DSDM®
CONSORTIUM

Rational. software

White Paper
DSDM and Rational Unified Process (RUP)

© Copyright DSDM Consortium 1999. All rights reserved

Background

The work on the Interoperability of the DSDM Framework & the Rational Unified Process (RUP) has arisen from the DSDM Community. A number of organisations that have been successfully utilising DSDM have been asking the DSDM Consortium for guidance on how to integrate RUP content to complement DSDM. As a result of such requests a team was formed to examine the Interoperability of DSDM with the Rational Unified Process. The intended audience of this paper is the DSDM community who are familiar with both DSDM and RUP. For those unfamiliar with RUP an introduction may be obtained from http://www-306.ibm.com/software/rational/

Introduction

DSDM and the Rational Unified Process have established themselves as the leading contenders in the arena of iterative development lifecycles. This makes them sound like competitors, which they need not be[1]. However, in the minds of many users it may seem that they are being forced into making a methods choice. Each comes as a self-contained package and it is not immediately obvious how to inter-operate the two. It is the premise of this paper that RUP is not a competitor to DSDM, but in fact may be complementary. The coverage of DSDM is wider than RUP, but relatively shallow whereas RUP is very detailed for the types of projects that it deals with most effectively.

This paper addresses the issue of inter-operation of these two processes from a DSDM perspective. It examines the differences and the similarities. This is necessarily subjective but is useful in pointing up the advantages to be gained from being able to combine the strengths of the two approaches. An example of the strengths of a combined approach is in the area of Component Based Development. The adoption of Component Based Development has led to the need for many organisations to have a software development approach that combines two styles of development. One must support the rapid assembly of solutions that use the services provided by components. The other approach must support the construction of those components and their supporting architecture. Business driven rapid application development is the corner stone of DSDM thus the component assembly approach may lend itself well to a DSDM process. The component development phase requires a more rigorous architecture-driven development approach, which it is characterised by RUP.

It is possible to characterise a process by any number of characteristics. Table one provides an approach to classification describing how the two approaches complement each other.

	DSDM and RUP
Principles	DSDM provides 9 principles which are supported in the RUP. It is therefore possible to map the principles. (see appendix one)
Lifecycle	Both processes support an incremental and iterative lifecycle and encourage an approach that encourages rapid delivery.
Process Model	The RUP process model provides a detailed definition of the underlying meta-model for the process. This meta model could be considered to be an implementation of the DSDM framework.
Terminology	Although both approaches use different terminology there is large amount of common ground (see glossary).
Roles	Both processes define clear and concise responsibilities for the process activities. DSDM has additional definitions of roles such as Ambassador Users and Visionary. However, RUP has more details on the Development Staff and their roles and responsibilities.
Techniques	DSDM provides a number of high level techniques, RUP provides a large amount of techniques including detailed guidance on how to apply them.
Guidelines	Both processes include guidelines on different aspects of the project lifecycle. RUP includes very complete guidelines on UML modelling, Requirements management, Development, Testing and Configuration management. DSDM includes support for facilitated workshops and timeboxing.

[1] Rational are full members of the DSDM Consortium.

© Copyright DSDM Consortium 1999. All rights reserved

Templates	RUP includes a number of general and tool specific templates. DSDM deliberately avoids templates to encourage wider usage.
Examples	Both processes provide project examples, DSDM providing these examples in the form of white papers, where RUP includes them in the on-line process.
Tools	RUP provides comprehensive documentation on how the Rational Tool set integrates with the process. This provides the practitioner with context sensitive help within the particular Rational tool. DSDM provides guidance on tools to assist people in choosing their own tool set.

The paper explores the ground rules for inter-operation, how you match the most appropriate process to your project needs, how you might mix-and-match techniques and how deliverables map between the two processes.

DSDM and RUP – Comparison

DSDM and RUP share a common approach to software development. Both use iterative development, and have a strong focus on developing software that meets the Users needs. Each process promotes continual testing throughout the software development process, configuration management and prioritisation of requirements.

So with all these similarities why should you consider looking at another method?

Why might you want to look at RUP when you have DSDM?

To quote the DSDM Manual,

"DSDM is more a framework than a method. It does not say how things should be done in detail, but provides a skeleton process and product descriptions that are to be tailored to suit a particular project or a particular organisation".

Many DSDM Users have seen this general guidance as advantageous, as it can be tailored to fit their environment and/or project. Some experienced DSDM Users may look for further detail and guidance in specific areas, and this is where RUP can provide assistance. RUP provides detailed descriptions of not just the 'what' needs to be done, but also 'how to' carry out its activities. It offers guidelines and templates. In addition, tool mentors describe how to apply the techniques defined in RUP to the Rational Tool Set, if appropriate. To take an example, a company may follow DSDM, but would like specific guidance on the development of an architecture. RUP can provide this detailed guidance. RUP is in its essence a detailed implementation of DSDM.

Note: This is not to say that DSDM cannot be extended – indeed there is a great deal of work done by the Consortium to extend DSDM with further guidance in a number of areas. This tends to take the form of white papers, full details of which can be obtained from the Consortium.

Why might you want to look at DSDM when you have RUP?

"Once a DSDM Practitioner, always a DSDM Practitioner"

It was suggested in an early meeting of the team that DSDM was a paradigm – you can approach software development with a DSDM frame of mind, almost regardless of the process. Indeed this could be applied to RUP. One of the early steps of RUP is to tailor it to meet the needs of a project – by creating what RUP calls a 'Development Case'.

If you are a RUP user and need to adopt a business-centred RAD approach then DSDM will provide many useful techniques. In this case it would be useful to supplement RUP with DSDM techniques by creating an appropriate RUP development case. For example, you may decide to use time boxing with RUP and MoSCoW prioritisation. Those experienced with DSDM, who have begun to use RUP may wish to take such an approach.

© Copyright DSDM Consortium 1999. All rights reserved

In considering why interoperability is important it becomes clear that we are examining particular usage models of each method. This enables us to use the method with those techniques that are beneficial to the project at hand. In most commercial organisations[2] new projects are mixed, some architecturally focused and some more customer focused. Examples include "component development" projects, "component consumers" and the ongoing backdrop of legacy development. **In an ideal world it would be nice to have all these techniques defined in a shopping basket or tool kit from which you could pick the appropriate techniques regardless of the process being utilised.**

Which Process Lifecycle?

So far, the discussion has focused on techniques. In reality, the techniques are practised in the context of an overall process or lifecycle. DSDM and RUP are very similar in that they advocate true iterative development (as opposed to pre-planned iterative or waterfall). Where they 'seem' to differ markedly is the actual process lifecycle; RUP's Inception. Elaboration etc. appears very different to DSDM's "cheese and 3 pizzas" model[3]. Close examination reveals that it is possible to map the two approaches.

