



Analysis of the Different Approaches for Software Project Management

Adeeti P. Gosavi, Jayshree B. Pandav
Assistant Professor, MCA Department,
G S Mandal's Marathwada Institute of Technology,
Aurangabad, Maharashtra, India

Abstract: To develop a project all activities in the project are supposed to be planned. Management involves activities like planning, organising, staffing, directing, and monitoring.

Traditional software development includes methods which are used from decades. Agility is dynamic, content specific, aggressively change embracing and growth oriented. There are various tools and techniques available for developing software projects. Various factors like team size, project size, requirement etc. are considered to choose method for developing software.

Keywords – Agile, Scrum, Extreme Programming, User stories, Spike solutions.

I. INTRODUCTION

The dictionary definition put a clear emphasis on the project being a planned activity [1].

Project Management is the process and activity of planning, organizing, motivating, and controlling resources, procedures and protocols to achieve specific goals in scientific or daily problems[2].

There are two types of project management techniques

1. Traditional Project management
2. Modern Project management

The traditional project management uses conventional methods in software project management process.

II. TRADITIONAL PROJECT MANAGEMENT

Developing software by following tools and techniques used by decades is said to traditional software development [2].

Traditional project management uses old methods in the management of process; these methods are introduced and used by decades. But these methods are not applicable for all types of projects.

Steps involved in the traditional project management are initiation, planning, design, monitoring, controlling and closing.

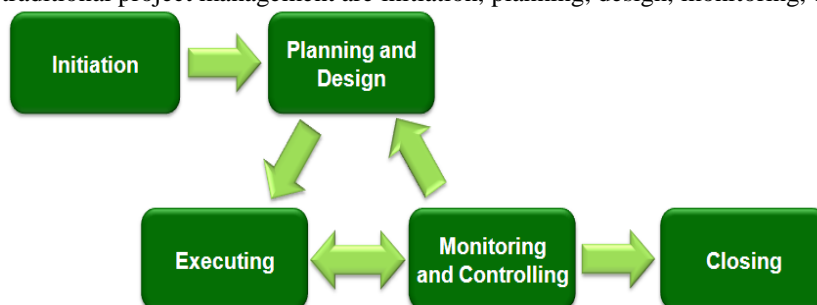


Fig. Typical development phases of an engineering project.

Some of the traditional management techniques are listed here

1. Waterfall model

Waterfall model follows sequential steps. Complete software requirements are gathered at initial steps. This model develops software systematically from one phase to another phase.

The most probable phases through which it progresses downwards are

- a) Requirements
- b) Analysis
- c) Design
- d) Coding
- e) Testing
- f) Maintenance.

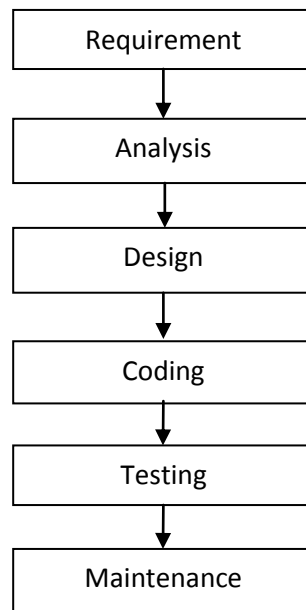


Fig. General steps in waterfall model

Developers(s) can move to next step only after completing the previous one. If any step faces problem it will not allow to move to the next step. Waterfall model is straight forward model which lets one know which step is in the progress.

Disadvantages: at the time of designing software project if there is change in requirements at any step, waterfall model does not allow step back to change requirements. Waterfall model is not at all cost effective, because it is not possible to find cost of every step as it is big. If in case change in further step will change previous step and it will waste the efforts, cost, time etc. factors required to design that step. Testing step comes very late in the waterfall model. So if testing fails one need to change previous steps, which is not possible in waterfall model.

2. Spiral Model

Spiral model is risk a driven process model.

