

## Build Your Project Using Dynamic System Development Method #5 of a Series, by Pavan Kumar Gorakavi, M.S., M.B.A, G.M.C.P, C.A.P.M.

Dynamic system development methodology is a non proprietary framework maintained by DSDM symposium [1]. DSDM symposium is a non-profit organization. DSDM is a vendor independent framework. DSDM is a systematic approach of handling a project in an effective and efficient manner. DSDM facilitates an interesting framework to develop functionality in better and amicable manner, deliver functionality efficiently and effectively, and satisfy the real requirement of the project.

### Life cycle of DSDM

Dynamic system development methods framework is developed in five phases [2][3], of which the first two phases are sequential in nature and the last three are incremental and iterative. The time interval also called as a time box is planned well before the game. Generally time boxes are from few days to few weeks. Life cycle of DSDM is illustrated in figure 1.

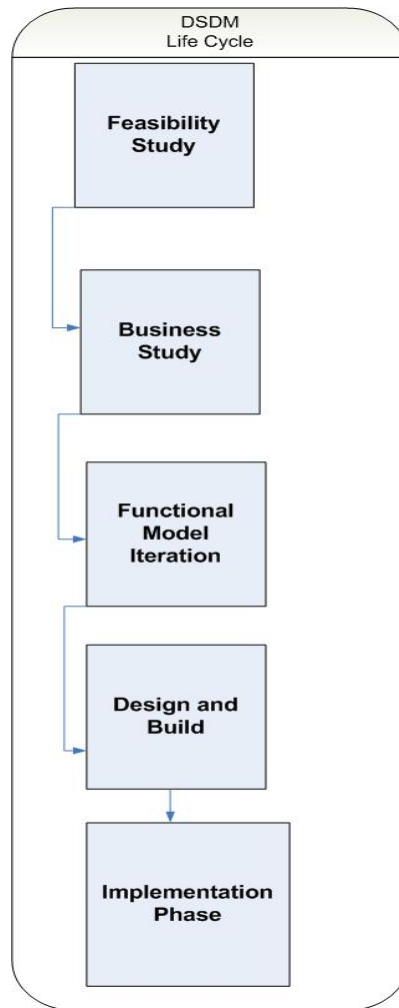


Figure 1: Life cycle of DSDM

### Step 1: Feasibility Study

Initial project suitability is assessed in this phase. This phase helps to identify the feasibility of a project. This phase should be necessarily short and should not be more than few weeks. This phase helps to identify the answer for some of the questions like:

- Is DSDM applicable to my project?
- What are the known dependencies of the project?
- Are there any technical challenges?
- Are there any resource constraints?
- Are there any organizational issues impacting the project?
- Are there any know risks, If so what are they?
- High level estimates of timescale and costs

The scope of the feasibility study is to gather required details about whether a feasible solution exists or not. Detailed analysis is performed in later phases. Generally in practice, management conducts a workshop which includes stakeholders, developers, testers and all necessary interested parties. The scope of the workshop is to have a high level walkthrough of the project requirements. Feasibility reports, prototypes, High level plans and risk logs result from this phase. The Feasibility report is a high level report that enables the project steering committee to decide on the project's future, and any need for further feasibility study. Feasibility prototyping is a methodology of developing a prototype as proof of concept.

### Step 2: Business Study

After performing feasibility analysis in step 1, it is time to analyze business and technological characteristics. The Business study provides the basis for all subsequent works. Business study should be necessarily short--not more than a few weeks. This phase leads to a detailed outline of business processes affected and their information needs. This phase tries to answer some of the key questions, such as:

- What is the business Scope of the Project?
- Any risks to be reassessed?
- What are the non-functional requirements (performance, constraints, subjective)?
- Outline the project backlog – future prototype deliverable
- Basis for technological developments
- Priorities of the requirements identified in step 1 and step 2.

This phase results in defining business area definition, prioritizing requirements, system architecture definition and development plan. Business area definition provides high level information about business processes, interested parties, and other dependent variables which support the proposed solutions. The Development plan contains a high level plan about the project and subsequent increments. Functional and technical architecture is illustrated in the form of System Architecture Definition. The architecture defines the coherence of hardware, software and other effecting components. This definition helps in understanding the technical architecture to be used, describing the target platform, and outline description of the software architecture.

### Step 3: Functional Model Iteration

This phase is incremental and iterative in nature. The phase has the objective of delivering a functional model consisting of both working software prototype and static models. This phase results in high level processing of information obtained in business study. Functional model iteration contains four activities. They are illustrated in figure 2, below.

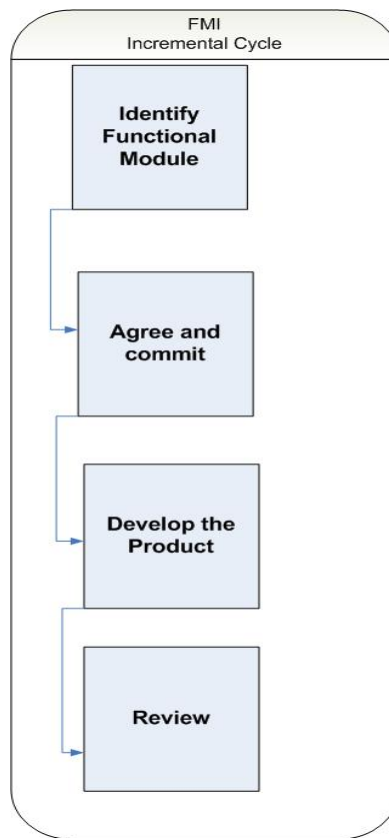


Figure 2: Iterative cycle of FMI

The Functional Model Iteration phase produces functional models, non-functional requirements, implementation plan, time box plan, and functional model review record. The functional model defines the solution that includes both documents and functional prototypes. Non-functional requirements are gathered during workshop sessions between developers and ambassador users. A Functional model review record captures the feedback on all parts of the functional model. An Implementation plan contains details about current system increments and a time box plan provides a recommended process schedule for a given time box.

