

# DSDM and Package Selection and Deployment

## Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>2</b>
1.1	AUDIENCE .....	2
1.2	CONTRIBUTORS .....	2
1.3	SECTION STRUCTURE .....	3
<b>2</b>	<b>THE CASE FOR USING DSDM FOR PACKAGE SELECTION AND DEPLOYMENT .....</b>	<b>4</b>
<b>3</b>	<b>THE DSDM PRINCIPLES .....</b>	<b>6</b>
<b>4</b>	<b>SPECIFIC RISKS WHEN SELECTING/DEPLOYING PACKAGES USING DSDM .....</b>	<b>10</b>
4.1	CRITICAL SUCCESS FACTORS FOR A DSDM DEVELOPMENT .....	10
4.2	OTHER RISKS/ISSUES.....	11
4.2.1	<i>Training .....</i>	<i>11</i>
4.2.2	<i>Roles .....</i>	<i>12</i>
4.2.3	<i>Contractual Issues .....</i>	<i>13</i>
4.2.4	<i>Non-Functional Requirements.....</i>	<i>13</i>
4.2.5	<i>The Suitability Risk List, additional risks and countermeasures .....</i>	<i>13</i>
<b>5</b>	<b>LIFECYCLES AND PATHWAYS.....</b>	<b>15</b>
5.1	PACKAGE DSDM LIFECYCLES .....	16
5.1.1	<i>Package Selection – Requirements well-understood/not excessively volatile .....</i>	<i>16</i>
5.1.2	<i>Package Selection – Requirements vague and changing.....</i>	<i>17</i>
5.1.3	<i>Package Implementation .....</i>	<i>17</i>
5.2	HOW DOES THE PACKAGE-DSDM LIFECYCLE DIFFER? .....	18
5.2.1	<i>Feasibility Study .....</i>	<i>19</i>
5.2.2	<i>Business Study .....</i>	<i>21</i>
5.2.3	<i>Functional Model Iteration (Package Selection - requirements vague and changing).....</i>	<i>24</i>
5.2.4	<i>FMI (Package Implementation).....</i>	<i>25</i>
5.2.5	<i>Design and Build Iteration Phase.....</i>	<i>27</i>
5.2.6	<i>Implementation .....</i>	<i>28</i>
<b>6</b>	<b>CONCLUSIONS .....</b>	<b>31</b>
	<b>APPENDIX A - GLOSSARY.....</b>	<b>32</b>

## **1 Introduction**

This White Paper has been produced in response to a considerable level of interest in the use of DSDM for systems where there is a significant element of third party packaged software. This interest relates to using DSDM for both the selection of such software and its implementation (deployment).

DSDM is often identified as a rapid and effective way of constructing a bespoke solution to a business problem. However, the DSDM framework and philosophy, the involvement of users, and approaches such as timeboxing, workshops and prototyping, can be valuable for both package selection and package deployment.

This White Paper gives guidance on how DSDM concepts can be used to support the effective use of package solutions, and considers using DSDM for:

- deciding whether appropriate packages exist
- selecting a package
- implementing a package with little or no bespoke work
- implementing a package, in parallel with considerable bespoke work.

The White Paper specifically excludes the consideration of large-scale package implementations such as SAP or other Enterprise Resource Planning (ERP) packages. The strategic and organisational change issues associated with these make them a special case, worthy of separate consideration elsewhere.

The focus of this paper is on applications software packages. These are interpreted as occupying a higher level in terms of business coverage than components (for which a separate White Paper exists). This paper will have relevance to the deployment of tailorable infrastructure packages but does not cover the strategic selection and deployment of these. It is also relevant to the selection, tailoring and delivery of applications packages in relation to a package integration strategy, in line with an agreed corporate data architecture. It does not cover the selection and deployment of operating systems, middleware and communications software. Nor does it specifically consider the deployment of packages that extend outside the traditional boundary of the enterprise.

The White Paper also excludes specific details of the contractual processes, which are the subject of other DSDM White Papers.

### ***1.1 Audience***

The readership of this White Paper should be familiar with the use of DSDM projects involving bespoke software, and need to gain the benefits of a DSDM approach for the selection and/or implementation of application software packages. It is intended to be useful to any member of a DSDM team, plus the management and sponsorship roles within the project

### ***1.2 Contributors***

Original Contributors

Phil Brown	Parity
Alastair Campbell	Jim Beam
Tony Jenkins	Parity (Task Group Co-Chair)
Jos Geurtsen	Deloitte and Touche Consultdata
Howard Jenner	Ives & Co
Dorothy Tudor	TCC (Task Group Co-Chair)
Paul Turner	Parity

This White Paper has been updated to reflect DSDM V4.2 by Dorothy Tudor in March 2003.

### ***1.3 Section Structure***

Section 2 gives the case for using DSDM in a package selection and deployment project is considered.

Section 3 considers the impact on the nine underlying principles of DSDM.

Section 4 considers the risks to be addressed, including the DSDM Critical Success Factors. Recommendations are given in relation to key issues/risks.

Section 5 covers the alterations and amendments to the DSDM lifecycle, products and roles that should be made when selecting and implementing packaged software.

Throughout the White paper, the perspective is principally that of a customer organisation seeking to procure and deploy a software package, rather than from the supplier's perspective.

## **2 The Case for using DSDM for Package Selection and Deployment**

DSDM is particularly suited to package selection/deployment because:

- Requirements are baselined at a high level early in the project. This gives a good framework for package selection, without being so detailed that no package will fit.
- Packaged software is unlikely to give better than a 70% to 80% fit to the detailed business requirements and a degree of compromise is usually needed. The 80/20 approach of DSDM, and the early MoSCoW prioritisation means that the focus is on the most important requirements from a business point of view.
- The requirements are elicited from a team with the right business representation and empowerment. This team is well placed to select (and later deploy) the best-fit package for the business.
- Facilitated workshops can be used effectively to: establish and prioritise requirements, shortlist and select a package; map original requirements to package-provided requirements; define and prioritise bespoke work.
- The use of timeboxing helps to avoid over-detailed initial requirements definition; to constrain the selection process and to control costs during package implementation.
- Prototyping may be used to establish initial requirements, if these are vague or not easily definable.
- If both customer and supplier know and use the DSDM approach already, this can be a powerful influence on successful communication between staff of both parties (a common language).