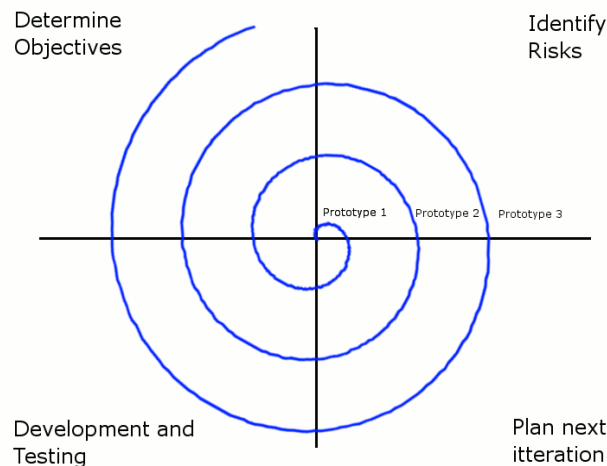


Fig. Spiral Model

Spiral model uses iterations. In first phase of planning, objectives are defined. Next phase analyzes alternatives and attempts to identify the risk involved. After analyzing risks focus is given on product development and testing. Finally, after successful testing product is delivered to the customer. An important feature of spiral model is that each phase is completed with a review by the people concern with the project. Review consists of all the products developed up to the review point and also includes plans for next phase.

III. AGILE PROJECT MANAGEMENT

Agile software development methodologies became popular in 1990s. The meaning of Agile is quick/rapid. Agile methods have been very successful for some types of system development [3]

1. Product development where a software company is developing a small or medium sized product for sale.
2. Custom system development within an organization, where there is a clear commitment from customer to be involved in the development process and where there are not a lot of external rules and regulations that affect the software



Fig. Agile methodology for software development

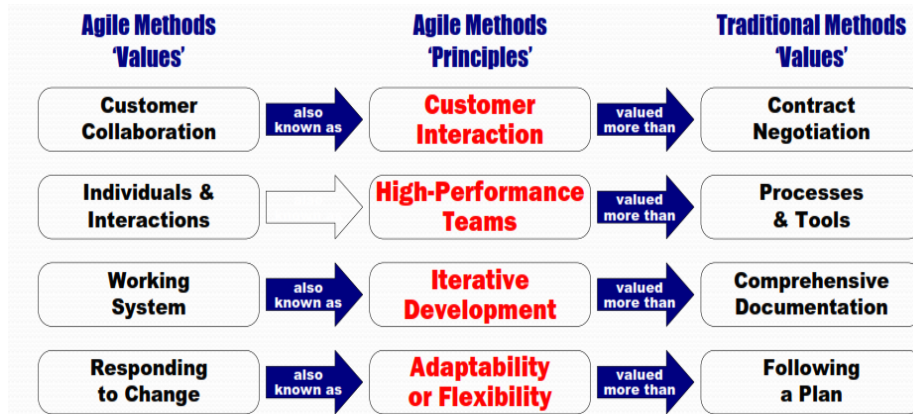


Fig. Manifesto for agile software development

Customer collaboration, individuals & interactions, working system and response to change are some of the values compared with traditional methods in above diagram.

Highest priority is to satisfy the customer through early and continuous delivery of valuable software. In agile methods changes in requirements are accepted even late in the development. In traditional methods changes in requirements is a tedious job. Agile techniques delivers software frequently after couple of weeks, or couple of months for getting feedback from customer time to time, rather than delivering software product finally.

A task accomplished by the team is reflected at regular intervals on which team decides further flow of development to become more effective and adjust development schedule accordingly.

Daily meetings are held to determine days work. Team members and customers communicate with each other openly and express their views. These views are taken into consideration at the time of developing software.

1. Scrum

Scrum is lightweight agile software development technology which concentrates particularly on how to manage tasks within a team-based development environment. Scrum manages and controls interactive and incremental projects.



Fig. The scrum process

A software development method originally proposed by Schwaber and Beedle in early 1990[2]

There are three different users in scrum

- 1) Product Owner: is a stakeholder who discuss his software product requirements with developers and provides necessary inputs to development team
- 2) Scrum Master: Work as a Team Leader for product development and is responsible for approving Scrum. Their main job is to remove obstacles, i.e. project related issues that might slow down or stop activity that moves the project forward.
- 3) Project Team: Consist of 5 to 10 members. The team consist of cross-functional, team members and involves individuals from a various disciplines like QA people, Programmers, User Interface Designers, and Testers etc.

