

COMPARATIVE STUDY OF SOFTWARE DEVELOPMENT METHODOLOGIES

SARANYA P¹, MONICA V², PRIYADHARSHINI J³, ASST. PROF. DEEPA N⁴

¹ School of Information Technology & Engineering, VIT University, Tamilnadu, India.

² School of Information Technology & Engineering, VIT University, Tamilnadu, India.

³ School of Information Technology & Engineering, VIT University, Tamilnadu, India.

⁴ Asst. Prof, School of Information Technology & Engineering, VIT University, Tamilnadu, India.

Abstract - This paper concentrates on the present condition of learning in the field of programming advancement systems. It intends to set the phase for the formalization of a product advancement system devoted to development oriented IT anticipates. The paper begins by delineating particular attributes in programming advancement extend administration. Overseeing programming advancement ventures includes methods and abilities that are exclusive to the IT business. Additionally, the product advancement extend director handles difficulties and dangers that are prevalently experienced in business and research ranges that include best in class innovation. Ordinary programming advancement stages are characterized and quickly portrayed. Improvement stages are the building pieces of any product advancement procedure, so it is essential to appropriately examine this perspective. Current programming advancement strategies are displayed. Improvement stages are characterized for each displayed philosophy. For each procedure a realistic portrayal is represented keeping in mind the end goal to better individualize its structure. Programming improvement procedures are thought about by highlighting qualities and shortcomings from the partner's perspective. Decisions are detailed and an examination. Bearing went for formalizing a product improvement procedure committed to development oriented IT anticipates is articulated.

Key Words: software development, project management, development methodology

1. INTRODUCTION

THE software development project management process also includes specific features. A software project manager has to undergo many challenges during the development of software. Apart from challenges and setbacks, there are benefits which ease the burden of management.

The requirement and planning for the software projects changes frequently from the initial planning and specs.

The following are some of the reasons for the change in requirements.

- The client identifies new business opportunities and decides to add those opportunities to the software which is being developed.
- Due to the lack technical knowledge the clients are not aware of the expected outcomes of the software development project. The requirement from the clients, project owners and stakeholders are difficult to understand for the technical software developers.
- Sometimes, software developers feel difficulty in implementing the expected functionalities due to lack of technical limitations to develop a software product.
- The real time context and data changes day by day thus making the development of the software to change. New technologies emerges day by day making the clients to go for it.

Changing the requirements has the negative on the project by making it difficult to complete it by given budget and expected deadline.

The software industry develops at a very fast rate and software is destined to process large array of data. For example. PHP server-side scripting language registered 16 releases on new and improved versions in 2014[1]. The project management should be aware of emerging new technology and changing business requirements in order to meet the client's requirements.

Software development project teams contains highly trained individuals who require huge compensation. Not only having the pressure from the clients also that project manager should be able to showing best interest on the highly trained individuals in order to avoid misunderstandings that may happen throughout the development of project.

Software development projects are not done by face-to-face interaction or under the same roof. It is a global process and software developers communicate with each other throughout the process. Task assignment is done through online management tools such as Base camp etc., Code version is ensured through the versioning tools like SVN. File sharing is done through tools like Google drive or many other shared drives. Online meetings and discussions about the development and developmental status are done through Skype conference calls.

Table 1. Software development projects characteristics

Characteristic	Positive Impact	Negative Impact
Frequently changing specifications	-	results in exceeding the project budget causes stress and discontent for the development team
High dynamics of technology and standards	generates new opportunities in terms of design and coddling	software can become obsolete by the time it hits the market software developers have to invest a lot of time in researching new technologies
skilled workforce	increases the likelihood of achieving innovative results	high cost generated by human resources
globally distributed teams	work can be performed around the clock	monitoring and control becomes more difficult
	cultural diversity nurtures creativity	integrating new code is more challenging

2. Software Development Stages

Software product is a process that has several stages of development.

