Dynamic Systems Development Method and Rapid Application Development

Dariusz Gall
School of Engineering
Blekinge Institute of Technology
daga03@student.bth.se

Richard Berntsson Svensson
School of Engineering
Blekinge Institute of Technology
dj20p@ny.com

Jimmie Johansson
School of Engineering
Blekinge Institute of Technology
jj76@spray.se

Abstract

When developing software it is a huge difference in which process model to follow. Different model is suitable for different projects and there different purposes. In this paper we will present the Dynamic Systems Development Method and the characteristics of this method as well as strengths and weaknesses and to what kind of project this method is suitable for. A small summary of the Rapid Application Development method is also presented together with similarities and differences from the Dynamic Systems Development Method. The result shows that the method is more suitable for in-house and market-driven development rather then contract-driven development. The major weakness with the method that we have found is that it is hard to plan and that the requirements are loosely defined. The major strengths are that the end-products tend to fit the business need as the user is much involved in the development process. This method also has a short time to market for development.

1. Introduction

This report is a part of an assignment in the course Project Management given by Blekinge Institute of Technology. The purpose with the assignment is to identify a development method and present the method. The presentation of the selected method consists of a summary, the characteristics of the method, a discussion of why the method have a specific focus were we relate to the book, Project Management for Business and Engineering and our own experiences. Further more the presentation consists of in which type of project this method is most suitable for and a discussion of strengthens and weaknesses of the method.

The rest of this report is organized as follow. In section a summary of Rapid Application Development is done. In section 3, a summary of Dynamic Systems Development Method is presented. First in the summary are the nine principles. Then the process overview is presented and followed by the roles in a Dynamic Systems Development Method project and finally a comparison with a more traditional project is done. In section 4 the characteristics of Dynamic Systems Development Method is presented together with our opinion why the method have this focus. The issues also relate to the course literature. The characteristics that are presented are, organizational and structure, systems and procedures and quality assurance. Section 5 covers our opinion of which type of project Dynamic Systems Development Method is suitable for. Section 6 presents strengthens and weaknesses of the Dynamic Systems Development Method. Finally in section 7, the conclusion is made.

2. Rapid Application Development – Summary

Rapid Application Development is an approach utilized in the field of Software Engineering. Rapid Application Development is a software development approach that utilizes in a development process tools of four generation, e.g. 4GL, CASE tools, etc., small teams, fast development methods, tights time-scales and resource constraints, iterative, evolutionary and participative prototyping and finally, intensive user involvement amongst many of activities of the process [1].

The author of the paper [1] wants to stress that the Rapid Application Development is not a bag with semi-rigour or monolithic methods. The focus of the Rapid Application Development is not steered on the rules in the software system development, which often is confusing for traditionalists. According to the paper [1], the management of and participation in the Rapid Application Development is compared in [1] to “white knuckle ride” and in case of bad selected and managed the Rapid Application Development’s team may lead to stress among the team and to an enterprise failure.

Moreover, following the statement of the author of book [2], the Rapid Application Development cannot be used to produce some software solutions, e.g. compound operation systems, the software for the space shuttle, etc., because CASE tools, in most cases, are not capable to create the highly complex and/or unique software enterprises. The aims of the Rapid Application Development are Information Systems with functionality requirements that meet particular user’s needs, as it is

stated in paper [3]. However, one can find some proposals of frameworks for embedded-distributed systems [4], or other areas of software systems.

The process is presented in paper [3], as a four-step application building-methodology. The idea of the Rapid Application Development is to create the prototype of a system and make improvements of the functionality, before it is placed in full-scale product. It is should be mentioned that both end-users and developers are strongly involved in the prototype development, for instance, each final prototype version shall be signed by the end-user [3]. The steps are: defining user requirements, building a system prototype, actual development and testing, and placing the system into production.

First step is performed to define and translate user needs to system specification. Users and developers are involved in this step because of two reasons: first is to validate whether they share the same vision of the system; another is to assure that business needs are on the first plan [3].

Next step is concerned in, already mentioned, a prototype building.

Whether the prototype is approved the heart of the matter, the system-building process, is run. Similarly to previous steps, users are also involved in the process, while the developers add required functionality, end-user is asked to make feedbacks about the product. The users' comments are incorporated directly into constructed system [3].

Finally, the system is placed into production and users are trained. Therein, the system's functionality is revised by developers and end-user and the system is validated against the requirements developed during the process. If the validation fails, necessary enhancements are begin to be developed.

Lastly, the testing of the system is performed and the system is reviewed [3].

Moreover, one shall remember that the Rapid Application Development approach has got iterative characteristics, since the activities, presented above, are only one cycle of the development process cycles, which are separated by the successive prototype developments.

However, every process has to be executed by the people and they, as it is emphasised in paper [1], are the key factor of the Rapid Application Development. The author of the article [1] writes that methodologies "can guide but cannot substitute" experienced developer in order to satisfactory result.

The typical size of team is 4 members, who are highly competent in given task's areas. The team consists of team leader, senior developer, developer, business analyst. The author of paper [1] recommends that all members should have got communication skills, and some business awareness. Naturally, senior developer and developer should have got proper development skills; moreover team leader should be able to manage the team, make proper motivation and has got analytical skills. The role of business analyst shall be strongly focused on business awareness, but the actor doesn't need to have e.g. technical or development skills.

According to [1], team's member can play several roles, but it is important that team's members are focused only on satisfaction of business needs, thus they shouldn't be distracted, e.g. by other projects. Moreover the author of paper [1] emphasises that stuff shouldn't be shuffled. The personal changes in the team only disturb and make noise, which influents negatively teamwork and in result the project.

The important issue in the Rapid Application Development, as it was emphasised couple times in the paper, is a user involvement in the development process [1], in order to meet user expectations [3]. It is often happens that a development process is too much focused on technical aspects of the enterprise, leaving behind the concerns for the user.

Author of the paper [1] enumerates the factors needed to satisfy this requirement; a proper design of the team, participation of the users and very good communication in order to fill understanding gaps between users and the developers. As it is presented in paper [3], the Rapid Application Development is run in proper way, whether the users are the integral part of the development process, e.g. the users ought to approve each development cycle.

One of the reasons of high performance of the Rapid Application Development approach is because the software product made within the process tends to meet the user expectations better, thus reducing amount of additional work prior to implementation.

