

## PROFESSIONAL APPLICATION MANAGEMENT BY PACKAGE AND ASP PROVIDERS.

*After the infrastructure management had been set up in a process-oriented manner in many organisations, the focus started to shift to application management and functional management. This can among other things be seen by the amount of attention that ASL<sup>1</sup> (the framework for application management) is receiving.*

*The topic also receives a great deal of attention in the management of standardised solutions (packages and ASP solutions)<sup>2</sup>. Traditionally organisations that supply these products are slightly further removed from the customer than suppliers of customised products, but on the other hand there is growing demand for transparent and manageable services. There is also a growing tendency for customers to use standardised solutions. Customers, who previously only worked with customised products, are now also expecting similar approaches in the case of package solutions.*

*As a result of the twenty-four hour economy and further standardisation it is therefore also becoming essential for package and ASP suppliers to properly handle these processes. For customers it is also becoming important to start placing demands on suppliers in this respect.*

*This article looks at this situation on the basis of a number of practical experiences and examples. In addition, it provides a number of tips for both suppliers and customers.*

**Authors:** Yvette Backer is a project leader at Getronics PinkRoccade Civility, where she is involved with the implementation of ASL. Among other things, she has written the ASL Management Guide. Jan-Jacob Sybenga is a project manager at Getronics PinkRoccade Civility. Remko van der Pols is a managing consultant at Getronics PinkRoccade. Among other things, he is one of the creators of the ASL framework and has written various articles and books about ASL, Business Information Systems Management and information planning.

### Reason

Within ICT it is possible to observe a trend towards standardisation and consolidation: various organisations are either replacing their existing customised products by packages or are collectively deciding to have a system developed that is suitable for all participating organisations. Suppliers are taking responsive action to this: standard solutions that have generally been adapted to specific branches or industries are becoming available to support more and more processes.

As a result of this, application management of packages is becoming increasingly important. Customers are now starting to expect the same quality from suppliers as they would in the case of customised products and they are also starting to make demands in terms of time and money. Furthermore, there is a growing tendency to outsource ICT services, also in the field of applications. ASP services (Application Service Providing) are becoming increasingly common.

ICT services are increasingly being seen as something that can be outsourced and that has to be managed on the basis of demand and supply. Normal requirements are starting to apply with respect to time, money, quality and results. This has an impact on package suppliers. The conclusion is therefore that package suppliers are clearly going to have to optimize their processes.

---

<sup>1</sup> This framework was first published in IT Beheer Jaarboek 2001

<sup>2</sup> See among other things the article by Donatz and Van Outvorst in IT Beheer Jaarboek 2003.

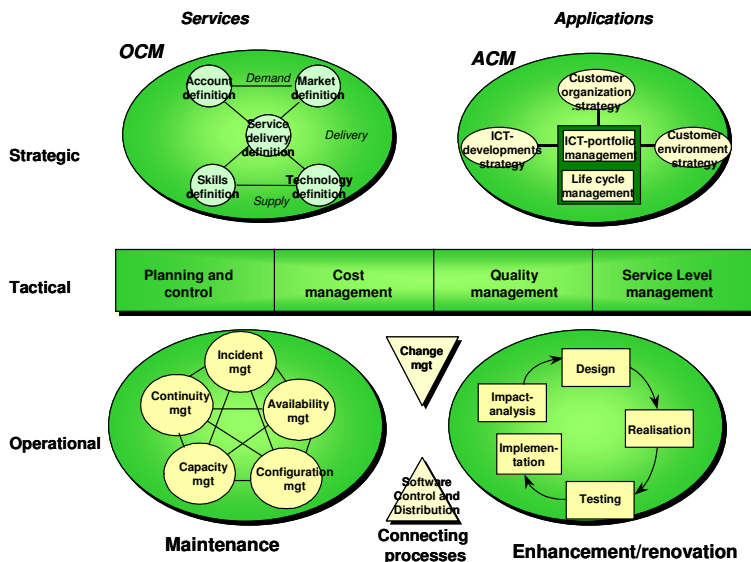


Figure 1. The ASL-model

## The concept of a package

There are many different interpretations of the concepts of a 'package' and a 'package supplier'. This is why it is important to distinguish the key element of these. Below are a number of examples of suppliers that refer to themselves as package suppliers.