2.1 Research

In this stage three main people are involved one is project manager, project owner, project team they set goals and all exchanges so many information. Here, project owner play important role who decides everything to allot the positions for each of them. Project owner decides the requirements and software product also helps to reach their goal. And project owner search of related goals people or companies to achieve their goals. Now, the project managers collects the requirements from the owner and sends to project team members and manager has a responsible to evaluate both business and technical perspective and he need to search for some characteristic and patterns. The project team is ready to evaluate the collected requirements from the project manager in technical perspective. The project team will have to research the frameworks, API's, libraries, versioning tools and hosting infrastructure that will be needed accordingly to design the software product.

2.2 Planning

Without the proper plan, assuming the strength and weakness of the project of software development is full of waste and meaningless. The perfect plan makes the project in a more positive way. The elements of this project are arranged in the order to develop software product. Before it starts the project it explains the overall flow of application. Future step is to break the flow to make smaller and easier for subassemblies. For each subassembly the detailed functionality are also explained. Database structure is designed based on the functionality which is required for the project. The overall flow of application, datastructures, sub assembly, the project manager along with their project team has to choose the right technology to develop their application. And the project manager has the responsibilities to select the best methodology and the work protocol for their project.

2.3 Design

In the software development of design stage the application of layout is created. When compare to desktop application, both web applications and mobile applications gives more importance to the layout. The applications depends range from rough and functionality driven and then to complexity and artistic. An accounting application only requires some basic designs and an online museum it requires high end work. In accounting application design

is enhanced and in online museum its functionality is designed to fit in the project. The graph design stage is overlapped and important because it shows preview for application before it properly designed. They come with new requirement and that we have to submitted to research and planning.

2.4 Development stage

In the development stage where code is written and software application is designed and created. Whenever it starts the project it starts with the development environment and testing environment and both should be synchronized in same way of protocol and using the synchronization protocol code is written in development environment and it uploaded in testing environment. Progress monitoring is one of important development. The project manager should update the entire work process to the owner of this project. On the testing environment the coders who were written need to upload code with bug free error. Software developers need to comment their code to easily understand for other users and developers.

2.5 Testing:

In the testing they are two types of errors one is programming errors and another one is design error both are identified and fixed. When programming error will occur actually programming error are in scenario and it crashes applications and it is not up to the design architecture. And it is full security then if application attacks vulnerable allow the attackers to use the private data. User may have problem with slow response time with the application that is also programming error. Design error is the project manager given some design that is not implemented properly by the project team. It is hard to fix and project owner is responsible for the design and requirements application.

2.6 Setup

In live environment the application is installed. It configuring resources like security, hardware, software, backup procedures and are also defined and tested. Software product includes copying the source code, importing the database, if required install third party applications, cron-job and configuring API's. When application is installed it will move towards the full testing cycle. Content is added to the application automatically whenever testing is completed.

2.5 Maintenance:

In this stage the software development is responsible for the application. To check whether it is working in a planned and correct parameter. Check the application which is monitoring firewall, mail, HTTP, FTP, MySQL and SSH error logs. In maintenance systematically testing functionalities feature are not identified in testing .They give opportunity to add many features which we left in some stages and after adding the features immediately update to all the stages of software development.

3 Current Software Development Methodologies

The software development methodologies having several set of rules and guidelines for the research process, planning, design, development, testing and maintained of software product .this paper have 8 of the most popular software development methods and their characteristics.

3.1 Waterfall

The waterfall model is believed that the first process model which was introduced in software engineering[2] .this model is introduced by Herbert d Pennington in 1956 and in 1970 whinstone Royce published the formal version of the model .the waterfall model is used for system development .this model is strictly sequential and top-down approach. Waterfall approach is for small, medium size of project. This model is start from requirement followed by design ,coding, implementation and maintenance of the product .this model is mainly focused on documentation .in each stage the documentation is made and it is used as input for next stage and output of the work product is find at the end of all process. The main principle of this model is one stage to be completed before the next stage is started it is followed in all the stages.