Difficulties may arise when using a DSDM approach, particularly in the following areas:

- working in mixed teams of supplier and customer staff, both during selection and whilst implementing the packaged software brings with it a mix of organisational cultures, organisational priorities and often geographic locations, which can make team working difficult;
- inflexibility of the package when prioritising requirements means that the requirements cannot be prioritised purely on business need. Care is needed that compromises made are tolerable;
- there may be a lack of software to match the initially prioritised requirements and the changes required might be fundamental organisational changes that require a focus wider than the individual project. However, even in this case, DSDM can be useful in prototyping business and organisational change as part of a higher-level infrastructure project – see the Business Process Prototyping White Paper;.
- lack of DSDM knowledge within the supplier or customer company: the learning curve of a new software selection or deployment approach may impede the process of selection and particularly of deployment. Specifically, if a supplier has its own “tried and trusted” method of implementing its software, but the customer insists on a DSDM approach (or any approach different from the supplier’s usual method) the

supplier may become confused between their usual approach and a new one which the customer is trying to impose, or may even abdicate responsibility for success of the project.

### 3 The DSDM Principles

In this section, each of the DSDM Principles is considered in turn and some of the major impacts of a package selection and deployment evaluated. The term “supplier” is used to distinguish the source of the third-party software, and “customer” to denote the company into which the deployment is to be made.

	<b>DSDM Principle</b>	<b>Considerations for both Selection and Deployment</b>	<b>Selection</b>	<b>Deployment</b>
1	Active user involvement is imperative	The customer’s Ambassador and Advisor Users need to be involved in both package selection and implementation. Continuity of personnel between selection and deployment is desirable (although not always possible).	Users must be prepared to identify requirements at a high enough level to allow flexibility, but must ensure that the true “must haves” are clearly understood and that packages not fulfilling these are removed from consideration.	Users need to be involved as Ambassador and Advisor Users. Both customer and supplier staff may occupy the role of developer.
2	DSDM teams must be empowered to make decisions	The DSDM team from the customer side must have both business and technical roles filled, for both selection and deployment.	The selection of a package inevitably requires compromise. Some requirements will not be met by a standard package. There must be clarity of business objective/intent from the Feasibility Study, e.g. whether to modify the package to obtain the best fit, or to change business processes/procedures to accommodate the package.	Levels of flexibility and empowerment must be clearly stated at the outset. In particular, there will need to be a degree of empowerment of supplier staff as well as customer staff. This needs to be clearly defined, understood and controlled.
3	The focus is on frequent delivery of products	In the selection stage, products can only be defined at a high, and somewhat generic, level. Attention needs to focus back onto the specific products, and these need to be defined more clearly within the plan, once the selection has been made.	The selection of one module of the package may begin an inescapable path to selection of the other modules. Thus, the selection process must consider the high level functionality of the rest of the package suite.  A package may be a monolithic product for which separation into individual increments is	If the package is monolithic, the scope for frequent delivery of small parts may be limited. The risk must be acknowledged and the suitability of DSDM considered carefully. If the package is modular, frequent delivery of increments may not present problems.

			not practical. If this is the case, the added risk of not being able to evolve the product and to learn must be taken into account. Alternatively, the package may be flexible, modular and parameterisable, in which case the DSDM approach is likely to be a better fit.	
4	Fitness for business purpose is the essential criterion for acceptance of deliverables	The first deliverable which becomes the focus of this principle is the selection of the package itself. Future assessment of “fitness for business purpose” must rely upon this selection.,	A package is likely only to address 70% - 80% of the business requirement, unless it is extremely flexible or the business requirement is (commercially) totally standard. It is essential that the skills in the selection team are sufficient to assess the fitness for business purpose (Note: There is not the control over which 70-80% is prioritised in the way that there would be with a custom-build, other than by selection of the best fit package.)	The deployment team need to be aware of the flexibility and constraints of the package, and have guidance on the extent to which the organisation wishes to customise the standard package, or accept the package and change the business. There are additional risks and costs in terms of ongoing maintenance and upgrade if the package is customised significantly to meet the requirements of an organisation.
5	Iterative and incremental development is necessary to converge on an accurate business solution	Since the development has already been done, this principle may only relate to iterative and incremental customisation and delivery. Even this may be difficult if the package is monolithic in nature. The concept of this principle should be built into the selection process.	Unless the package is extremely flexible and parameters can be set to customise its functionality, this principle may be compromised by the package selected.	Incremental deployment of parts of the package should be considered, assessing fitness for purpose and reassessing regularly the business case. Flexibility for subsequent change of the package, and the impact of not being able to incrementally adjust the functionality should be assessed. In the case of a small package, with a narrowly confined business impact, there will be limited risk. For an enterprise-wide suite of packages, impact may be great, and should be fully assessed.
6	All changes during development are reversible		All “changes” to the expected functionality are still “reversible” whilst several packages are being considered. However, once a selection has taken place, reversal past this point is	Since there may be little or no development accompanying the deployment of a package, this may not be an issue. However, the version control aspect of this principle may become significant if a supplier who holds many different versions, for other customers, is performing much

			usually impossible, or at least prohibitively costly. Therefore the passing of the point where the selection is made (and details of the decisions made up to and including that point) should be fully agreed at Executive Sponsor level and formally recorded.	customisation. The supplier's version control/configuration management procedures should be carefully audited if this is likely to be an issue.
7	Requirements are baselined at a high level		Requirements must still be baselined at a high level for package selection. The high level focus on <i>what</i> is required rather than <i>how</i> it will be implemented is the correct approach for a package selection. However, it is essential that the MoSCoW prioritisation very clearly identifies the mandatory features. The difficulties that may arise with non-functional requirements are further explored in the Risks section of this paper.	The baselined set of high-level requirements will have formed part of the original contract. However, if the package is known to be unable to deliver any of these, there may be little flexibility to provide them. Thus, the selection process must be a careful one, and considerable ingenuity may be needed at deployment. The flexibility about how a requirement is achieved may, again, be very limited. This presents a risk that the timebox constraints will not be achievable. The implications of this in business terms must be fully explored and the risk assessed.
8	Testing is integrated throughout the life cycle	The focus of testing is different for a package than for a purely bespoke development. Part of the selection process involves "testing" that the package is fit for purpose by visiting users of the software and trialling. Later testing can then focus on interfaces and areas of change.	The whole selection process is concerned with testing: testing whether any of the packages under consideration can meet the business requirement, what the business requirement is and how flexibly this can be interpreted. There is no reason why testing should not be integrated throughout the lifecycle. The early selection stages are tests of the "fitness" of the software for the business purpose.	Software built and implemented during deployment stages should be tested in the same way as any software implemented, with the focus on fitness for business purpose (validation). However, if much customisation or bespoke work is done, those areas that have been changed must be verified and validated. Whilst site visits to other users of the software should confirm the correctness and stability of the code in the body of the package, interfaces and parameterised areas should be tested fully, with the focus on DSDM testing principles (benefit-directed testing, etc.)
9	A collaborative and co-operative approach between all stakeholders is	This can be a very hard principle to control, since the number of	During the selection process, it is inevitable that suppliers and customers will have their	Once the package has been selected, work often begins on deployment whilst the contractual issues are still being finalised - the business deadline



