

ANALYSIS OF SOFTWARE PROJECT MANAGEMENT METHODOLOGIES

Tanya Rastogi, Manika Bisht, Manjula R

Computer Science and Engineering, VIT University, Vellore

tanyarastogi2@gmail.com

Abstract

In order to build a project, all project activities must be planned. Management involves activities like monitoring, planning, directing, staffing and organizing. The objective of this paper is to compare various methodologies and analyze them based on different aspects and to find which among them is the most significant or complete. New concepts and technologies are happening very rapidly and as a result, as the time progressed, the methods that were developed earlier seemed to be incomplete and did not fulfill the requirements of up-to-date technologies and hence new methods are being evolved. Most of them are based on old methods with some changes or additions and some might be new concepts. There are many techniques and tools available for developing software projects. Various factors like requirement, project size, team size, etc. are considered before choosing a method for developing software.

Keywords: Agile Software Development, Adaptive Project Framework, Methodologies, Project Management, Rapid Application Development, Software Project, Traditional Software Development.

Introduction

Project Management is a process. Activities and tasks are assigned, organized, planned resources and implemented within a given period and budget. Several software project management methodologies are used in managing a software project. This helps managers in making decisions about which methodology to use according to their needs. The aim of this paper is to research on the different management methodologies. Information on each of the methodology is discussed with emphasis on their advantages, disadvantages and key points. Methodologies being discussed are Crystal Methods, Adaptive Project Framework, Rational Unified Process (RUP), PRINCE2, Agile Software Development, Rapid Application Development (RAD), Joint Application Development (JAD), Information Technology Infrastructure Library (ITIL) and Waterfall (Traditional).

SOFTWARE PROJECT:

A project is a definite assignment, which is a compilation of numerous functions done in order to accomplish a purpose. A Software Project is the thorough process of software development from requirement assembling to testing and safeguarding,

conducted according to the implementation procedures, in a quantified amount of time to accomplish planned software product.

PROJECT MANAGEMENT- Project management is the implementation of procedures, methods, information, expertise and understanding to achieve the project objectives. The key objective of project management is to achieve all of the project aims within the given constraints.

SOFTWARE PROJECT MANAGEMENT METHODOLOGIES

A. Adaptive Project Framework

In this, project scope is flexible. Also, price and time are constants for the task. Therefore, during the project implementation, in order to get the maximum business worth from it, the project scope is adjusted. It's a new level of agile development. It is not a "one-size-fits-all" approach—it continuously adjusts to change. APF uses a just-in-time development methodology. It is client-driven, client-focused and outputs maximum business value. Crux of adaptive project management: based on knowledge from the performance, projects are managed and this knowledge is acquired and evaluated through measurable approaches. Adaptive management

processes can be passive and active. The aim of active adaptive management is to use research to find out the top administration approach. The goal of passive adaptive management is to unite the process of gaining knowledge with existing administration methodologies.

The only existing kind in the domain that usually vigorously repels adjustment is human beings. Specifically, task supervisors often do not recognize that adaptive approaches will mostly result in better outcome than traditional project management processes, where before initiation, the project approach is finalized. Most of the organizations have adaptive management procedures and approaches. Most of the software development groups frequently utilize values of the agile method. The field of project management is controlled by traditional project management processes till date.

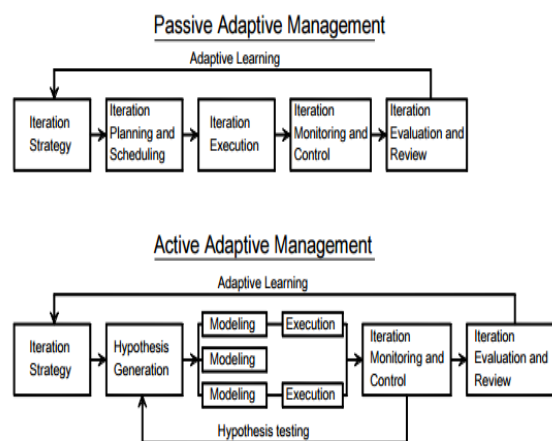


Figure 1: Passive versus Active Adaptive management

B. Agile Software Development

Agile software development approach is used for an assignment which desires tremendous responsiveness in its requests. The main qualities of agile are its agile requirements, short-termed delivery series (sprints), fewer preventive task control, stress on actual interaction and dynamic team culture. This approach attempts to deliver continuous, quick transfer of outcome to the user. While conventional methods like the Waterfall method or other linear procedures need exhaustive supplies which are outlined in the start where the completion produce is similar to the technique it has been described at the beginning. There is no evidently distinct end produce at the beginning with Agile. There is nevertheless a well-organized ranking procedure, but the flexibility, non-static desires, regular communication and continuous modification method is a portion of the process and culture. It is most frequently utilized in software growth but can also be useful in further developments.

