

# A Survey of the Use of Agile Methodologies in Different Indian Small and Medium Scale Enterprises (SMEs)

Nomi Baruah

M.E. student

Computer Science and Engineering Department  
Thapar University, Patiala

Ashima

Assistant professor

Computer Science and Engineering Department  
Thapar University, Patiala

## ABSTRACT

Prior to the use of Agile Methodologies in software industry, the software projects were chaotic and unpredictable. As a result the processes which are best suited for well-defined problem domains cannot manage such software projects. It had been seen that with highly detailed user interface designs, specifications and plans, the software developed turned out different from its original intent because customer requirements change frequently. The crux of Agile Methodologies is that the changes in the requirement can be managed software even during the development cycle of the software development. A survey of 18 SMEs catering software market has been carried out for finding software development scenarios. Eight Agile Methodologies had been studied and surveyed which I found effective in current scenario of software development. These are Extreme Programming, Scrum, Crystal Methodologies, Rational Unified Process, Adaptive Software Development, Feature Driven Development, Dynamic Systems Development Method and Lean Development. And it has been discovered that majority of SMEs follow Scrum for their software development. And other SMEs are also developing their software projects with the help of Scrum variants like scrum and waterfall model as they are using Scrum in requirement gathering, management level processes, and development using waterfall model. This paper summarizes the trends followed for software development in SMEs and it has been shown graphically also.

## General Terms

Agile Methodology, Agile Software Development, Small and Medium Scale Enterprises (SMEs)

## Keywords

Adaptive Software Development, Crystal Methodology, Dynamic Systems Development Method, Extreme Programming, Feature-Driven Development, Lean Development, RUP, Scrum.

## 1. INTRODUCTION

The software development is expanding multidimensionally in different fields. Now a days the software projects are becoming more complex as compare to years. As a result the customer are not able to visualize the output of the software, which results in change in the requirements. Previously, the software development approaches were not able to implement the late changes in the requirement of a software project. Agile Methodologies finds its scope in configuring requirement changes even lately in a software project.

## 2. AGILE SOFTWARE DEVELOPMENT

An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with "just enough" ceremony that produces high quality solutions in a cost effective and timely manner which meets the changing needs of its stakeholders[9][10].

Agility, for software development organization, is the ability to adopt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment [12]. An agile process is one that readily embraces and supports this degree of adaptability. So, it is not simply about the size of the process or the speed of the delivery; it is mainly