The tools used in the process are another important part of the Rapid Application Development. They are special designed to speed the development time [5]. The Rapid Application Development tools are associated with drag and drop functionality, automatic code generations, which saves the programmer's time by adding chunks of code that are common for a written solution, or by providing libraries with necessary functionality. Moreover one should mention about the CASE tools that support the architectural and design part of the process by decreasing the time of the activities duration and improving the quality of theirs result.

Concluding, the Rapid Application Development approach was designed to support development application that meets the users expectations in short time. The Rapid Application Development process can bring very big growth comparing to other development methods, provided that appropriate projects are selected. Moreover, one should notice that implementation of the Rapid Application Development philosophy often is very

difficult, “due to Rapid Application Development recognition that no two projects are alike” [1].

2.1. Similarities and differences from Dynamic Systems Development Method approach

Both Rapid Application Development and Dynamic Systems Development Method are based upon short time to market together with correct business needs. Both methods uses small teams for development and the working process are iterative and incremental prototype building with a lot of user involvement to assure that the business needs are met. A prototype should be built and then improved and when the user signs upon it, the product should be built in large scale. Both methods involve the user intensively in the working process to keep a business focus to avoid going into technical issues. What differentiates Rapid Application Development from Dynamic Systems Development Method is that the testing is performed in the end of the current iteration. In Dynamic Systems Development Method, the testing is an ongoing process along with the ordinary development. Unit testing is performed during each timebox along with system and integration testing and acceptance testing.

3. A summary of Dynamic Systems Development Method

The Dynamic Systems Development Method is a framework for how to deliver a product as quickly as possible and with a guiding of how to control the process at the same time. Dynamic Systems Development Method do not have any defined techniques, instead there is advised paths of structured approaches as well as object-oriented. This development method consists of every part in systems development, from the first idea to the project to that the last solution of the project. The whole idea with Dynamic Systems Development Method is to produce what the market needs, within in time and budget and as fast as possible. Dynamic Systems Development Method is based upon nine principles which will be described in more detail.

3.1. The nine principles

The Dynamic Systems Development Method is based upon nine principles [6] that should be followed. These core principles guide the project towards success. It is recommended that all nine principles are followed. The main issues for the principles are strongly connected to user involvement and to meet the businesses needs.

3.1.1. Active user involvement. The most important principle is active user involvement during the whole project. Preferable is that a couple of knowledgeable user’s takes part in a development team and guides throughout the projects lifetime. The users will make sure that no misunderstandings of what the system should do takes place so that the system passes the acceptance tests at the end. The users should have knowledge of the systems purpose and aims and also have deep understanding of the business needs.

3.1.2. Dynamic Systems Development Method team must be empowered to make decisions. Another principle is that Dynamic Systems Development Method teams must be empowered to make decisions. This means that the team can take decisions during development for speed up the time for development. However, the team cannot do what ever they like. They need guidelines and rules and cannot overrun the projects budget. Decisions they can make is for example, changing the meaning of a requirement, change the order of the prioritization of the requirement or making changes in the solutions made for the system.

3.1.3. Focus on frequently delivery of products. Teams in a Dynamic Systems Development Method framework should make frequently delivery of products to ensure that the project is steering in the right direction. This is a third principle. The customer sees that a part of the product goes along with the expected view of the system. By deliver many parts a let the customer inspects those parts, the team will be sure of that the end-product pleasuring the customer.

3.1.4. Fitness for business purpose is essential for acceptance of deliverables. When developing systems with the Dynamic Systems Development Method it is more important to make sure that the right thing is developed then that it is developed in the best way and with best quality. Fitness for business purposes is the fourth principle and means that operational functionality is sufficient if it is acceptable by the customer. It means that the product will be more correct to the need of the business rather than having a lot of advanced features.

3.1.5. Iterative and incremental development is necessary to converge on an accurate business solution. The fifth principle, iterative and incremental development is necessary to converge on an accurate business solution, says that project changes corrective direction as the lifecycle continue to meet the need of the business. As systems are developed incremental and iterative, it is easier to correct errors in an early stage that do not meet the business needs and notice changes in the business needs. By this way systems will be more accurate to the business need in the end.

3.1.6. Changes during development are reversible. This sixth principle means that even if some errors is introduced to the system during the development process it will be possible to backtrack this error to a specific point in the development process. This does not mean that huge part of the system will be garbage. If the principles that deliver frequently to the customer are followed, then only the recent work is in need of changes.

3.1.7. Requirements are baselined at a high level. This principle means that all conducted requirements in the early stage become a high level scope for the project. Then detailed requirements can be conducted during the iterative process of the project.

3.1.8. Testing is integrated throughout the lifecycle. The eights principle says that testing is done along with the lifecycle of the project. Software components are tested by developers for technical purpose and by users for functional purpose and acceptance testing is performed incremental throughout the lifecycle. Integration tests are performed as soon as possible when some component is integrated with some other component just to make sure that something new does not damage earlier developed components. By testing throughout the lifecycle, the team can guarantee the quality of the system as it continues.

3.1.9. A collaborative and co-operative approach between all stakeholders is essential. This principle aims to remove the feeling of, us and them from the project. It is important that not only developers and users cooperate but also that parts of the organization and the customer organization can cooperate. The whole Dynamic Systems Development Method builds upon fast delivering of products and the fast development is in no good if the rest of the organization does not work fast with the customer organization.

3.2. Process overview

The dynamic systems development method has five phases [6, 9]. These phases are feasibility study, business study, functional model iteration, systems design and build iteration and implementation, see figure 1 [6]. When a project is developed with the dynamic systems development method, the feasibility study and the business study must be done sequentially. These two phases is the ones that decide the ground for the rest of the project. The last three phases are iterative and incremental. It is during these phases that the actual development is done in the project. The dynamic systems development method is divided into something that is called timeboxes. A timebox is a time period that can last from a couple of days to a couple of weeks. Timeboxes duration is planned before the timebox starts, as well as what is supposed to be produced.

In the feasibility study it decides if the dynamic systems development method is the best and most suitable method to develop the project with. It is not just the project that is controlled to see if it is suitable to use the dynamic systems development method, it is also the organization in the company as well as the people. Since dynamic systems development method is used for products that has to been developed as fast as possible, the feasibility study should not last for more then a couple of weeks. During the feasibility study a report is produced together with an outline plan for the development and a fast prototype of the product.

