ARCHITECTING PROJECT MANAGEMENT

for Enterprise Agility…

July 14 to 16, 2016,

NIMHANS Convention Centre, Bengaluru

RISK MANAGEMENT - DSDM

Organization Design For Enterprise Agility

Paper ID: PMIBC-16-1-002

Author: Ms. PONSASIKALA PONNIAH

CONTENTS

ABSTRACT

The purpose of this paper was to ascertain the extent to which risk management practices are incorporated into agile development projects. Businesses are increasingly operating in extremely turbulent environments necessitating the need to respond and adapt to change more quickly and improve overall time to market.

From an Automotive Systems Development perspective, this has triggered a new wave of development, the most notable of these being agile methods. A principle objective of agile methods is to reduce well-known risks and its methodology deployed for this research involved a case study of a change management consultancy firm dedicated to the use of the Dynamic Systems Development Method (DSDM).

Dynamic systems development method (**DSDM**) is an agile project delivery framework, primarily used as a software development method. DSDM originally sought to provide some discipline to the rapid application development (**RAD**) method. DSDM became a generic approach to project management and solution delivery. DSDM is an iterative and incremental approach that embraces principles of agile development, including continuous user/customer involvement.

The **MoSCoW** method is a prioritization technique used in management, business analysis, project management, and software development to reach a common understanding with stakeholders. The DSDM Consortium is a not-for-profit, vendor-independent organisation which owns and administers the DSDM framework. The principles in DSDM directs the team in the attitude they must take and the mindset they must adopt in order to deliver consistently.

Keywords: Risk Management, Agile Methods, DSDM (Dynamic Systems Development Method), RAD, MoSCoW

INTRODUCTION

In Automotive Systems Development [ASD], the rapidly growing use of agile methods shows the urgency of organizations to adapt to change at a more speedy and efficient pace. Agile methods are known for their use of iterative development, active user involvement and their acknowledgement of the need to incorporate changing system requirements and "focus on generating early releases of working products using mostly collaborative techniques".

This is a stark contrast to the traditional model for systems development which promotes "**elicitation and freezing of requirements in advance**" with no overlap between project phases of analysis, design and implementation.

A principal objective of agile methods is to reduce well-known risks associated with common ASD project failures by for example, accepting that requirements will change.

In 2001, a group of practitioners of light weight methods met to explore what was common to their approach to Software development. Arie Van Bennekum represented DSDM.

The Dynamic Systems Development Method is considered to be the first "**truly agile method**" where "it is preferred to fix time and resources and then adjust the amount of functionality accordingly".

This highlights the element of flexibility in DSDM with regard to adjusting system functionality where system requirements are open to change. However, no matter what the nature of change, there will always be associated risks involved.

The primary objective of this paper was thus to develop a better understanding of risk management practices in agile ASD projects and the level of formality with which these practices are executed. Specifically, this paper focuses on three main elements of risk management, namely risk identification, estimation and evaluation.

DETAILS OF THE PAPER

Agile Manifesto

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Principles

The strength of DSDM lies in its principles; it is recognised that a framework will not hold the answers for all situations, and solutions that adhere to the Principles will most likely be robust.

The Principles are:

- Active user involvement is imperative
- Teams must be empowered to make decisions
- The focus is on frequent delivery of products
- Fitness for business purpose is the essential criterion for acceptance of deliverables.
- Iterative and incremental development is necessary to converge on an accurate business solution
- All changes during development are reversible

- Requirements are baselined at a high level
- Testing is integrated throughout the lifecycle
- A collaborative and co-operative approach between all stakeholders is essential

Lifecycle

DSDM recommends a complete lifecycle starting with Pre-Project work all the way through to Post-Project. It recommends iterative and incremental working but is product centric rather than activity based:

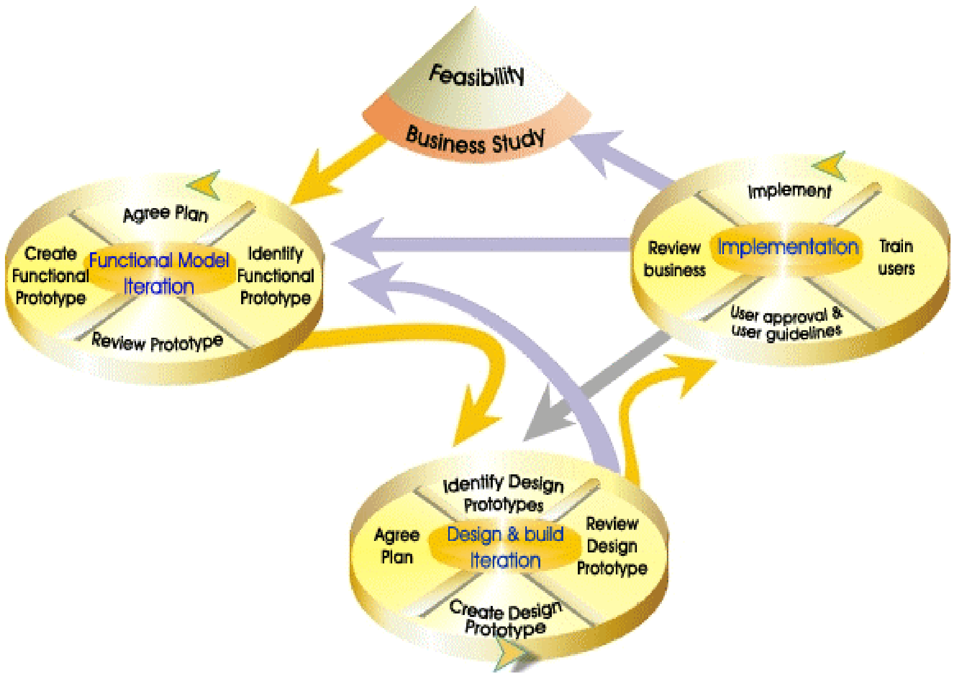


Figure 1: Lifecycle

DSDM Phase Descriptions

Pre-Project

Projects do not exist in a vacuum: they need to be set up correctly from the outset to ensure success. Pre-project activities will depend on local working practices for the initiation of projects. The evaluation and prioritisation of projects within an organisation's portfolio is beyond the scope of DSDM.

Key Products:

· Initial definition of the business problem to be addressed

· An outline scope for the investigation to take place during the Feasibility Study

· Decision to proceed with the project

· Visionary and Project Manager assigned to the project

· Initial plans for the Feasibility Study

· Confirmation of alignment of the project with the appropriate strategy

· Budget and resources allocated for the Feasibility Study at least and preferably the Business Study as well - with outline budget/resources approval for the development phases.

· The initial project governance in place, e.g. Project Board or Steering Committee

Feasibility Study

The objectives of the Feasibility Study are to establish whether a proposed development can meet the business requirements of the organisation; to assess the suitability of the application to DSDM development; to outline possible technical solutions to the business problem; and to obtain firstcut estimates of timescale and costs.

The Feasibility Study should only go to the level of detail required to assess whether a feasible solution exists or to select the most appropriate one. The detail of the requirements, risks, plans, costs, etc. for the solution will be developed in the later phases.

Key Products:

· Feasibility Report

· Feasibility Prototype (optional)

· Outline Plan

· Risk Log

Business Study

The objectives of the Business Study are to scope the business processes to be supported; outline the future development in terms of prototyping deliverables (defining which are incremental and which, if any, are throwaway) and prototyping controls; identify representatives of the user classes for prototyping activities; prioritise the requirements of the proposed system; reassess the risks of the project; provide a firm basis for technical development to proceed; and scope the non-functional requirements, in particular to decide the maintainability requirements.

Key Products:

· Business Area Definition

· Prioritised Requirements List

· Development Plan

· System Architecture Definition

· Updated Risk Log

Functional Model Iteration

The objectives of the Functional Model Iteration phase are to demonstrate the required functionality using a functional model consisting of both working software prototypes and static models (e.g. class models and data models); and record the non-functional requirements which may not be demonstrated by the working prototype.

Key Products:

· Functional Model including Functional Prototypes

· Non-functional Requirements List

· Functional Model Review Records

· Implementation Plan

· Time box Plans

· Updated Risk Log

Design and Build Iteration

The objectives of the Design and Build Iteration Phase are to refine the Functional Prototypes to meet the non-functional requirements; and engineer the application so that it demonstrably satisfies user requirements.