1. An organisation supplies a standard program with limited possibilities for making adjustments. They refer to this as a *package*. The functionality is modified and extended for a group of customers; this group of customers determines which modifications are going to be implemented. However, the ownership and user rights continue to be vested in the supplier.
2. An organisation supports the payroll records for several organisations. It processes the changes, takes care of the calculations, prints out the payslips, sees to the financial transactions and the transfers. One standard information system is used for this.
3. An organisation supplies software for financial and logistical records (such as SAP). In order to be able to use this it is necessary to make extensive adjustments. Components are often added to the software to ensure that it links up with the specific situation at the customer.
4. An organisation supplies information systems for home and room rentals. It runs these within its own organisation. Every municipality has its own regulations. Because the regulations differ per municipality, the supplier has specific software for each customer organisation, in addition to software that is used collectively.

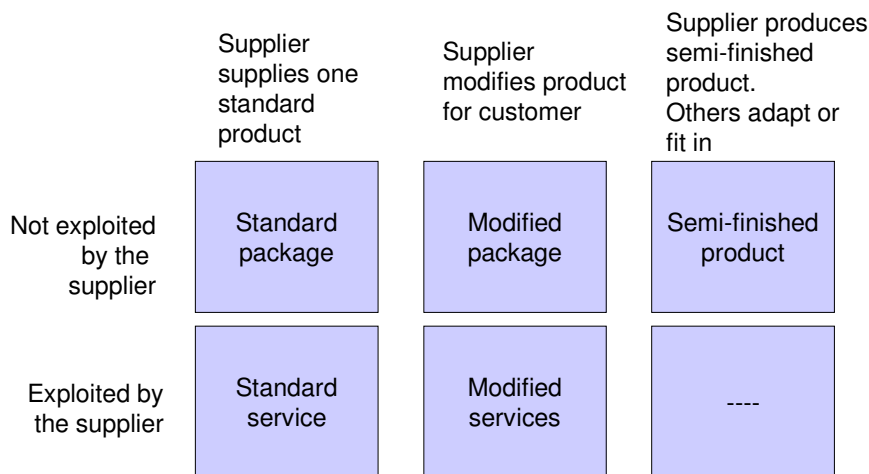
Each of these organisations says that they supply packages or package services. They often have quite a different picture of the other suppliers. The organisation referred to in the first point may say, for instance, that the organisation referred to in the third point does not supply packages, because a great deal still needs to be assembled. While the latter organisation may feel that it is supplying a package because it determines the functionality itself.

These examples show that there can be a lot of differences when it comes to what people mean by a package or standard product. Examples of fundamental differences of this kind include:

- Some organisations see to the exploitation themselves (so-called ASP-service), others do not.

- Some packages are really standard and ready for use. Other packages can be regarded as semi-finished products, which still need various adjustments.
- In some situations there are two parties involved in the management and maintenance (package supplier and the end user); in other situations there are several (the user organisation, the package supplier and the organisation that implements, modifies and manages the package).

For this reason it is a good idea to take another look at the main features of the possible forms. A number of forms have been summarised in Figure 2. These have been worked out in more detail below.



**Figure 2. Forms of packages and package service**

It is possible to distinguish a number of basic differences in the service that makes the structure of the specific 'package organisation' different. These differences have a major impact on the service provided by the supplier.

## Exploitation

One important distinguishing issue in relation to the service is whether or not the package is exploited by the supplier. If the package supplier is running his own package, customers feel that the supplier is responsible for the link between the infrastructure and package and that this should be perfect. In situations like this, the management organisation has to play an active role in the operational aspects and also has to set up processes to safeguard them.

If the operational aspects take place elsewhere, at the customer organisation for instance, it is far more difficult for the package supplier to gain insight into them: after all, the software is running on an infrastructure which is not really visible to and manageable by the package organisation.

## Concreteness of the end product

Other factors that play an important role include the number of suppliers involved in the development and the extent to which it is possible to speak in terms of a concrete end product. We can distinguish three situations here:

- **Situation 1:** There is a standardised set of software. The supplier therefore recognizes one unique collection of software (apart from the various versions/releases). See example 1. This software could possibly offer different functionalities at specific customers by means of splitting it into modules and/or parameterization, but the software nevertheless stays the same.
- **Situation 2:** There is a common basis in terms of software, but in addition there are customer specific programs (customised), such as reporting and calculation modules that have been developed by the package supplier. See example 2. The combination of basic and customised software is unique per customer. The supplier therefore has a separate system for each customer as it were.
- **Situation 3:** From a business process point of view, the software supplied is a semi-finished product. It is possible to speak of a common point of departure, but before the system can be used to support the business process a certain amount of assembly, adjustment and customisation work will need to be done. Although the package supplier supplies a development tool for this, he leaves the customisation work up to the customer or another supplier. In the latter case there are therefore two suppliers for the user organisation (the supplier of the customised software and the package supplier).

Example 1.

A supplier supplies a package that offers functionality to support the business processes of the Population Affairs department of municipalities. This package comes in the form of standard modules, from which the customer can choose. Depending on the situation at the customer, such as the size of the municipality, a certain standard functionality (module) may or may not be purchased.