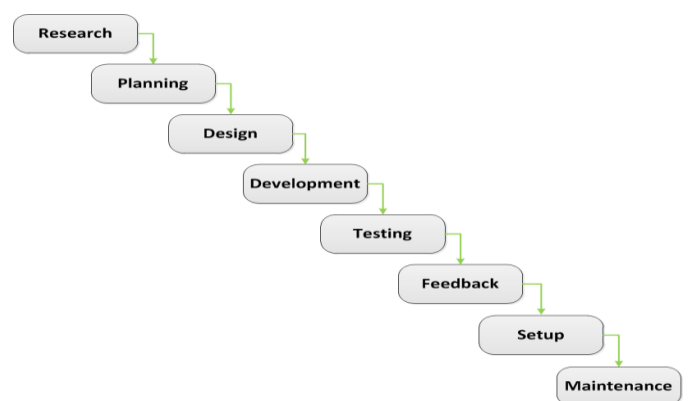


Fig-1 Waterfall methodology

3.2 Prototyping

The working model of software product with some functionality. It is mainly for satisfying the user requirements and not the developer requirements in software development. It also helps to understand the user specifics as early as possible at the starting stage it have information to build prototype. The prototype is used as interface between the development team and project team. After the prototype is build the feedback is collected from the user whether it satisfies their needs .it is for large project and not for the static requirements.

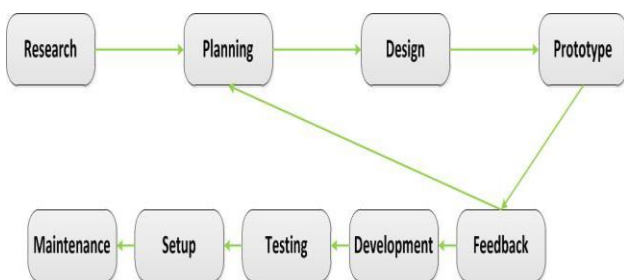


Fig- 2 Prototyping methodology

3.3 Incremental model

Incremental model can be used when the requirements are clearly defined. The fully defined requirement are divided into small modules .this model is similar to water fall model .each module passes through requirement, design, implementation and testing phases. Software developed quickly in software life cycle. It is more flexible and cost is less expensive to change requirement and scope. Easy to test and debug in smaller iteration. Customer responds to each module. Initial delivery cost is less. Easy to manage risk and are identified during iteration. It needs good planning and design. It needs a clear definition of entire system before it can be broken down into modules. It costs higher than waterfall model.

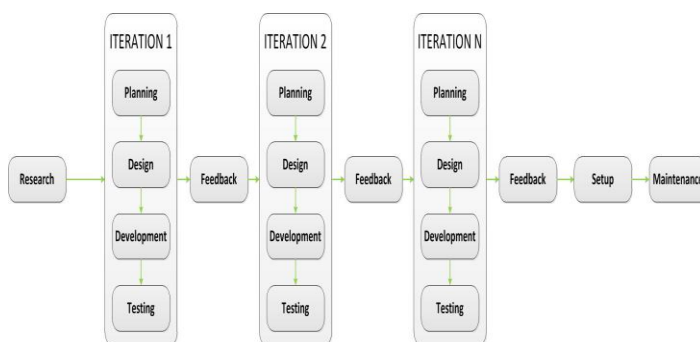


Fig-3 Iterative and incremental methodology

3.4 Spiral Model

It is similar to incremental model .it has four phases: planning, risk analysis, development and evaluation. A project repeatedly passes through these phase is said to be spiral model. In planning requirements are gathered. In risk analysis phase, a process is used to identify risk and other alternative solution[5]. A prototype is produced at end of this phase. If any risks are identified then alternative solutions are suggested and implemented. In development phase, software is developed with testing at the end of phase. Evaluation phase allows evaluating output of project to date. The spiral model is used in medium and high risk projects[3

]. It also used when user is not sure about need and the requirements are complex. Risk analysis is used to identify risk. It is good for large projects. Proper risk evaluation software is produced early. Changes are implemented faster. It is costly for smaller. Risk analysis requires expert peoples. Documentation is more when it has intermediate phases.