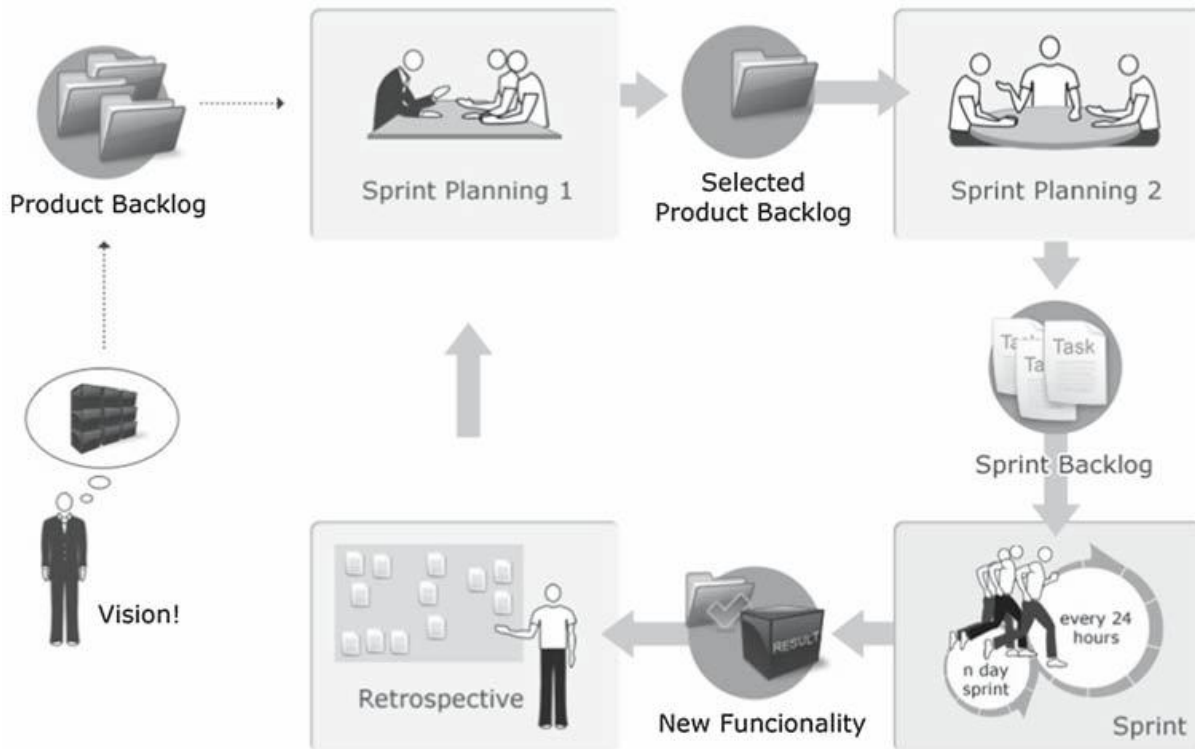
Eight Agile Methodologies had been studied and surveyed. They are Extreme Programming, Scrum, Crystal Methodologies, Rational Unified Process, Adaptive Software Development, Feature Driven Development, Dynamic Systems Development Method and Lean Development.

### 2.1 Scrum

Ken Schwaber first described Scrum in 1996 [1] as a process that accepts that the development process is unpredictable, formalizing the "do what it takes" mentality, and has found success with numerous independent software vendors. According to Schwaber, Scrum starts with the premise that software development is too complex and unpredictable to be planned exactly in advance[16]. The Scrum approach was developed for managing the systems development process. The industrial process control theory to system development ideas are applied so that flexibility, adaptability and productivity are reintroduced. The Scrum is implemented as an iterative, incremental skeleton [16].

The Scrum Process Overview is given. (see Figure 1). A high-level vision is developed in a Scrum-based project. A list of known requirements is stored in a Product Backlog list. The customer, sales and marketing division, customer support or software developers can give their requirements. The requirements are prioritized and are divided into small time-box iterations known as Sprints and the effort needed for their implementation is also estimated.

The Sprint carries out every task in a Scrum. The development time period in a sprint is 30 days. The Product Backlog list is constantly updated with new items of requirement



**Fig 1: Scrum Process Overview [16]**

-ents. The Scrum team reviews the updated Product Backlog in each iteration. After deciding what has to be done, the Sprint Backlog is developed. During the execution of each Sprint, a daily meeting of 15 minutes are held to monitor work progress and schedule other meetings, if necessary. During everyday Scrum, each team members need to answer three questions in the meeting. They are: What have you done on this project since the last daily Scrum meeting? What will you do before the next meeting? Do you have any obstacles? At the end of each Sprint, the result is presented to the stakeholder by the team.

When it is found that the stakeholders had no more requirements, the system is now ready for release and preparation is done which includes task such as integration, system testing and documentation.

The roles that are described by Scrum are Scrum Master, Product Owner, Scrum Team, Customer, User and Management [16].

## 2.2. Extreme Programming (XP)

The evolution of XP is as a result of the problems caused by the long development cycles of traditional models [2]. XP was initially developed to get the job done [6]. The XP life cycle can be described in five phases. They are Exploration, Planning, Iterations to Release, Productionizing, Maintenance and Death (see Figure 2)[2].