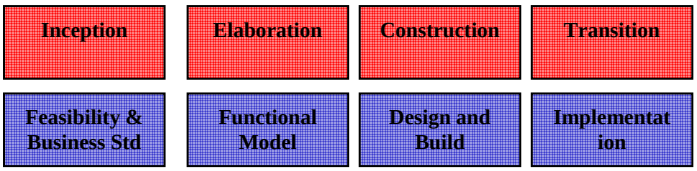

Figure 1

Figure 1 illustrates a possible mapping. Each phase has a number of objectives – these correlate with DSDM product quality criteria and the RUP milestones. It is therefore possible to consider RUP as a possible implementation of the DSDM framework. However, the framework alone is not enough for a project manager to start their project

What project managers require is some kind of assistance to help them choose the most appropriate techniques and associated iteration structure for their project. One approach suggested picks up on a DSDM idea – the "Suitability Filter". The Suitability Filter is already used at the start of a DSDM project to ensure that use of DSDM is appropriate. A small extension of this idea would enable a Suitability Filter which would ask the project manager the right questions in order to decide which techniques and associated iteration structure. To avoid confusion and to provide a name that is more appropriate the extended suitability filter is called "The Process Design Assistant". An example of the actual assistant is described in Appendi*x 2: "The Process Design Assistant"*.

Hybrid Projects

Because RUP is an implementation of a DSDM framework it is possible projects may not wish to adopt all the parts of RUP on every part of their project. Examples of such projects include Component Based Development and Scientific Development. The term "hybrid projects" is adopted here to describe these types of projects. A hybrid project may consist of several strands each focusing on a particular aspect of the project e.g. business process re-engineering, or creating a component. Each project would adopt the appropriate process style (DSDM, RUP or, perhaps, other); the interconnections between the strands would be managed by the project

This type of approach to project is very familiar to DSDM which encourages all project teams to be small and focused. The ability to mix different strands is addressed in a DSDM white paper but the addition of RUP is a useful addition. RUP includes sufficient rigour to make these complex projects less of a risk than would be the case in vanilla DSDM. This hybrid approach is consistent with the

[2] The authors have first-hand experience in a large investment bank

[3] Colloquial expression for Feasibility, Business Study, Functional Model Iteration etc.

© Copyright DSDM Consortium 1999. All rights reserved

adoption of a 'Development Case' within RUP. The development case would describe the hybrid project process, describing which aspects of RUP or other approaches are being used.

What has to be considered in the context of hybrid projects is that constituent strands will need to exchange deliverables in order for the project as a whole to complete. This raises the question of the compatibility of deliverables between DSDM and RUP.

Exchange of Deliverables between DSDM and RUP

Exchange of deliverables between the two may be less of an issue than might be expected because we are still assuming that within the RUP or DSDM process framework an individual strand may practise techniques from the other method. Thus, for example, a DSDM project may produce a UML-format deliverable which will, of course, be perfectly acceptable to a RUP project. What is clear is that the project must set some overall standards for deliverables and for configuration management of those deliverables.

It can be argued that, at a high level, all projects have similar deliverables. Thus, for example, you would expect to find a requirement specification and a project plan amongst the deliverables of just about any project. What differs from project to project is the form these deliverables might take: a requirement specification could be a set of use cases or, alternatively, the minutes of a JAD workshop; a project plan could be an MS Project plan or, alternatively, a set of timebox specifications. When inter-operating DSDM and RUP it is important to recognise the high-level generic purpose of each deliverable so that it can be mapped between the processes. See Appendix 3: *"Interchange of Deliverables between RUP and DSDM Projects"* for further detail on this topic.

Techniques in RUP, DSDM and other Methods

So far, we have considered the potential for mixing and matching techniques from RUP and DSDM but we have not considered, in detail, the techniques which might be so utilised. In fact, when this topic was analysed we also considered an in-house framework methodology in use within the one of the author's business. The reason for this was that we could see techniques which were absent from both RUP and DSDM. These techniques, in the main, relate to project funding and governance whereas RUP and DSDM both tend to assume a fully funded and justified project as their starting point.

The actuality of mixing and matching techniques is not a major problem. Project managers may choose to employ whatever techniques they find most appropriate within their overall chosen process framework. However, what this assumes is that a project manager is aware of all the available techniques and understands how and when to employ them to best effect. Project managers have to be methods-literate before they start the project. To combat this, the recommendation here is that the Process Design Assistant, proposed earlier, be extended to embrace techniques. Given sufficient input data about the characteristics of a project it should be possible to recommend suitable techniques that the project can deploy to best effect. So, for example, if the project has a fixed deadline MoSCoW scheduling and timeboxing would be recommended.

There is scope to extend this idea even further. All the methods reviewed here are delivered to their practitioners via web pages. It is a relatively easy exercise to superimpose further pages which can help project managers navigate all the hypertext available across a set of methods to access guidance relevant to their particular project. However, it would be burdensome to do this on a project-by-project basis to create truly project-specific guidance. As an alternative, it may be possible to recognise some common project "patterns" and have sets of guidance customised for each type.

The logical conclusion of this train of thought is that we really want to lift the whole development lifecycle concept by a meta level. Instead of considering processes we need to consider **processes for generating processes**. Instead of tools which provide a fixed set of guidance for one process we need tools which will **automatically customise processes and guidance according to project needs**. This is clearly not a short-term recommendation but it is, nevertheless, a very exciting proposition.

© Copyright DSDM Consortium 1999. All rights reserved

Programme Management Methodology

Both RUP and DSDM are essentially project lifecycles. Yet we have identified that there is value in running projects within the framework of a programme. Whilst we can tolerate process differences from project to project, the overall programme framework needs to be consistent, and sympathetic with the underlying project processes. In a sense we are fortunate that no programme management methodology has yet been defined for RUP or DSDM because it leaves an opportunity to provide one which works well for both.

A good start point for programmes is to look at the portfolio of techniques to decide which are important to apply at the programme level. The following list specifies these:

- ✓ Requirements Management
- ✓ Business Case Development
- ✓ Risk / Issues Management
- ✓ Dependency Management
- ✓ Quality Management
- ✓ Configuration Management
- ✓ Programme Metrics
- ✓ Milestone Reporting / Progress Visibility
- ✓ Suitability Filter
- ✓ Project closure, process and benefits review
- ✓ In addition to the above there is potential for other programme-wide standards which will need to be adhered to across all projects.

It is possible to define a programme management methodology in a similar fashion to a project methodology. It will consist of defined stages and steps to get the programme underway and, perhaps more importantly, a set of cross-lifecycle activities which must be undertaken to manage the projects effectively on an ongoing basis.

Summary of Conclusions

This work arose after a number of DSDM Users expressed the need to understand how they could complement DSDM with some of the detailed guidelines of the Rational Unified Process. The two should not be considered as competitors. RUP provides detailed guidelines and could be considered a detailed implementation of DSDM. In reality, many projects bridge the competence zones of the two approaches and would be better served by an approach which mixes techniques from both. This leads to the conclusion that there is definite value in examining particular usage models of each, pulling in to the usage model those techniques that are beneficial to the project at hand

Each approach can be regarded as an underlying process framework which hosts a number of techniques. Combining the process frameworks is probably a pointless exercise. Each serves certain needs well so it is better to choose the appropriate framework for each strand. A Process Design Assistant is proposed to enable projects to choose the most appropriate techniques for the project at hand. This would support the practitioner in the creation of a development case.

Proposals are made for a Programme Management Methodology to assist with large projects. Neither RUP nor DSDM currently define such a methodology so there is much to be gained from having a jointly agreed and promoted version.