In the business study the projects business and technology is analyzed. Even this phase is supposed to be fast, there is not time for separately analysis of people. Instead a workshop with skilled people is used to present the business process and the users that will be affected or using the product when it is completed. The findings of the users will help the project to involve the customer at an early stage. Another important outcome from the workshop is the collected facts about the product. The affected users and the presented business processes are described in something that is called the business area definition. Besides the business area definition, there are two other outcomes from the business study. The first one is the system architecture definition. Since some parts of the product will start to be developed in the next phase, the first system architecture needs to be done. The second outcome is the development plan. This plan consists of prototyping strategy, testing strategy and a configuration management plan.

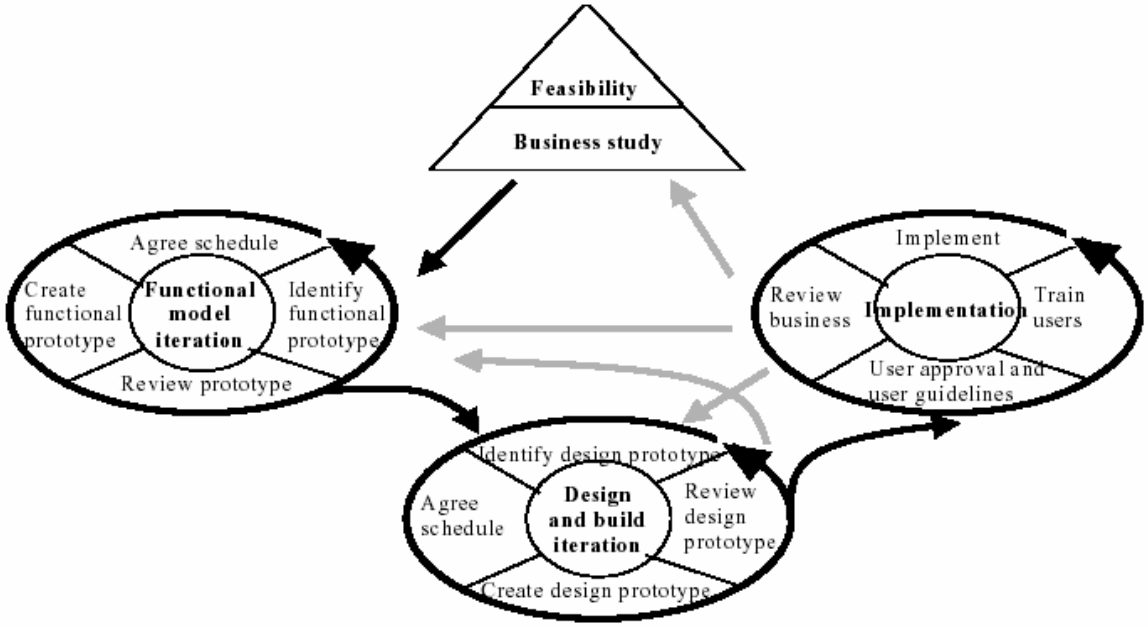

Figure 1 Dynamic systems development method process overview

The functional model iteration has four different activities. First identify the work, find a way to perform the work, perform the work and finally check that the work was performed in a right way. These four activities are also included in the next phase, which is the design and build iteration. In this phase, analysis as well as coding is done. This means that first is a prototype built and then it is used to improve the analysis model. The output from this phase is functional model that consists of the prototypes code and the analyze model. Besides this output, there are four other outputs from this phase. First a list of the identified and prioritized functions, a document to collect the users opinions about the product, all non-functional requirements that has been identified during this phase and in the business study phase. The last output is an implementation plan.

In the design and build iteration the product is built to a high standard. During this phase the product is also tested. The tested product will probably not fulfill all the requirements or non-functional requirements, but all agreed requirement for this timebox will be implemented and passed the test cases.

The last phase is the implementation phase, in this phase the product is delivered with the required documentation and is installed in its actual environment. Another part of the implementation phase is to train the users and then hand the product over to them. This phase has four possible outcomes. The first outcome is that all requirements are implemented in a satisfied way. The second outcome is a new area of business functionality has been found during the development of the product. The third outcome is some lower prioritized functionality had not been implemented. The last outcome is that one or more of the lower prioritized non-functional requirements were excluded from the product. All outcomes except the first one, is represented with the grey returning arrows in figure 1.

3.3. Project roles

A dynamic systems development method team has different roles. But, there are three that is consider as the key roles. These roles are the visionary, the ambassador user and the technical coordinator. Of course there are developers, testers and so forth too. The team should not be bigger then six people, but at least two.

The visionary is the one that makes sure that the project is on the right path and does to lose its original business objectives. The visionary starts to work with the project from the beginning, which means that he is in the project in the feasibility study and the business study and is in the project to the end. The visionary is the one with the initial idea to the project.

The ambassador users' most important task is to give the project the knowledge from the users and to spread information from the project team to the rest of the users. The ambassador user is actually a real user that will sue the product when it is completed. To find a person with the right knowledge to this role is an important task for the project to succeed.

The technical coordinator is responsible for the system architecture and that the work is done with high technical quality. He is also responsibly for the configuration management.

3.4. What differentiate a Dynamic Systems Development Method project from a more traditional one?

A Dynamic Systems Development Method project is similar to a traditional project. The project manager carries out duties like planning, monitoring of the work and progress, make sure the team works effectively and keep the risks in the project on distance. The difference is how the project manager performs his duties. The project manager must make sure that an empowered team can continue to work effectively. He has to plan for resources in a project. An example is that he has to make sure that testing environment have been set up in time for testing. In Dynamic Systems Development Methods, testing is not that far in time from implementation.

Planning is much done in four time-phases during the project. In the pre-project phase, the activities are planned. An outline plan is created in the feasibility of the project to prove the projects success. The development plan is created in the business phase of the project and provides more detailed information of what will be done and the prioritization of the work. At the end of the functional iteration phase, the implementation plan is created to describe the work to move the system from the development environment into the business that will use it.

How project managers monitor the project also differs a lot from traditional projects. The project managers do not follow up the time for certain activities, but monitor the high prioritized requirements and if they are implemented satisfactory during a timebox. The project manager uses no Gantt chart but prioritized requirements for monitoring a project. To prove progress to high level management, the project managers show that the work done for each timebox is made or not made. This shows the progress of the project in an easy way. For the project manager to get more detailed information of the progress of the work, meetings is held each morning for about fifteen minutes with the whole team to get a grip over the situation.

The teams that work in a Dynamic Systems Development Method will work more concentrated and many developers are not used to this way of working. Some critical activities can force the team to work overtime. However, the team members seem to like this way of working and have no problem with working overtime. With good estimates and avoiding risks, this overtime will not take place.