In exploration phase, the XP team listens to the customers about the requirement of their software and the technical team understands the business context for the software and the required output and major features and functionalities. The

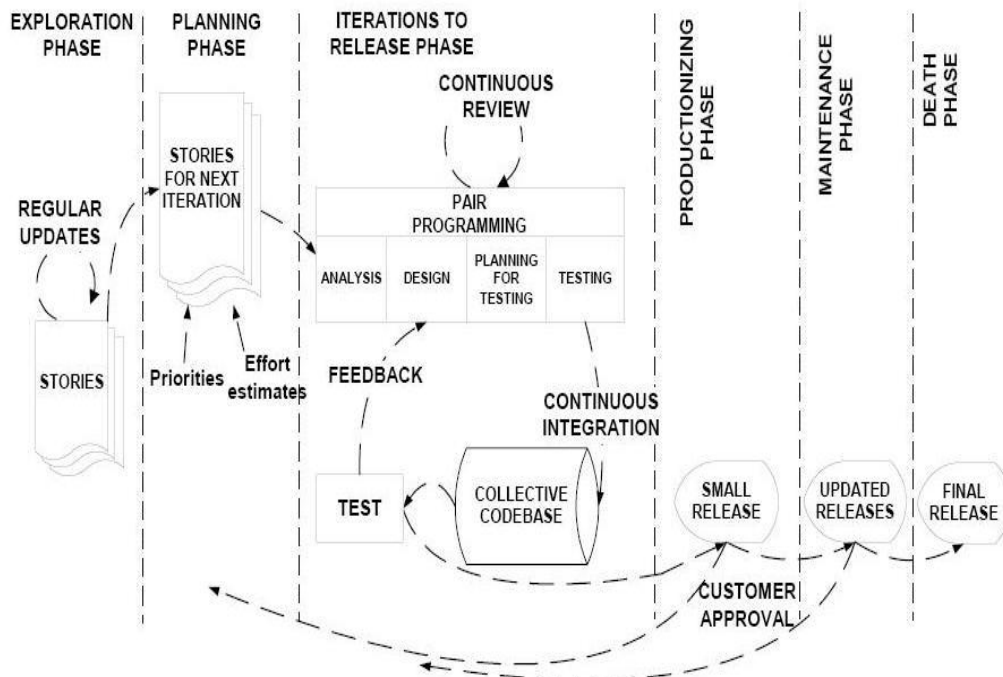
requirements of customers lead to creation of user stories which describes the required output, features and functionality for the software that will be built. The customer writes the stories and is placed in the index card. The XP team members assign a value to the stories based on the priority on overall business value of the feature and function. The schedule set in planning stage is broken down to several iterations before the first release.

In productionizing phase, extra testing is done and checking of the performance of the system before it is released to the customer. In maintenance phase, an effort is required for customer support tasks. When the customer has no stories to be implemented, it enters into death phase.

The roles that are described by XP are Programmer, Customer, Tester, Tracker, Coach, Consultant and Manager (Big Boss) [2].

## 2.3. Crystal Family of Methodologies

The crystal family of methodologies includes family of methodology and is used for selecting the most suitable methodology for each individual project [4]. The Crystal family members are marked with color indicating the 'heaviness' of the project. Crystal Clear and Crystal Orange are the two family members that have been constructed and used. The darker the color the more heavier is the methodology. The project size and criticality defines as which methodology color of Crystal is best suited. There are currently three main crystal methodologies constructed: Crystal Clear, Crystal Orange and Crystal Orange Web need to follow the seven properties of the methodologies.



**Fig 2: Life Cycle of the XP Processes [2]**

Every project which wants to follow Crystal Clear and Crystal Orange need to follow seven properties. They are: Frequent Delivery, Reflective Improvement, Osmotic Communication, Personal Safety, Focus, Easy Access to Expert users and Technical Environment with Automated tests, Configuration Management and Frequent Integration [5].