There is a real need to be able to mix-and-match techniques from the two approaches (and other sources) within a chosen process framework. Whilst this is easy to achieve for the cognoscenti it is safe to assume that many project managers will not have either enough time or may not have sufficient knowledge to be able to derive benefit from such an approach. Current documentation of the two provides no assistance. We need to seek ways to provide the majority of project managers with processes, tools and documentation which are a best fit with their project needs. This opens up the exciting prospect of taking methods to a new plateau – the **meta method**. (*In doing so, methods will only be following the same lifecycle as many other technologies - after case tools came meta-case tools; after HTML came XML*). In the same vein, deliverables can be defined at the meta level as generic

© Copyright DSDM Consortium 1999. All rights reserved

deliverables; the actual deliverables for a given project will be defined by populating the meta deliverables according to the techniques in use and project requirements.

The significant factor in all of the above is that it is driven by customer need. This paper includes significant customer input and the route forward needs to continue to combine the efforts of all stakeholders.

Contributors

David Tuffs, Warburg Dillon Read; Jennifer Stapleton, DSDM Consortium; David West, Rational Software; Zoe Eason, Rational Software.

© Copyright DSDM Consortium 1999. All rights reserved

Appendix One

Mapping the DSDM Principles

RUP realises all of the DSDM principles either all or in part. Those where the DSDM guidance is stronger are marked with an asterisk. It would be appropriate to use the full strength DSDM principle within a RUP project. E.g. The inclusion of Ambassador Users within a RUP team.

DSDM Principle	RUP Realisation
* Active user involvement is imperative.	Throughout the process user involvement is encouraged. Examples include workflow details such as Analyse the problem, Understand Stake Holder Needs, Manage the scope of the system and Test. Techniques such as How to set up and Run Use Case Workshops, or more general Requirements Workshops are explained in detail.
* DSDM teams must be empowered to make decisions.	The RUP defines key roles and responsibilities and a requirements management process. This allows the team to make decisions within the scope of their responsibilities. Full details of the teams empowerment could be provided as part of the development case.
The focus is on frequent delivery of products.	Each iteration in the RUP Lifecycle will deliver something that is tangible and of value. The focus is on delivery of models and executables , that may be internal or external. Iterations are planned around the needs of a project – so you can tailor the process to include more frequent iterations if appropriate.
* Fitness for business purpose is the essential criterion for acceptance of deliverables.	If further understanding of the business is required, RUP provides detailed guidelines on Business Modelling. In addition, the Requirements Management aspect of RUP takes into account the needs of the Stakeholders. Quality is crucial throughout RUP with the definition of a quality plan. This quality plan is realised throughout the process with workflows such as testing and phase milestones.
Iterative and incremental development is necessary to converge on an accurate business solution.	The RUP supports the management and control of an iterative and incremental approach to software development. The detail of this support is in the Iteration and Project management workflows.
All changes during development are reversible.	The ability to manage and 'survive' change is fundamental to any iterative and incremental development process. RUP includes a number of mechanisms to manage change. RUP has a workflow dedicated to configuration and change management.
Requirements are base-lined at a high level.	During the inception phase requirements are scoped. This provides a firm foundation for the later stages of the development process. During elaboration requirements are taken through one or more iterations that add further detail. A prioritisation mechanism which takes into account various aspects of the requirement such as User Priority, difficulty, Cost, Impact on architecture allows you to scope each iteration to implement the appropriate Requirements. All requirements are baselined at the start of an iteration – any changes are reviewed during the iteration assessment.
Testing is integrated throughout the lifecycle.	Because RUP is a iterative approach, each iteration will have a testing element within it. This encourages testing to be undertaken throughout the project not just at the end.
* A collaborative and co-operative approach between all	There are a large number of roles defined with the RUP. Each role has a defined set of responsibilities – Because any one activity or phase

© Copyright DSDM Consortium 1999. All rights reserved

stakeholders is essential.	requires more than one role to contribute collaborative working is required. Techniques to support this collaboration such as requirements workshops and Use Case Analysis Workshops are defined.

© Copyright DSDM Consortium 1999. All rights reserved

Appendix Two - Process Design Assistant

What Is the Process Design Assistant (PDA)?

The Process Design Assistant (PDA) provides the practitioner with an approach to determine which mechanisms and techniques are appropriate for their project. It provides a set of questions which must be asked about the project. The answers will provide guidelines on which mechanisms and techniques are appropriate to the project.

How do I use the PDA?

Step 1 - Mechanism Selection

The first step is to review the Mechanisms table, there is a cut down version provided as an example below although the full table can be seen in Table One. Down the left of the table are the questions that should be asked about a project:

- Is the project large?
- Is the project considered complex?
- Does the project implement strategic foundations?

…etc

If the answer to any of the questions is yes, my project does exhibit these qualities then looking along the row we can identify the appropriate mechanisms – the numbers range from zero to three, the higher the number the more appropriate the mechanism.

	Iterative Development	**Architecture Driven**	**Deadline Driven**	**OO**	**Component Based**	**Risk Driven**	**Legacy**	**Leverages Components**
Large	**3**	**2**	0	1	**3**	**3**	**2**	**3**
Complex	1	3	0	2	3	2	0	0
Small	3	0	2	0	0	0	1	2
Flexible Solution	3	3	2	2	3	2	1	3
100% Solution	1	1	0	0	0	2	0	0
Strategic Foundations	0	3	0	2	3	1	2	1
Technical Software	0	3	0	2	1	2	0	1
Demanding non functional reqs	0	3	0	0	1	3	1	1
Vague requirements	3	0	3	0	0	2	0	0
rigid requirements	0	0	0	1	1	1	0	0
Fixed deadlines	2	0	3	0	0	3	0	2
Variable deadlines	3	0	0	0	0	2	0	0

For example: "Is the project considered by your organisation to be large?"

If the answer is yes then we can extract some mechanisms that would be appropriate for our project from the table.

© Copyright DSDM Consortium 1999. All rights reserved

Highly Recommended Mechanisms (Score of 3)	Recommended Mechanisms (Score of 2)
Iterative Development	Architecture Driven
Component Based	Geographically Dispersed
Risk Driven	Legacy Integration
Leverage of Components	

Step 2 - Technique Selection

Step 2 refers to a second table, again a cut down version of the table is provided as an example below, although the full table can be seen in Table Two. Once the mechanisms are selected it is then possible to determine which techniques apply to these mechanisms. Down the left of the table are the mechanisms defined previously. By looking at the identified mechanisms then looking along the row we can identify the appropriate techniques – again the numbers range from zero to three, the higher the number the more appropriate the technique.

	Time Boxing	Business Roles	MoSCoW	Prototyping	Configuration Management	Facilitated Workshops	Architectural Framework	Use Cases	Modelling (System)	Risk Analysis
Iterative Development	**2**	0	**3**	**3**	**3**	1	1	0	1	**3**
Architecture Driven	0	0	0	1	2	1	3	1	3	3
Deadline Driven	3	2	3	0	1	0	0	0	0	3
OO	0	0	0	0	0	1	3	3	3	0
Component Based	0	0	0	0	3	2	3	0	3	2
Risk Driven	1	0	2	3	0	0	0	2	1	3
Legacy	2	2	3	0	2	0	0	0	0	3
Leverages Components	0	0	0	2	3	0	1	3	1	0

If we have found Iterative development to be a useful mechanism then it is possible to determine which techniques might be appropriate: -

Highly Recommended Techniques (Score of 3)	Recommended Techniques (Score of 2)
MoSCoW	Time-boxing
Prototyping	Dependency Management
Risk Analysis	Quality Management
Configuration Management	Regression Testing
Incremental Testing	Milestone Reporting

Note: the above selection also refers to the more complete table two.

