© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 06 Issue: 02 | May-June 2015

Investigation into Agile Methodology Practices in Software Development Process

Poonam Punia
MCA
Email id- *poonamgill25@gmail.com*

Abstract: *In the traditional software development processes, the customer's first touch to the developed software is in the end of the project, start falling apart. Customers expect the software to conform to their needs, even though they are not always able to define them exactly beforehand. New approaches are required and these new approaches are called Agile Methodologies to remedy these problems. Agile Methodologies based development has been widely accepted in both academia and industry for building software systems. The comparison of agile methodologies is also done on the basis of various parameters. So here a methodology is proposed which exhibits the features like small team size, having design phase, more adaptable to changes, work week according to the project requirement, proper documentation and supporting code-refactoring with code ownership.*

Keywords: AMDD, TDD, DSDM, BDD, CC, EVO, IID, FDD, JIT.

I. Introduction

Agile software development is a group of software development methods in which needs and solutions evolve through collaboration between cross-functional teams, self-organizing. It promotes adaptive planning, evolutionary development, continuous improvement, timely delivery, and encourages rapid and flexible response to change.

II. Agile principles

The Agile Manifesto is based on 12 principles:

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstance

III. Predecessors

Incremental software development methods trace back to 1957. In 1974, E. A. Edmonds had written a paper that introduced an adaptive software development process. Concurrently the same methods were developed and deployed by the New York Telephone Company's Systems Development Center under the direction of Dan Gielan. In 1970s, Tom Gilb started publishing the concepts of evolutionary project management (EVO), which has evolved into *competitive engineering*. During the mid to late 1970s, Gielan lectured extensively throughout the U.S. on this methodology, its practices, and its applications.

A collection of *lightweight* software development methods evolved in the mid-1990s in reaction to the

perceived *heavyweight* waterfall-oriented methods, which critics called heavily regulated, regimented, and micro-managed; although some proponents of these lightweight methods contended that they were simply returning to earlier software practices. These lightweight methods included: from 1994, unified process and dynamic systems development method (DSDM); from 1995, scrum; from 1996, crystal clear and extreme programming (aka "XP"); and from 1997, adaptive software development and feature-driven development. Although these originated before the publication of the Agile Manifesto in 2001, they are now collectively referred to as agile methods; and often abbreviated loosely as *Agile*, with a capital *A*, although this is progressively becoming deprecated.

IV. Agile practices

Agile development is supported by a group of concrete practices, covering areas like requirements, design, modeling, coding, testing, project management, process, quality, etc. Some notable agile practices include:

- Agile model-driven development (AMDD)
- Agile modeling
- Backlogs (Product and Sprint)
- Behavior-driven development (BDD)
- Cross-functional team
- Continuous integration (CI)
- Features-driven development (FDD)
- Information radiators (scrum board, task board, visual management board, burn down chart)
- Iterative and incremental development (IID)
- Pair programming
- Planning poker
- Refactoring
- Scrum events (sprint planning, daily scrum, sprint review and retrospective)
- Test-driven development (TDD)
- Agile testing
- Time boxing
- Dynamic system development(DSD)
- Crystal Clear (CC)
- Story-driven modeling
- Retrospective
- Velocity tracking

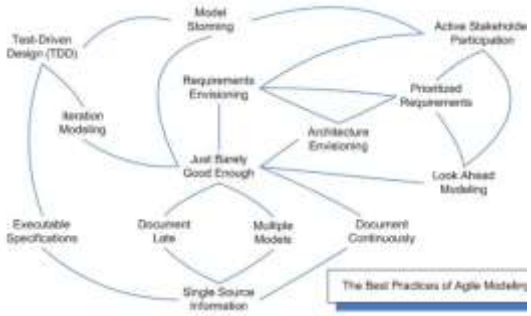

Fig 1.