• The Process

Stake holders requirements are collected and named as product backlog which need not to be precise.

Product backlog is being prioritised by customer

: items of importance to the project/business, i.e. those items that add immediate and significant business value, are bubbled up to the top.[4]

The development team transform the requirements in terms of a software product called as a 'Sprint Backlog', comprises of Product Backlog items that can be completed within a 30 day period. After 30 days have elapsed, the team should have a 'potentially shippable product' increment which will provide something of value to the business

The Product Owner, the Scrum Master and the Project Team makes an initial pass over the Product Backlog items where they work out roughly how long each item will take. Initially, these are estimates, best guesses. [4]

Testing and documentation are on-going as the product is constructed Work units occurs in sprint and is derived from a backlog of existing prioritized changing requirements [2]

2. XP (Extreme Programming)

XP technology concentrates more on software engineering process and addresses analysis, development and test phases which improves the quality of end product.

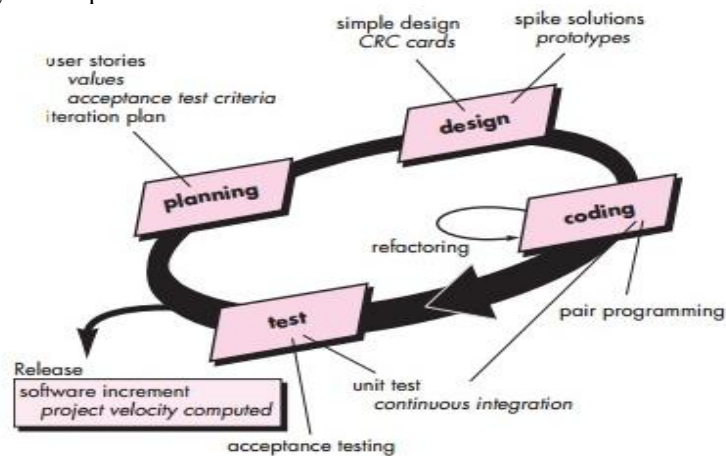


Fig. Extreme programming [roger]

• XP Planning

Starts with listening to customer requirements, which in turn leads to creation of 'user stories' that describes required output, features, and functionality and are tasks for development team[2]. Customer assigns a priority to each story. Agile team assesses each story, estimates efforts and resources required, assigns cost i. e. development weeks. If more than 3 weeks are required to develop a story, customer is asked to split it into smaller stories. Stories are grouped together for a deliverable unit.

After first increment 'project velocity is calculated as number of stories implemented during the first release and is used for defining subsequent delivery dates for other increments.

• XP Design

XP design strategy encourages the use of CRC (class-responsibility-collaborator) cards in an object-oriented context[2]. CRC identify and organize the classes that are relevant to the current software increment.

XP uses spike solutions — a design prototype for a particular user stories is created, implemented, evaluated and is used for difficult design problems.

• XP Coding

Propose the formation of a unit test for a user story before coding begins. So development team can focus on what must be implemented to pass the test. New versions of software may be built several times per day and releases are delivered to customers roughly every two weeks [4]

• XP Testing

As there is a formation of test cases before development, the developer(s) have to thoroughly understand the specification to write them. All components are unit tested generally in an automated test environment. Regression tests are conducted to test the working of current and previous components when integrated together. Finally acceptance test cases are assisted and confirmed by the customers for the stories to be implemented.

IV. CONCLUSION

Every model is having some advantages and disadvantages. It is somehow critical to choose a model for developing software. To choose software development methodology depends on type of software being developed, capabilities of development team. The software which is to be delivered must be error free and satisfy customer requirements, so beyond traditionalism and agility an importance goes to testing and goes to testing and validation. Preferring agile technology is more effective as it responds to regressive changes which software development is supposed to.

REFERENCES

- [1] Bob Hughes & Mike Cottrell, Software Project Management.
- [2] Roger Pressman, Software Engineering a Practitioners Approach, sixth and seventh edition, MacGraw Hill Publications.
- [3] Ian Somerville Software Engineering ninth Edition, Pearson Publication.
- [4] Craig Murphy, Adaptive project management using scrum, Methods and tools winter 2004.