Why use the PDA?

From asking some very simple questions it is possible to develop a good picture of how the project is going to be approached. Going through these tables provides the practitioner with a definition of

© Copyright DSDM Consortium 1999. All rights reserved

both the important mechanisms that apply to the project (i.e. iterative development) and the techniques that will support them.

It is possible that contradictory answers will be provided. If this is the case then it is very likely that the project should be broken down into 'streams', each stream will be separate, but will managed within the project as a whole. The interchange of deliverables from the streams will be managed by the introduction of meta-deliverables (see Appendix two).

The PDA is only an assistant and does not replace the intelligence of the practitioner. It should be used to support the definition of the way in which the project will be organised. The benefit of this Assistant is that it does not assume that a Project Manager is aware of all the available techniques or that he understands how and when to employ them to best effect.

Is the PDA Complete?

The PDA is described more fully in tables one and two, however it is not complete, it is the first draft. It is very likely that both the questions and the weightings will need to be changed for particular organisations or industry sectors.

Once we have these Techniques defined what next?

These techniques should be used within an overall process framework. DSDM Practitioners may feel more at home using the DSDM 'cheese and 3 pizzas' model. However, if the techniques shown to be relevant are key aspects of the Rational Unified Process then you may wish to adapt the RUP framework and create a DSDM style Development Case.

© Copyright DSDM Consortium 1999. All rights reserved

Mechanism Selection

	Iterative Development	Architecture Driven	Business Driven	Deadline Driven	OO	Component Based	Small Team	Geographically Dispersed	Outsourced	Risk Driven	Legacy	Leverages Components
Large	3	2	0	0	1	3	0	2	1	3	2	3
Complex	1	3	0	0	2	3	0	0	0	2	0	0
Small	3	0	2	2	0	0	3	0	0	0	1	2
Flexible Solution	3	3	3	2	2	3	1	0	0	2	1	3
100% Solution	1	1	2	0	0	0	0	0	1	2	0	0
Strategic Foundations	0	3	2	0	2	3	0	0	0	1	2	1
Technical Software (see Note 1)	0	3	1	0	2	1	0	0	0	2	0	1
Business Engineering	3	0	3	2	0	0	2	2	0	1	3	1
Commercial	3	1	3	2	0	1	0	2	1	2	3	1
High UI Content	3	0	3	2	3	0	1	1	0	0	0	2
High Algorithmic Content	0	0	1	0	0	0	2	0	0	0	0	0
Demanding non functional reqs	0	3	2	0	0	1	0	1	0	3	1	1
vague requirements	3	0	3	3	0	0	2	0	0	2	0	0
rigid requirements	0	0	2	0	1	1	0	0	2	1	0	0
fixed deadlines	2	0	3	3	0	0	2	0	2	3	0	2
variable deadlines	3	0	3	0	0	0	0	0	0	2	0	0

Table One

© Copyright DSDM Consortium 1999. All rights reserved

Technique Selection

	Time Boxing	Business Roles	MoSCoW	Prototyping	Business Impact Analysis	Stakeholder Involvement	Facilitated Workshops	Architectural Framework	Use Cases	Modelling (System)	Risk Analysis	Measurement
Iterative Development	2	0	3	3	0	0	1	1	0	1	3	0
Architecture Driven	0	0	0	1	0	0	1	3	1	3	3	1
Business Driven	1	3	2	2	3	3	2	0	2	0	3	0
Deadline Driven	3	2	3	0	2	2	0	0	0	0	3	2
OO	0	0	0	0	0	0	1	3	3	3	0	0
Component Based	0	0	0	0	0	3	2	3	0	3	2	1
Graphically Dispersed	0	3	0	1	3	3	1	2	0	3	3	0
Outsourced	0	3	0	0	0	3	0	1	1	2	3	3
Risk Driven	1	0	2	3	2	3	0	0	2	1	3	2
Legacy	2	2	3	0	2	2	0	0	0	0	3	1
Leverages Components	0	0	0	2	0	3	0	1	3	1	0	0

Table Two

Technique Selection (further columns)

	Business Case Development	Dependency Management	Project Close / Review	Business Benefit Review	Configuration Management	Quality Management	Incremental Testing	Regression Testing	Capacity Testing	Milestone Reporting	Business Process Modelling
Iterative Development	0	2	0	0	3	2	3	2	0	2	0
Architecture Driven	0	0	0	0	2	3	0	0	2	0	1
Business Driven	3	1	0	2	0	0	0	2	1	3	3
Deadline Driven	0	3	0	0	1	3	3	0	0	3	0
OO	0	0	0	0	0	0	2	0	0	0	1
Component Based	2	3	0	2	3	3	3	3	3	0	3
Graphically Dispersed	2	3	0	2	3	3	3	3	2	3	2
Outsourced	0	3	3	0	0	3	0	3	1	3	0
Risk Driven	1	3	0	0	0	0	2	2	0	2	0
Legacy	2	2	0	1	2	3	0	3	0	1	0
Leverages Components	0	3	0	1	3	1	2	0	0	0	1

Appendix Three

Interchange of Project Deliverables

Not all software projects are the same! The usefulness of the techniques and framework that the techniques reside on depends on a number of factors including:-

1. The problem domain.
2. The skills and experience of the development team.
3. The organisation (including culture, legislation, and existing processes).

It is therefore the requirement of any project manager to tailor the process they are using depending on the nature of the project. The 'tailoring' comes from experience and skill of the project manager, often (if the environment is new, or the project has a large amount of unknowns) this tailoring is difficult, the result being that there is no overall process being followed. The intention of any software engineering process is to support the development team, not stifle or constrain it – The process is meant to provide a memory jogger or checklist that will help ensure that the end game will be achieved. Documentation plays a key role in the support for the development process providing a mechanism to both assess the project and to communicate how the software product works or should work. This appendix describes how in a hybrid project the documentation will be configured to allow interchange between the strands.

Once the project process and techniques to be used are determined, it is possible to determine the project deliverables. A project deliverable is defined as something which contributes to the project solution and is visible and able to be checked.

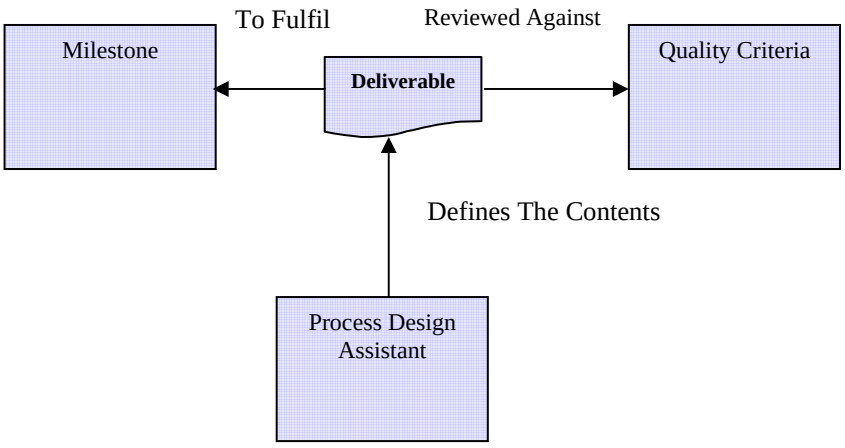

Figure 1

The figure illustrates the relationship between a deliverable and the process design assistant.

