Method engineering 2016

Assignment D: Draft Paper

Dynamic Systems Development Method (DSDM)

Name: Kariem Slegten

Studentnumber: 5767202

TA: Daniel Alami

Introduction

The Dynamic Systems Development Method (DSDM) is a framework for developing software in an agile way (*aydal, 2005*). It has clearly defined phases, sub-phases, roles and principles that enable software development teams to work efficiently. It became a popular way for software developers to work in the 1990's and bears many similarities to other agile methods such as Scrum and Extreme Programming. This is also the reason that it has not been a very active study subject recently, as most researchers and businesses prefer to work with Scrum. The DSDM framework has three main phases: pre-project, project-lifecycle and post-project. These phases can be divided into four phases of the project status:

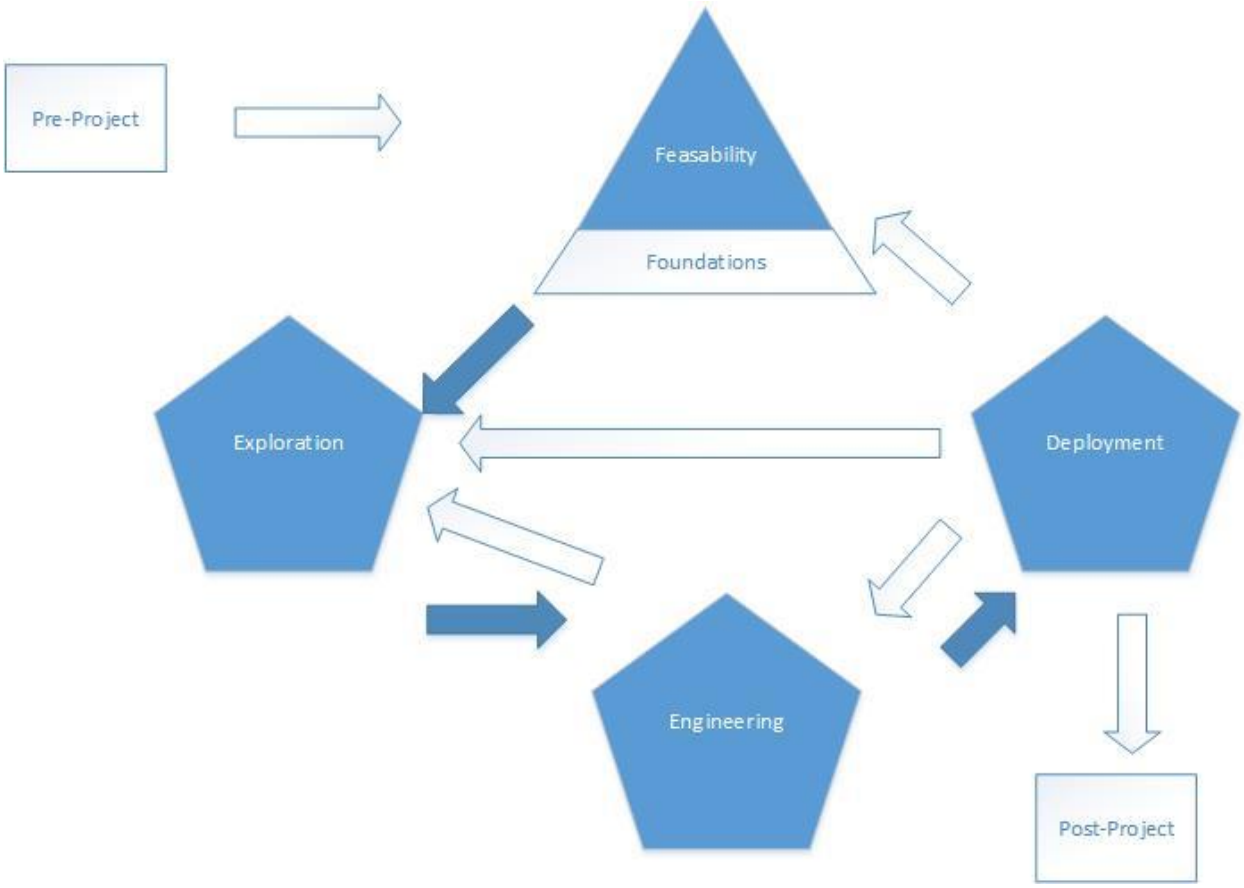


Figure 1: DSDM lifecycle (after DSDM HANDBOOK,2008)