Key Products:

· Time box Plans

· Design Prototypes

· Design Prototyping Review Records

· Tested System

· Test Records

Implementation

The objectives of the Implementation phase are to place the Tested System in the users' working environment; train the users of the new system; determine the future development requirements; and train operators and support staff.

Key Products:

· User Documentation

· Trained User Population

· Delivered System

· Increment Review Document

Post-Project

The Post-Project phase contains the activities that occur once the project team have disbanded. These include support and maintenance activities and (optionally) a Post-Implementation Review to assess the system in use.

The objectives are to keep the Delivered System operational; assess whether or not the proposed benefits of the project as stated during its initial phases have been achieved; enable development processes to improve; and review the Delivered System in use.

Key Products:

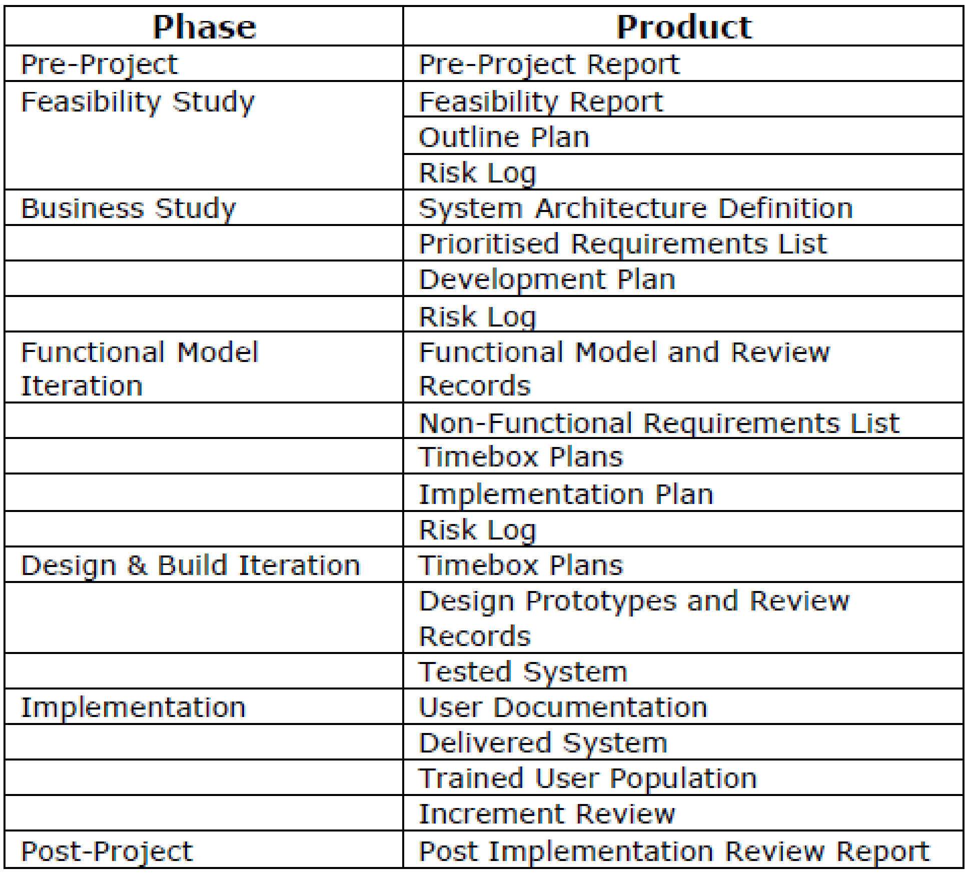
· Post-Implementation Review Report

· Change Requests

· New releases of the Delivered System in response to Change Requests

Products

The phase products relevant to the purposes of this White Paper are:



Phase	Product
Pre-Project	Pre-Project Report
Feasibility Study	Feasibility Report
	Outline Plan
	Risk Log
Business Study	System Architecture Definition
	Prioritised Requirements List
	Development Plan
	Risk Log
Functional Model Iteration	Functional Model and Review Records
	Non-Functional Requirements List
	Timebox Plans
	Implementation Plan
	Risk Log
Design & Build Iteration	Timebox Plans
	Design Prototypes and Review Records
	Tested System
Implementation	User Documentation
	Delivered System
	Trained User Population
	Increment Review
Post-Project	Post Implementation Review Report

Figure 2: Phases of development

People

DSDM recognises that development of any kind within a business involves people; those that have to do the development and those that are impacted by change.