Two risks that are most common are that the system will not meet the business needs and the other one is that the project will run over time or budget. These risks are taken care of if the principle, frequently deliver of products is followed to minimize the business need problems and timeboxes for minimizing the budget and time over run. In the end of the business study, analyse of how the project will success or fail are done and might end up with project termination. Otherwise the project continues and the requirement prioritization list gives a help knowing the important requirements and the related risks. These risks will then be monitored by the project manager.

4. Identified characteristics in Dynamic Systems Development Method

The following characteristic are those we have identified for the Dynamic System Development Method. We present each characteristic and explain why we think there is a focus on this characteristic and how it is related to the course literature.

4.1. Structure and Organization

Organisation is a system that consists from humans and physical elements interacting in order to accomplish stated goals. The systems are described from perspective of structure and behaviour [10, p.433]. This chapter is devoted to present the structure, i.e. elements and relationships that are set among the system's elements, of the Dynamic Systems Development Method.

4.1.1. The Roles. Among the development process, several roles are distinguished, some of them were already presented in above chapters, and thus some adding only is going to be made.

The roles in the Dynamic Systems Development Method are constructed in this manner that they guarantee that the characteristics of the process stated in the nine rules of the Dynamic Systems Development Method will be met.

The roles in the Dynamic Systems Development Method are divided in groups, which are related with the activities performed during the process. In particular, one can find roles connected with team, project and workshop parts of the process [11]. The team roles are: ambassador user, advisor user, team leader, developer; the project group of roles include: executive sponsor, visionary, project manager, technical co-ordinator, and finally the group responsible for workshops contains roles of: workshop sponsor, workshop owner, participant, facilitator, scribe and observer. The following part is

going to outlines the responsibilities and characteristics of enumerated roles.

4.1.1.1. The Visionary. The role of the visionary was already presented in this paper. Moreover, the actor of the roles is responsible for supporting the translation of the vision into working practice, monitoring progress against the original vision and resolving conflicts across business area that are in the visionary's focus. The visionary role is also responsible for making available of user resources, farther they should make contribution into key requirements, design, and review sessions [11]. In order to fulfil these responsibilities, the visionary role's performer should be an excellent communicator, an excellent awareness of business goals, finally in order to have some detailed description of one's vision; one should have a high-level awareness of technological possibilities [11].

According to the Nicholas [10, p.492], one can find some similarities between the visionary and role of the project controller described in the book, because the project controller assists the project manager in planning, controlling, reporting and evaluation tasks, which overlaps the visionary's competences on the field of "translation of the vision to working practice", and project monitoring.

Moreover, one can find also some connection between the visionary and the customer liaison role, since the last one is responsible for customer and user's technical representative [10, p.493].

As the visionary is a connection of the project controller and customer liaison roles, the role of visionary might be seen as someone whose main object is to control whether the project meets the original user's demands, and if project brings the functionality really needed by the users. This is stated in the principle of the user involvement in development process [6, p.13].

4.1.1.2. The Executive Sponsor. The visionary is not always the purse-holder and most important decision maker in business area [6, p.46]. In the Dynamic Systems Development Method process, the role of the executive sponsor has commission in business issues related with the software enterprise. The executive sponsor is a high level executive role, since one is responsible for ensuring decision-making process for escalated project issues is effective and rapid, and responds for risen issues, furthermore the executive sponsor is obligated to get funds and other resources required for the enterprise inception. Finally, one is responsible for giving example of a commitment and availability throughout the development cycle, and at the end one monitors the continued business case for the project [11].

The role of the executive sponsor requires abilities of commitment appropriate funds and resources, political awareness and decisiveness and business knowledge [11].

The role of the executive sponsor is partially analogous to the role of the project champion, according to the book [10, p.495]. The project champion organises as the executive sponsor resources necessary by the project teams. Moreover the project champion "firmly believes in the project" [10, p.495], which is equivalent of the commitment made by the executive sponsor, and being "readily accessible […] and readily willing to rally support" [10, p.495] is the same as availability requirement put on the executive sponsor.

The care about the resources and commitment of the executive sponsor are features that are necessary to realize the time-requirements related to the Dynamic Systems Development Method process.

4.1.1.3. The Project Manager. The project manager role, if one follows the opinion from the book [6, p. 49], differs from the one played during the traditional development process with less strict timescale. The skills required in case of traditional project managers remain the same; however there are slightly altered responsibilities of the project manager. The basic skills of the project manager as careful planning, monitoring of progress, keeping the team working effectively, awareness of the risks to be managed [6, p. 49], and other enumerated in the Nicholas book [10, p.478]. The differs are in issues that are faced up by project manager; e.g. other problems in the planning process, thus another manner of the planning is carried out; different aspects in the progress measuring, hence the way the project is monitored is changed; the style of the team management, since the teams are more empowered than in traditional projects; or sorts of the risks that have to be over passed [6, p.49].

4.1.1.4. The Technical Co-ordinator. The last but not least role in the project group is the task of technical co-ordinator, whose characteristics one can find in previous chapter. The analogous one can find is between this role and role of quality assurance supervisor presented in the book [10, p. 493].

The role of the technical co-ordinator plays crucial part in fulfilling the assurance of the quality of the product developed using Dynamic Systems Development Method. Moreover, the technical co-ordinator role is created in order to ensure that the iterative and incremental development and the backtracking of all changes made during the development are possible, since ones is responsible for software architecture [6, p.47], and change management.

4.1.1.5. The Ambassador User. Moving to the team's roles, the first one presented is the role of the ambassador

user. It was described in previous chapters; however some accomplishment shall be made. The ambassador plays crucial role in the software design decisions. The ambassador brings user's ideas and inputs and by driving system design using the contacts with the team leader or technical co-ordinator [11].

As it is mentioned in the presentation [11], the ambassador has to have got desire, authority and knowledge, to be able to guaranty that the right system is built. Among many responsibilities the ambassador user takes care about reviewing the documentation, and the delivered software, providing user documentation, and taking care of acceptance tests [6].

In the relation to the roles presented in the Nicholas, one can found that the role of the ambassador user is the partially answer to issue of the user's commitment in the project, since by the role of the ambassador user the users have got opportunity to "share in making decisions" [10, p.546].

Moreover the role of the user ambassador is established in order to achieve the high quality information sharing and exchange, which is necessary for "facilitating […] development process" [10, p.546]. The good communication is a base for success project as the Nicholas claims [10, p.546].