Example 2.

A supplier supplies a standard package to support the finance department. Components of the package are customised according to the wishes of the customer and maintained by the supplier.

	<b>Situation 1</b>	<b>Situation 2</b>	<b>Situation 3</b>
Number of users	Large number of users	Limited number of customers	Very many users
Functionality	Well-defined functionality	Functionality with strong deviations in various places	Basic functionality
Relative influence customer	Low	High	Low
Communication between customer and supplier	Average	High	Low

**Table 1. Differences between the various situations**

These differences (see Table 1) do provide a basis for structuring the management and the management organisation, but nevertheless do not give a decisive answer about what is considered to be a package and what is not. The borderline between customised work and a package can be even more diffuse. Examples 3 and 4 illustrate this.

Example 3.

An ICT department has been set up at holding level, at an organisation consisting of a holding company and a number of regionally oriented operating companies. This department is developing an information system for the operating companies. This information system has to link up with the processes of the various operating companies, but one is aiming for a generic solution. Only one information system is being built.

The ICT department is in fact acting as a package supplier. However, because the operating companies and the ICT-department belong to the same holding company, the department does not see itself as a package supplier.

#### Example 4.

Out of cost considerations, a package supplier has outsourced the development of the software for a system to an organisation in India. The package supplier acts as the customer and supplies detailed specifications, which the Indian organisation subsequently uses to produce an automated system or introduce changes to such a system.

The organisation in India sees the contract as a customised job: there is one customer who acts as a customer and provides an accurate description of what needs to be done.

The customers of the package supplier on the other hand see him as a package supplier. After all, he is supplying a standard solution.

## Definition of a package supplier

The fact that they come in so many different forms makes it impossible to give a clear or unequivocal definition of the concept of a 'package' on the basis of one criterion. However, there is another possibility for giving a definition, which covers the concept of a package or a package service. It is the following criterion:

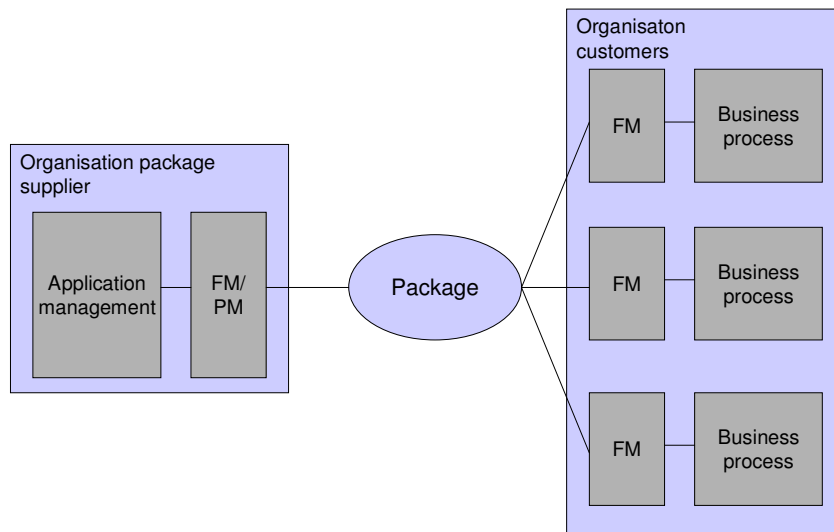
It is possible to speak of a **package** when the supplier has decision-making powers with respect to the functionality and the software, and the supplier is / belongs to an organisation other than the user organisation.

Such a definition appears to be a long way off the idea of a package or standard service. But the main difference between customised products and packages nevertheless appears to be related to which organisation takes the decisions. In the case of a customised situation the user organisation determines (and pays for) the functionality, in the case of a package situation it is the supplier who decides.

If ownership is vested in the supplier, he will try to sell the package several times. A user organisation will not give another organisation decision-making powers, as long as they are or continue to be the only user. This is because if the ownership and the decision-making powers are vested in the supplier, the user organisation will not have any control over the costs. Furthermore there are no other customers to share the costs with and thus keep them down (one of the advantages of a package).

One consequence of this definition, which is considered to be the most important difference between a package and a customised product, is that the supplier's organisation in fact performs the functional management. This means that two types of management are performed at package suppliers: application management and functional management. Package suppliers also refer to functional management as 'product management', 'product consultancy' or 'users helpdesk'.

Whenever we refer to 'package suppliers' in this article, we are referring to all organisations that supply some form of service related to packages such as suppliers of 'starter kits', suppliers of customised products and services related to packages, ASP providers, etc.