	essential	<p>stakeholders may be increased (the software supplier, and often separate hardware and network suppliers).</p> <p>Care should be taken to ensure communication, buy-in and control of this principle throughout the whole of the selection and deployment process.</p>	<p>own agendas. The supplier wants to sell and often the people doing the selling are on commission. They are not usually the people who have to deploy the package once contracts are signed. This can lead to an over-optimistic presentation of the package's features and its fit to the business requirements. The buyer must beware!</p> <p>The contract must be explicit about the collaborative working approach to be expected.</p>	<p>may require that deployment is not delayed. However, whilst the supplier is still in a commercially “vulnerable” position (i.e. the customer has not yet purchased, and may pull out of the deal), this may affect both openness and commitment. A clear “letter of intent” is one way of improving this situation (see separate white paper on contractual issues).</p> <p>When customisation of the package is needed, a mixed team of supplier and customer staff often work together. Issues of loyalty and different company priorities emerge. The Executive Sponsor (customer management) has no direct control over the resourcing of supplier staff (see later recommendations for duplicated roles).</p> <p>Once deployment is underway and contracts are signed, problems may still arise in that the supplier has other customers, perhaps vying for the same development and deployment resources. Whilst this project may be top priority to the customer’s business, it may be secondary to the needs of other customers from the supplier’s point of view. This may mean the loss of promised resources from the supplier, which may in turn lead to delay in deployment. Delay in deployment rather than dropping of functionality may be inevitable as there may be little flexibility in what can be omitted from the package.</p> <p>The best defence against these risks is to identify them and make them visible to all stakeholders as early as possible in the project. Contractual commitments can then be gained to mitigate the risks.</p>
--	-----------	--	--	---

## 4 Specific Risks when Selecting/Deploying Packages using DSDM

### 4.1 Critical success factors for a DSDM development

DSDM Critical Success Factor	Package Selection	Package Deployment
Acceptance of the DSDM philosophy before starting work.	The decision to use DSDM for the selection process does not necessarily mean that the deployment will be a DSDM project. The Suitability Risk List should be applied to selection and deployment separately.	The decision to use DSDM for deployment of the package may be a prerequisite determined early in the project and made a condition of the ITT that suppliers will work in this way.
Decision-making powers of users and developers in the development team.	The selection team is initially wholly “customer” staff. This team needs to have appropriate business and technical staff, embodying both managerial and operational skills, to enable an informed selection to be made.	The team for deployment of the selected package may be entirely different from that for selection, although some continuity is recommended from both supplier and customer.  The level of empowerment of developer team members must be limited to a level comfortable to customer management, and control mechanisms must be in place to monitor the operation of this empowerment.
Commitment of senior user management to provide significant end-user involvement.	The end-user involvement is needed both during selection and deployment.	The end-user involvement is needed both during selection and deployment. The developers in this team are usually from the supplier organisation.
The organisation should be amenable to the delivery of systems in an incremental fashion.	The requirement for incremental delivery of the package software must be specified in the ITT. This may exclude some suppliers/packages.	Suitable increments for deployment will usually be limited by the package structure. This may mean the Timeboxing of MoSCoW priorities may be less easy (to include some must, should and could have requirements in each timebox, for example.)
Easy access by developers to end-users.	At least two sites (customer and developer) are inevitable. The choice of supplier (and the decision who to invite to tender) must include consideration of the ease of access of customer and supplier team members.	Arrangements may need to be made to locate supplier staff on the customer’s premises for a lengthy period of time.  Additionally, “virtual” means of communication may need to be considered (virtual project room on the internet, teleconferencing, etc.)
The stability of the team	The team for package selection is usually small compared with the team or teams required for deployment. Continuity between the selection and deployment must be	The allocation of the supplier members of the team is not directly in the customer’s control. Some stipulation about supplier resource allocation should be part of the contractual

	considered during the selection process.	agreement.
Development team skills	The developer skills needed during package selection are wide, as many software and hardware configurations may need to be analysed and evaluated.	The deployment “developers” may be mostly implementers who do not write any code. An alternative scenario would be that several teams are required, including supplier developers, supplier implementers and customer developers (perhaps to write interfaces between the package and in-house systems)
The size of the development team		As above, there may be several teams, implementing and developing. The role of the Technical Co-ordinator becomes very important. This may be a shared role, with a customer-side and a supplier-side Technical Co-ordinator.
A supportive commercial relationship	The relationship between customer and supplier staff will be one of the key factors in the success of deployment. During selection, the ITT must make the expected close working relationship a selection criterion. Site visits to suppliers’ previous customers can assist in the assessment of the relationship achieved in prior deployments.	It is essential that a supportive approach be maintained throughout deployment. This must be monitored.  The relationship must accommodate the evolution of the system’s requirements without imposing onerous change management overheads or invoking contractual problems.
The development technology should be suitable for the DSDM approach.	The requirement for this should form part of the ITT	

## 4.2 Other Risks/Issues

There are certain other project issues that prove problematic or deserve particular attention. Key issues include:

- Training;
- Project roles;
- Formation of teams;
- Contractual issues;
- Non-functional requirements.

### 4.2.1 Training

Training in DSDM may be needed for both supplier and customer staff and across many different departments. If DSDM is to be used effectively, it needs to be understood and embraced by:

- the business sponsor and senior user management;
- the project/programme management;
- the team that gathers requirements, creates the ITT and makes the selection;
- the procurement staff and legal staff who create and negotiate the contract;
- the supplier staff who respond to the ITT;
- the supplier staff who deploy the package, or perform bespoke work;
- the testers, including those who perform the contractual acceptance testing.

These different factions must appreciate the impact that DSDM has on their processes. For example, those who define contractual terms must do so in a way that does not conflict with the DSDM deployment approach. Those who implement the software must understand the implications of the timeboxed approach, whilst working within the need for “fitness for business purpose”.

Advice on the appropriate way to address contractual issues is given in other DSDM White Papers.