RUP Inception / DSDM Feasibility and Business Studies Phase

At the end of the feasibility phase is the first major project milestone or **Lifecycle Objectives Milestone**. At this point, you examine the lifecycle objectives of the project, and decide either to proceed with the project or to cancel it or to undertake another feasibility phase.

The evaluation criteria for the feasibility phase are:

© Copyright DSDM Consortium 1999. All rights reserved

- Stakeholder concurrence on scope definition and cost/schedule estimates
- Requirements understanding as evidenced by the fidelity of the primary use cases
- Credibility of the cost/schedule estimates, priorities, risks, and development process
- Depth and breadth of any architectural prototype
- Actual expenditures versus planned expenditures.

The above milestone maps to the DSDM quality criteria :-

1. Is the problem definition in line with the needs of senior user management?
2. Is the scope of the project sufficiently clear for it to be refined within the Business Study?
3. Are the business objectives to be met by the development clearly defined?
4. Is the solution to the problem, as laid out in the major products to be delivered and in the objectives of the project, feasible in both technical and business terms?
5. Is the case for using DSDM sound, i.e. does the application meet the criteria for using DSDM as laid out in the suitability filter. NB This is particularly interesting because it is important to define the culture of the project. Can the project be driven by time constraints or is the functionality not negotiable.
6. Does management accept what has been included and excluded from the scope?
7. Are all associated systems and their interfaces identified? Is any impact on those systems acceptable?

Deliverables that may be included :-

DSDM Product	**RUP Artefact**
Feasibility Report Feasibility Prototypes Outline Plan	Vision Document Business Case Project Plan Use Case Model Risk List Glossary

Content Element	**Example Realisation**
To outline the problem to be addressed by the new system	If the system is business process driven a series of business processes may be described to provide context. Once this model is stable it is possible to produce a domain model. RUP Artefacts Include :- Business Use Cases and appropriate realisations and a Domain Model,
To define the scope of the project or set of projects	System Use Case and Actor Model and any packaging of use cases.
To give a preliminary indication of any areas within the scope which may be desirable but not essential.	An initial ranking of the Use Cases in terms of their importance to the business.
To state, at least in outline, the Business Case for the project(s) - including where possible expected costs, benefits, assumptions and risks (whether quantifiable or not).	A RUP business case artefact provides descriptions of both the problem context, financial forecasts and the Return on Investment.
To define the major products to be delivered by the project.	These products are described in the use case model and the packaging of those use cases.
To report on the suitability of particular techniques for use on the project, which may vary for each solution.	A description of the overall development process and techniques that will be used. A description of the focus of the project should also be provided (i.e. Prototype or Architecture driven).
To document the objectives of the project including process performance criteria	A list of all the non functional requirements with particular reference to requirements such as robustness and performance.
To document high-level technical and business	Constraints and safety requirements are part of

constraints, e.g. timescale, hardware and software platforms.	the non functional requirements.
To identify whether the system may be safety-related or if there may be any product liability issues.	
To describe at a high level the business processes and data that are expected to be automated.	Business Use Cases may be used to describe the high level business processes. Parts of these business processes will be automated – These automated functions will be described in system use cases.
To identify at a high level the interfaces necessary to existing data and applications.	System use cases describe the input and output of the system – The actors reference in this description are the external systems or people that are using the system.
To identify which business processes and/or systems (whether automated or not) might be impacted by the new system and which might need to change in order to accommodate it.	Any impact should be described in the use case model with those impacted systems described as actors.
To define the expected life of the computer system and hence the requirements for maintainability.	The overall maintainability should be described in terms of the life of the product and the change-ability of that product.

RUP Elaboration / DSDM Functional Model Iteration Phase

The end of this phase is the second important project milestone, the **Lifecycle Architecture Milestone**. At this point, you examine the detailed system objectives and scope, the choice of architecture, and the resolution of the major risks.

The main evaluation criteria for the elaboration phase involve the answers to these questions:

- Is the vision of the product stable?
- Is the architecture stable?
- Does the executable demonstration show that the major risk elements have been addressed and credibly resolved?
- Is the plan for the construction of sufficient detail and fidelity, and is it backed up with a credible basis of estimates?
- Do all stakeholders agree that the current vision can be met if the current plan is executed to develop the complete system, in the context of the current architecture?
- Are actual resource expenditure versus planned expenditure acceptable?

The project may be aborted or considerably re-thought if it fails to reach this milestone.

The above milestone maps to the DSDM quality criteria :-

1. Are the business context, business process and business objectives defined and agreed?
2. Have all the currently identified requirements been prioritised (including non-functional requirements)?
3. Have all the priorities been assigned in collaboration with the users?
4. Have high-level acceptance criteria for the Delivered System been defined?
5. Are the business areas clearly documented, including high-level information needs that are affected by the system?
6. Is the envisaged boundary of the proposed new system realistic in the timescales?
7. Are all classes of users affected by the new system identified?
8. Are the information and processing requirements of the proposed system defined at least in outline?
9. Is it still clear that the business needs are being addressed by the proposed new system?
10. Is the person responsible for each business process identified? Can they commit the necessary resources and time?
11. Are the business context, business process and business objectives defined and agreed?

12. Have all the currently identified requirements been prioritised (including non-functional requirements)?
13. Have all the priorities been assigned in collaboration with the users?
14. Have high-level acceptance criteria for the Delivered System been defined?
15. Are the business areas clearly documented, including high-level information needs that are affected by the system?
16. Is the envisaged boundary of the proposed new system realistic in the timescales?
17. Are all classes of users affected by the new system identified?
18. Are the information and processing requirements of the proposed system defined at least in outline?
19. Is it still clear that the business needs are being addressed by the proposed new system?
20. Is the person responsible for each business process identified? Can they commit the necessary resources and time?
21. Are all major business events identified?

Deliverables that may be included :-

DSDM Product	**RUP Artefact**
Business Area Definition System Architecture Definition Prioritised Requirements List Outline Prototyping Plan Non Functional Requirements List	Iteration Plan Software Architecture Document Project Plan and Iteration Plan Use Case Model Risk List Analysis Classes Design Sub-Systems Design Classes Packages Interfaces

Business Area Definition

Content Element	**Example Realisation**
To identify the business needs that should be supported by the proposed computer system.	The mapping of the high level needs of the system to features and then the use cases.
To refine the Outline Business Case (documented in the Feasibility Report) to include benefits, risks, costs and impact analyses.	Adding more detail to the business case with reference to the how the system is going to be delivered and what that means in terms of benefits, risks, costs and how it will effect other systems. All these details should be added to the use case supplementary specification and requirements specification.
To outline the information requirements of the business processes that will be supported.	A description of the business component model, the services it provides and any dependencies between components. In UML terms these will be described as sub-systems and interfaces.
To identify the classes of users impacted by the development and introduction of the proposed system.	A list of all the actors (roles) combined with a stakeholder list will determine which roles will be interested in this development.
To identify the business processes and business scenarios that need to change.	The description of the use cases (basic and alternate courses) provides information on the detail of what the system does. The mapping of these to the business use cases help determine the level of impact.
To clarify all interfaces with other systems (human or automated).	The system use case list provides a detailed description of all the affected systems.
To verify that the proposed development process is working.	To ensure that the process is working documented evidence of a process review will be provided.