DSDM subscribes to many agile principles like active user involvement, empowered teams and frequent delivery of products. But there is more focus on the techniques used in DSDM development. Some examples of these are MoSCoW prioritization, prototyping and Facilitated workshops. An important technique used is Timeboxing: a set interval for software development iterations after which deliverables have to be presented, preferably in the form of working software. This is similar to sprints in Scrum.

DSDM was first created in 1995 by a consortium that wanted to explore different ways develop software. The DSDM Consortium is a non-profit organization that promotes the DSDM method. Some early revisions were done in 1995 and 1996 after which the method was used all of the world. In 2007, the DSDM consortium released DSDM ATERN 4.2.(DSDM REFERENCE) This method contains much of the original method but incorporates elements of modern software development.

Example

Since no real examples are readily available; this chapter will describe an example of a generic software development company implementing DSDM as their main method for software development. It is very important that a few roles are filled. The executive sponsor is someone on the board that act as the project champion, in the case of the example this is the CIO of the company. There is also a visionary, someone who is responsible for the early identification of essential requirements and for the initialization of the project. This is one of the product managers of the company. An ambassador user is someone with a deep understanding of the target audience that provides the development team with feedback. This is an outside consultant who has a lot of knowledge of the user base. Apart from the developers and testers, there is also the role of project manager, someone who is responsible for the successful completion of the project and day-tot-day management.

So, a feasibility study is performed and after the business study is complete, the company has identified high-level functional and non-functional requirements and designed a software architecture and a development plan. The project manager and the CIO are largely responsible for this phase. They create a deliverable called the feasibility assessment and it details a high-level costs and benefits evaluation. In the second phase the project leaders and the project development team work closely with the outside consultant to develop a functional model and make sure that they have all areas of attention covered. Once development commences in the third phase, the roles of the visionary and executive sponsor are reduced. The project is developed in iterations and the outside consultant is constantly involved to make sure that the software meets the business needs of the intended users. During the final stage, the finished system is presented to the users and feedback in incorporated into the system in several iterations.

Concerning deliverables, each phase has certain documents or models that have to be delivered. Some examples are a feasibility report, feasibility prototype, global outline plan, functional model, user documentation and a functional prototyping review document. One of the most important deliverables the design prototype. This is a basic version of the software stripped of all but the bare functionalities that are required. This is usually created in the first few time-boxes in order to test the system and show the stakeholder working software.

Related literature

The DSDM Method was created in 1995(*Thelen & Smith, 1996*) by a non-profit group in the UK. It is mainly based on the theory of Rapid Application Development (Martin, 1991) and a full evolutionary model including predecessors and successors can be seen in Figure 2. The DSDM method came about mainly through the DSDM consortium and not through scientific literature, although a very good book has been published containing the basic principles of DSDM (*Stapleton, 1997*). The DSDM Consortium has also published a very good handbook (*DSDM Consortium, 2008*), containing the DSDM Atern 4.2 method. In 2001, the DSDM consortium subscribed to the agile manifesto (*Fowler & Highsmith 2001*), and as part of the so called Agile Alliance, they dedicated themselves to four principles: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan.

