about membership, visit www.aslbislfoundation.org.

More information

Visit www.aslbislfoundation.org or send a request to info@aslbislfoundation.org.

Appendix 3

Sources and more information

Van der Pols, R. (2009). *ASL 2 – A Framework for Application Management*. Van Haren Publishing. ISBN 978 90 8753 313 7

This book contains the definitive description of the ASL 2 framework and major terms of reference.

Van der Pols, R. and R. Sieders. (2014). *ASL*2 Self-assessment*. Van Haren Publishing. ISBN 978 90 8753 740 1

Juurlink, A. (2011). *Applicatieportfoliomanagement voor IT-complexiteitsreductie – Management Guide* (Dutch). Van Haren Publishing. ISBN 978 90 8753 669 5

Van der Pols, R. (2005). *Strategisch beheer van de informatievoorziening met ASL en BiSL* (Dutch). Academic Service. ISBN 90 395 2210 3

Van der Pols, R. (2009). *Modern leveranciersmanagement* (Dutch). Academic Service. ISBN 978 90 1258 106 6

Backer, Y. and R. van der Pols (2012), *BiSL Pocket Guide* (2nd revised edition). Van Haren Publishing. ISBN 978 90 8753 711 1

This book describes the BiSL framework in the same way as in this book. The case in this pocket guide is GUC, until recently ISPM's biggest customer.

Van der Pols, R., R. Donatz and F. van Outvorst (2012), *BiSL, a Framework for Business Information Management* (2nd revised edition). Van Haren Publishing. ISBN 978 90 8753 702 9
This book contains the definitive description of the BiSL framework.

In addition to these publications there are various articles and white papers about ASL and related topics. These can be downloaded from the ASL BiSL Foundation website: www.aslbislfoundation.org. They are also available via the Van Haren Publishing e-Knowledgebase: www.vanharen.net.

Appendix 4

The ASL 2 process model

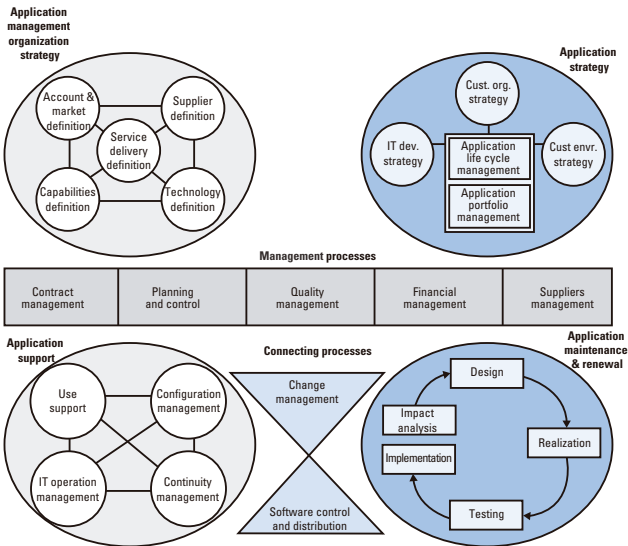


Figure B4.1 The ASL 2 process model

Index

A

acceptance test 64

Account & market

definition 117

agreements 84

Application management

organization strategy 26

application landscape 111

Application lifecycle

management 111

Application maintenance and

renewal 23, 51

Application management 13

application management,

business case 97

Application management

organization strategy 115

application objects 77

Application portfolio

management 111

Application strategy 24, 103

Application support 21

application support

activities 21

Application support cluster 33

Application support

processes 35

Application support

variables 35

ASL 18

ASL 2 21

ASL 2 framework 21

ASL 2 - Implementation 125

ASL BiSL Foundation 139

ASL, process clusters 21

availability 42

B

business information

management 14

C

- Capabilities definition 118
- chain computerization 28
- Change management 73
- changes 74
- componentization 28
- components 28
- conditions 86
- Configuration management 39
- Connecting processes 23, 73
- Continuity management 46
- Contract management 84
- Customer environment
 - strategy 109
- Customer organizations
 - strategy 109
- customer-supplier
 - relationship 27

D

- Dependability 42
- Design 57

E

- Efficiency 43
- expectations 86

F

- Financial management 96, 97
- functionality 85
- functional system test 64

G

- Growth scenario 130

I

- ICT developments strategy 108
- Impact analysis 53
- Implementation 68
- infrastructure management 13
- integration test 64
- interface 84
- ISO 16350 141
- IT operation management 41