The Nicholas statements about communication are partially included in the 9th principle of the Dynamic Systems Development Method, which declares that "a collaborative and co-operative approach between all stakeholders is essential" [6, p.19], and in opinion of the authors this is one the main features of the user ambassador role.

Further, as the ambassador takes place in the user testing, hence one contributes in fulfilment of the 8th principle, testing throughout the lifecycle [6, p.19].

4.1.1.6. The Advisor User. The knowledge of business process that is automated in details is brought by the role of advisor user. One's special ability should be able to communicate a practical knowledge of the business area. The holder of this role is probably somebody who is going to use the software system when it is complete [11]. The future user shall provide information of the user's request; moreover one participates in the prototyping and review process, and is responsible for checking whether the prototype is acceptable enough to be processed in further stages of development cycle.

And again the advisor user is the role that brings the user commitment and supporting described in the book of Nicholas [10, p.546] and in the book [6, p.91] into the project, since advisor users are the source of the business details. Thereby the advisor users are very important in order to achieve the 9th principle of the Dynamic Systems Development Method [6, p.19], and further the advisor users are necessary to fulfil the first principle of the Dynamic Systems Development Method, i.e. imperative user involvement, which gives authority to the users to decide how the system should look like [6, p.13]. The involvement is made as well in order to get the understanding of the system functionality and awareness of the business case for the product [6, p.91].

4.1.1.7. The Team Leader. The tasks of the team leader is to ensure that the team works as a whole, more about the team building process is presented later, and meets the stated objective. Moreover, the team leader is responsible for delivery of part of the system related with the team, change control, and project documentation. And finally one should organise the prototyping and review session between users and developers [11].

According to the book [6, p.91], the team leader assume the responsibilities attached to the project manager within Dynamic Systems Development Method development project, if one team is present, or assume the same responsibilities within one's team [10, p.481], if the multi team project is considered.

The role of the team leader give sureness of performing the team's task in time, thus one's role is important in bringing the 3rd principle of the Dynamic Systems Development Method in to the live. Seeing as this principle claims that the Dynamic Systems Development Method project will be success in effective control of activities and working performed in short timescales [6, p.16].

Obviously team leader as a team member brings in the reality the 2nd principle of the Dynamic Systems Development Method, i.e. one is able to make some decisions related to development tasks within one's team, e.g. the prioritization of requirements, altering the final detail of the technical solution, or arbitrate whether the quality parameters of the interim products of developments are acceptable at the current stage of development, etc. [6, p.15].

To sum up, the team leader may be perceived as the manager of the part of the project agreed to accomplish by the manager's team.

4.1.1.8. The Developer. The developer is responsible for the part activities that are the closest to built system. The developer has to model and interprets the users' requirements about the system using the documents of the business study. The models and requirements have to be transformed into prototypes and code.

Because of the small size of the team structures in the Dynamic Systems Development Method, the developers shouldn't be varied among the IT roles played in during development, i.e. someone who is skilled in programming, shall be able also see beyond this and understand the user requirements, or developers should have also some analytic, designer or tester skills [6, p.80].

The core skills shall be represented by the developers, since some members are stronger in one area, by cooperation among the members, every developer's skill will be intensified by others capabilities [6, p.80].

The important characteristic of the team's members is flexibility. The Dynamic Systems Development Method's developers shouldn't protective of their area of work. Moreover the developers that are able to change their views quickly about requirements of the enterprise are the most welcome one [6, p.80].

To sum up, the developers are the equivalent of the role of the project engineer described in the Nicholas' book [10, p.492].

As the developer is the fundamental role that deals with project implementation, one has strong influence on project time-to-market characteristics, and quality assurance about the developed software, thus one's role is performed in order to achieve the idea of the 3rd principle [6, p.16]. Additionally the testing is performed by the developers during the project lifecycle, thus it fulfils the 8th principle of the Dynamic Systems Development Method. Further the developers also are the main agents of the 2nd principle, which claims that team members should be given the authority to make decision about theirs part of developed system, e.g. what requirements mean in practice, altering the fine detail of the technical solution, etc. [6, p.15]. Moreover, with respect do the content of the book [6, p.80] the developers should have got communication skills, thus their role was also tailored to meet the expectancies of the necessity of collaboration and co-operation among all stakeholders in the 9th principle of the Dynamic Systems Development Method [6, p.19].

4.1.1.9. The Workshop's Roles. The workshop are the meetings, during these meetings brought people who have the right knowledge debates about the requirements applied to developed system [6, p.61]. The roles that participate in the workshops were mentioned at the beginning of this section.

The main idea of the workshop is to achieve the consensus about particular issues of the project. The role of the facilitator is established in order to help in this process, thus the important work during the workshop is made by the facilitator.

The facilitator is responsible for managing the workshop process. Further the facilitator is catalysts for the workshop's communication and preparation. The duty of the facilitator is to agree the workshop with the Project Manager, to familiarization the participants with the business area, and finally facilitating and reviewing the workshop against its objectives [11].

Moreover, the authors of the book [6, p.48] claim that it is good idea to have the facilitator that is not involved in the current project, because otherwise gaining the consensus might be very difficult to do. Further, the external facilitator will not make any pressure on the workshop's participants in order to agree on particular project issue, which will be against of the Dynamic Systems Development Method principles.

The workshop are organised in order to improve the communication and commitment, described by the Nicholas in the book [11, p.543], of the stakeholders. Thus, by the workshop the promises of the 9th principle of the Dynamic Systems Development Method are implemented in the project life. Obviously the 1st principle is in use, when the workshops are organized, since they are the best time and place to present the comments about the project's requirements.

4.1.2. Project Structure. The team is a basic part of the Dynamic Systems Development Method organisation's structure. The size of team varies from the minimum, two persons, i.e., one do technical work, and one user to check whether developed software meets user's expectations, to maximum six persons; however the top limit is established because of the increasing of the process productivity when the teams' size is greater [6, p.45].

The typical project within the Dynamic Systems Development Method has got one or two teams, however the larger project can be realized by 6 teams maximally. This restriction of number of the teams is caused by the problem of unmanageable the greater number of the teams. However, as the author of the book [6, p.46] claims, this does not mean that Dynamic Systems Development Method might not be used in larger project, in the book [6, p.47] example of enterprise that involves more than 150 is given, however in that case the trick are used, i.e. one role is played by several persons who divide responsibilities among them and more formal communication channels is added.

The project realized within Dynamic Systems Development Method by many teams may cause the necessity of creation teams specialized in some sort of the project's tasks, for instance, the co-ordination team, which faces up the synchronization of the project's assignments; or the testing team [6, p.47].