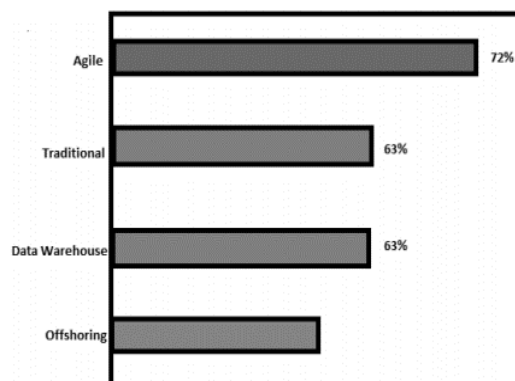


Figure 2: Software development success rate [13]

Table 1: Description of Main Agile Development Methods with References

Agile method	Description
Crystal methodologies	The utmost agile technique, Crystal Clear, centers on interaction in minor groups emerging software that is not life-important [8].
Dynamic Software Development Method (DSDM)	It splits tasks into three phases: before-task, task life-cycle, and after task [9].
Feature driven development	Pools model-motivated and agile progress with stress on iterative policy [10].
Lean software development	It involves seven principles: remove spare, intensify knowledge, choose as late as feasible, provide as fast as doable, enable the group, build reliability, and realize the complete [11].
Scrum	Emphasizes on assignment supervision in circumstances where is hard to design early, with a significance on opinion methods [9].
Extreme Programming	Concentrates on finest exercise for progress. Involves twelve exercises [12].

Agile methodologies include:

1. Extreme Programming (XP)

Extreme Programming was presented by Kent Beck in 2000. It is one of the first agile activities to be proposed. The goal of extreme programming is reducing the cost of requirement changes. XP emphasizes on programmer welfare, fine scale feedback, shared understanding and uninterrupted activity. There is no detailed needs description or software design built in XP.

2. Scrum

Scrum was developed by Jeff Sutherland in 1993. It is an incremental and iterative agile software growth structure to manage artifact growth. It expresses "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", tests conventions of the "traditional, sequential approach" to artifact growth, and encourages physical co-location, thus, enabling teams to self-organize.

3. Crystal Methods

Alister Cockburn established Crystal Methodologies in 2000. They concentrate on habitability and efficiency for project safety. The task procedures are set at a low importance in crystal technique. This process centers extra on interaction, group communication, people and group associate services instead of the processes.

4. Dynamic Systems Development Method (DSDM)

DSDM is an incremental and iterative approach which embraces agile growth principles, comprising incessant customer/consumer participation. DSDM repairs time, expense and attribute at the onset. Eight principles underpin DSDM: deliver on time, focus on the business need, certainly not bargain excellence, collaborate, construct incrementally from stable fundamentals, demonstrate control, interconnect constantly and distinctly and develop iteratively.

5. Feature-Driven Development (FDD)

Feature Driven Development (FDD) was created by Jeff De Luca and Peter Coad. It conglomerates few of the procedures renowned in business into one approach. FDD method centers on the software qualities of the technique which are the key trainer of

the whole expansion course. It enforces rigorous guidelines and coding standards to find defects in the system and supports steady constructs on a weekly or daily basis to enhance newly devised qualities to the base structure.

6. Lean Development

Lean software development (LSD) is a combination of lean IT values and exercises and lean manufacturing to the software expansion area. Customized from the Toyota Production System, a pro-lean subdivision is budding from inside the agile group. Seven main values of lean software growth are: create Knowledge, eliminate waste, respect People, build quality in, deliver fast, defer Commitment and optimize the whole.

C. Information Technology Infrastructure Library (ITIL)

ITIL is an abbreviation for Information Technology Infrastructure Library. It is a set of methods for IT service management (ITSM) that emphasizes on supporting IT facilities with the corporate requirements. It is a collection of top methods in project management. Starting from the organizational management level ITIL covers a broad aspect of project management. It aims at improving information technology service delivery. It provides many benefits to the organization by improving customer satisfaction and productivity and reducing costs. Thus, ITIL provides a framework to define the IT operational management procedures and IT organizational structure. This facilitates the firm to better organize the IT infrastructure and operation in a standard way. ITIL adopts a 'lifecycle' approach to IT services, with emphasis on methods for facility evolution, facility plan, facility procedure, facility design and continual facility enhancement.