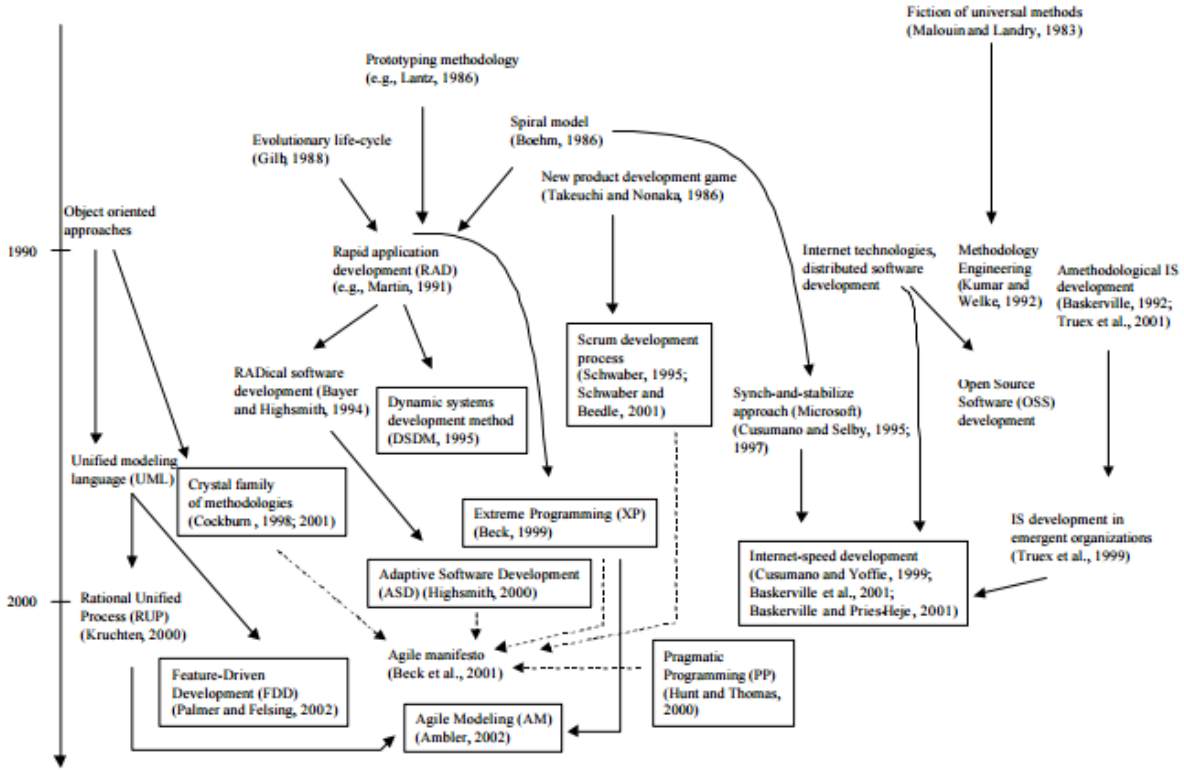


Figure 2: Evolution map of agile methods. (*Abrahamsson et al, 2003*)

DSDM often pitted against other agile methods to do comparative analyses (*Qumer & Henderson-Sellers, 2008*)(*Dybå & Dingsøyr, 2008*). These papers find that although the tools used in DSDM are not as powerful as in Scrum for instance, the DSDM method does pose a more complete method because it contains very specific roles and principles. Next to the main paper discussed in this paper, there is a recent paper that investigates the applicability of DSDM in UX design (*Plonka et al, 2014*). The paper concluded by stating two new roles that would have to be added to DSDM to make it more compatible with UX design.

The main paper that is reviewed in this paper reviews security aspects in the DSDM method(*Sani et al*, *2013*). They give a thorough description of the DSDM method, software development security principles and conventional security attacks. They then proceed to do a systematic literature review to check in how many papers these security principles are mentioned in relation to DSDM. They concluded that there is no need to extensively research the security aspects of DSDM since there is only one paper that mentions this as a problem. They mention in their future work that they intend to release a secure DSDM model anyway, based on their own experiences with DSDM.