**Figure 3. Functional management in two places in the case of packages [Donatz c.s.]**

**Example 5.**

An organisation performs the management and maintenance of payroll systems for the government. This application management organisation sets up the management and maintenance as a customised product. Within the government there is a department, which issues instructions to this application management organisation. The users are educational establishments in the Netherlands.

At one point the government decides that it is not a task of the central government to provide payroll services.

A decision is taken to 'outsource' the customer organisation within the ministry to the application management organisation. The financing is transferred from the ministry to the users. The whole of the application management organisation and the customer organisation very soon start to act as a supplier of payroll services: it has become a package supplier.

It is therefore possible to distinguish two types of functional management in the use of packages: the functional management at the customer/user and the functional management at the supplier (see also [Donatz c.s.]

## **Differences and similarities between packages and customised products**

Now that the concept of 'package' has been defined, it is possible to take a look and see what the generic differences between a package and customised service are for the process clusters of ASL.

### **Maintenance**

In the case of packages (in all possible forms) it is very important to deliver good products. If faults occur, the same report often comes in from various people at the same time. So more incidents occur, as a result of which more time is needed to record and handle these. In addition the possibilities for taking corrective measures are more limited, especially for suppliers who do not see to the exploitation of the packages themselves.

It is also more difficult to solve problems: the customer often does not have the version management of the packages or of the infrastructure completely up to date. It then becomes more difficult for the supplier to establish the exact cause of the problem remotely.

The communication process between supplier and end users does not run quite as smoothly and is less direct. And because more parties (because more end users) are involved, more coordination is required.

#### Example 6

The helpdesk of a package supplier is being swamped with reports: after the migration to a new operating system users are no longer able to gain access to the application.

It turns out that the hardware supplier has told users that the last version is no longer going to be supported and that users should migrate to a later version.

The package supplier failed to inform the user organisation about the impact on its software on time and furthermore did not provide enough information about a plan to migrate the package.

## Development and design

Suppliers of standard systems often have to deal with a large number of deviating demands from various customers. To solve this one often makes the system more generic and flexible. You also tend to see this in customised products, but it is more common in the case of packages.

On the other hand the functionality is often more complex in the case of customised systems. In the case of customised situations one has to deal with customers, who in principle can have all their demands turned into modified functionality. In the case of packages, ownership is vested in the supplier, who will be more focused on how easy it is to maintain the package.

Just like in the case of a customised situation, it is also very desirable to have user participation here. But in this case, the organisational aspects of the communication and coordination are primarily in the hands of product management and in principle are not within the scope of application management.

## Steering processes

The demands that are being placed on package suppliers and suppliers of customised products are getting more and more similar all the time. Unlike in the past, customers are far less willing to accept that a system does not work. This was already the case for customised products. But people are now also far less willing to accept that something does not work in the case of packages and standard services.

Higher demands are also being placed on the completion times of new releases, for instance. This means that in the case of packages, the application management organisation has to pay more attention to steering processes. The way in which this steering takes place is largely the same as it is for the application management of customised products.

- Tactical processes

A lot of package suppliers automatically pay more attention to strategic processes. It is only logical for them to do so: if a system is end-of-life, the supplier has to start making some major investments (in a customised situation, the user organisation generally pays for this). A new package often also means that the contract is going to be reviewed, which means that there is a risk that customers are going to start looking around for other packages.

For package suppliers it is therefore essential to keep an eye on the long-term perspective and to control this themselves. In the case of customised situations, the application management often adopts more of a wait-and-see attitude.

## Structure of the application management organisation in the case of packages

None of the differences outlined in the previous paragraph are fundamental (with the exception of the ownership of the information system). Differences will arise when structuring the processes of the application management. But these differences are very much related to the service that is provided, and to the situation that the supplier is in.

When structuring the management within a package organisation, it is more important to identify the nature of the responsibilities and work involved and therefore what kind of tasks need to be performed. Figure 4 provides help on this.