DSDM advice includes a full list of roles and, more importantly, responsibilities, including representatives of the end-user community. If this information and the advice given in the Techniques section below is followed, all stakeholders in the outcome of the development will be involved at the appropriate time.

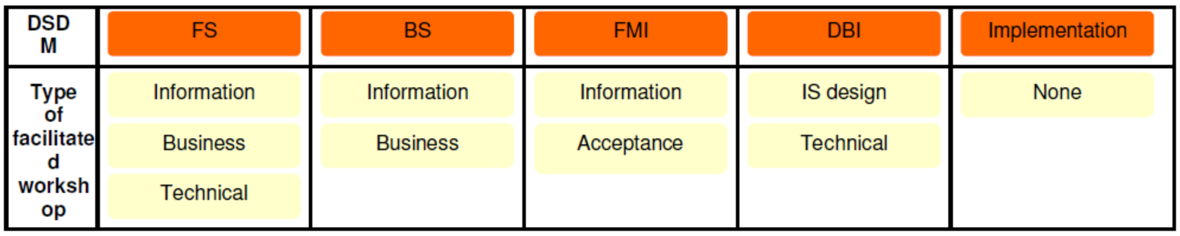
Techniques

There are many techniques that may be used during any development and any that conform to the DSDM Principles may be applied on a DSDM project. The following paragraphs list with explanations the techniques that DSDM particularly recommends for use

Facilitated Workshops

Facilitated workshops are an important technique in keeping a project moving along quickly and in the right direction both in business and technical terms. They are used throughout DSDM projects to both create products and make decisions quickly incorporating a wide range of stakeholder views

1. IS requirement definition (Information)
2. Business information benefits (Business)
3. Technical system operation (Technical)
4. Acceptance test planning (Acceptance)
5. IS design



DSDM	FS	BS	FMI	DBI	Implementation
Type of facilitated workshop	Information	Information	Information	IS design	None
	Business	Business	Acceptance	Technical	
	Technical				

Figure 3: Facilitated Workshop

Prototyping

Prototyping is fundamental to successful DSDM projects. This technique enables real user involvement in development activities from a very early stage and thereby enables the users in the project team to steer the developers in the right direction needed to achieve maximum business benefit. Prototyping encourages creativity. However, it needs to be controlled

Modelling

Guidance is provided about what to model, when and how, while enabling each organisation and project to determine the best set of models to produce in particular circumstances: one size does not fit all!

Configuration Management

Configuration Management is a key factor in managing the evolving products (both software and documents) that are created throughout the DSDM lifecycle from project initiation to the completion of the final delivery.

The DSDM techniques to weight importance are the MoSCoW rules:

Must have, Should have, Could have & Want to have

Support Environments

The support environment is an important factor in enabling fast delivery:

A poor toolset or a non-interoperable technology infrastructure will hinder the creativity of developers within the teams.

Risk Management

In an ASD context, Boehm highlighted the concept of managing risks and giving them priority as far back as 1988. Ten years later, Hall described Boehm as being "the father of software risk management." Boehm proposed a move away from the '**staged' SDLC** to a more iterative or incremental process and this proposed concept in software development was an attempt to lower project risks. Boehm's aim was to eliminate any software difficulties or risks mainly by deriving "**risk-driven documents**" and "incorporating prototyping as a risk-reduction option". It resulted in what was called the **Spiral Model** that essentially created "a risk-driven approach to the software process rather than a primarily document-driven or code-driven process".

Many associations can be made between Boehm's proposals above and that which has developed in the approaches deployed by agile methods. While many authors highlight distinct approaches and frameworks for dealing with risk management (ranging from formal to informal), its basic fundamentals remain the same and are consistent across disciplines. The early practitioners of risk management – who outline the three main elements of risk assessment as (i) Risk Identification, (ii) Risk Estimation and (iii) Risk Evaluation.