If the impetus for the use of DSDM has come from the customer organisation, it must be remembered that the supplier staff will potentially need DSDM training, in order to accept the DSDM philosophy and to have the correct approach to such concepts as timeboxing, prioritisation and facilitated workshops. It may not be possible, or even desirable, for the customer to pay to train the supplier staff. The imposition of a qualification in the ITT that suppliers must have DSDM expertise should be considered.

If the package supplier wishes to use DSDM on a customer site where the customer staff have no knowledge of DSDM, the supplier must consider whether:

- the customer has DSDM expertise already;
- the customer staff should (or practically could) be trained – possibly at the supplier’s expense;
- there is sufficient autonomy for the supplier in the deployment of the software for the project risk to be low if the customer staff do not fully understand DSDM.

Consideration should also be given to awareness education (rather than specific training) being given by roadshows, presentations etc. to a group wider than just the DSDM core team.

#### 4.2.2 Roles

The specific individuals who occupy the DSDM roles during selection are likely to be different from those with responsibility for deployment. On the supplier side, it is rare that the individuals who sell the product return to implement it. Continuity of key customer personnel should be sought and a mechanism must be in place for the transfer of knowledge from the selection phases to the deployment phases on both supplier and customer aspects of the team.

There may, in some projects, be involvement of staff from several package suppliers during deployment. Here the “collaborative and co-operative approach” becomes a higher than usual risk. To mitigate this risk, it may be necessary to mirror DSDM roles for the supplier and customer, from Project Manager downwards. It is also advisable to have a steering committee for the project, with the Executive Sponsor (customer organisation) as chair, and “Supplier Sponsors”, i.e. appropriate senior management from the supplier organisation(s), with the power to resolve resourcing, policy, prioritisation, contractual and business issues within the supplier organisation.

#### 4.2.3 Contractual Issues

In some organisations, package selection and contract negotiation is traditionally the province of a completely separate Procurement Department. Problems can result when the teams for contractual negotiation and deployment are different. A contract may result that constrains package deployment and customisation by imposing too much rigidity of requirement (too many “must-haves”). The DSDM selection team should include a procurement co-ordinator, in addition to other business and technical roles.

#### 4.2.4 Non-Functional Requirements

During package selection, essential non-functional requirements must be part of the selection process. However, it is usually impossible to actually test these in the customer’s own environment, with appropriate transaction loading, until deployment takes place. The best safeguard is to attempt to obtain benchmarks for the performance of the package during the selection process, and also to arrange to see the package in use by another customer of the supplier in circumstances as similar as possible to their own. The importance of these non-functional requirements as part of the minimum usable subset of requirements should be a clear part of the selection process and the contract.

#### 4.2.5 The Suitability Risk List, additional risks and countermeasures

The DSDM Suitability Risk List identifies the key risks that apply to any DSDM project. These relate to non-adherence to the DSDM principles and the lack of an environment that provides the critical success factors for a DSDM project. All of the “standard” risks included in the DSDM Suitability Risk List still apply in the “package” case. Several extra risks are also introduced. The Suitability Risk List may need to be extended to cover these specific extra risks. These are described below, with suggested countermeasures.

	<b>Risk</b>	<b>Possible Countermeasure</b>
	Slow procurement practices (particularly if affected by European Union/World Trade Organisation (EU/WTO) regulations) will endanger rapid deployment principles and may	Remove the procurement cycle from those elements of the project to be handled by DSDM, i.e. it becomes a hybrid project.

	<b>Risk</b>	<b>Possible Countermeasure</b>
	render the principle of “active user involvement” difficult to comply with.	
	Traditionally selection and procurement practices involve a different team and skill set from those needed for deployment/ tailoring of the package. This may lead to lack of communication and loss of “buy-in” from the team.	Some members of the team are involved throughout requirements elicitation, package selection, procurement, tailoring and deployment, e.g. Team Leader, Ambassador User.
	The chosen package may not be easily tailorable/ extensible.	Use DSDM for requirements elicitation phases only; then revert to traditional approaches for deployment. (Note: This may involve significant changes to business working practices).
	The chosen package may not be easily partitioned, thus making incremental build/modification difficult.	Accept the need to implement as a complete package. If necessary, allocate parallel, but closely coupled, DSDM teams to work on the tailoring and deployment phases.
	The internal structure of the package software may not be available, so causing difficulties in modelling extensions and system interfaces.	If possible, reverse engineer the models, particularly the data/object/class model from the package software set. Otherwise insist on continuous support from a supplier expert while “bespoking” or building interfaces.
	The package supplier’s staff may have little knowledge of or do not “buy in” to the DSDM approach.	Arrange attendance on DSDM training courses for key supplier staff (or choose a better supplier!)
	The known “cheap and cheerful” toolset used to elicit requirements pre-selection and procurement is unable to adequately represent the functional features required from the package solution.	Look for “easy” development tools that fit the area under consideration e.g. workflow, document management or ask the package supplier to provide an “on-loan” copy of their software.
	Requirements change <u>after</u> the package has been purchased, and these changes cannot be cost-effectively handled by package tailoring or bespoke “add-ons”.	Reject non-mandatory changes of this type post contract, but note the requirement for possible future projects (which could be initiated before the end of the current project).

## 5 Lifecycles and Pathways

The guidance below is intended to form a framework for decision-making, and for structuring and planning the different types of DSDM package selection and deployment project. Further tailoring of the frameworks presented here will be inevitable at the individual project level.

The approach described here as *Package DSDM* considers the DSDM lifecycle separately for package selection and package deployment (package implementation). It allows for the fact that DSDM may be used **just to select** the package, but not to implement it. DSDM may also be used to **implement** a previously selected package, chosen by means other than DSDM.

A number of decision points are defined in the lifecycle where a “breakout” of the Package DSDM lifecycle into bespoke development may be necessary, usually because of the perceived lack of a suitable package.

The following broad categories of project emerge:

### Package Selection

- **Requirements are well understood or not excessively volatile**, i.e. the requirements are well known, in sufficient detail, and are unlikely to change significantly during the project;
- **Requirements are not well understood or are subject to change**, i.e. the requirements are initially vague, or are likely to be subject to change during project, as user representatives become more aware of the possibilities.

### Package Deployment

- **Little or no bespoke work will be needed to deploy the package;**
- **Considerable bespoke work will be needed**, either to the package itself, or in order to provide interfaces etc to implement the package. Note: Bespoke work may be by supplier or customer staff, or mixed or parallel teams of both.

In addition to the above broad categories, the cases of **selection only** (deployment via a non-DSDM route) and **deployment only** (package already selected) are covered.

Further, two styles of deployment are supported: **single-shot** and **incremental** (module by module).