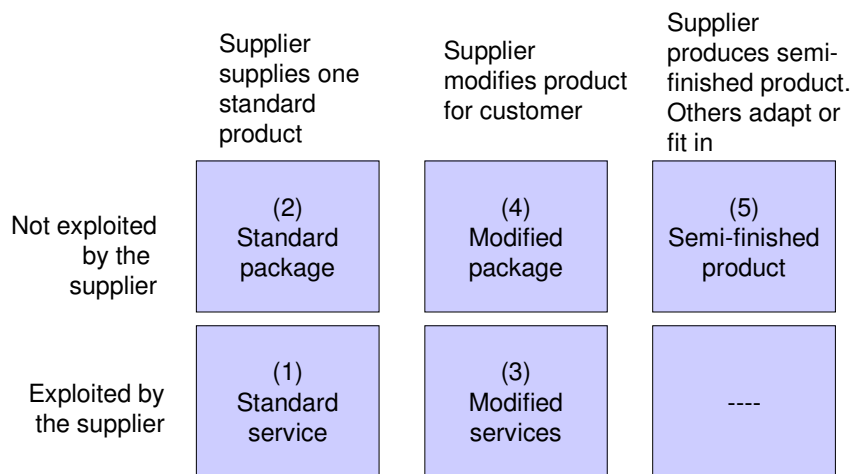


Figure 4. Forms of packages and package service<sup>3</sup>

### Situation 1: standard service

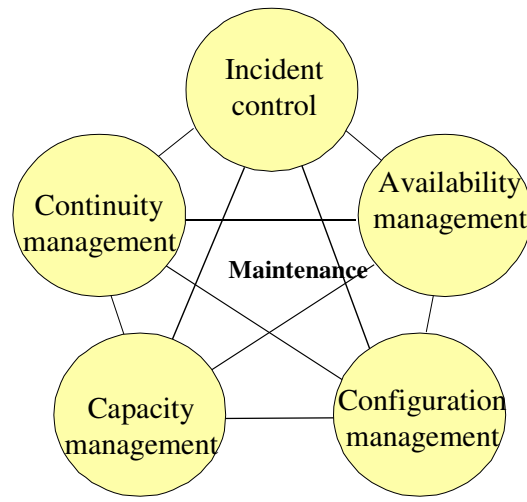
In the case of a 'standard service' it is possible to speak of a fixed set of software for which the supplier in addition to taking care of the management and maintenance also takes care of the exploitation. ASP therefore fits into this category.

In a situation like this, the supplier will structure and implement the processes in a way that is similar to the way that this is done in a customised situation, in which the supplier also takes care of the technical management. Customers will therefore want to reach agreements with the supplier on aspects such as performance, opening hours of the system, and reliability that are analogous to a similar customised situation. The SLAs will be similar to a large extent.

This means that it is possible to apply the ASL-processes in the structure of these without any problems.

<sup>3</sup> We still have not come across a supplier of semi-finished products who is also involved in exploitation





**Figure 5. The management processes within ASL**

## Situation 2: standard package

The situation is different when the supplier does not take care of the technical management himself but when the customer organisation sees to this instead, for instance. In this case, the package organisation is not really in a good position to steer on performance or reliability, for instance. The activities in the management ellipse of ASL (Figure 5) are therefore primarily seen to by the customer. The maintenance activities of the application management are primarily taken care of by the customer.

The supplier organisation will nevertheless also have to spend time on these processes: if the customer organisation is unable to solve the bottlenecks, the supplier will have to do so. Questions are dealt with at the package supplier by a kind of 'functional management' helpdesk, which generally operates under such names as a customer service desk, etc. This helpdesk will nevertheless frequently have to fall back on the application management organisation that either developed the software or is familiar with how the software works and is therefore in a position to track down and solve internal causes of problems.

It is important for the package organisation to be aware of the fact that it needs to reserve time and capacity at the application management organisation to deal with bottlenecks of this kind.

This form of management is far more reactive than it is in the case of customised products: in the case of capacity management, for instance, the package supplier is not in a position to monitor whether the tables are slowly starting to fill up. The customer will have to do this himself.

To ensure that this does not happen too often, package suppliers take measures in a variety of fields:

- Minimum infrastructure configuration requirements are agreed beforehand. This means that minimum configuration requirements are identified and that no support is provided if the environment does not meet these requirements.
- The various releases and systems are tested internally on various platforms and in various different environments.
- Recommendations are issued on the way in which the package should be used: what it can and cannot be used for.

- Monitoring and indication tools are supplied, so that the customer organisation is able to exercise some control itself and the customer organisation's helpdesk is able to access basic information in the event of a breakdown.
- Customers are kept informed at the various levels of the organisation about the impact of changing environmental factors, new developments, release planning, market signals, etc.
- Proactive communication takes place about problems that have been observed at other customers.

Incidentally, it is important for the package supplier to properly set up the configuration management. In a customised situation this is a minor process – after all, one only has to deal with one customer and one production release of the system – for a package supplier who is not seeing to the exploitation of his package himself, it is of utmost importance to know what the configuration at his customers looks like.

In an ideal situation this should be taken one step further, by linking the 'local' management processes to the package supplier's management processes. Very few of these links currently exist, but the first steps are being taken in this field.