D. Joint Application Development (JAD)

JAD is a procedure utilized in the prototyping lifetime phase of the Dynamic Systems Development Method (DSDM) to gather commercial needs while evolving fresh knowledge techniques for an association. "The JAD process also includes approaches for enhancing expediting development, user participation, and improving the quality of specifications." It comprises of a yard where "IT specialists and knowledge workers meet, sometimes for several days, to review and define the business requirements for the system." [2] The people attending contain great level executive officers who will guarantee the invention

offers the required results and knowledge at completion. This provides “a management process which allows Corporate Information Services (IS) departments to work more effectively with users in a shorter time frame.”[3] JAD reduces costs and time related with requirements elicitation method. It permits corporations to modify their systems evaluation course into vigorous ones like Double Helix, a procedure utilized for mission-critical job. It also helps in bringing experts together providing them with a chance to comprehend opinions of others, share their views and foster the awareness of job possession. Combination of CASE instruments into JAD factories offers systems experts with considered and prepared to use prototypes and improves session productivity.

E. PRINCE2

PRINCE2 encompasses organization of a project, control and quality management with review and consistency to align with project objectives. It is a certification programme for practitioners who are qualified and accredited through training. PRINCE2 focuses on dividing projects into controllable and manageable stages. It has significantly offered an easily scalable and adaptable method for every kind of project management. Transforms and uncertainty are the significant factors strengthening the execution of PRINCE2 methodology by professional organizations. PRINCE2 is a process-driven project management approach. It is built on seven themes, seven values and seven procedures. The seven principles are: Learn from experience, continual business reasoning, manage by stages, definite parts and tasks, concentrate on products and tailor to suit the project environment and manage by exception. The seven themes are: organization, corporate situation, risk, quality, strategies, progress and variation. The themes and principles come into play in the seven processes.

Using PRINCE2 provides ability to manage project risk and business and more effectively and greater control of resources. It will benefit:

- 1) Individuals seeking greater employment prospects and leading project management skills
- 2) Organizations
- 3) Executives/Directors (senior responsible owners) of projects, and
- 4) Project managers

F. Rapid Application Development (RAD)

Rapid application development (RAD) is a software growth practice which utilizes nominal preparation in support of quick prototyping. An archetype is an operational standard which is functionally comparable to a constituent of the outcome.

The operative components are advanced in match as archetypes in RAD model and are merged to build the whole result for earlier release. As there is no thorough preplanning involved, changes are simpler to include within the growth procedure. RAD schemes trail incremental and iterative standard and need little groups including area specialists, client delegates, developers and other IT assets functioning increasingly on their prototype or component. In order for the ideal to be effective, it needs to be ensured that the developed models are returnable.

It allows swift release since it lowers the total expansion period due to reusability of parallel development and the components. RAD works well only if the client is dedicated to attain the directed model in the bounded interval and high skilled engineers are available. Commitment lacking on either side will lead the model to fail.

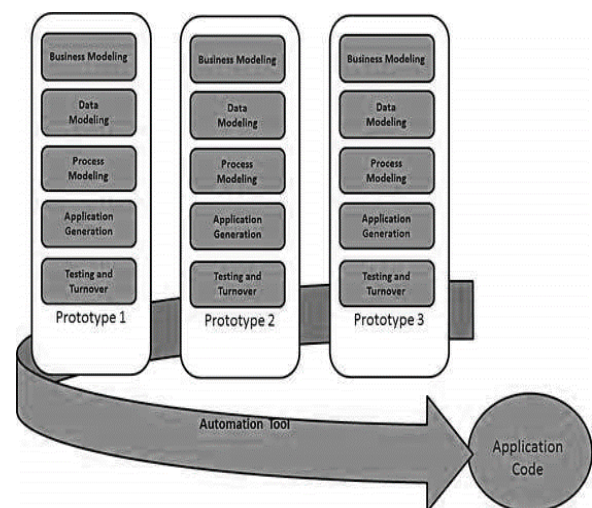


Figure 3: RAD Model RATIONAL UNIFIED PROCESS

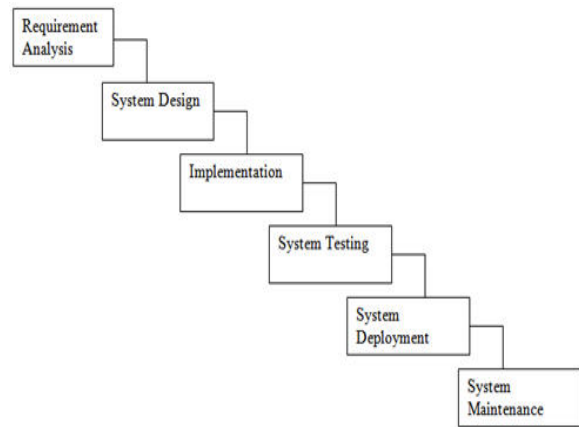
The Rational Unified Process is a Software Engineering method with an aim of confirming manufacture of top class software meeting the complete requirements of it's customers in a foreseeable and reasonable agenda and budget [7]. It allows a systematic approach for assignment of tasks and responsibilities within an organization. It maintains four steps in development with certain number of iterations which much be satisfied before taking on the next step of development. In the beginning step, developers establish the scope and