It is assumed that:

- some bespoke elements may have to be written to supplement the package functionality and to support other system interfaces;
- the package may support considerable tailoring of its functional and usability features, through parameterisation;
- modifying business practices to be compliant with the package paradigm may be acceptable.

### 5.1 Package DSDM Lifecycles

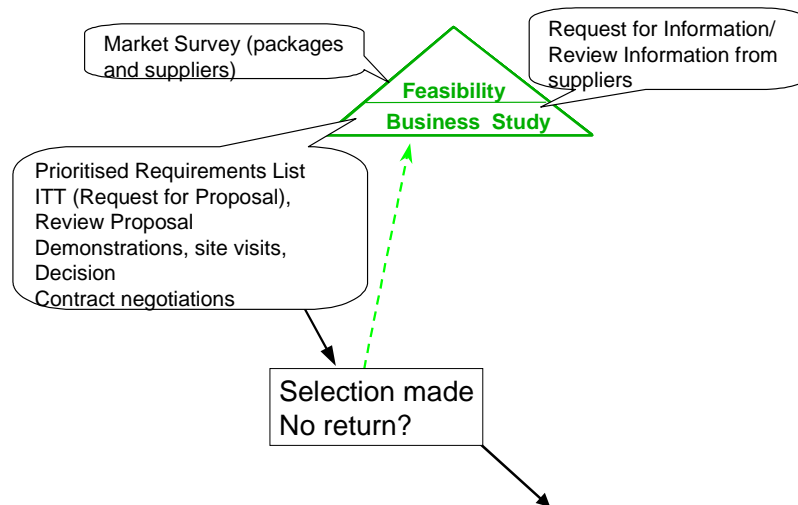
A number of possible lifecycles are shown diagrammatically below. The “speech bubbles” indicate extra/modified activities or products within the standard lifecycle phases.

The Pre-Project and Post-Project phases are not illustrated in the diagrams, but are briefly considered in the text. They are unlikely to be significantly different for the package route.

Package Selection is shown separately from Package Implementation, but it is likely that it would lead onto Package Implementation, either via DSDM or some other approach. Equally, Package Selection may be achieved by an approach other than DSDM, but Package Implementation may follow the DSDM framework. The diagrams show, separately:

- Package Selection – Requirements well-understood/not excessively volatile
- Package Selection – Requirements vague and changing
- Package Implementation (whether standard package or including bespoke work)

#### 5.1.1 Package Selection – Requirements well-understood/not excessively volatile

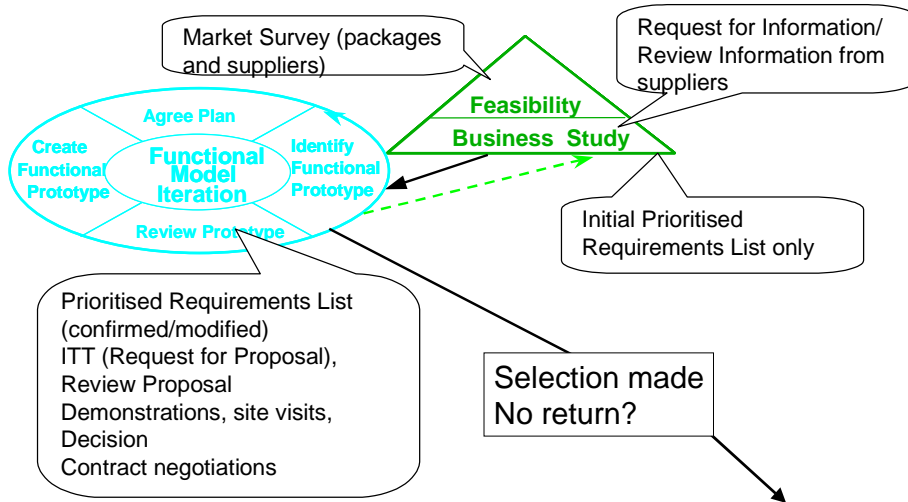


**Figure 1: Package Selection – Requirements well understood/not excessively volatile**

In this scenario, high level requirements can be listed and prioritised, suppliers can be identified, invited to tender for the work, and package selection can be made during the Business Study or even Feasibility Study, without need for prototyping of requirements. This would be the case, for example, where the business requirement is standard and many packages exist on the market already (for example, packages for accounting software).



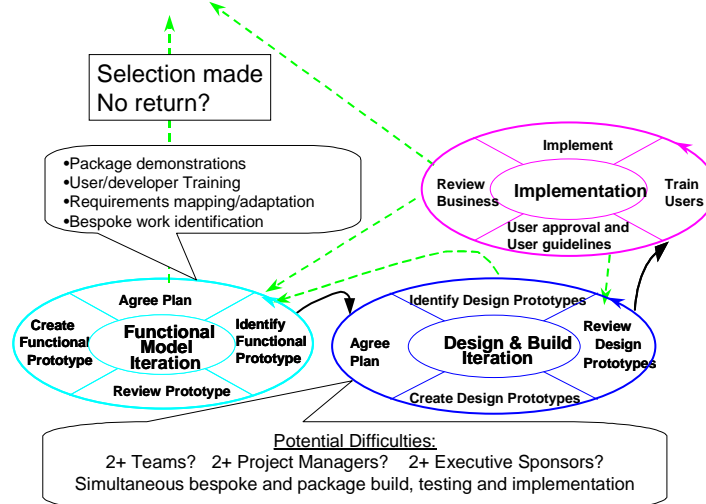
### 5.1.2 Package Selection – Requirements vague and changing



**Figure 2: Package Selection – Requirements vague and changing**

In this scenario, requirements are not sufficiently well understood during Feasibility and Business Study, but can be clarified by prototyping and workshops in an iterative way. Here a Functional Model Iteration (FMI) is undertaken to establish and prioritise requirements. This needs to be at a sufficient level of detail to convey a need to a supplier, but not to actually build the software from (as would be the case with conventional FMI).

### 5.1.3 Package Implementation



**Figure 3: Package Implementation**

In this scenario, we are assuming that the package selection has already been made, either via DSDM or any other approach. We enter the project with a chosen package and

supplier. Although “all changes are reversible” we are unlikely to want to reverse the selection decision at this point!

Even if an initial FMI, had been undertaken during package selection, the Package Implementation begins with a further iteration of the FMI. The purpose of this is to ascertain what the package can do in detail, to map the gaps between that and the business requirement. It is also necessary to ensure initial training in the package for customer staff that will be involved in customising and deploying it and to make decisions about how the gaps are to be filled and where organisational change may be necessary.