## ASL - operational

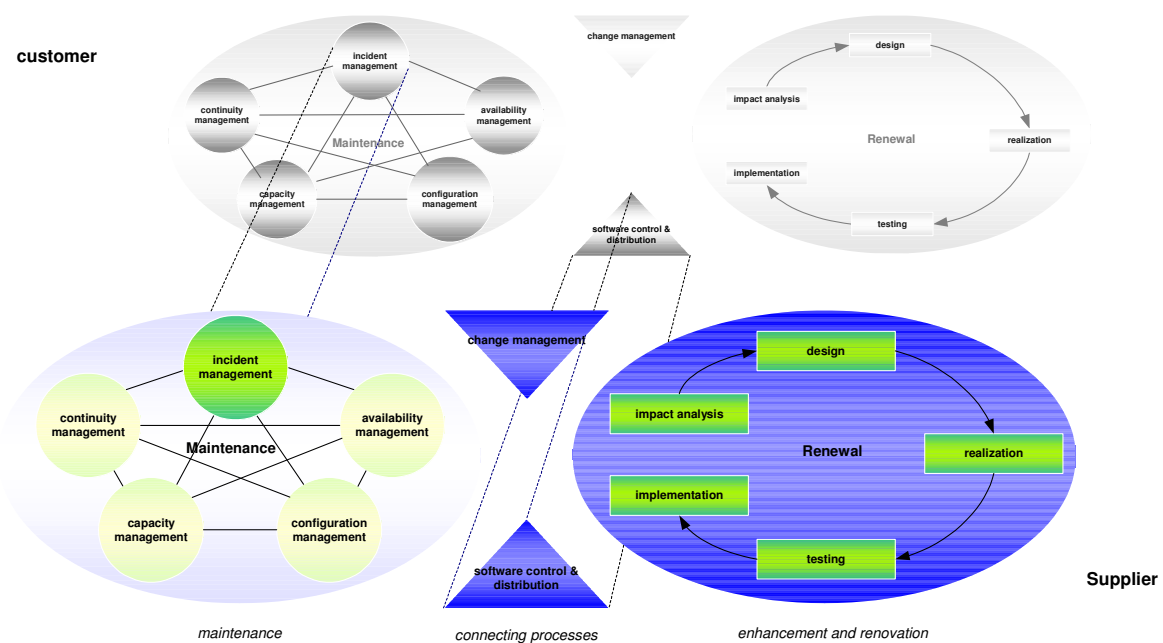


Figure 6. The link between the management processes at the customer and those at the supplier

### Situation 3 and 4: modified package/modified service

This situation is similar to the previous one in many respects. You have a supplier who supplies a system, which means that there is no ambiguity when it comes to who has knowledge about the functionality and software: the supplier can be tackled on his expertise in relation to the product/service provided.

One complication in comparison with the previous situation is that there are far more sources. Because the supplier has built functionalities per customer, one as it were gets a system per customer. This complicates the connecting processes within ASL to a great ex-

tent. Software control and distribution has to take account of various releases, but of also various sources per customer within a release. This therefore places very high demands on the software management.

Change management and impact analysis also become rather complex processes as a result: a change has to be held up against the various situations that exist at different customers. For every change one therefore not only has to consider whether it is suitable for customer a, but also how what its implications are for the sources supplied to customer b and c.

This also has a major impact on the test process. Ideally speaking the test process should be completed in full for each customer. In the above-mentioned situations it was possible to suffice with testing the uniform package on the various platforms, now the 'unique' version for each customer comes on top of that.

This is a situation that a supplier would prefer to avoid. The possibilities for doing something about this are often fairly limited: this situation generally arises when a supplier has a limited number of customers and when these customers have a relatively big influence on the functionality.

### **Situation 5. Semi finished products.**

There are two ways to go about solving the problems referred to in the above-mentioned example:

- keep the functionality of the package uniform by limiting the functionality or by incorporating generic (adjustable) functionalities. The result of this is a uniform set of software (situation 1 or situation 2).
- increase the possibilities, by no longer providing all round solutions but only the basic elements (those processes that are virtually always the same at every customer). The basic elements are assembled for a specific customer on the basis of the requirements of that customer and adapted to the customer's own situation with the aid of a development tool that is supplied. You then supply a starter kit as it were.

In the latter case we are dealing with semi finished products. In situations like this it is not uncommon for a third party to see to the assembly and modifications. This in turn has an impact on the service.

For suppliers of semi-finished products it is virtually impossible to exercise any control over the use of the package in the end situation. The package may be modified, functionalities may be added to it, its internal workings may be changed. It is therefore more common for suppliers to supply packages like this 'as-is' and not take any responsibility for their use.