The members of the teams should be able to perform the core of what is necessary to do. However, it is may happen that in order to perform some particular task, the external specialists in that area will be called. Example of the specialist is the user interface designer, who will care about crafting the appropriate, e.g., easy to use, with maximal ergonomics, etc. Obviously, the type and size of the support will depend from the sort of the problem and the skills of the project's teams [6, p.47].

To sum up, teams are the fundamental part of the Dynamic Systems Development Method process. The basic team in the Dynamic Systems Development Method

contains at least two members; the user, and the developer.

4.2. Systems and Procedures

The characteristics of systems and procedures in Dynamic Systems Development Method are the nine principles [6] and the five phases [6] during the project. These principles and phases are briefly described in the summary. For a Dynamic Systems Development Method project to be successful, it is important to follow all nine principles and carry out each phase of the method in a careful manner. The method follows a certain way of working. All these steps can be read about in chapter 4, Dynamic Systems Development Method process overview. For the functional model iteration and design and build iteration there are four activities [7] that we will describe a bit further. Those activities are, identify the work to be performed, decide upon how to perform the work, perform the work and finally check that the work was performed in a right way. During the functional model iteration, a prototype is created. The prototype should be identified and decision on how to build the prototype should be taken. Then the prototype should be build and a review on the prototype should be made to verify that the prototype was built in a correct manner. In the design and build iteration, the prototype should be turned into a real system. The system should then be identified and decision on how to build the system is taken. Then the system is built and review of the system is made to assure that the system is correctly built. In the last iteration, implementation, [7] the product is packed together and documentation is written together with an inspection document. Then the users are trained to use the product and then they give their approval of the product.

As mentioned before, for the principles it is important to apply them all on the project. However, those principles that we think are the most important ones are active user involvement, focus on frequently delivery of product and testing throughout the project lifecycle. These are the ones that the whole Dynamic Systems Development Method is based upon.

Planning is made in the end of the business study. The work that has to be done and prioritization is being clearer. The development plan [7] is put together which includes schedules for timeboxes and strategies for testing and configuration management. Detailed plan of each timebox is made later on by the responsible team for the timebox. Non-functional requirements are connected to each timebox so that each team has to think about them. In the end of the functional model iteration, an implementation plan [7] is produced to address the issues for moving the product from development into user's operational environment.

In Dynamic Systems Development Method there are mainly two risks. The first risk is that the product will not meet the users need and the second one is that the project will overrun its budgeted cost. These risks would be minimized if the principles are followed.

When monitoring a project that uses the Dynamic Systems Development Method t is not time to focus on. Rather than using a Gantt chart the prioritized requirements should be used. Instead of time the project manager should focus on the highest prioritized requirements and how many of them that has been implemented during a period of time. When reporting to upper management it is more of an interest to know the progress for a timebox. In the end of a timebox the project manager knows if all activities that are assigned to a timebox are completed.

4.2.1. The focus on the characteristics. Why Dynamic Systems Development Method focuses on the different principles are because of that these principles are the most important aspects for a successful product development. What Dynamic Systems Development Method focuses on is fast development of correct product that fulfills the business needs. The principles, active user involvement, frequently delivery of products, fitness for business purpose, high level requirements and iterative development driven by user feedback, are important in terms of develop the right product. A product that actually fits the users needs. The principles empowered teams and high level requirements are principles that should make the process faster during the development phases. The principles mentioned are representing the most of the principles and they are directly connected to the most important aspects of Dynamic Systems Development Method.

The different phases are also important. The planning is only rough made in the beginning and is being more precisely when timeboxes are being worked on. The meaning is that you should plan when knowing exactly what to do and by this getting more accurate estimations. In the early phases, plans are rough but are more precisely as the lifecycle goes on. We think this way of working would be good but also you do not get the whole picture of the work as in a more traditional project. A project plan [10] in the beginning of a traditional project includes all activities and the time for each activity to be completed. This feel like a more secure way of working if monitoring of the activities and follow up is made to re-estimate activities if needed. The project plan in a Dynamic Systems Development Method contains the

Risk management also feels a little insecure in a Dynamic Systems Development Method in contradiction to a more traditional one. Basically what Dynamic Systems Development Method says is that if the core principles are followed then the project will not fail. We

agree on that part if risks only are defined as meeting the business need and complete project without overrun the projects budget. We think risk management in a traditional way [10] feel more secure as the risks are covering a wider perspective.

Monitoring of a project though, we think is performed in a good manner. As the development cycle is divided into timeboxes and each timebox are connected to certain activities and requirements it is easy to track if all are completed in the end of the timebox. In a traditional project you can follow up progress on a weekly basis as you calculate the number of completed activities. In a Dynamic Systems Development Method the focus of monitoring is on the timeboxes as they contain the specific work for a specific period.

Why the Dynamic Systems Development Method is based upon the phases of work is that this method is focused on iterative and incremental development and the working phases are more accurate for this kind of development.

4.3. Quality Assurance

Dynamic Systems Development Method do not have the time to take care of all identified defects in the systems as more traditional projects has, because the Dynamic Systems Development Method purpose is to deliver fast and with 'good enough' [6] quality. This means that, if the system fulfils the requirements and the users can use the system without calling the IT staff to fix defects all the time. Then the quality is 'good enough' and the system can be delivered. The quality thinking in Dynamic Systems Development Method is not the same as it is in more traditional development methods. Dynamic Systems Development Method uses five different techniques [8] to ensure the quality in the system. These techniques are facilitated workshops, user involvement, reviews, testing throughout the lifecycle and that quality once built in into the system is kept.

Facilitated workshops are used to minimize the misunderstandings within the team and to make certain that the requirements are always kept in the mind from the beginning of the project. In the Dynamic Systems Development Method, facilitated workshops can be used in any development phase.

The user involvement is meant to be as high as possible to ensure that everybody can express their view and to reduce the misunderstandings between the users and the team. The idea is to have an ambassador user that works full-time with the team. This is not always possible, but at least a good contact and continues flow of information and feedback during the whole project time is required. It is important to receive information from the users as well as feedback to be able to produce a system with 'good enough' quality.

Reviews are used on the produced code, the system and the documents. The purposes with reviews are to make sure that it follows the quality criteria's that has been defined in the project.