V. Agile Model Driven Development (AMDD) Lifecycle

Envisioning

- Identify the High-level scope
- Identify initial "Requirement Stack"
- Identify an architectural vision

Iteration Modeling

- Modeling is part of iteration planning effort
- Need to Model enough to give good estimates
- Need to plan the work for the iteration

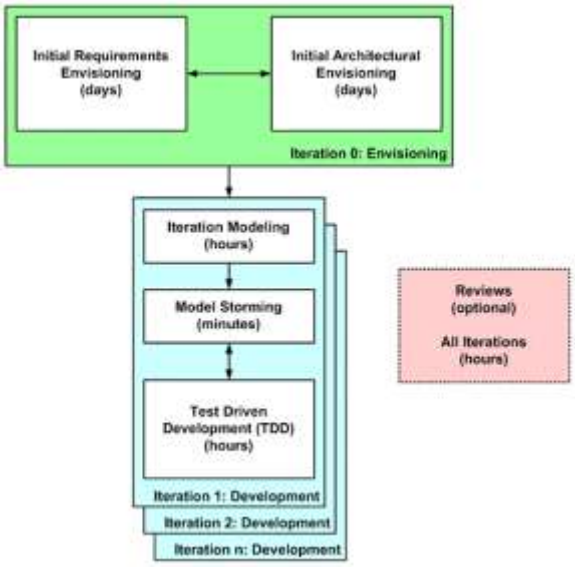

Fig.2

Model Storming

- Work through specific issues on a JIT manner

- Stakeholders actively participate
- Requirements evolve throughout project
- Model just enough for now, you can always come back later

Test Driven Development

- Develop working software via a test-first approach
- Details captured in the form of executable specifications

VI. Feature Driven Development

The Feature Driven Development method was originally conceived by Peter Coad and his colleagues as a practical process model for object oriented software engineering. Stephen palmer and John Felsing have extended and enhanced Coad's work, describing an adaptive, agile process that can be applied to moderately sized and larger software projects.
It has some common features with XP such as short iterations, customer representative on the team & using the notion of features which are comparable to the XP. FDD feature short iteration cycles i.e. two weeks. Prioritization is encouraged in FDD and it is ensured that most valuable features are delivered.

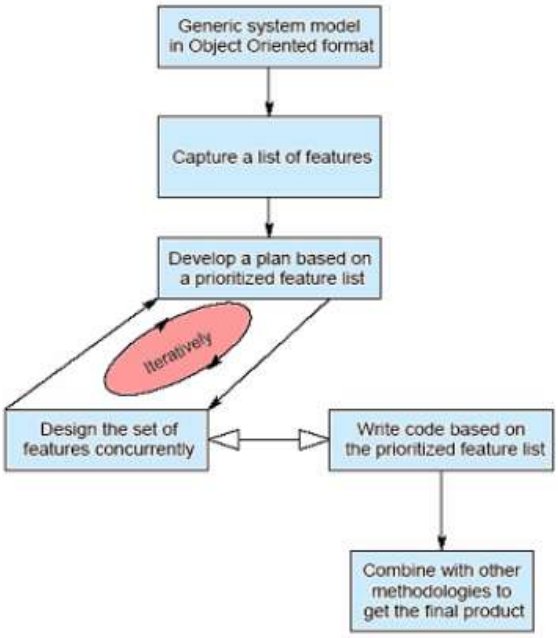

Fig. 3

VII. Dynamic Systems Development Method

The Dynamic Systems development Method (DSDM) is a lightweight software methodology which has its origins in the U.K. Here time is fixed[19,22] for the life of a project, and resources are fixed as far as possible. It means that the requirements that will be satisfied are allowed to change .DSDM is concerned on business value and customer solutions. The life cycle of DSDM consists of five phases. In DSDM, implementation indicates the transition from the development environment to the operational environment rather than coding. The system is handed over to the users.