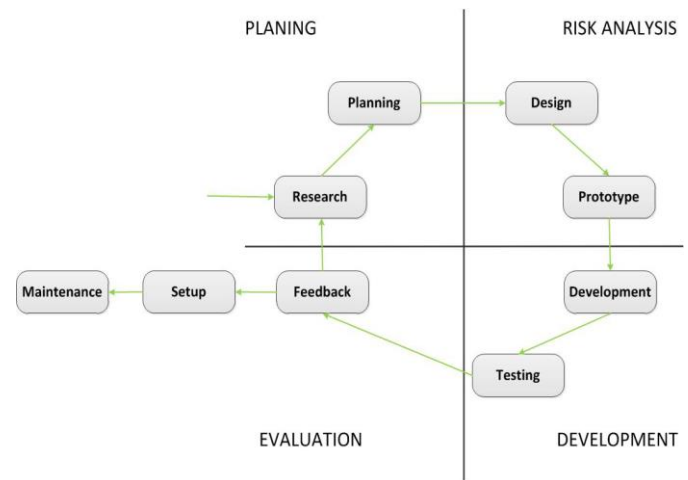


Fig-4 Spiral Methodology

3.5 Rapid Application Development

Rapid application development as a system development methodology that was developed to respond the delivery system very fast[6]. In this model mainly concentrated on development then the design and planning .this method is for small, medium and large project. The developers and user will work together in development of project the knowledge gathered from the development is used as feedback for requirement and design .the combination of join application development (JAD) and case tools to

convert the user needs into designs. In rapid the multiple projects can be done within time and budget.

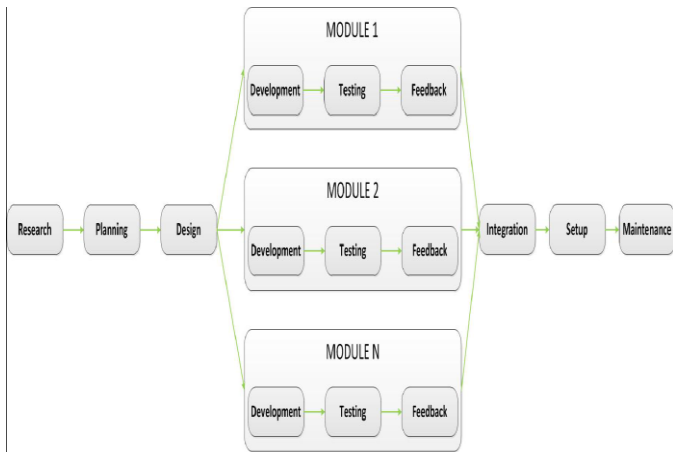


Fig-5 Rapid application development methodology

3.6 V-model

V model is derived from the waterfall model. In which each phase is validated with their requirement[7]. The user will give the feedback after application is developed in acceptance testing .it is suitable for small and medium projects.

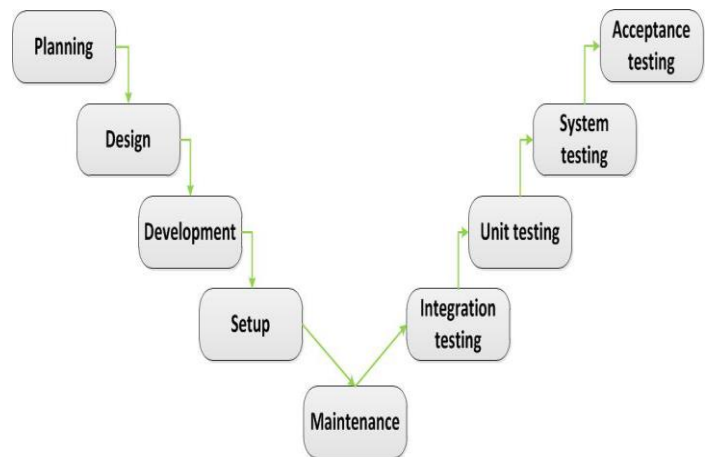


Fig-6 V-Model methodology

3.7 Scrum

Methodology for developing the application in complex environment .the requirements are partition and prioritized by the user requirement and it is known as stories. Product backlog is made by stories [8].this methodologies is work based on time box approach in development cycle and it is called sprints. it will take four weeks and end with a working version of the application the sprint backlog is made by all stories .daily scrum –is process is assessed in daily meeting for 15 min to know the undone task by the individual or project manager. The team efforts are track by scrum master the feedback is collected after the end of each sprint. It is suitable for small, medium and large scale project.