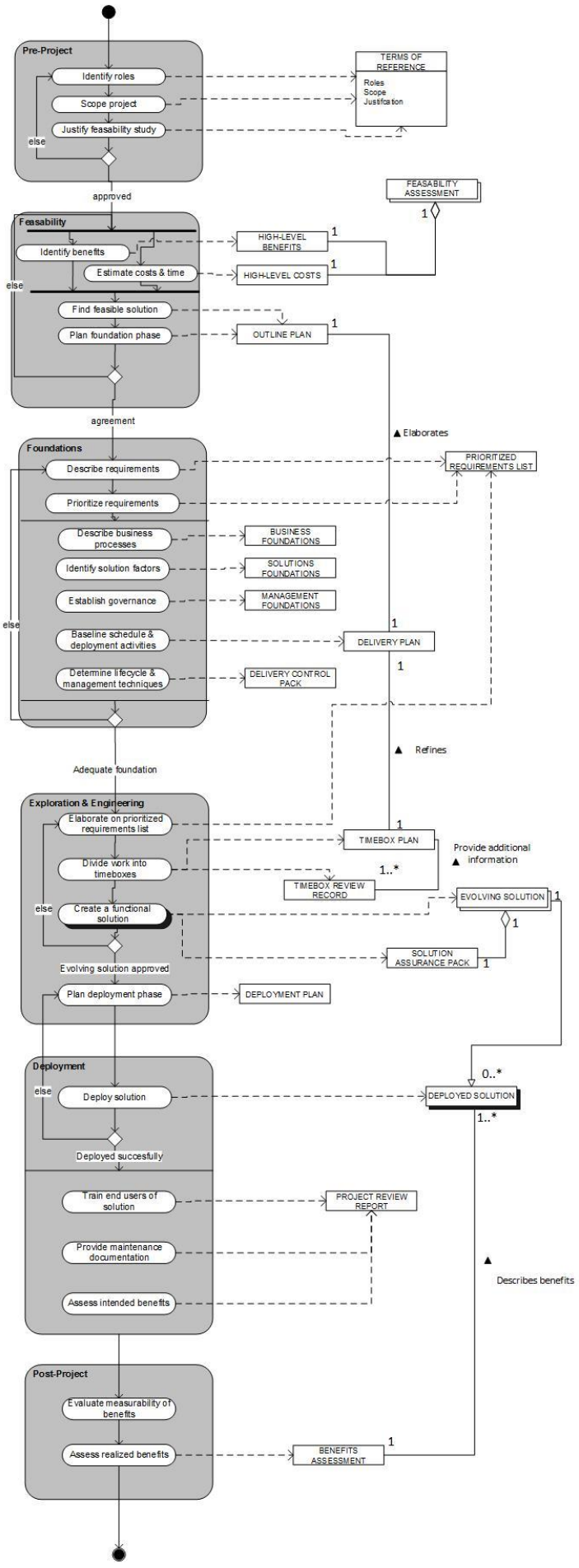
Process-deliverable diagram

A process-deliverable diagram is a model of a technique or a method that links development processes to products that are to be produced. On the left hand side activities and sub activities are modelled in order of which they occur (*Weerd & Brinkkemper, 2008*). It is also possible to indicate decisions points of how to continue and a clear project start- and end state are given. On the right hand side deliverables are modelled and their relations are stated using the UML class diagram notation. It is possible to create Open, Closed and Standard concepts depending on the nature of the deliverable. Each deliverable has to be linked to an activity or sub activity.

DSDM PDD

Shown in Figure 3 is the PDD for the DSDM method. It states five main phases with subsequent sub activities, these are listed in Table 1. These sub activities produce a number of deliverables, these are explained in Table 2. The phases and concepts are derived from the DSDM ATERN handbook (*DSDM Consortium, 2008*).