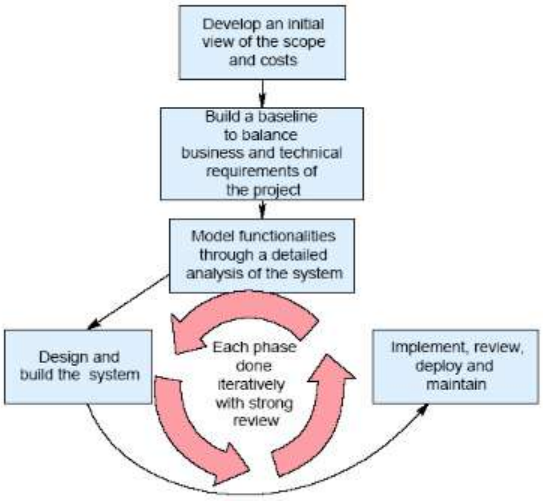

Fig. 4

Scope

DSDM has been applied to mini and major projects and mainly applicable to the development of user interface intensive systems and complicated business applications, but can be used for non–IT projects. Team size differs from two to six people but there may be many teams in a project.

VIII. Crystal Clear (software development)

Crystal Clear is a member of the **Crystal** family of methodologies as described by Alistair Cockburn and is considered an example of an agile or lightweight methodology.
Crystal Clear can be applied to teams of up to 6 or 8 co-located developers working on systems that are not life-critical. The Crystal families of

methodologies focus on efficiency and habitability as components of project safety. Crystal Clear focuses on people, not processes.

.
Crystal Clear requires the following properties:

- Frequent delivery of usable code to users
- Reflective improvement
- Osmotic communication preferably by being co-located

Crystal Clear additionally includes these optional properties:

- Personal safety
- Focus
- Easy access to expert users
- Automated tests, configuration management, and frequent integration

IX. Criticism

Agile methodologies can be very difficult for large organizations such as multinational banks and governments to constantly adopt, for reasons ranging for lack of sponsor buy-in to agile, to refusal to notice agile consultants' advice on co-located teams - and particularly in the case of governments, stereo type procurement and project management policies that assume non-agile methodologies.

Agile methodologies can be incompetent in big organizations and certain types of projects. Agile methods seem best for developmental and non-sequential projects. Many organizations believe that agile methodologies are too extreme and adopt a hybrid approach that mixes elements of agile and plan-driven approaches. However, DSDM is an agile methodology that in fact mixes elements of agile and plan-driven approaches in a organized and disciplined way, without sacrificing the fundamental principles that make agile work.

The term "agile" has also been criticized as being a management fad that simply describes existing good practices under new jargon, enhance a "one size fits all" mindset towards development strategies, and wrongly emphasizes method over results.

References

1. A.Khan, S.Babloo," A Tale of two Methodologies: Heavyweight versus Agile", The Tenth Australian World Wide Web Conference, from 3-7 July, 2004.
2. Addicam.V.Sanjay," Overview of Agile Management & Development Methods", The Project Perfect White Paper Collection, 2005.
3. Alexandre Magno Figueiredo," An Executive Scrum Team", IEEE Agile Conference 2009.
4. Asif Qumer, Brian Henderson-Sellers," Six Aspects of an agile software development methodology" ,Proceedings of European and Mediterranean Conference on Information Systems , June 24-26 2007.
5. Badr, I, Rapid Development through Agile Modeling. Telelogic AB, 2006. Viewed 04 June 2006. http://www.telelogic.com.
6. C.Balan et.al. , "The Need To Adopt Agile Methodology In The Development Of Cyber Forensics Tools, IEEE, 2010.
7. David Fox et. al., "Agile Methods and User-Centered Design: How these two methodologies Development Methodology", Journal of software, Vol. 3, No. 4,April 2008.
8. Glen Litton Van der Vyver et. al., "Agile methodologies and the emergence of the agile organization: A software development approach waiting for its time", 7th Pacific Asia Conference on Information Systems, 10-13 July 2003.
9. Hamed Yaghoubi Shahir et. al., "Improvement Strategies for Agile Processes: A SWOT Analysis Approach", IEEE-Sixth International Conference on Software Engineering Research, Management and Applications, pg. 222-223, 2008.
10. Hesam Chiniforooshan Esfahani et. al., "Adopting Agile Methods: Can Goal-Oriented Social Modeling Help", IEEE, 2010.

11. James E. Tomayko, "Engineering of Unstable Requirements Using Agile Methods".

12. Jeff Sutherland et. al., "Distributed Scrum: Agile Project Management with Outsourced Development Team", IEEE Proceedings of the 40th Hawaii International Conference on System Sciences-2007.