In frequent delivery, the developers frequently deliver the usable code to the customers. Reflective Improvement describes the reflective improvement of the processes which defines how can we make this suit us better. The Osmotic communication defines having the development teams close to each other so that the people don't have to go a long way to raise or answer questions i.e. all the teams of a particular project are placed in the same geographical location. The Personal safety refers to the team members that they feel themselves to be a part of the team and feel able to contribute without fear or reprisal. The focus defines that the members totally concentrates on the task, they have in hand. Easy Access to Expert users defines that if the expert users are easily available to the team, they can answer questions and deliver feedback on quality and design decision. Technical Environment with Automated tests, configuration Management and Frequent Integration defines that a proper technical environment should be present so that testing and configuration management /version control tasks do not have to be done by hand and will make life easier for developers.

In crystal clear, the main roles are sponsor, senior designer programmer, designer programmer and user [5].

## 2.4. Rational Unified Process:

The Rational Unified Process or in short RUP is a software design methodology created by the Rational Software Company. RUP is an industry oriented software development methodology which is iterative in nature. It is basically for object oriented system and is developed to complement Unified Modelling Language (UML). The development of software is basically inclined towards object-oriented concept [8].

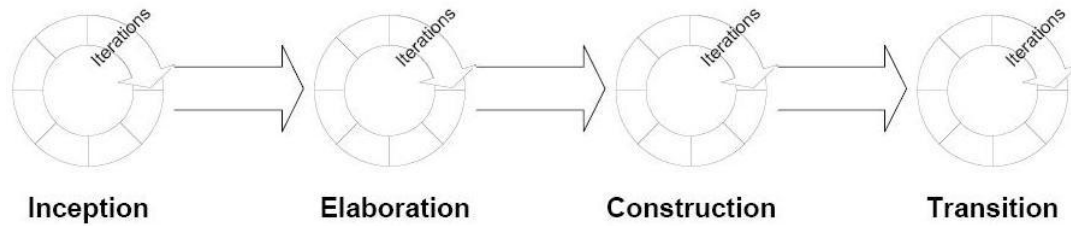
There are four phases in the life span of a RUP project. They are Inception, Elaboration, Construction and Transition. These phases are split into iteration. (see Figure.3 )

In Inception phase, every stakeholder for e.g. end user, purchaser, or contractor are considered so that their needs are reflected in the life cycle objectives of the project. Critical use cases are also identified which drives the functionality of the system.

In Elaboration phase, the software architecture is being described out here. The project plan is defined by analyzing the problem domain. The requirements and the plans are assumed to be stable in the elaboration phase. The phase identifies and describes most of the use cases and all users. The prototype of the architecture has been created after the identification of the architecture of the software in this phase. At the end of the phase, it visions the expenditure of the resources with respect to what was planned initially.

In Construction phase, the components that are remained out and the application features are developed, and are integrated

to product and are tested. This phase is also considered as



**Fig 3: RUP Phases [11]**

manufacturing phase. It emphasizes on managing resources, controlling costs, schedules and quality.

In Transition phase, the product reaches to a stage where the user community is able to use it with no more requirements for the product.

RUP defines 30 roles as workers. They are Architect, Designer, Design Reviewer, Configuration Manager, etc[11].

**2.5.Adaptive Software Development (ASD)**

The Adaptive Software Development (ASD) was developed based on rapid application development. It emphasizes on the continuous adaptation of the process.ASD addresses the economy of increasing returns i.e. the economy characterizes high speed and high change which other traditional approaches are unable to handle it[7].ASD methodology encourages incremental, iterative development with constant prototyping.

There are three-phase cycles in ASD (see Figure 4). They are Speculate, Collaborate and Learn. They are shown in the fig below. The Project Initiation phase defines the objectives and goal of the project. The overall schedule, as well as the schedules and objectives of the development cycle are fixed in the Project Initiation phase. The ASD is component-oriented i.e. they focus on the results and quality. The Collaborate phase undergoes adaptive development cycles where several components undergo development concurrently. The quality

review is done with the presence of the customer as a group of experts which focuses on the functionality of the software developed during the cycle. The final stage of ASD is the Final Q/A and Release stage. The sequence of the phases are not so important in ASD what’s important is the learning of the lessons.