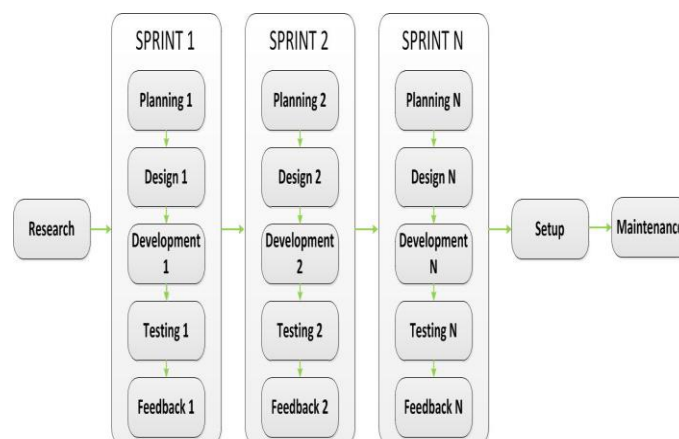


Fig-7 Scrum methodology

3.8 CLEANROOM

In the clean room process based on the software development on formal methods and the software tool support mathematical formalism which includes model

checking, process algebra and Petri nets. Specifying and designing a software product is box structure method. Team review performed verification on specifications. In

the incremental implementation under statistical quality control it is an iterative approach and the quality is measured against the standards. Quality is responsible for failure and it return to design phase in the statistically sound testing it is carried out by the experiment. Particular subsets of inputs and outputs are tested and selected. This sample is then statistically analyzed to produce an estimate of the reliability of the software, and a level of confidence in that estimate.

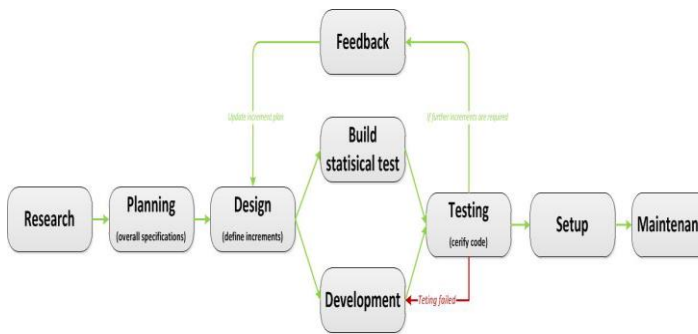


Fig-8 Cleanroom methodology

Table 2. Software development methodologies characteristics

Methodology	Characteristics	Strength	Weakness
Waterfall	<ul style="list-style-type: none"> - It is linear sequential life cycle model. - each phase has its own specific deliverables 	<ul style="list-style-type: none"> - easy to use - easy to understand each phase 	<ul style="list-style-type: none"> - risk of changing requirements - delivery of working code is late in project - it has low tolerance in design and testing errors
Prototype	<ul style="list-style-type: none"> - it build more demo versions - users are actively involved - prototype are discarded later - code is important than specification 	<ul style="list-style-type: none"> - exact identification of requirement - can get feedback earlier - missing functions can be identified earlier 	<ul style="list-style-type: none"> - it may lead to increase application complexity - more programming efforts are needed
Iterative and incremental	<ul style="list-style-type: none"> - initial model extend iteration - it gives importance to design than documentation-project owner or user involve actively 	<ul style="list-style-type: none"> - Gets feedback from project owner - it revise entire application and function - early code delivery 	<ul style="list-style-type: none"> - every iteration resemble small scale of waterfall projects
Spiral	<ul style="list-style-type: none"> - it focuses on objectives ,alternatives and constraints - it has four major phases: planning ,risk analysis, development and evaluation - It requires risk analysis - it evaluates m any alternatives 	<ul style="list-style-type: none"> - Early code delivery in project - Risk is minimized - documentation is strong 	<ul style="list-style-type: none"> - For risk handling it requires cost - it depends upon risk analysis