System Architecture Definition

Content Element	Example Realisation
To provide a common understanding of the technical architectures to be used during development and implementation.	A detailed description of the technology that will be used and how it will be used. Description of all appropriate patterns (such as persistence, error handling, etc..). Inclusion of any standards necessary.
To describe the target platform and (if different) the development platform.	Use of technologies such as COM and CORBA should be described and the overall deployment and process models such be defined.
To describe how the functional requirements map to the software architecture.	A list of use cases and how they map to the software architecture. It is important that the use cases that are crucial to the software architecture are developed / proved to ensure that the architecture is workable. In RUP this mapping is described in the use case view of the architecture.
To give an outline description of the software architecture (i.e. the major software objects or components - both process and data - and their interactions).	The layering of the model should be defined with each layers responsibilities expressed. All business components will be described, as will all the analysis mechanisms.

RUP Construction Phase / DSDM Design and Build Iteration Phase

At the end of the construction phase is the third major project milestone (**Initial Operational Capability Milestone**). At this point, you decide if the software, the sites, and the users are ready to go operational, without exposing the project to high risks. This release is often called a "beta" release.

The evaluation criteria for the construction phase involve the answers to these questions:

- Is this product release stable and mature enough to be deployed in the user community?
- Are all the stakeholders ready for the transition into the user community?
- Are actual resource expenditures versus planned still acceptable?

Transition may have to be postponed by one release if the project fails to reach this milestone.

The above milestone maps to the DSDM quality criteria :-

1. Does the design and build timetable still fit in with business needs?
2. Do the cost and effort estimates (both developer and user) look realistic?
3. Are the necessary resources (both developer and user) available to meet this plan?
4. If relevant, are the procedures for hand-over to maintenance and support staff clear?
5. If relevant, have the requirements for data take-on and/or system cut-over been adequately considered?
6. Does the Functional Model match the users' needs as elicited during discussions and prototyping sessions?
7. Is it within the scope of the development as defined in the Business Area Definition?
8. Are all parts of the Functional Model mutually consistent?
9. Does the model contain the minimum usable subset?
10. Are all essential aspects of integrity and security contained within the Functional Model?
11. Are the requirements for system administration visible?
12. Are all static models (e.g. data models) consistent with the Functional Prototype, and vice versa?
13. Does the model give confidence that the right levels of performance, capacity and maintainability will be achievable?
14. Is any necessary supporting documentation available and to an adequate standard
15. Does the system satisfy all the user-defined acceptance criteria?
16. Are the developers satisfied that the system is sufficiently robust to be put into full operation?
17. Has the system been tested at an appropriate level, considering its intended use?

18. Is there evidence that all the essential requirements (functional and non-functional) have been tested and, where necessary, demonstrated to the users?
19. Have any and all safety-related and product liability aspects of the system been properly validated?
20. Has all functionality that is provided to support implementation been adequately tested (in particular, has account been taken of any need for data conversion/uploading software)?
21. Are all components of the Tested System traceable to the Functional Model?
22. Are all components rejected in the design review documents omitted from the Tested System?
23. Is the system documentation consistent with the software?
24. Have tests been appropriately documented (for example does each test identify the requirements and business rules addressed by the test)?
25. If appropriate, have test specifications been reviewed?
26. Are records available to show that all required tests have been performed and that the user involvement in that testing is as required?
27. Have all problems noted during testing been properly identified and recorded?
28. Have regression tests been performed appropriately?
29. Do the Test Records contain sufficient detail to enable the tests to be run again in future?
30. Is user guidance available to users in an appropriate format (e.g. electronic documents, paper documents, and help facilities)?
31. Does the user guide offer a complete and unambiguous step-by-step guide to using the Delivered System?
32. Does the user guide cover all the functionality within the system as delivered?
33. Does the user guide explain how the system interacts with other systems, manual or otherwise?
34. Where there are different classes of user, does the user guide explain who should read what?

Deliverables that may be included :-

DSDM Product	**RUP Artefact**
Functional Model Implementation Strategy Test Record Functional Model Review Records Development Risk Analysis Report Tested System User Documentation	Iteration Plan Implementation Model Integration Build Plan Test Scripts Implementation Sub-systems Components Test Components Test Subsystems

Functional Model

Content Element	**Example Realisation**
To provide a cohesive demonstration of the functionality and data requirements to be met, including all currently known constraints.	The specification of the unit of construction. This may include an object and component model and use case realisations for a use case (functional requirement).
To demonstrate the feasibility of achieving the non-functional requirements.	In addition to the description of the realisation of the functional requirements a mapping to the overall non-functional requirements such as performance and security it made.

Implementation Strategy

Content Element	**Example Realisation**
To refine the project plan for the later stages of the development.	The project plan may be updated with a view to the actuals of development.
To define the costs and effort in more detail, enabling management to reassess the costs and benefits of the development.	

.

Implementation Model

Content Element	Example Realisation
The implementation model is a composite, comprehensive artefact which encompasses all artefacts needed to build and manage the system in the run-time environment.	Inclusion of source code, build descriptions, DB schemas and how they map to the overall topology of the system.

Test Procedures, Test Scripts and Test Results

Content Element	Example Realisation
The purpose of the Test Scripts is to implement and execute the test procedures in an efficient and effective manner.	Test procedures will map to the functional and non functional requirements and can be at many different levels within the software architecture.

RUP Transition / DSDM Implementation Phase

At the end of this phase is the fourth important project milestone, the **Product Release Milestone**. At this point, you decide if the objectives were met, and if you should start another development cycle. In some cases this milestone may coincide with the end of the inception phase for the next cycle.
The primary evaluation criteria for the transition phase involve the answers to these questions:

- Is the user satisfied?
- Are actual resource expenditures versus planned expenditures acceptable?

The above milestone maps to the DSDM quality criteria :-

1) Have any changes made to the Tested System been properly authorised, implemented and tested?
2) Does the system work as required in its target environment?
3) Does it appear to operate to the required service levels?
4) Are there any unforeseen problems in the system's placement in the target environment that remain unresolved?
5) Have all data loading and conversion activities been completed successfully?
6) Have all configuration items been properly archived?
7) Are all configuration items identified?
8) Is the correct version of each configuration item recorded?
9) Are all known outstanding problems recorded?

DSDM Product	RUP Artefact
Delivered System	Deployment Plans
Project Review Document	End User Support Materials
	Release Notes
Trained User Population	Training Materials
	Defects

User Documentation

Content Element	Example Realisation
To describe to the users how to use the Delivered System.	The Use Cases provide a good start point for this material.

Installation Deliverables

Content Element	Example Realisation
The purpose of the installation artefacts is to enable someone to install the product.	Description of component dependencies, install scripts and any issues associated with the installation process.

Release Notes

Content Element	Example Realisation
The purpose of the Release Notes is to describe the release.	Any known defects and problems will be described in this documentation.