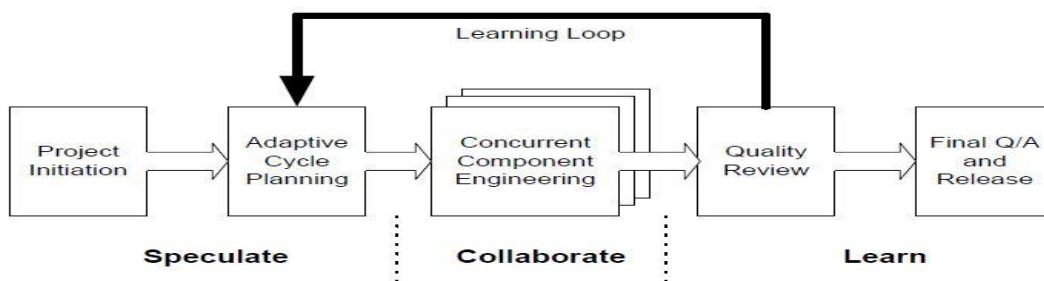
The roles of ASD are Executive Sponsor, a facilitator to plan and lead the session, a scribe to take the minutes, the project manager, and customer and developer representative [5].

**2.6.Feature Driven Development (FDD):**

Feature Driven Development is an agile and adaptive process that focuses on the design and building phases. The FDD approach is designed to work with any software development project .The approach doesn’t require any specific process model and follows iterative development.

The FDD approach consists of five sequential processes. The designing and building is carried out in these processes which are iterative in nature and have quick adaptations to late changes in requirements and business needs(see Figure 5)[11].

The first phase known as Develop and Overall Model, the domain experts begins developing the overall model, where the scope, contexts and requirements of the system are already known to the domain experts. Different domain areas are being created from the overall domain so that the domain members had a detailed walkthrough for each domain areas.



**Fig 4: The ASD Life Cycle Phases [7]**

In Build a Features List, a comprehensive feature list is being built for the system that is to be developed from the walkthroughs, objects models and existing requirement

documentation. The sub feature sets are created from the overall feature set which represent different activities within specific domain areas.

In Plan by feature, the features are planned out so that a high-level plan is created with the priority and dependencies of feature sets and is assigned to Chief Programmers.

In Design by Feature and Build by Feature, the Class Owners selects a small group of features from the feature set(s) and feature teams in order to develop selected features. The design by feature and build by feature are iterative when the selected features are produced. Multiple feature teams are designed and built concurrently. The features that are completed after successful iterations are promoted to the main build. The iterations again begins with the designing and building of new group of features are taken from the feature set.

The roles that are present in FDD are Project Manager, Chief Architect, Development Manager, Chief Programmer, Class Owner, Domain Expert, Domain Manager, Release Manager, Language Lawyer/Language Guru, Build Engineer, Toolsmith, System Administrator, Tester, Deployer and Technical Writer [13].

### **2.7. Dynamic Systems Development Method (DSDM)**

DSDM is a framework for RAD development which is non-profit and non-proprietary. The main concept behind DSDM approach is that it prefers to fix time and resources and then accordingly adjust the functionality [17].

DSDM consists of five phases (see Figure 6). They are feasibility study, business study, functional model iteration, design and build iteration, and implementation [17].