Risk Identification

Risk identification is the reduction of descriptive uncertainty which involves "**surveying the range of potential threats**". This element of risk assessment involves detecting issues which could jeopardize or threaten the success of a project. Chapman states that "**the risk identification and assessment stages have the largest impact on the accuracy of any risk assessment**." It is therefore the most important stage of risk management. Of particular importance to ASD is the early identification of risks where "identifying and dealing with risks early in the development lessens long-term costs and helps prevent software disasters". Furthermore, it is important that risk identification carries on throughout the project's lifecycle. Therefore, risk identification is an ongoing, continuous process that requires regular screening and monitoring.

An important aspect of risk identification is categorizing the risks organizations encounter. According to Coppendale, "**depending on the size and the complexity of the project there might be between five and fifteen**" categories of risk. Categories attempt to group certain types of risk under a particular heading and in doing so "can help you find global risks that can be solved together". There are many sources of risk, some of which include senior management, the client or customer, the project team, organization of the project itself and even laws and standards which directly impact the project. Such sources can be placed into their respective categories as being either internal or external risk as follows:

	Risk Category	
	Internal	**External**
Sources of Project Risk	Senior Management	Acts of Nature
	Project Team & Management	The Client
	Organisation of the Project	Laws and Standards

Table 1. Categories and Sources of Project Risk

The risk factor is calculated by the severity and impact of the issue.

Risk Factor = Probability * Impact

The Impact can be from [0 to 4]

The Probability can be from [0.1 to 0.9]

The two most dominant sources of internal risk identified across the literature are Senior/Project Management and Project Team. The **dominating external source of risk is the client**. Some internal risk factors include project conflict and resource boundaries, which can be linked to sources of project team and senior management risk respectively. The following table shows some of the most dominant risk factors identified by Wiegers, who categorizes these factors by sector:

Project Sector	Risk Factor	% of Projects at Risk
MIS	Creeping User Requirements	80
	Excessive Schedule Pressure	65
	Low Quality	60
	Cost Overruns	55
	Inadequate Configuration Control	50
Commercial	Inadequate User Documentation	70
	Low User Satisfaction	55
	Excessive time to market	50
	Harmful competitive actions	45
	Litigation expense	30

Table 2. Most common risk factors for various project types

On analysing the table, some direct link between sources and risk factors is evident. For example, the sources of 'management' risk or 'organisation of the project itself' could be linked to the risk factor of 'inadequate configuration control' due to a flaw in the project's arrangement and organisation. However, the most interesting correlation is that of the ten risk factors listed in Table 2, at least six of these can be linked to the client as a source of project risk.

Finally, a dominating feature in recent literature deserves recognition where there is strong support among authors that an actual 'source of risk' can provide a '**source of opportunity**.' Chapman and Ward state "it is only once risk is seen as a good thing people begin to look for opportunities." Very few people would acknowledge 'opportunity' as being a facet of risk as naturally there are negative connotations associated with risk.

In a general sense the above ideas represent something we all know and understand about risk and the nature of taking gambles – people and organisations usually undertake risks with the aim of benefiting from potential opportunities . Taking on any form of risk can be a daunting task but as DeMarco & Lister state, "**if a project has no risks, don't do it**."

Risk Estimation

At this stage it is hoped that the project team have identified all potential risks and they can now move on to estimating those risks. Risk estimation is the reduction of measurement uncertainty where "the values of the variables describing the system are determined, the various consequences of an event occurring are identified" and finally, "the magnitude of the risk is determined".

In ASD environments there are many generic risk factors (such as creeping user requirements) but very few instances of projects operating under similar circumstances. Therefore, allocating future estimates is undoubtedly different to those allocated for past events as there are so few exact comparable projects conducted in the past.

Risk estimation attempts to estimate "the chance (or probability) of potential loss" as well as "the exposure to potential loss i.e. the consequences or magnitude of the identified risks".