L

- legacy 105

M

- Manageability 43
- Managing processes 24, 81

P

- performance 86
- Planning and control 89
- Proactive communication 37
- proactivity 31
- problem management 94

Q

- quality 93
- quality aspects 42
- Quality management 93

quality of organization 93
quality of the process 93
quality of the product 93
Quality scenario 129
quality system 93

R

reactive communication 37
Realization 60
releases 74
Reliability 43
Result scenario 129
rules of engagement 84

S

Security 47
Service delivery definition 120

services 85
Software control and
distribution 77
Supplier definition 120
Supplier management 99

T

Team scenario 130
technical system test 64
Technology definition 119
Testing 63

U

unit test 64
Use support 36

ITIL and ISO 20000



English
€39.95
excl tax

Foundations of ITIL®

Encompasses all of the implications of the 2011 refresh of ITIL, this ITIL Foundations book looks at ITSM Best Practices, focusing on the Lifecycle approach, and covering the ITIL Service Lifecycle, processes and functions for Service Strategy, Service Design, Service Operation, Service Transition and Continual Service Improvement.

ISBN HARD COPY 978 90 8753 674 9

ISBN EBOOK 978 90 8753 923 8



English
€29.95
excl tax

Passing the ITIL® Foundation Exam

A complete and thorough explanation of all key concepts for ITIL Foundation Exam, this title contains sample questions and practical examples. Endorsed by APMG, it has been designed to follow the official ITIL Foundation Exam syllabus.

ISBN HARD COPY 978 90 8753 664 0

ISBN EBOOK 978 90 8753 912 2



English
€17.50
excl tax

ITIL® - A Pocket Guide

A concise summary to ITIL®, providing a quick and portable reference tool to this leading set of best practices for IT Service Management. Available in: English, German, French, Dutch

ISBN HARD COPY 978 90 8753 676 3

ISBN EBOOK 978 90 8753 925 2



English
€15.95
excl tax

ISO/IEC 20000:2011 - A Pocket Guide

A quick and accessible guide to the standards ISO 20000-1:2011 and ISO 20000-2: 2012, essential for all IT service organizations.

ISBN HARD COPY 978 90 8753 726 5

ISBN EBOOK 978 90 8753 787 6

Van Haren Publishing is a leading international publisher, specializing in best practice titles for IT management and business management. Van Haren Publishing publishes in 14 languages, and has sales and distribution agents in over 40 countries worldwide: www.vanharen.net

Copyright protected. Use is for Single Users only via a VHP Approved License. For information and printed versions please see www.vanharen.net

Leading ITSM publications



English
€39.95
excl tax

Metrics for Service Management: Designing for ITIL

This title is the sister book to the global best-seller Metrics for Service Management. Taking the basics steps described there, this new title describes the context within the ITIL 2011 Lifecycle approach.

ISBN HARD COPY **978 90 8753 648 0**

ISBN EBOOK **978 90 8753 649 7**



English
€39.95
excl tax

The ITIL® Process Manual

Covers the basic approaches to the fundamental processes – companies will find the concise, practical guidance easy to follow and implement.

ISBN HARD COPY **978 90 8753 650 3**

ISBN EBOOK **978 90 8753 651 0**



English
€39.95
excl tax

The Service Catalog

Practical guidance on building a service catalog, this title focuses on IT community relationship with the business and users. Including useful templates on key documents such as OLAs and SLAs, this is definitive guide for all those delivering this tool.

ISBN HARD COPY **978 90 8753 571 1**

ISBN EBOOK **978 90 8753 572 8**

Application Management: ASL®



ASL® 2 - A Framework for Application Management

This book is the official manual of ASL 2, an evolutionary improved version of ASL that was introduced in 2001. ASL, Application Services Library, is a public domain process framework for application management.

ISBN HARD COPY **978 90 8753 313 7**

ISBN EBOOK **978 90 8753 822 4**



ASL® 2 - A Pocket Guide

This pocket book offers a generic introduction and additional background information of ASL 2, illustrated by case study texts

ISBN HARD COPY **978 90 8753 643 5**

ISBN EBOOK **978 90 8753 803 3**