the business case of the project. In the elaboration step, they analyze the needs of the project in high detail and establish an architectural base. In the construction step, they make the application design and the source code. In the transition step, they deliver the complete product to the customers or the users. At the completion of each iteration/step, RUP provides a prototype. RUP is easily adaptable because of its documentation using Unified Modeling Language (UML). It is a well established structure for agile project management which helps in delivering quick and effective results focusing on planned objectives and increasing distribution of actual business profits and also having a check of time, risk, price and value. It resolves the project risks paired with clients' based requirements with precautionary changes and request management. There is very little need for integration as there is integration taking place after each step throughout the development process.

TRADITIONAL SOFTWARE DEVELOPMENT

The waterfall model is a progressive growth method, in which growth is seen as rolling gradually down (like a waterfall) across numerous stages, usually: Requirements examination causing in a software requirements description, implementation, integration, software plan and analysis, if there are various subsystems, maintenance and implementation (or connection). This classical is an established work tactic related to software engineering. Revising and revisiting any prior phase once it has been completed is discouraged by strict waterfall approach. This "inflexibility" in a wholesome waterfall model has led to disapproval by followers of additional more "flexible" prototypes. It has been extensively accused for numerous giant administration jobs running over time, over budget, and occasionally lacking to bring desired needs because of the Big Design Up Front method.

The extended waterfall model with prototyping is spiral methodology. Rather than using the waterfall model for large projects, this method is used.



Waterfall Model - © www.SoftwareTestingHelp.com

Figure 4. Waterfall Model

I. CONCLUSION

Each method has its own pros and cons. It is somehow necessary to choose the right methodology for developing a software. Choosing the methodology depends on a number of factors like capabilities of development team and type of software being developed. Every project management approach has some advantages and disadvantages. The software that needs to be delivered should satisfy customer requirements and be error free. It should go beyond agility and traditionalism and also give importance to testing and validation. Agile methodology is more preferable as it is more effective and responds to regressive changes that development of software is supposed to.

REFERENCES

1. Haag, Stephen; Cummings, Maeve; McCubbrey, Donald J. (2006). "Phase 2: Analysis". Information Management Systems for the Information Age. McGraw-Hill Ryerson. ISBN 978-0-07-281947-2.
2. Jennerich, Bill, "Joint Application Design: Business Requirements Analysis for Successful Re-Engineering", (1990).
3. A B M Moniruzzaman, Dr Syed Akhter Hossain, "Comparative Study on Agile software development methodologies", Global Journal of Computer Science and Technology, Volume 13 Issue 7 Version I, (2013).
4. KudaNageswara Rao, G. Kavita Naidu, PraneethChakka, "A Study of the Agile Software Development Methods", Applicability and Implications in Industry, International Journal of Software Engineering and Its Applications Vol. 5 No. 2,(2011).

5. MalikHneif, Siew Hock Ow," Review Of Agile Methodologies In Software Development", International Journal of Research and Reviews in Applied Sciences, Volume 1, Issue 1, (2009).
6. Rational Unified Process: Best Practices for Software development Teams, Rational Software White Paper TP026B, Rev 11/01.
7. Mrs. Kalaivani S., Mrs. Kavitha S., "Comparative Study On Software Project Management Models", International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 04, (2015).
8. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Addison-Wesley, 2004.
9. Linda Rising and Norman S. Janoff,"The Scrum software development process for small teams", IEEE software, 17(4), (2000), 26-32.
10. Kajko-Mattsson, M.; Lewis, G.A. Siracusa, D.; Nelson, T. Chapin, N. Heydt, M. Nocks, J. Snee H., "Long-term Life Cycle Impact of Agile Methodologies", 22nd IEEE International Conference, (2006), 422-425.
11. M. Poppendieck, T. Poppendieck, "Lean Software Development – An Agile Toolkit for Software Development Managers", Addison-Wesley, (2003).
12. K. Beck, "Extreme Programming Explained: Embrace Change", Addison-Wesley, (2000).
13. Manish Kumar, Santosh Kumar Singh, Dr. R. K. Dwivedi,"A Detail Study of Agile Software Development with Extreme Programming", International Journal of Advanced Research in
14. Computer Science and Software Engineering, Volume 5, Issue 10, (2015).