It is crucial in this case therefore, more so than in situation 1, to supply a good product. This means that it is even more important to properly design, build and test these semi-finished products. It also helps to design the systems in a very modular way. Because the connection with the customer is fairly weak, it is essential to use upgrade-kits to implement automated changes for the benefit of subsequent releases. This is because in this case the upgrade process is even more difficult than it is in the other situations.

#### **Example 7**

An SAP solution has been implemented for the finance department at a municipal council. The municipal council in question has enough in-house knowledge to make further adjustments to the package. The outcome is a specific solution, which is no longer controlled by the supplier. The supplier does not have any insight into the developments at the municipal council and will only take responsibility for the starter kit he supplied.

This means that ultimate responsibility for performance is not vested in the supplier but in the municipal council's ICT department. So if there are any problems it is up to this department to prove that the supplier is responsible for them.

## Conclusions and analysis

In short, it is possible to conclude that there are differences in application management for customised systems and for packages, but that these differences are not fundamental ones. The most important differences between the two situations are related to ownership and are largely evident in the domain of functional management and/or product management. This is where the big question as to what the application has to be able to do comes in and how much it can cost. It is also where the power struggle between users/customers and the party responsible takes place. It is also where the business case lies.

It is possible to conclude that there are differences in the various forms of service. These differences are:

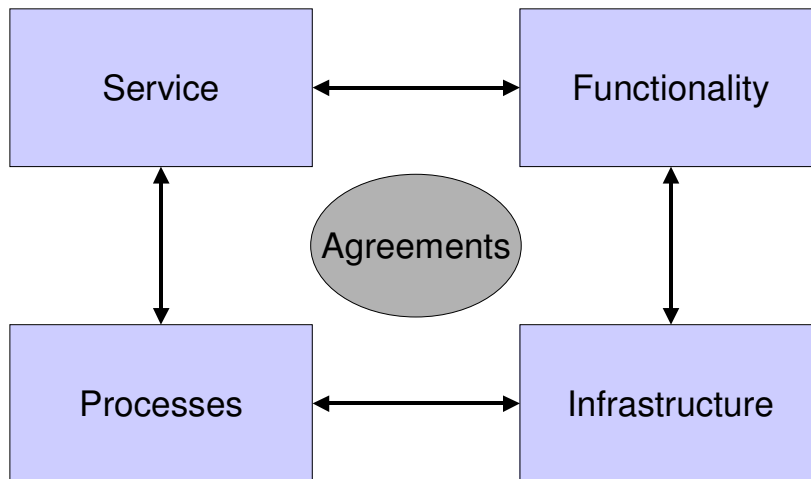
- It is not always possible for the supplier to properly set up the management within the application management. This is the case when the supplier does not perform the technical management or is unable to exercise any control over it.
- The change management, software control and distribution and the impact analyses are generally more difficult, because more releases need to be supported. Unlike in the case of customised work, there are generally fewer possibilities for the supplier to control this. This is not a fundamental difference, but requires more time and attention in the case of suppliers of packages and standard services.
- The maintenance and innovation processes also require extra attention: the impact analyses, the design and the realisation have to take account of the fact that there are several releases and also of the fact that it is often necessary to support several platforms. So more attention is required, to prevent a situation in which inadequate products are supplied. The costs involved in dealing with this and putting this right are considerably higher than they are in the case of customised situations.
- Life cycle management is essential too. When an existing package is replaced by a new package this generally also involves a new contract and therefore also invitations to tender. The supplier then in fact opens up the door to his competitors.

This means that it is also possible to conclude that as far as the service is concerned and the demands that are placed on this, the differences between package and customised situations are getting smaller all the time.

## Tips and tricks

We can give a number of recommendations, tips and tricks on the basis of these conclusions and the different situations.

Figure 7 contains the most important areas of special attention that the supplier and his customers should come to an agreement on.



**Figure 7. The decision issues in the case of packages**

## Suppliers

### Type of service

The most important conclusion that can be drawn is that suppliers have to make a clear choice as to what type of package/package service they supply. The five main categories are shown in this article.

Each situation not only leads to the various processes being structured differently but also to differences in the details contained in SLAs for instance. We often see that this has not been thought through properly beforehand: one then agrees to SLAs, which one cannot make good, or agrees to SLAs that could have been a lot stronger.

### Structure of the processes

The need to start providing more professional services and products was mentioned in the introduction. Process structuring is unavoidable in this. Before making a start on the structuring it is however essential to determine what kind of organisation you are and what kind of service you provide. This was discussed in more detail in the previous point.

The management organisation has to be structured on the basis of this strategic position. The most important aspects of this were indicated in paragraph 5. In addition it appears that the principles and the practices of ASL for instance are a very useful point of departure. On the one hand it is important to fill in the details of these processes because of customer satisfaction, and on the other because these activities have to be carried out anyway. Proper feedback to application management also leads to better products.