#### Step 4: Design and Build Iteration

This phase refines the functional prototype developed in step 3 to meet functional requirements. This phase primarily engineers or develops the system to satisfy user requirements. A Tested product is the major outcome of this phase. This Design and build iteration consist of four activities.

1. Identify the requirements of the module.
2. Plan and commit to the requirement.
3. Develop the module, and
4. Validate the functionality of the module.

Design and Build Iteration results in a time box plan, tested system, design prototype, and test records. Design prototype is a interim product that addresses technical issues. This does not include functional module activities. A Tested system is the final system which can be moved to operational use. Test records have the statistics about product quality.

### Step 5: Implementation phase

This phase covers the transition from the development environment to operational environment. The major objective of this phase is to place the tested system in the users' working environment and train the individuals. Future development requirements are also framed in this phase. An Incremental review document is used for illustrating planning activities for subsequent increments.

### DSDM Team Players

There are different team players playing an effective role while practicing DSDM. Stakeholders include executive sponsors, Ambassador user, Advisor user, visionary, project manager, coordinator, Team leaders, developers, testers, scribe, and facilitators.

Executive Sponsors are a high level designation for that person or those persons who are committed to the project. They are the ones who own the system and are responsible for it. Ambassador user is responsible for providing key inputs to business requirements and design sessions. Visionary is responsible for driving vision of the project. Advisor user is responsible for providing information on request. Scribe is a stakeholder role who is responsible for recording, managing and assisting with necessary documentation. The Organizational architecture of DSDM is illustrated in figure 3.

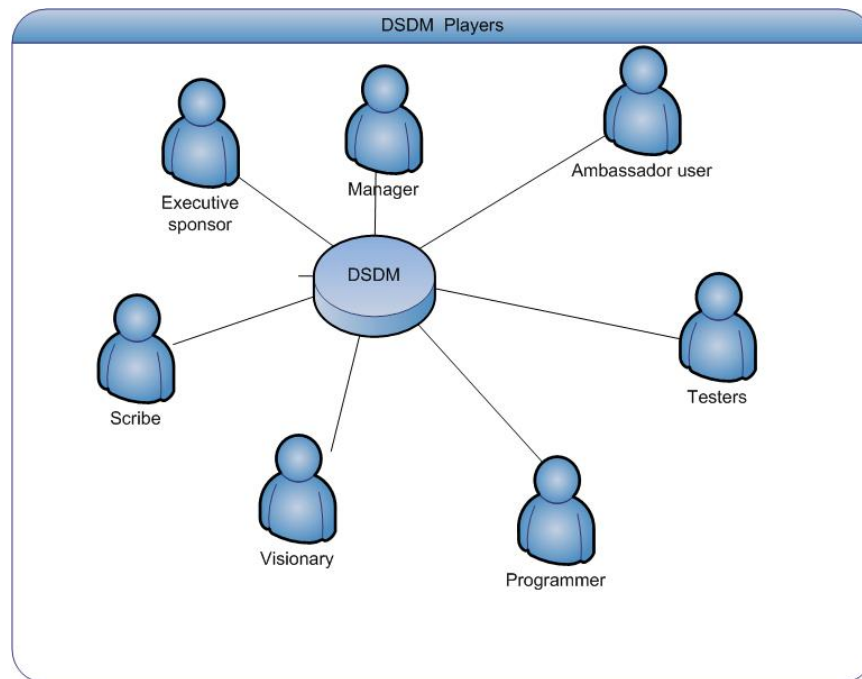


Figure 3: Different Team Players of DSDM

### Conclusion

DSDM is a useful approach that helps to develop a robust system. The methodology helps keep a project from failing and also helps in developing requirements-satisfied systems. DSDM implementation aims at on-time delivery and within budget. If proper analysis is performed, the DSDM approach can yield superior results. Further details about DSDM can be obtained from [www.dsdm.org](http://www.dsdm.org)

### References:

- [1] [www.dsdm.org](http://www.dsdm.org)
- [2] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, Juhani Warsta. Agile software development methods, VTT Publications.
- [3] Russo, Barbara, Marco Scotto, Alberto Sillitti, and Giancarlo Succi. "Chapter 6 - Other Agile Methods". Agile Technologies in Open Source Development. IGI Global.

### About the Author

Pavan Kumar Gorakavi is working as a Senior Software Developer in Dallas, TX. He is settled in Dallas, TX with his family (wife Swapna Gorakavi and son Anish Gorakavi). He is VP - programs for *asapm* Young Crew. He is also acting as Associate Director [Marketing] for PMI-ISSIG. Pavan earned his Bachelor's degree in computer science from Jawaharlal Nehru Technological University and Masters in computer science from Lamar University. He did his MBA from University of Texas at Dallas and GMCP from Southern Methodist University. Pavan holds SUN, IBM and PM Institute certifications.

Pavan Gorakavi authored a book on 'Artificial Intelligence' published by Rahul publications - India, and 'Digital Electronics' published by Subhash publications, India. His research interests are Artificial Intelligence, Agile methodologies, and Software development in ADA, Prolog and Java. You can reach Pavan at [gorakavi@gmail.com](mailto:gorakavi@gmail.com).

### About this Series

This article is the fifth in a series by Mr. Gorakavi on Agile, posted on the *asapm* website; watch for the others in the series. And, although the concepts of Agile are most-common in Software Development projects, increasingly Agile and Lean PM methods are also turning up in many other project areas, including Engineering and Manufacturing, where some assert they actually originated.