In feasibility study, the project is assessed for the suitability of DSDM. In business study phase, the analysis of essential characteristics of business and technology is done. The first

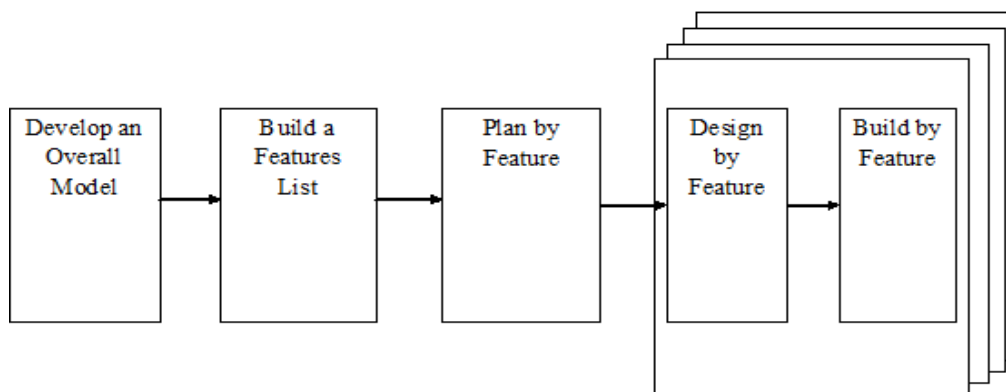
iterative and incremental phase is the functional model iteration phase. At first the functional prototype is created and the developers gain experiences from them and are used in improving the analysis model. The output results in a functional model which contains the prototype code and the analysis model. In this phase, testing is also a very mandatory part. At different stages in the phase, there are further four outputs. They are Prioritized functions, Functional prototyping review documents, on-functional requirements and Risk Analysis of further development. In design and build iteration phase, the system is built in this phase. In final implementation phase, the system is shifted to the production environment.

The DSDM have Developers and Senior Developers, Technical Coordinator, Ambassador User, Adviser User, Visionary and Executive Sponsor as roles in the approach [17].

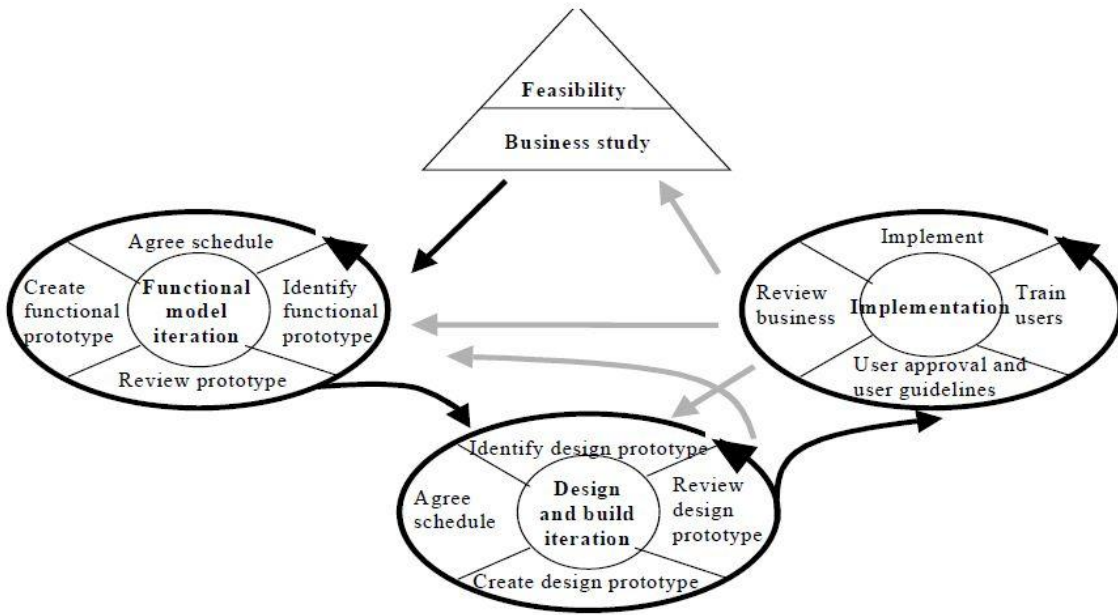
### **2.8. Lean Software Development**

The objective of Lean Software Development is to achieve breakthrough quality, savings, speed, and business values in a project. There are seven lean principles that have revolutionized the manufacturing and research and development (R & D) [14]. The seven principles are described as follows.