Figure 3: PDD of the DSDM



Phase	Sub-activity	Description
1. Pre-Project	1.1 Identify roles	Identify a Business Sponsor and Business Visionary. Other roles such a project manager and development team also need to be identified.
	1.2 Scope project	Set a business scope in which the project will operate. These are captured in the TERMS OF REFERENCE.
	1.3 Justify feasibility study	The feasibility study phase has to be planned, scoped and resourced.
2. Feasibility	2.1 Identify benefits	Identify the benefits likely to arise from the proposed solution. Sub activities 2.1 and 2.2 make up the FEASABILITY ASSESSMENT.
	2.2 Estimate costs & time	Deliver first-cut estimates of costs and timescale for the project.
	2.3 Find feasible solution	Outline possible solutions including strategies for sourcing and project management.
	2.4 Plan foundation phase	To plan and resource the foundation phase, described in the OUTLINE PLAN.
3. Foundations	3.1 Describe & prioritize requirements	Formulate high-level requirements and describe their priority and relevance to the business need. These are stipulated in the PRIORITEZED REQUIREMENTS LIST.
	3.2 Describe business processes	Describe the business processes to be supported by the proposed solution. Described in the BUSINESS FOUNDATIONS.
	3.3 Identify solution factors	Identify information used, created and updated by the proposed solution. This leads to the SOLUTIONS FOUNDATION.
	3.4 Establish Governance	Establish appropriate governance and organization for the project. Described in the MANAGEMENT FOUNDATIONS.
	3.5 Baseline schedule & deployment activities	Baseline a schedule for development and deployment activities for the solution. Makes up the DELIVERY PLAN.
	3.6 Determine lifecycle & management techniques	Describe the solution development lifecycle for the project along with techniques to be applied in managing the project and for the communication process. This information is stated in the DELIVERY CONTROLL PACK.
4. Exploration & Engineering	4.1 Elaborate on prioritized requirements list	Elaborate on the prioritized requirements captured and described in the foundations phase.
	4.2 Divide work into timeboxes	Divide work on requirements into timeboxes, which will be the foundation of the development process. Involves both the TIMEBOX PLAN as the TIMEBOX REVIEW RECORD.

	4.3 Create a functional solution	According to the TIMEBOX PLAN, create a functional solution that demonstrably meets the needs of the business. The results are an EVOLVING SOLUTION and a SOLUTION ASSURANCE PACK.
	4.4 Plan deployment phase	Consider all elements to deploying the evolving solutions and plan accordingly. Described in the DEPLOYMENT PLAN.
5. Deployment	5.1 Deploy solution	Deploy the solution (or increment of it) into the live business environment. This newly created entity is called the DEPLOYED SOLUTION.
	5.2 Train end users of solution	Train the end users of the solution and/or provide documentation that support the live operation of the solution in the business environment. The results of sub activities 5.2, 5.3 and 5.4 are all captured in the PROJECT REVIEW REPORT.
	5.3 Provide maintenance documentation	Provide documentation and/or training to technical staff that will be responsible for maintenance on the solution.
	5.4 Assess intended benefits	Assess whether the DEPLOYED SOLUTION is likely to enable the delivery of intended elements of business benefit.
6. Post-Project	6.1 Assess realized benefits	To assess whether the benefits described in the FEASABILITY ASSESSMENT have been achieved through use of the DEPLOYED SOLUTION. This report is called the BENEFITS ASSESSMENT.

Table 1: Activity Table for the DSDM method

Concept	Description
TERMS OF REFERENCE	A high-level description of the business with a proposed solution including scope, intended roles and a justification for the feasibility phase. Most likely a short one- or two page document but can also a an email or a verbal agreement.
FEASABILITY ASSESSMENT	This document provides a high-level overview of expected costs, benefits and risks of the project. Details these attributes from a business and technical perspective to determine feasibility. Made up from HIGH-LEVEL BENEFITS and HIGH-LEVEL COSTS.
HIGH-LEVEL BENEFITS	An estimation of the expected benefits of the project.
HIGH-LEVEL COSTS	An estimation of the expected costs of the project.
OUTLINE PLAN	An overview of the project from a project management and solution delivery perspective. Also details the plan for the foundation phase.
PRIORITEZED REQUIREMENTS LIST	This list details all the requirements that need to be addressed in order for the business need to be met. This is a prioritized list that is constantly updated throughout the foundations and exploration & engineering phases.