There are clear advantages to getting customers involved in the testing, the process of connecting management processes at the customers up to management processes of the application organisation, and the life cycle management.

### Functionality

The functionality of a package or service is an important item of course. Package suppliers run a big risk of going 'too far' when it comes to meeting individual user requirements. Sup-

pliers with few customers especially run this risk. The big question here is whether one still designs functionality that fits into the bigger whole, or whether one is fitting customised products into the package. In that case the supplier ends up in situation 3 or 4 ('modified package', 'modified service').

It is possible to avoid this by incorporating parametrisering and adjustment options in to the package. This increases the size of the tests (one has to test all possible combinations), but gets around the extremely complex maintenance required in the case of diverse versions for various customers and it promotes the growth possibilities of the package. If there is no alternative ensure that the specific customised work is clearly separated from the package. A clear insight and a clear description with respect to the goal and the use of the package is an essential condition for this.

### **Infrastructure**

The selected infrastructure plays an important role in the costs of the application management. A too restricted choice of the types of infrastructure restricts the market, too much of a choice forces up the costs of application management. It not only means that all these types of infrastructure have to be 'managed', but also that applications need to be developed and tested for them.

To manage the infrastructure, it will be necessary to have knowledge of all the possible infrastructures that are used by the users of the package in practice. If the supplier is not taking care of the exploitation himself, it will be difficult to keep this up to scratch.

It is essential to draw up clear limiting conditions with respect to the infrastructure and to communicate possibilities and impossibilities of the package in its environment in order to create the right limiting conditions for the exploitation of the package.

## **Customers**

### **Types of service and requirements**

Purchasers of packages or standard services should also ask themselves the question as to what exactly it is they want. Customers have a tendency to want to combine the advantages of various variants. They for example want the flexibility of 'situation 3' at the price of 'situation 1'. In practice this generally works out in the short term but never works out in the long term.

### **Maintenance structure**

As part of the selection process a customer may ask the package supplier to explain or demonstrate how he has set up the maintenance structure (application management and functional management).

The way in which he has set up the maintenance structure, also affects the processes and the quality of these at the customer.

It is not just the supplier who has to ensure that processes are well set up. The customer (if he is taking care of the exploitation himself) also has to properly set up his management processes. Calling out the supplier to discover a fault in the infrastructure, may be part of the contract (however often it is not), but if the system does not work it is going to cost the company money.

It is therefore important to arrange good infrastructure support and to also give the supplier insight into this environment. It generally pays to make agreements with the supplier explicit and link up the processes between the customer and supplier as much as possible.

### **Functionality**

The basic choice in favour of a particular package is that the operating process is adapted to the information system. This means that a customer also 'buys' shortcomings: the information system will not meet all of the wishes and requirements of the operating process or the

organisation. It is nevertheless ill advised to insist that a package should meet all your wishes and requirements as this would make it too complex and would ultimately to high costs.

It pays off to ask for generic or broadly applicable functionalities as much as possible. This is why it is advantageous to work together with other customers in the form of a *user group*. This not only ensures that there is a unequivocal and powerful body to see to the communication with the supplier, but can furthermore promote the exchange of knowledge and experiences relating to processes and the support of the package as well as uniformity of use.

### **Infrastructure**

Suppliers tend to limit the number of choices when it comes to infrastructure. It is in the interest of suppliers to do so as it keeps down the costs for the supplier and ultimately also the costs for the customer.

It is essential to follow the recommended configuration: this is something that is now generally being adhered to and which is also in the customer's interest. Even though it might occasionally be cheaper in the short term not to follow this configuration, the chances of errors and problems occurring are greater as a result. This not only causes problems for the operating process but also results in additional exploitation costs.

## References

Donatz, Outvorst, "Functioneel beheer bij pakketten". In *IT Beheer Jaarboek 2003*, Ten Hagen en Stam, ISBN 90-440-0576-6

Van der Pols, Remko en Meijer-Veldman, Machteld E.E., "ASL, the next generation application management". In *IT Beheer Jaarboek 2001*, ten Hagen & Stam, Den Haag 2001.

Deurloo, C. D., Meijer-Veldman, M.E.E., en Van der Pols, R., "Model voor Functioneel Beheer". In *IT Beheer Jaarboek 1998*, ten Hagen & Stam, Den Haag 1998.

Backer, van der Pols, *Managementguide ASL*. Van Haren Publishing, 2003, ISBN 90-77212-04-3 ISBN 90-77212-03-5

Van der Pols, *ASL, a framework for application management*. Van Haren Publishing, 2004, ISBN 90-77212-051