	before starting planning stage		
Rapid application development	<ul style="list-style-type: none"> - to focus more on develop than planning tasks - It follows time box approach 	<ul style="list-style-type: none"> - the development of application is faster - code can be reused 	<ul style="list-style-type: none"> - weak documentation - development cost is high - issues in code integration - application is converted into small modules
V model	<ul style="list-style-type: none"> - In every development stage, testing is introduced - it gives importance to maintenance 	<ul style="list-style-type: none"> - decrease in bug rate - easy to understand and use 	<ul style="list-style-type: none"> - small change lead to change still more - it relies on initial set of requirements and specification
Scrum	<ul style="list-style-type: none"> - Iterative development - time box approach is called sprints - daily meeting progress is known as scrum - has self-organizing team - tasks managed by product backlogs and sprint backlog 	<ul style="list-style-type: none"> - product delivered in short time - it enables fast feedback - Change in rapid adaptation 	<ul style="list-style-type: none"> - poor documentation - experienced developer are required - difficult to estimate overall effort required to implement large projects and cost cannot be determined accurately
clean room	<ul style="list-style-type: none"> - iterative development - box structure method - using mathematic models in quality control - statistical approach to Testing 	<ul style="list-style-type: none"> - considerable reduction in bug rate - higher quality software products 	<ul style="list-style-type: none"> - increased development costs - increased time to market for software product - requires highly skilled highly experienced developers

4 Conclusion

Software development methodology uses two methods: heavyweight and lightweight. Heavyweight requirement cannot be changed and software requires detailed planning. It is easy to use and to do implementation. The documentation has detailed information and has clear information about each and every stage of project. The project manager can easily use heavyweight methodology to track, evaluate and report about project. The project owner involved only in research and planning stages. lightweight are suitable in project where requirements are not clear and changes can be done .it follows incremental approach where software are delivered in many iteration .lightweight are easy to change .it provides delivery of working code ,self-organizing teams and adaptive planning. The project made effort in all stages when its input and feedback is critical for the success of project. Choosing a software development methodology need to check profile of project owner ,project details, complexity of project ,developers technical skill and budget .in some cases there is no other methodology is suitable for profile

of project. The experienced team and project, manager together gives a best methodology. A new methodology is needed to make innovative software project.

REFERENCES

- [1] <http://php.net/ChangeLog-5.php>
- [2]W. W. Royce, "Managing the development of large software systems: Concepts and techniques", IEEE WESCON, vol. 26, no. 8, pg. 1-9, 1970
- [3]J. E. Cooling, T. S. Hughes, "The emergence of rapid prototyping as a realtime software development tool", Proceedings of the Second International Conference on Software Engineering for Real Time Systems, 18-20 Sep. 1989, Cirencester, UK, Publisher: IET, 1989, pg. 60-64
- [4]C. Larman, V. R. Basili, "Iterative and Incremental Development: A Brief History", Computer, vol. 36, no. 6, pg. 47- 56, 2003, doi:10.1109/MC.2003.1204375
- [5]B.W. Boehm, "A spiral model of software development and enhancement", Computer, vol. 21, no. 5, pg. 61-72, 1988, doi: 10.1109/2.59
- [6]J. Martin, Rapid application development, Publisher: Macmillan Publishing, 1991, pg. 788, ISBN:0-02- 376775-8

[7]S. Mathur, S. Malik, "Advancements in the V-Model", International Journal of Computer Applications, vol. 1, no. 12, pg. 29-34, 2010, doi: 10.5120/266-425

[8]L. Rising, N. S. Janoff, "The Scrum Software Development Process for Small Teams", IEEE Software, vol. 17, no. 4, pg. 26-32, 2000, doi:10.1109/52.854065

[9]A. Spangler, "Cleanroom software engineering-plan your work and work your plan in small increments", IEEE Potentials, vol.15, no. 4, pg. 29 - 32, 1996, doi: 10.1109/45.539962