BUSINESS FOUNDATIONS	This provides information about the business that is fundamental to the success of the project and needs to be understood by all relevant stakeholders.
SOLUTION FOUNDATIONS	This document provides information about the solution that is relevant to the success of the project and needs to be understood by all relevant internal stakeholders. It contains elements like business area definition, system architecture definition, development approach definition and a solution prototype.
MANAGEMENT FOUNDATIONS	This product describes the essential governance and organizational aspects of the project. It also describes how the DSDM principles are applies throughout the project.
DELIVERY PLAN	The DELIVERY PLAN elaborates on the schedule as outlined in the OUTLINE PLAN.
DELIVERY CONTROL PACK	This is a live document that contains information about the status of the project, comprising of documents, logs and reports from the project management.
TIMEBOX PLAN	This elaborates on the objectives provided for each timebox as stated in the DELIVERY PLAN.
TIMEBOX REVIEW RECORD	The TIMEBOX REVIEW RECORD is a record where information is stored concerning the execution of the intended timeboxes. Concerns progress, feedback and outstanding issues.
EVOLVING SOLUTION	The nature of the concept depends entirely on the progress of the project. It can be a rough sketch in early stages but near the end of the project it can be a fully operational software application with all required documentation.
SOLUTION ASSURANCE PACK	This is a collection of elements that support the solution such as test units and review records.
DEPLOYMENT PLAN	The DEPLOYMENT PLAN is a detailed plan for the deployment phase. It tends to focus on individuals rather than on products to be delivered by the solutions team.
DEPLOYED SOLUTION	An instance of the EVOLVING SOLUTION set in a live business environment.
PROJECT REVIEW REPORT	This is a constantly evolving report that is updated at each increment. It contains an increment review record, a benefits enablement summary and an end of project assessment.
BENEFITS ASSESSMENT	The BENEFITS ASSESSMENT details how the solution has been used to realize benefits and what these benefits are. In cases where benefits are gained over a longer period of time, the assessment is updated periodically.

Table 2: Concept table for the DSDM method

References.

Aydal E.G. 2005. Extreme programming and Refactoring for Building Secure Web based Applications and Web Services, MSc Project, University of York.

Gregory, P., Plonka, L., Sharp, H., Taylor, K., & Keynes, U. K. (2014, June). Bridging the gap between research and practice: the Agile Research Network. In *Proceedings of the 13th European Conference on Research Methodology for Business and Management Studies: ECRM 2014* (p. 161). Academic Conferences Limited.

Stapleton, J. (1997). *DSDM, dynamic systems development method: the method in practice*. Cambridge University Press.

DSDM Consortium. (2008). DSDM Atern: the Handbook. *DSDM Consortium*.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003, May). New directions on agile methods: a comparative analysis. In *Software Engineering, 2003. Proceedings. 25th International Conference on* (pp. 244-254). Ieee.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, *9*(8), 28-35.

Thelen, E., & Smith, L. B. (1996). *A dynamic systems approach to the development of cognition and action*. MIT press.

Martin, J. (1991). *Rapid application development* (Vol. 8). New York: Macmillan.

Qumer, A., & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering.*Information and software technology*, *50*(4), 280-295.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*,*50*(9), 833-859.

Plonka, L., Sharp, H., Gregory, P., & Taylor, K. (2014). UX design in agile: a DSDM case study. In *Agile Processes in Software Engineering and Extreme Programming* (pp. 1-15). Springer International Publishing.

Sani, A., Firdaus, A., Jeong, S. R., & Ghani, I. (2013). A review on software development security engineering using dynamic system method (DSDM).*International Journal of Computer Applications*, *69*(25).

van de Weerd, I., & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. In M.R. Syed and S.N. Syed (Eds.), *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications (pp. 38-58)*. Hershey: Idea Group Publishing.