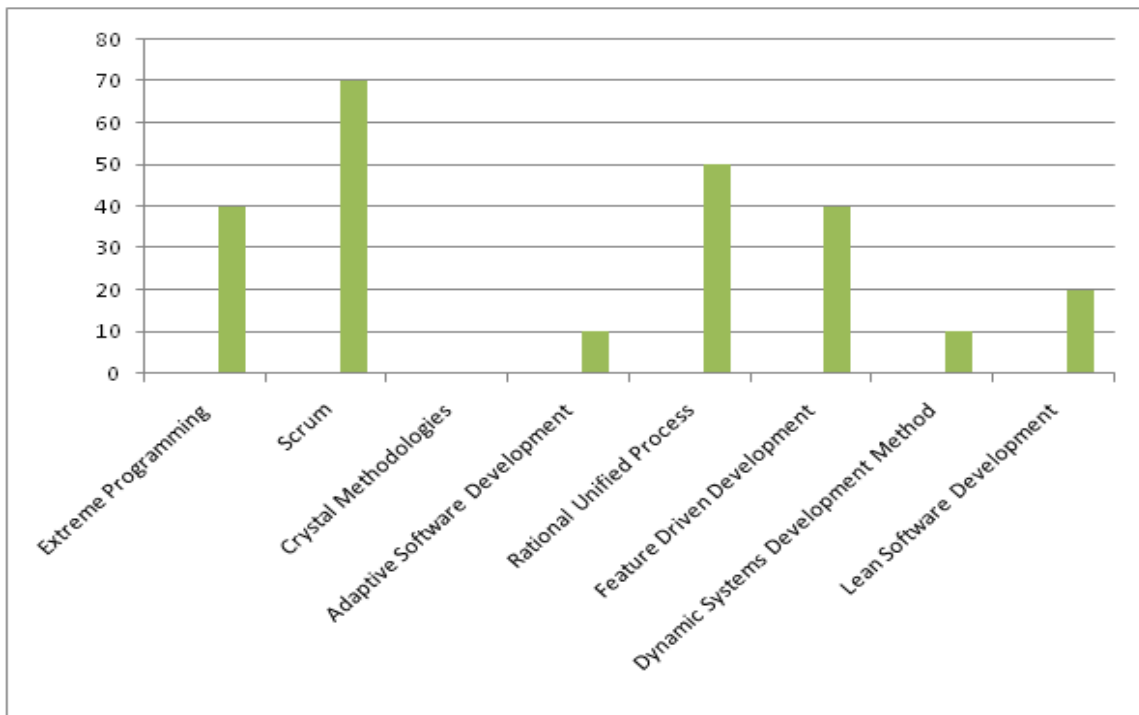
Elimination of waste defines anything that doesn't add value to a product i.e. value as perceived by the customer. Amplify learning helps software development teams to improve the software development environment. When the domains involve uncertainty, it is better to take decisions as late as possible because decisions that are made by delaying are based on fact and not on speculation. Deliver as fast as possible speed assures what customer needs today and not what they wanted yesterday. Empower the team by involving



**Fig 5: Processes of FDD [13]**



**Fig 6: DSDM Process Diagram [17]**



**Fig 7: Percentage of use of Agile Methodologies in SMEs of India**

by involving the people i.e. the developers who is actually doing the work in order to achieve excellency in work. Building integrity in a system defines that a user gets from the system what he/she actually wants and to get integrity for products in the market depends on customer perception overtime. It is necessary to focus equally on different diverse

areas of the system so that it results the overall system performance [15].

### **3. SURVEY OF THE USE OF AGILE METHODOLOGIES**

A survey is carried on the use of Agile Methodologies in 18

SMEs of India. It has been seen that the SMEs are using to a great extent the agile methodologies in order to improve their productivity and quality. Eight Agile Methodologies had been studied and surveyed. They are Extreme Programming, Scrum, Crystal Methodologies, Rational Unified Process, Adaptive Software Development, Feature Driven Development, Dynamic Systems Development Method and Lean Development.