In Dynamic Systems Development Method, testing is not something that is done at the end of the project. Testing is something that is done throughout the whole lifecycle of the project. Within each timeboxes different tests are performed like, unit tests, integration tests, system tests and acceptance tests. Unit test is a test that each developer writes for the part of the system that he works on. Integration tests is a test for integrate the current part of the system with a part from previous timeboxes. The system test is a test where the overall systems functionality is tested. These three tests is performed before the timebox ends and this part of the system is delivered to the customer. The last test is the acceptance test, which is a test that the ambassador user puts together for the rest of the users. All tests from unit tests to the last acceptance test should be planned before the production of the system starts.

Building in quality into the system and then make sure that the quality is still there throughout the project is important. To make sure that the quality still is there is up to the configuration management and the change control. To be able to build in quality, one of the three levels of maintainability must be achieved on the project. Which maintainability level the project should use, is defined in the business study phase. The three levels [6] are, must be maintainable from the first delivery, do not have to be maintainable from the beginning but after the delivery and the last level is that the system is short-term fix and thereby it do not have to be maintainable. The first level is the one that takes most time in the beginning of the project. The second level is faster with the first delivery, but in the long term it takes more time then level one. The third level is the most difficult to achieve of the different levels.

4.3.1. The focus on the characteristics. Why Dynamic Systems Development Method has a different focus on quality then more traditional development methods, is first of all that they deliver the first part of a working system faster. More traditional methods produce more products [8], like intermediate design. This leads to more reviews, test and inspections. More traditional methods has two quality-oriented aims [10], produce a correct end-item in a correct way. Dynamic Systems Development Methods aim is to deliver a product that is needed and when it is needed and with 'good enough' quality, see section 4.3 for an explanation of 'good enough' quality.

In traditional methods they have a focus on fixing as many errors, problems and mistakes as possible [10]. Dynamic Systems Development Method can not have this focus because of the fast deliveries. Instead they accept

that a product can be a little bit defected [6], as long as they can deliver with 'good enough' quality, no more or less. In our opinion this thinking of quality can be a little problematic, for example, customers that have quality as an important factor would not like this approach for software development.

Dynamic Systems Development Method has a big focus on user involvement. This is one of the most important quality factors in this method. The focus is that an ambassador user should be able to give information and feedback constant during the projects lifecycle. This is done to avoid misunderstandings and to present each others views. In our opinion this is an important and positive factor. This leads to fewer change requests and to a product that is more accurate for the users need and more user-friendly.

5. Which type of project is the method suitable for?

Or opinion is that Dynamic Systems Development Method is most suited for market driven and in-house development rather then contract-driven development. As the method has a short time to market as well as intensive user involvement that can secure the business needs, the method should be very accurate for developing new products for the market.

The argumentation for that the method is accurate for in-house development is that the team is small and can if needed be much focused on difficult problems that are hard to solve. This small team can be dedicated to the problem and finding the best solution for it. Therefore it can be good to run a project of Dynamic Systems Development Method in parallel to the main project and they should focus on more hard solved problems that consist in the main project. You then have the user on site as it is the main project, you build prototypes to test with and there is better communication within the small team and the creativity should be higher.

Why this method is not that appropriate for contract-driven development is because that in contract-driven development, projects runs for a given period of time. Two parts agree upon a contract which can be hard to fulfil from the suppliers perspective. Also it is hard to agree upon a contract with high-level requirements. High-level requirements leads to that it is hard to give an exact cost for the product as you can not say how much time it will take to develop the product. All functionalities and non-functional requirements that are agreed upon in the beginning can be hard to fulfil as the process model is iterative and incremental. This can lead to that non-functional requirements will not fit the user's needs and all functionality can be hard to have time to implement as it from the beginning does not consist a detailed plan of what to implement. This is at least our own experience, iteration is for a given period of time but what actually will be implemented during that time is hard to pronounce.

6. Strengths and Weaknesses

In this section we will present strengths and weaknesses with the Dynamic Systems Development Method. The strengths are mostly based upon literature while the weakness are based upon own experiences and discussions between the authors of this report.

6.1. Strengths

The strengths with the Dynamic Systems Development Method are that this method gives a short time to market [6] for new products. As the method has incremental and iterative development it means that in the end of the current iteration there is a developed product with complete functionality. Then in the next iteration there will be an extension of the functionality. The basic means of this way of develop is that at the end of anyone of the iterations, there should be a full working product.

One other strength with the Dynamic Systems Development Method is that the user should be involved in the development process very intensively [6]. This gives the project a good asset in terms of always being able to ask the user about functionality questions. As the user is involved in the development process, the developed product should fit the users need in a very good manner instead of having a product with just technical elegance. The user have good experiences from his business and provides a good support for the development team as he can give answers directly and questions marks of functionality could be straitening out directly. The user and developers uses prototypes [6] to communicate the functionality. This is a good way of really show that the understanding of each others is correct. The user can sometimes have problem with express his thoughts while developers can tend to talk in a more technical way. This can easily be solved with prototypes as user can try the functionality and in an easier way then explain differences from his vision and also give approval for functionality that is in line with his thoughts. As there exist frequently delivering [6] of product in a Dynamic Systems Development Method there also exists frequently feedback from the customer to the developers. This also gives the product a more accurate look when deciding on the users needs as the customer can follow the development changes in the product.

When working with Dynamic Systems Development Method, team should be very small. This provides more

informal communication and less bureaucracy. This means that information travels faster between team members and the team is also empowered [6] that means that they can take faster decisions upon solutions to problems. This reduce the overhead time of solve problems as communication does not have to travels on formals ways and decisions would not have to be taken by upper management. Our opinion is that this way of working is more stimulating then if you have to grab someone else to discuss solutions and have to wait on some decisions from upper management and so on. This gives the developers a chance of speeding up the process and also a chance to be more creative.

Some other positive aspect with the method is that in the beginning of a project, in the feasibility study, it is first analyzed if the Dynamic Systems Development Method is the best way of work in the project. Basically, there will be an assessment of how appropriate the method is for the current project and its type. If the result is that this method is not the most appropriate way, then the project does not use the Dynamic Systems Development Method as a process model. If it turns out to be an appropriate method, then they continue on to the business study. Also in the feasibility study, judgment of whether or not the right people are chosen to work in the project. As the team is small, it should then consist of skillful people that can work together in a good manner.