Appendix Four - Glossary

During the work on the Inter-operability of DSDM and the Rational Unified Process (RUP) the varying terminology across the two processes became apparent. This Glossary aims to clarify the terminology referred to throughout the series of papers describing this work. In addition to the definition of a term, references to DSDM or RUP are included where appropriate.

Note: The HTML links to RUP will only work if this document is placed in the installation directory of the Rational Unified Process.

Definitions

General Definitions

Term	Definition
Deliverable	**Deliverables** are the things that are produced or used, during a project. Examples of deliverables include documents, models, prototypes, test plans and software components In the RUP a deliverable is known as an artefact. In the DSDM a deliverable is known as a product.
Dynamic Systems Development Method (DSDM)	DSDM provides a framework of controls for building and maintaining systems which meet tight time constraints and provide a recipe for repeatable Rapid Application Development (RAD) success. The framework not only addresses the developer's view of RAD but also that of all the other parties who are interested in effective system development, including the users, project managers and quality assurance personnel.
Joint Application Design (JAD)	A forum for knowledgeable and empowered staff from business and IT to make decisions and produce products through consensus, controlled and enabled by an impartial facilitator. Also known as a 'Facilitated Workshop'
Programme	A "Programme" is a linked set of projects focused on a common objective. e.g. business process re-engineering, or creating and exploiting components.
Rational Unified Process (RUP)	The Rational Unified Process is a Software Engineering Process. It provides a disciplined approach to assigning tasks and responsibilities within a development organisation. Its goal is to ensure the production of high-quality software that meets the needs of its end users, within a predictable schedule and budget. The Rational Unified Process captures many of the best practices in modern software development in a form that is tailorable for a wide range of projects and organisations.
Technique	A defined approach to a particular aspect of a project. Examples of techniques include, Iterative Development, Time boxing, Requirements Management. Note: Please refer to the Definitions in the Techniques table for specific explanations of each technique referenced.
Unified Modelling Language (UML)	The Unified Modelling Language (UML) is the industry-standard language for specifying, visualising, constructing, and documenting the deliverables of software systems (Visual Modelling). See www.rational.com/uml for more information

Definitions relating to Techniques

Technique	Definition
Architecture Driven	This technique provides a methodical, systematic way to design, develop and validate architecture. This approach contains specific activities aimed at identifying architectural constraints and, architecturally significant elements, as well as guidelines on how to make architectural choices. The focus is on proving the architecture early in the development process. RUP References: • Features: Architectural Emphasis • Core Workflow: Analysis and Design: Workflow Details: Architectural Design • Overview: Introduction: Key Concepts: What is Software Architecture? DSDM References: • Part 2 Topics: Chapter 13 DSDM Product Descriptions: PD4 System Architecture Description
Business Case Development	A Business Case provides the necessary information from a business standpoint, to determine whether or not it is worth investing in this project. This technique provides advice on defining the Business Case. RUP References: • Workers & Activities: Project Manager: Activity – Develop Business Case • Artefacts: Management Set: Business Case DSDM References: • Part 2 Topics: Chapter 13 DSDM Product Descriptions: PD1 Feasibility Report and PD3 Business Area Definition • Part 2 Topics: Chapter 15 DSDM Roles: Executive Sponsor and Visionary
Business Impact Analysis	In many cases the implementation of a Project will have a direct impact on the Business area affected by the Project. The project may materially affect Business processes, staff, locations, etc.; it might also indirectly affect other processes and systems peripheral to the main Project. For these reasons, this technique ensures that Business impacts are separately specified and agreed by all affected stakeholders. DSDM References: • Part 3 Taking DSDM Further: Chapter 26 The DSDM Business Environment
Configuration Management	A supporting process whose purpose is to identify, define, and baseline items; control modifications and releases of these items; report and record status of the items and modification requests; ensure completeness, consistency and correctness of the items; and control storage, handling and delivery of the items. (ISO) RUP References: • Core Workflows: Configuration & Change Management DSDM References: • Part 2 Topics: Chapter 22 Configuration Management Guidelines
Dependency Management	The purpose of dependency confirmation is to ensure that all activities upon which the project depends, but which are not under the direct management of the project, are fully agreed and understood. Very few projects are completely self-contained; there are invariably services and products that must be acquired from outside the project at some stage in its life. These dependencies must be discussed and agreed before a project plan can be put together to deliver the project solution.

Facilitated Workshops	A forum for knowledgeable and empowered staff from business and IT to make decisions and produce products through consensus, controlled and enabled by an impartial facilitator RUP References: • Work Guidelines: Requirements Workshop • Work Guidelines: Use Case Workshop • Work Guidelines: Brain Storming • Work Guidelines: Use Case Analysis Workshop • Work Guidelines: Business Object Modelling Workshop DSDM References: • Part 1 The Foundations: Chapter 8 Facilitated Workshops • Part 2 Topics: Chapter 23 Facilitated Workshops
Functional requirement driven project (opposite of Architecture-driven project)	Projects may be driven by one of the two often opposing sets of requirements. The first set of requirements are those that are derived from the business or user community (i.e. the functionality the system must provide). The second set of requirements are often defined by the management (both technical and commercial) of the organisation. Projects that are driven by the functional requirements may be tactical rather than strategic in scope and thus may deliver a different level of quality than their strategic contemporaries. DSDM References: • Part 1 The Foundations: Chapter 5 Project Culture • Part 1 The Foundations: Chapter 7 Timeboxing and Prioritisation
Metrics -based estimation / control	A metric is a measure of the development process e.g. number of classes, number of defects. Metrics provide us with the ability to measure a project to evaluate how close or far we are from the objectives we had set in our plan in terms of completion, quality, compliance to requirements, etc. Metrics can assist with estimating for new project effort, cost and quality, based on past experience. We also measure to evaluate how we improve on some key aspects of performance of the process over time, to see what are the effects of changes. RUP References: • Artifacts: Management Set: Measurement Plan: Guidelines: Metrics • Artifacts: Management Set: Measurement Plan • Core Workflows: Project Management: Concepts: Metrics • Workers and Activities: Project Manager: Develop the Project Plan • Core Workflows: Test: Concepts: Key Measures of Testing DSDM References: • Part 2 Topics: Chapter 17 Measuring and Estimating DSDM Projects
Milestone Reporting/ Progress Visibility	This technique monitors the progress of the project at defined stages referred to as 'Milestones'. Once a milestone is reached the objectives of the phase leading to the milestone are assessed to establish if the goals have been met. The subsequent course of action depends upon how well the objectives were met, but may involve continuation of the project as planned, corrective changes may be made to the plan, or the project may be cancelled. RUP References: • Core Workflows: Project Management: Concepts: Project Life Cycle • Core Workflows: Project Management: Concepts: Iteration and Release • Workers and Activities: Project Manager: Develop the Project Plan • Workers and Activities: Project Manager: Evaluate the Iteration