A questionnaires concentrating rigorously on software process requirement engineering, software process models, software teams, software metrics, software planning and development, software process improvement models and software development documentation. The questions were filled up by Software Developers, Software Project Managers, Software Product Managers and Project Leaders. The figure shows the percentage of use of different agile methodologies in different SMEs of India(see Figure 7) .It has been seen that about 70 percent of the industries are using Scrum. Most of the SMEs use the agile tools.But in case of Scrum, it is seen that many industries use the concept of Scrum, they don't use any Scrum tool.

#### **4.CONCLUSION**

This paper explored the answer to the trends being followed by small and medium scale enterprises in India for software development. Limited resources, costly Software Process Improvement Model proved a bottleneck in success for SMEs. As shown in Fig.7: Percentage of use of Agile Methodologies in SMEs of India, Scrum is favourite for most of the SMEs in India.Extreme Programming is used by those SMEs which undergoes refactoring in their organization. And these organizations are doing well in Software Process Improvement (SPI) also with Scrum instead of limited resources. Only the different Scrum Team Requirement ask for more human resources in SMEs which they are managing by making smaller teams i.e. they make all different teams needed but allotting less number of persons per team. The Agile Methodologies produce small software release with rapid cycles. As the customers and developers are in communication with each other, the customers frequently know about the progress report of the software that is being asked by them to be made and if they want any changes in the requirement they can ask the developer .The main advantage of Agile Methodologies is that the customers can ask for late changes in requirement also. As the market demands changes frequently the customer can change the requirement of the software in accordance with the market demand. Concluding, small and medium software organizations have found a superb development methodology i.e. Agile Methodology

#### **5. REFERENCES**

[1].Advanced Development Methods.1996.Controlled chaos: living on the edge, <http://www.controlchaos.com/old-site/ap.htm>.

- [2].Beck, K. 1999.Embracing Change in Extreme Programming, IEEE Computer, Vol.32 (10), pp.70-77.
- [3].Cockburn ,A.1998.Surviving Object-Oriented Projects: A Manager's Guide. Addison Wesley Longman.
- [4].Cockburn, A, 2002. Agile Software Development. Boston, Addison Wesley.
- [5].Cockburn, A, 2002a. Agile Software Development. Boston, Addison Wesley.
- [6].Haungs, J. 2001..Pair Programming on the C3 project, Computer, Vol.34 (2),pp.118-119.
- [7].Highsmith,J.A. 2000. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. New York, NY, Dorset House Publishing.
- [8].Jacobsen,I.,Christerson M.,Jonsson,P. and Overgaard,G.1994.Object-Oriented Software Engineering :A Use-Case Driven Approach. Reading, MA, Addison-Wesley.
- [9].Larman,C.2004.Agile and Iterative Development:A Manager's Guide.Addison-Wesley
- [10].Larman,C. and Basili,V.2003.A History of Iterative and Incremental Development.IEEE Computer,Vol.36(6),pp.47-56.
- [11].Kruchten, P.2000.The Rational Unified Process: an Introduction. Addison-Wesley.
- [12].Kruchten, P.2001.Agility with the RUP. Cutter IT Journal.Vol.14 (12), pp.27-33.
- [13].Palmer, S.R. and Felsing,J.M.2002.A Practical Guide to Feature-Driven Development .Upper Saddle River, NJ, Prentice Hall.
- [14].Poppendieck M. and Poppendieck T.2003.Lean Software Development: An Agile Toolkit.Addison-Wesley.
- [15].Schwaber,K. and Beedle,M.2002.Agile Software Development with Scrum,Upper Saddle River,NJ,Prentice Hall.
- [16].Schwaber, K.2004. Agile project management with Scrum, Microsoft Press, Redmond.
- [17].Stapleton, J.1997.Dynamic Systems Development Method-The method in practice. Addison-Wesley.