The chance of potential loss is essentially the process of attaching a probability of occurrence to any identified risk. As Hall states "estimation is the appraisal of risk probability and consequence." Probability is categorized as being greater than zero and less than one hundred while consequence is decided relative to cost, schedule and technical goals. If an event is certain to occur it has a probability of exactly one or one hundred percent. According to McManus, probability data should be used to compute the risk. When no actual data on probabilities exist, estimates by individuals most familiar with the project, its risk factors and overall problems are a good substitute

Key Success Factors

To determine the important factors management needs to control prior to starting and during DSDM practice is crucial to the project's success. The DSDM consortium has compiled 10 most important factors from its member experiences:

1 Acceptance of the DSDM philosophy before starting work

2 The decision making powers of users and developers inside the development team.

3 The commitment of senior user management to provide significant end user involvement.

4 Incremental delivery.

5 Easy access by developers to end-users.

6 The stability of the team.

7 The development team skills

8 The size of the development team.

9 A supportive commercial relationship.

10 The development technology

DSDM requires a fairly advanced corporate culture to be successful in the long run. Some success factors might not even apply to many projects, i.e. "**supportive commercial relationships**" are not usually present in Open Source Projects.

CONCLUSION

The DSDM framework is a straight forward framework based on best practices to start implementing a project structure, its strengths being simplicity, extendibility, proven in the past but not claiming to be the solution to all kind of projects. The fact that the very authors of DSDM issue a warning concerning the single-minded use of DSDM where it does not suite the purpose provides much more **credibility to the concept** then many other similar methodologies.

The weakness of DSDM is, like with many other structured approaches, the relatively high barrier to entry (apart from the licensing costs). Switching to DSDM is neither cheap nor fast, and requires a significant cultural shift in any organization, because suddenly deliverables are replaced with tasks. I suppose for many people accepting the fact that their project will be on time and on budget, but not necessarily delivering the initially requested functionality will be hard to accept.

DSDM clearly presents itself as the most mature agile development method, while many agile methodologies are rather **programming methodologies then process models**. The select few agile methodologies share common ground with DSDM, ie. SCRUM which as well as DSDM, promotes team empowerment. The Crystal Methodologies in turn assume to be a good collection of best practices as well, whereas DSDM does a much more comprehensive job in showing its evolving character by versioning their framework after every revision by the DSDM consortium.

Shell, Loyds Bank Insurance Services, British Telecom, British Airways, Deutsche Bahn, Hewlett-Packard, Renault, the city of Los Angeles; the list of companies using DSDM is long. Many other organizations may already use DSDM, but are not yet ready to publicly commit to DSDM, even the worlds largest software vender, Microsoft, borrows ideas from DSDM when publishing his own semi-agile solution framework currently in version 3.1 complete with 25 000 certified practitioners.

Agile methodologies are very visibly on the rise, as well as another promising concept, Offshoring. Xansa used DSDM to organize its offshore development in India; ThoughtWorks has been using offshoring to India with agile methods as well, proving that the highly profitable practice of offshore development and onshore engineering works with agile methods, good enough to present a business case.

Further research into agile methodologies and offshoring could provide actionable evidence for business which IT and development efforts they should relocate and whether agile methods are supporting the change.

REFERENCES

1. Boehm, B. W.: Software Risk Management: Principles and Practices, IEEE Software 8(1), 32--41 (1991)

2. Chapman, C., Ward, S.: Project Risk Management: Processes, Techniques and Insights, John Wiley & Sons Ltd. (1997)

3. Chapman, R. J.: The Effectiveness of Working Group Risk Identification and Assessment Techniques, International Journal of Project Management, 16(6), 333--343 (1998)

4. Coppendale, J.: Manage Risk in Product and Process Development and Avoid Unpleasant Surprises, Engineering Management Journal, 5(1), 35--38 (1995)

5. Grey, S.: Practical Risk Assessment for Project Management, John Wiley & Sons Ltd. (1995)

6. Hall, E. M.: Managing Risk: Methods for Software Systems Development, Addison Wesley (1998)

7. Smith, P., Pichler, R.: Agile Risks/Agile Rewards. Software Development, 13(4), 50--53 (2005)

8. Wiegers, K. E.: Know Your Enemy: Software Risk Management, Software Development, 6(10), 38--42 (1998)