	• Artifacts: Management Set: Status Assessment DSDM References: • Part 1 The Foundations: Chapter 3 Overview of the DSDM process • Part 2 Topics: Chapter 11 The DSDM Development Process Framework (e.g. preconditions) • Part 2 Topics: Chapter 13 DSDM Product Descriptions (e.g. PD 14 Project Review Document)
Model Driven Development	Modelling helps the development team to gain a good understanding of the business and the developing system and can significantly aid communication. In understanding the problems, accurate models can be produced which reflect the realities of the business world. This technique uses these models to drive the development of the system. It provides assistance in the evolution of these models and the traceability defined between them. RUP References: • Artifacts: Reqs Set: Use Case Model: Guidelines Use Case Model • Artifacts: Design Set: Design Model: Guidelines Design Model • Artifacts: Design Set: Design Model: Guidelines Test Model • Artifacts: Implementation Set: Implementation Model: Guidelines Implementation Model DSDM References: • Part 2 Topics: Chapter 22 Modelling techniques used within DSDM
MoSCoW	MoSCoW is an acronym where the capital letters stand for Must have, Should have, Could have, Won't have this time. These provide a technique for prioritising requirements. RUP References (Note: The info in RUP looks at prioritising requirements generally) • Core Workflows: Req Management: Workflow Details: Managing the Scope of the System DSDM References: (Note: The info in DSDM looks specifically at using MoSCoW) • Part 1 Foundations: Chapter 7.2 The MoSCoW Rules
Prototyping	A technique where a component that is produced to assess whether or not the system will be fit for purpose. A prototype need not be complete and tested with respect to all its related functional and non-functional requirements, the aim is to try out some aspect of the project to prove its ability to meet the needs of the Users. Examples include architectural prototypes to prove some new technology and User interface prototypes to try out the User Interaction with the system. RUP references: • Core Workflows: Project Management: Concepts: Prototyping • Workers & Activities: User-Interface Designer: User Interface Prototyping DSDM References • Part 1: Chapter 9 Prototypes • Part 2 Topics: Chapter 12 Controlling Prototyping
Prioritised Testing	An approach to testing where the test activities are prioritised based on the business goals. Priority should be given to those system features that support: • Overall business process performance (i.e. business processing cycle times); • Large processing volumes (i.e. very frequently occurring events);

	• Labour intensive or complex business tasks; • The human computer interface, particularly if the computer system will be visible to the *customer's* customer (e.g. a front-office application, an Internet application or a kiosk). DSDM References: • Part 2 Topics: Chapter 23.3 The Testing Approach • DSDM White Paper on Testing
Quality Management	A technique which addresses the Quality aspects of the project. The "quality" of the computer system will often be defined in terms of the way in which that system meets or exceeds agreed upon capability and support required by the user. The aim of this technique is to ensure "fitness for purpose" of the system. RUP References: • Core Workflows: Test: Concepts: Quality • Core Workflows: Test: Concepts: Key Measures of Testing • Core Workflows: Project Management: Concepts: Metrics DSDM References: • Part 2 Topics: Chapter 19 Quality in DSDM • "Dynamic Systems Development Method and TickIT: Guidance to software developers using DSDM to meet the requirements of ISO 9001" (ISBN 0 580 27081 5)
Requirements Management	Requirements Management is a systematic approach to • eliciting, organising, and documenting the requirements of the system, and • Establishing and maintaining agreement between the customer and the project team on the changing requirements of the system. Where a requirement is a condition or capability to which the system must conform. RUP References: • Features: Requirements Management • Core Workflows: Requirements Management DSDM References: • Part 1 The Foundations: Chapter 2 The underlying principles (particularly principles 2 and 7) • Part 1 The Foundations: Chapter 3 Overview of the DSDM Process • Part 1 The Foundations: 7.2 The MoSCoW Rules • Part 1 The Foundations: Chapter 8 Facilitated Workshops • Part 2 Topics: Chapter 13 Product Descriptions PD3a Prioritised Requirements List • Part 2 Topics: 19.4 Non-functional requirements • Part 2 Topics: Chapter 23 Facilitated Workshops
Risk / Issue Management	A risk is an event that may occur to impact a project; an issue is an event that has actually occurred. The purpose of Risk Management is to actively control all the risks facing a project or the implementation of the solution it is delivering. This includes: • Identification of risks that may threaten the project. • Assessment of the impact of those risks. • Management of those risks through specific counter measures aimed at minimising the business impact as a result of the risks materialising RUP References: • Core Workflows: PM Workflow: Concepts: Risks • Artifacts: Management Set: Risk List: Guidelines • Workers & Activities: Project Manager: Identify Risks

	DSDM References: • Part 2 Topics: 18 Risk Assessment • Appendix A: The Suitability Filter • DSDM White Paper on Risk Management
Project Closure	This technique examines the formal close down of the Project. The key activities that should be included are: • Confirmation on whether the benefits defined in the Business Case are being achieved, and to set actions to achieve them if they are not being met (Business Benefit Review). • A review of the project processes in order to feed any lessons learned into future projects (Post Implementation Review). DSDM References: • Part 2 Topics: Chapter 13 Product Descriptions PD 14 Project Review Document
Risk Based Development	This approach focuses on reducing risk during the early stages of development. Risk drives the iteration plans; iterations are planned around addressing specific risks, attempting to either bound the risk or reduce it. The risk list is periodically reviewed to evaluate the effectiveness of risk mitigation strategies, which in turn drives revisions to the project plan and subsequent iteration plans. RUP References: • Core Workflows: PM Workflow: Concepts: Risks • Artifacts: Management Set: Risk List: Guidelines • Workers & Activities: Project Manager: Identify Risks
Stakeholder Involvement	A Stakeholder is an individual who is materially affected by the outcome of the project. This technique examines how to involve these individuals in order to ensure their needs of the project are defined and met. RUP References: • Core Workflow: Requirements Management: Workflow Details: Understand Stakeholder Needs DSDM References: • Part 1 The Foundations: Chapter 8 Facilitated Workshops • Part 2 Topics: Chapter 16 Project Planning • Part 2 Topics: Chapter 23 Facilitated Workshops
Suitability Filter	The Suitability Filter consists of a set of criteria for helping the practitioner to determine how suitable a project is to apply a DSDM or RUP Framework. The suitability factors present in the filter are based on the DSDM critical success factors and other project situational factors. At a second level the filter determines which RUP & DSDM techniques are appropriate to be used within the context of the chosen Framework. The criteria are intended as guidance material only. The Suitability Filter referenced is an extended form of the Suitability Filter as defined in DSDM: • Appendix A – The Suitability Filter
Time boxing	A period of time with a fixed end date in which a team produces, checks and agrees a deliverable (partial or complete) or set of such deliverables. DSDM has an overall time box for the project which contains nested time boxes for day-to-day management and control.

	DSDM References: • Part 1 Foundations: Chapter 7.1 Timeboxing
Team Dynamics	This technique focuses on achieving the best team structure for your development. The decision as to how to compose the team depends on both personalities and practicalities. It is important to give consideration to what mix of team will be most likely to produce a good result quickly. DSDM References: • Part 2 Topics: Chapter 14 DSDM Project and Team Structures
Use Case-Driven Development	A Use Case is a sequence of actions which returns a result of value to the Actor. An Actor is something external that interacts with the system. This technique takes the use cases defined during requirements capture and utilises them as the basis for the entire development process. For example, they are used as a basis for planning an iterative development. They form a foundation for what is described in user manuals. In analysis & design use-cases are realised in a design model. During test the use cases constitute basis for identifying test cases and test procedures. That is, the system is verified by performing each use case. RUP References: • Feature: Use Case Driven Development • Overview: Introduction: Key Concepts: Use Cases