The Dynamic Systems Development Method provides a good way of handling changes in the requirements. If some change will be made there is only a need to track functionality from the last delivery and forward. If the functionality is past of the last delivery, the functionality is fully working and is not in need of being changed or removed. In this case new functionality is added that fits the new requirement in a good way. If the changes of functionality are in the current working iteration, changes are made to the functionality. In this kind of process model, changes are very easy to make to the product in contradiction to a more traditional project where you in an early stage decide upon functionality and solutions. If changes of functionality have to be made, chances of that it will be large changes in the system is huge. Changes in functionality is unavoidable and all agile methods says that changes is no problem but are more then welcome in these kind of projects.

Some parts of the testing strategies are good in the Dynamic Systems Development Method. System-testing, were the user tests the product and gives feedback on what are good and also possible changes, is very good. The team will then be given feedback from the user on the system. If unit-testing and integration testing is done in a correct manner, these testing strategies are also very good. To have continues testing throughout the development lifecycle is good as the quality should then be higher then if the testing is done afterwards.

We think that the Dynamic Systems Development Method has many good aspects when developing software but it is important to know how the process works to be able do follow the working process correct.

6.2. Weaknesses

In Dynamic Systems Development Method the requirements for the systems is not on a detailed level. The requirements are instead on a high level [6], like in our last project where we used extreme programming as development method. Our opinion about problem with high level requirements is that it is harder to make a detailed planning and estimations for the whole project. Not being able to plan the whole project in detail, like a traditional project plan where activities and tasks are estimated for the whole project, can lead to problem with project monitoring and to see if the project has a progress. This was the problems we had in our extreme programming project. It also hade the effect that we could not minimize the uncertainties in the project, because we did not know if and where the risks in the system were. If the project do not have an overall project plan for the activities and tasks, it is hard or impossible to see dependencies between activities from different iterations and to see how a schedule overrun of one activity affects the activities in the next iteration. This was at least our experience in the extreme programming project. We also experienced that it is hard to know exactly what will be delivered as the projects end-item. In the real world we think that it could be hard to know how long time it would take to complete a project with Dynamic Systems Development Method since you do not have an overall detailed project plan from the beginning as they have in more traditional development methods.

Another weakness with Dynamic Systems Development Method as we see it is the quality assurance. Dynamic Systems Development Method is about fast deliveries with ‘good enough’ quality [6] and if there are some defects or errors in the system that do not effect the users, then they can be in the system [6]. This might not be a problem for the moment for the users, but what if the customer wants a new feature and the defects in the system affects the new feature. Then the defects should been taking care of, but sometimes there has to be a lot of re-factoring to be able to remove the defects just because the defect was not removed from the beginning, this happened in our extreme programming project and it took a lot of time to fix. Another problem with quality and defects is that in Dynamic Systems Development Method as well as in extreme programming, the developers should not think of problems that can occur in the future. Just develop what is needed to develop and leave the problems for the future, if they happen they

happen. Even this is related to our own experience in such a project.

Since Dynamic Systems Development Method has user involvement [6] as an important part in the method, which is a strength. But, if the person that has the key role as ambassador user [6] do not have the right knowledge that is needed for this role, if he do not give the right information and feedback to the developers or if he can not be have a continues flow of information exchange with the developers, then the project has a big problem. We see this as a weakness that if this role is not properly used, the whole project will have big problems and difficulties to deliver a good product.

7. Conclusions

To conclude, the organisation of the Dynamic Systems Development Method process is designed in order to meet the explicit features outlined in the principles of the Dynamic Systems Development Method process. The structure of project and teams is chosen fulfil the objectives of active user involvement, empowerment of the teams, frequent delivery of products, interactive and incremental development and others. However, one should remember that the people are the important part of the Dynamic Systems Development Method process. They make the biggest contribution in development effort.

The nine principles in Dynamic Systems Development Method are important to follow otherwise the project would not be successful. If not all principles are applied, the whole project could be jeopardized instead. There are five phases in Dynamic Systems Development Method. The first two phases are done sequentially and in these phases the ground for the whole project is set. The last three phases is where the systems are actually developed and implemented. These phases are iterative and incremental. There is no overall detailed project plan in the Dynamic Systems Development Method. The detailed planning is instead made for each timebox before the beginning of each timebox. Monitoring in this development method is not done in the traditional way of checking its progress. Instead the progress is checked by checking what is done in each timebox and compares that to what was planned to do in the timebox.

When it comes to the quality of the system, Dynamic Systems Development Method delivers a product just with 'good enough' quality. This means that their still can be defects and errors in the system. But, they do not affect the users need for the moment. Tests are performed during the whole lifecycle of the project.

The Dynamic Systems Development Method is most suitable for market-driven and in-house projects. But, it is not suitable for contract-driven projects because it is hard to say what is going to be implemented within a specified time period.

Dynamic Systems Development Methods strengthens is that they have fast deliveries, user involvement where the user ambassador gives information and feedback from the users to the project. Requirement changes s easy to handle. The weaknesses with Dynamic Systems Development Method are that there is no overall detailed project plan, which can give problems with for example, to see dependencies between activities. Another weakness is that there still are defects or errors in the system when it is delivered. But, there errors do not affect the user for the moment.

8. References

[1] A.J. Mortimer, "Project management in rapid application development", *Project Management for Software Engineers, IEEE Colloquium*, 1995, 5/1-5/3.

[2] Martin, J., *Rapid Application Development*, Macmillan Publishing Company, New York, 1991.

[3] R.T. Compton, "Minimizing waste with *Rapid Application Development*", Strategic Finance, 2002, p. 50-53

[4] R. Obermaisser, P. Peti, "A framework for rapid application development of distributed embedded real-time systems", EUROCON 2003 Computer as a Tool. The IEEE Region 8, 2002, p. 80-84

[5] G. Sharon, "*Rapid Application Development* Java tools lure business users", Computerworld, 1997, p. 57, p. 60

[6] J. Stapleton, "DSDM: Business focused development", Addison and Wesley, Great Britain, 2003.

[7] M. Clifton, "What is DSDM", http://www.soe.ucsc.edu/~brianh/papers/ProgQualHanksMcDowell.pdf visited 18 October 2004

[8] "DSDM Manual Version 3", http://htsa.ie.hva.nl/~helpdesk/DSDMManual/p2quality/qulty01.htm visited 18 October 2004

[9] P. Abrahamsson, O. Salo, J. Ronkainen, Agile software development methods, VTT Technical Research Centre of Finland, Finland, 2002

[10] J.M. Nicholas, Project management for business and technology, Prentice-Hall inc., New Jersey, 2001.

[11] J. Venneman, “DSDM Quick Reference http://www.dsdm.org/en/resources/suitcase.asp. Presentation-DSDM Suitcase”,