The Design and Build Iteration may have elements of conventional software design and build or may be more concerned with parameterisation of the package and alteration to methods of working to accommodate the package. There may be complexity of software being built in parallel by supplier and customer developers. There are also the complexities of managing a project with two (or more) organisations, with their different management structures, priorities, locations etc. that would also apply to the previous and subsequent phases.

## ***5.2 How does the Package-DSDM Lifecycle differ?***

Some extensions and changes are necessary to the default DSDM approach.

The aim has been to minimise these changes to retain compatibility with the “standard” lifecycle; changes are highlighted where they are necessary to produce an integrated approach.

In particular, in the selection phases of the lifecycle, extra activities have been introduced to deal with a package solution:

- I. Request for Information (RFI)
- II. RFI Review
- III. Requirements Definition (Req. Def.)
- IV. Request for Proposal (RFP)
- V. RFP Review
- VI. Visit(s)
- VII. Decision
- VIII. Complete Contract (CC)

**Activities I and II:** A request from suppliers for initial information, and confirmation of willingness to tender (RFI) and the RFI Review would normally take place (as part of a package selection) in the Business Study phase;

**Activity III:** Requirements Definition would either be completed during Business Study (where requirements are well understood), or from an initial set of Functional N Model Iterations (where requirements need investigation through prototyping). This produces the Prioritised Requirements List, which forms a major part of the Invitation to Tender (ITT) issued to suppliers.

**Activities IV to VIII:** Specifically related to Package Procurement, these would occur after either Business Study (requirements were well understood) or FMI (requirements needed investigation through prototyping). They would lead into Package Deployment, which would normally begin with FMI, to drive out requirements in relation to the package now selected.

A variant of the FMI “standard” phase can be used to elicit global requirements ahead of the “Request for Proposal” activity. This uses “cheap and cheerful” tools (or an “on loan version of a potential package solution) to create early visualisations of a set of functions that meet the initially perceived requirement. The objective is, through the use of at least two iterative prototyping cycles, to improve the quality of the requirement used in the package selection/procurement process.

Below, we consider each of the lifecycle phases, focusing first on selection, and then on implementation of the package.

In each case we consider:

- The approach;
- Products, and whether they are:
  - new (N)
  - augmented (A), or
  - diminished/different emphasis (D)
- Roles (in addition to standard DSDM roles).

### 5.2.1 Pre-project

#### Approach

The Pre-project phase addresses the initial definition of the business problem to be addressed, and an outline scope for the investigation to take place during the Feasibility Study. This may include the decision to consider package software, or even state this as a constraint. The Visionary and Project Manager will normally be assigned to the project and initial plans for the Feasibility Study Decision will be made. An initial high-level market survey of packaged software may be needed to determine whether to proceed with the project.

Budget and resources will be allocated for the Feasibility Study at least and preferably the Business Study. Outline budget/resources approval for the development/deployment phases will be stated. Initial project governance will be established; this may change on selection of the package. Required governance may form part of the Invitation to Tender.

#### **Products and Roles**

As for standard DSDM Pre-project.

### 5.2.2 Feasibility Study

#### Approach

The Feasibility Study has the additional objective of ascertaining the existence of suitable packages and suppliers. The standard DSDM Feasibility Study Phase is augmented with a Market Survey of software packages available, and an initial investigation into the suppliers who should be contacted (to determine their stability and suitability to be contacted.) Selected suppliers are contacted with a Request for Information (RFI). A “Feasibility Prototype” product may be built to establish the scope of the project, or it may be augmented or replaced by one or more demonstrations of packages on the market, to give an impression of the features and styles available. The augmented Suitability Risk List is applied for the first time during this phase.

#### Products

Standard (S); New (N); Augmented (A); Diminished/Different (D) DSDM Products

Product	Status S, A, N, D.	Differences/Features
Feasibility Report	A	Extended Suitability Risk List  Alternative routes through the project lifecycle that are acceptable, dependent on bespoke or package decisions.
Feasibility Prototype (optional)	D	May consist of demonstrations of packages already on the market, or may be created in a tool not intended for a final build of the product.
Outline Plan	D	Definition of deliverables for later project phases will be less detailed than is possible in a bespoke project  Configuration Management and Change control may be defined by the supplier of the package, who is unknown at this stage. Success criteria for this can be defined here.  Actual supplier resources cannot be identified, although criteria for inclusion in an ITT can.
Risk Log	S	
Market Survey Report	N	Evaluation of the results from a market survey of package solutions and suppliers in the area of business under consideration.

Request For Information/Willingness to Bid	N	This is a request to suppliers for initial information (prior to sending a full ITT) and includes a Willingness to Bid Request
--	---	--

## **Roles**

These are unchanged, with the possible addition of:

- Procurement Specialist
- Contract Specialists

### 5.2.3 Business Study

#### **Approach**

There are additional pre-conditions to be taken into account for the Business Study:

- The decision to follow a package “route” has been made;
- The level of flexibility of management to change organisational procedures to fit a package, or to customise;
- The decommit criteria if a suitable package cannot be found are defined.

The standard DSDM Business Study Phase will initially be at a high level (which is performed in all cases except when the package solution has already been chosen at Feasibility Study or as part of another project/programme). A more detailed Business Study is used both when a mainly bespoke route is chosen and where, although a package route is preferred, the requirement is perceived as too vague for entry immediately into a selection process.

#### **Products**

Standard (S); New (N); Augmented (A); Diminished/Different (D) DSDM Products

Note: The products marked \* may be produced here, or completed during an initial FMI phase.

<b>Product</b>	<b>Status S, A, N, D.</b>	<b>Differences/Features</b>
Business Area Definition (BAD)	D	The BAD is somewhat diminished. It cannot name specific resources on the supplier side at this stage, although it can define customer roles and responsibilities. It can define the high-level data and functionality required, but cannot define the underlying structures of the eventual package.
Prioritised Requirements List (PRL)	A	Where the requirement is not well understood/subject to change, only an initial PRL can be produced at this stage, to be augmented by

<b>Product</b>	<b>Status S, A, N, D.</b>	<b>Differences/Features</b>
		an initial FMI prototyping exercise, before package selection takes place.
System Architecture Definition (SAD)	D	This can define criteria for the Hardware/software/networking platform and add constraints of current system. However, it will need to be revisited after the chosen package is determined.
Development Plan	A	This is a two-stage plan: initially, it will specify the schedule required for package selection and implementation, and the in-house resources available. After selection (and as part of the negotiation process) it will be amended to fit the chosen software and augmented with supplier resources and agreed timescales/ timeboxes. .
Revised Plan	N	If contracts are decided at Business Study stage (and not after an initial FMI), this will be produced here. This may be a totally new plan, produced as a result of negotiations and discussions between supplier and customer, or may be an augmentation of the Development Plan. However, it will now have specific human (and other) resources allocated and timeboxes and products more fully defined.
Risk Log (updated)	S	
Selection Criteria and Weightings	N	As a separate exercise from producing a PRL, it is essential that the specific selection process and criteria are determined. This will ensure fair and equitable assessment of all suppliers' proposals, on receipt. These will be such criteria as: <ul style="list-style-type: none"> <li>• Functional Match</li> <li>• Costs</li> <li>• Technology</li> <li>• Company Profile</li> <li>• Response Quality/ Enthusiasm for DSDM approach</li> </ul>

<b>Product</b>	<b>Status S, A, N, D.</b>	<b>Differences/Features</b>
Proformas for Supplier Visits, Reference Site Visits and Demonstrations	N	<p>This is an interim product, but is necessary to ensure a fair and objective assessment process. It should cover:</p> <ul style="list-style-type: none"> <li>• Pro-forma for Supplier Visits</li> <li>• Schedule for Supplier Visits</li> <li>• Documented results of Visits</li> <li>• Documented areas of continuing concern</li> <li>• Pro-forma for Reference Site Visits</li> <li>• Schedule for Reference Site Visits</li> <li>• Documented results of Visits</li> <li>• Resolved issues with suppliers following Visits</li> </ul>
*Invitation To Tender (ITT)	N	<p>This is derived from the PRL and contains details of functional and non-functional requirements, plus hardware, infrastructure and other requirements and constraints. It would also seek information about the supplier (stability, customers, training provided etc.) and place certain requirements on the return of a proposal to supply.</p>
*Draft Contract (N)	N	<p>This may be included in the ITT, or may be for internal use at this stage, until suppliers have been short-listed.</p>
*Agreed Contract	N	<p>If contracts are decided at this stage (and not after an initial FMI), the supplier and customer both must agree and sign the final contract.</p>
Selection Decision	N	<p>This should contain details of:</p> <ul style="list-style-type: none"> <li>• Eliminated bidders and reasons</li> <li>• Potential suppliers, in order of preference</li> <li>• Areas of concern, including “no suitable package” decision.</li> </ul>

## **Roles**

These are unchanged, with the addition of:

- Procurement Authority
- Contract Specialists

#### 5.2.4 Functional Model Iteration (Package Selection - requirements vague and changing)

##### Approach

The Functional Model Iteration Phase may effectively occur as two separate phases:

- one during Package Selection, with a number of iterations whose main purpose is to elicit further requirements by building prototypes to demonstrate potential software solutions;
- the other during Package Implementation, with the aim of using iterations to:
  - demonstrate functionality, and identify errors and omissions;
  - drive out further, detailed, specific non-functional requirements.

The FMI phase defined below is used to drive out/clarify requirements in order to produce a sufficient ITT to make a selection.

FMI (Package Implementation) is considered separately later.

##### Products

Standard (S); New (N); Augmented (A); Diminished/Different (D) DSDM Products

Product	Status S, A, N, D.	Differences/Features
Functional Model (including Functional Prototypes)	D	For FMI during Selection, the functional model will show functionality required in the package. Functional prototypes will be built without the intention of carrying them forward to the finished product. Testing criteria will need to be defined, with a view to selection of the best package.
Non-functional Requirements List	S	For FMI during Selection, key non-functional requirements required in the package must be identified. Testing criteria will need to be defined, with a view to selection of the best package.
Functional Model Review Records	S	
Implementation Plan (including	D	This will be considered from the point of view of constraints and requirements for package selection,



the strategy for data take-on, if appropriate)		to be included in the ITT as a part of the requirement. Data take-on could also be included in the ITT if it is thought to be a major selection determinant, and will certainly need to be revisited once a package has been selected, or is in the final stages of selection.
Timebox Plans	A	Timebox plans will need to be established for the selection process.
Risk Log (updated)	A	<p>The purpose of this is to assist management in deciding the future of the project. This should identify major risks of the package approach, including reliance on third party software and support. These factors will feed into the ITT and the selection criteria.</p> <p>The product will need to be revisited once a selection has been made, to reassess risk and identify countermeasures.</p>

#### **Additional, Augmented DSDM products:**

The products below may also emerge from FMI during the selection process, and are a duplicate of those that would be produced during Business Study, if requirements are well understood, and there is no need for the FMI stage during Package Selection. For more details, please see under Business Study above.

- Prioritised Requirements List (enhanced)
- Invitation To Tender (ITT) (N)
- Draft Contract (N)
- Agreed Contract (N)
- Revised Plan (N)
- Selection Decision (N)

#### **Additional Roles**

- Supplier Management
- Supplier Technical (Hardware/software)

#### **5.2.5 FMI (Package Implementation)**

##### **Approach**

The FMI stage during Package Implementation is concerned with Functional Demonstration, in order to:

- Give the customer familiarity and initial training in the package
- Map functions/data/objects to identify areas not covered
- Implement control structures (e.g. define configuration items to be managed)

The default package will usually be installed, and used for demonstration, education and training. Prototyping workshops will be used to determine customisation to the package and business procedures.

### Products

Standard (S); New (N); Augmented (A); Diminished/Different (D) DSDM Products

Product	Status S, A, N, D.	Differences/Features
Functional Model (including Functional Prototypes)	D	<p>For FMI during Implementation, the functional model will represent the software package chosen. This model should be mapped against the Functional model derived before the selection was made, to identify gaps between the provisions of the package and the business requirement.</p> <p>A new Functional Model then emerges, based on how the package will be used and customised, and what bespoke work is needed.</p> <p>Testing criteria will need to be defined, with a view to testing the package, particularly in areas of change, areas of bespoke work and interfaces between the package and other company systems.</p> <p>There may also be large elements of bespoke work, which need more conventional, detailed prototyping and functional model analysis.</p>
Non-functional Requirements List	S	For FMI during Implementation, key non-functional requirements required in the package must be tested.
Functional Model Review Records	S	
Timebox Plans	A	Timebox plans will need to be established for Supplier teams, customer teams and mixed teams.
Implementation Plan (including the strategy for	A	This is now developed in relation to the chosen package. It is similar to the standard DSDM product. However, there may be a greater element

data take-on, if appropriate)		<p>of organisational structure/ role change in the package environment.</p> <p>Training plans may need a different approach form those for an in-house built system, due to cost and supplier geography.</p> <p>Data take-on will need to be revisited once a package has been selected, or is in the final stages of selection.</p>
Risk Log (updated)	N/A	This should now be revisited to identify major risks of the chosen package and supplier. A full risk analysis should be performed to reassess risk and identify countermeasures.

### Additional Roles

- Supplier Management;
- Supplier Technical Authority (Analysis);
- Supplier Trainers;
- Supplier Technical (Hardware/software).

### 5.2.6 Design and Build Iteration Phase

#### Approach

Many levels of complexity are possible at this stage, from the very simple one-team structure to a complex scenario, consisting of several parallel teams. If there is considerable bespoke work, one scenario could be:

- Supplier team implementers, with ambassador/advisor users and customer-side testers, implementing the parameterised standard parts of the package;
- Supplier team of developers, with Ambassador/advisor users and customer-side testers, building and testing bespoke code;
- Customer developers and testers, with ambassador/advisor users, building interfaces and aspects not covered by the package.

#### Products

Standard (S); New (N); Augmented (A); Diminished/Different (D) DSDM Products

Product	Status S, A, N, D.	Differences/Features
Design	A	Aspects of the system that are bespoke will follow

Prototypes (and Design Prototyping Review records)		the standard DSDM approach. Other areas, covered by the packaged software will have to take particular account of: <ul style="list-style-type: none"> <li>• Re-prioritised requirements</li> <li>• Change Request Logging</li> <li>• Errors and resolutions</li> <li>• Sign off/acceptance</li> </ul>
Tested System	A	Bespoke aspects will need a more conventional DSDM approach, but contractual issues still apply if the bespoke work is being done by a third party.  Modified business processes and/ or procedures may be a more significant proportion of the project than with a non-packaged solution.  Acceptance of timeboxed deliverables by users may have to be more formal.
Test Records	A	For package elements, modified parameters, interfaces and changed areas should be the priority areas for testing.

### Additional Roles

- Supplier Management
- Supplier Trainers
- Supplier Technical (Hardware/software)

### 5.2.7 Implementation

#### Approach

This is as for standard DSDM, with added complexity of co-ordinating supplier and customer staff in joint implementation.

#### Products

Standard (S); New (N); Augmented (A); Diminished/Different (D) DSDM Products

Product	Status S, A, N, D.	Differences/Features
User Documentation, any supporting	A	This may be partially in existence as documentation to cover the standard package. However, it will need to be augmented to cover changes made in this

procedures		particular implementation of the package. It needs to be clear where the standard documentation applies and where it does not! Bespoke areas, and interfaces also need documentation
Delivered System, together with supporting build, delivery and acceptance records	A	This may be part of the contractual product to be supplied. Requirements for its delivery need to be included in the contract for the package supply.
Trained User Population (including operators and support staff)	S	Differences here relate to who does the training (supplier or in-house) and how the interfaces between bespoke areas and packaged areas are handled.
Increment Review Document	S	This may need to be a more formal review than for a totally in-house project. It may need to be performed very soon after the implementation, in order for money to be released to the supplier.

#### **Additional Roles**

- Supplier Management
- Supplier Trainers
- Supplier Technical (Hardware/software)
- Contract Specialist

#### **5.2.8 Post-project**

##### **Approach**

The main objectives of the Post-project phase are to keep the Delivered System operational and to assess whether the proposed benefits have been achieved. In a package deployment scenario, this may have added importance, since contractual conditions may have to be met by the supplier for final payment to be made.

The Post-Project handling of change requests would be a continuation of change control established during earlier phases, subject to contractual agreement on the supplier/customer responsibilities for such changes.

The location of the package and bespoke source code would be subject to contractual agreement, and may include an Escrow agreement (source code retention by a third party).

**Products**

Post-Implementation Review Report (with possible contractual implications)

Change Requests and New Releases of the Delivered System in response to Change Requests

**Roles**

On-going post-implementation roles and responsibilities must be clarified, for both supplier and customer.

## **6 Conclusions**

DSDM can be effectively used to allow package selection and/or deployment projects to benefit from an iterative, incremental and user-driven system implementation approach. Additional risks need to be addressed, and some extra roles will need to be considered.

## **Appendix A - Glossary**

### **Software Package**

A **Software Package** is a program or set of programs designed to meet the requirements of a particular class of user.

### **Application Software Package**

An **Application Software Package** (also referred to as an “Application Package” or just “Application Software”) is a program or set of programs and data structures designed to meet the needs of particular classes of users to support the scope of a specific business problem; or business or organisational function. The application may be defined in terms of the industry to which a user organisation belongs and/or to the aspect of its operation that the package addresses. For example, an application package may support a wide variety of banking functions, or the entire operation of a travel agency or may support a function such as payroll. Generic Application Packages exist, for example word processing packages, spreadsheets. Much of the operation of a package will usually be fixed but the user can often modify aspects of the working of the package, such as the appearance of the user interface, the frequency and format of reports, and with some packages the ways in which internal data processing is carried out, by the setting of parameters.

### **Component**

A component is a self-contained unit of computer logic, together with its supporting data structures, with predictable outputs for a given input or set of inputs. Copybooks, include files and dynamically linked libraries were all steps on this journey. More recently, object oriented approaches have generated objects consisting of processing rules and data which can be “bolted together” as self-contained to form complete Applications to address business needs.

Components can be assembled from multiple sources to construct an application. Some may be acquired as pre-existing software some may be purpose-built in-house.

Two basic ideas underlie CBD. Firstly, that application development can be significantly improved if applications can be quickly assembled from pre-fabricated software components. Secondly, that an increasingly large collection of inter-operable software components will be made available to developers in both general and specialist catalogues, reducing the need for code to be custom-built.

### **Tailable infrastructure packages**

Tailable infrastructure packages are applications packages with considerable flexibility and scope. They are designed to cover and integrate many or most of the functional areas of an organisation. They allow the proficient package deployment specialist to select the ways in which they operate, both at user interface and more fundamental processing levels, by effectively bolting together, or selecting/de-selecting different components of the overall package. By the use of parameters, they can be configured to support the



business; they may be flexible enough to effectively equate to an applications development environment.

### **Middleware**

Middleware is software that acts as an intermediary between an operating system and applications software. Its purpose is to make the task of the applications programs (and those who develop them) easier and more standardised. Middleware performs frequently used, complex tasks and tasks requiring co-ordination between a number of applications programs. Applications programs usually call middleware routines by using special statements in their coding. These are interpreted by the operating system or by some other piece of software such as a compiler, to activate the appropriate middleware routine. Typical examples of middleware are teleprocessing monitors and database management systems.