

Comparative Analysis between Sequential and Iterative Project Management Approaches

Aishwarya Vijay Patil

B.Tech in Information Technology from Rajarambapu Institute of Technology, Affiliated to Shivaji University, Maharashtra, India

Abstract - Since the past few decades, there is tremendous development in I.T Field. With the emerging of various cutting-edge technologies, the way of project management is also changed rapidly. New project management methodologies like PRINCE2, Agile, Rapid Applications, Spiral, etc. are evolved to meet the changing requirements and modern scenario. But this all methodologies are based on two principal software management approaches recognized as sequential and iterative. These approaches, more generally referred to as software development models, forms a foundation for multiple project management methodologies. Before understanding various methodologies and determining which is better for a given project, it's crucial to analyze the model on which these methodologies are based on. This paper discusses two main approaches of project management: sequential and iterative, to give a deeper insight. The purpose of this paper is to analyze both approaches, understand their features and drawbacks and finally identify which approach is better suited for which kind of project.

Key Words: Project Management approaches in I.T, Software Project Management Methodologies, Sequential, Iterative, Waterfall, Agile, SDLC, I.T

1. INTRODUCTION

In today's era, project management is the initial and most vital element of the software development process. Project management drives the whole process in the right direction. Proper management facilitates smooth strategic alignment of goals and plans for complex and multi-faceted projects. Project management assures that balance is maintained between available resources, scheduled timeline, cost estimated along with the quality of the project. Every I.T. company follows a well-defined project management methodology. There could be a single methodology followed for all projects in the organization or each project can have a different style of management based on need.

There are various types of methodologies like agile, waterfall, spiral, etc. These methodologies may follow a sequential or iterative approach but all of these methodologies go through the identical well-defined software development process. This process is nothing but the software development life cycle (SDLC). The software development life cycle is split into non-parallel distinct phases which are Requirement elicitation, Design,

Implementation, Testing, Deployment, and Maintenance [1]. Figure 1 describes flow of SDLC.

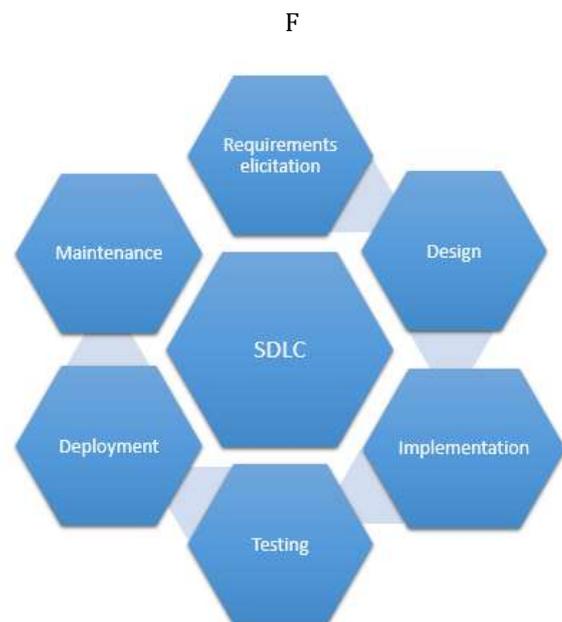


Fig-1 SDLC.

Following are SDLC phases:

1.1 Requirements elicitation

This is the beginning round of the software development process where all requirements are assembled from several stakeholders like client, end-users, vendors, etc. by the business analyst. These requirements are examined, validated and prioritized by the team. This phase comprises interviewing customers, brainstorming, creating questionnaires, etc. These business needs are documented with precision in a requirement specification document.

1.2 Design

The requirement specification document from requirement elicitation phase is utilized as input in this phase. All functional and non-functional requirements are analyzed by senior manager and architecture to sketch software architecture, high-level and low-level design of the product. This includes specifying software and hardware requirements, technology to use, network, databases, user

interfaces, system interface, etc. Software design description document (SDD) is drafted in this phase which describes “how” of the software [2].

1.3 Implementation

This is the phase where the actual product built takes place. The information in SDD is used to develop the product. Initially, the product is generated in small chunks by different developers. After ensuring that each unit works fine, these units are integrated to develop the whole product. Source code is the yield of this phase.

1.4 Testing

The testing team conducts testing to ensure each piece of product operates as expected and without any defects. Aim of this phase is also to ensure all requirements in requirement specification documents are met. Testing phase involves various stages as follow [3]:

- Requirement Analysis
Quality Assurance (QA) team understands the requirements in terms of what they will be testing and estimates the testable requirements.
- Test Planning
In this stage, the testing process overview is defined and test schedules are planned.
- Test Case Development
In this stage, the testing team drafts the detailed test cases.
- Test Environment Setup
This involves setting up software and hardware required for testing of the product.
- Test Execution
In this stage, the testing team start executing test cases and records test case as pass or fail as per the results and if any flaw is located corresponding test case with the bug is reported to the developer. After modification from development team retesting is perform.
- Test Cycle Closure
This stage is performed when testing and retesting is completed. Test closure report & Test metrics are outlined.

1.5 Deployment

After the product passes all the test cases and accepted by the client, it is deployed client-side in their respective environment by installing the application on client servers.

1.6 Maintenance

This phase is performed after the product goes live and carried out throughout the life cycle of the product. This covers error correction, enhancement of performance and quality, adding new functions and removal of obsolete ones, introduction of new features and optimization of the software [2].

All software management approaches go through these phases but in a diverse pattern or different manner.

2. Sequential Project Management Approach

In this approach, all the phases are carried out in linear fashion. Next phase cannot be started until the previous one concludes because the output of the previous phase is input to the next phase. Due to this reason, any delays in the prior phase causes a delay in later phases.

The best example of project management methodology which uses a sequential approach is the waterfall. In natural waterfall, where once the water has flowed through the edge of the cliff, never returns to the top of the hill [2]. Similarly, in waterfall methodology, phases cannot be revisited. The only way to revisit a phase is to start the process again with the opening phase [4]. Well, this is justified because waterfall model was discovered in the manufacturing and construction industry where there is a need for things to fall sequentially. E.g. one cannot start building roof until walls are constructed. In construction and manufacturing industry hardly any requirements are altered once implemented is started. But in software industry, this is not the case as client requirements often change. So in methodologies with a sequential approach, excess time is spent on planning and it is essential to map out a clear plan.

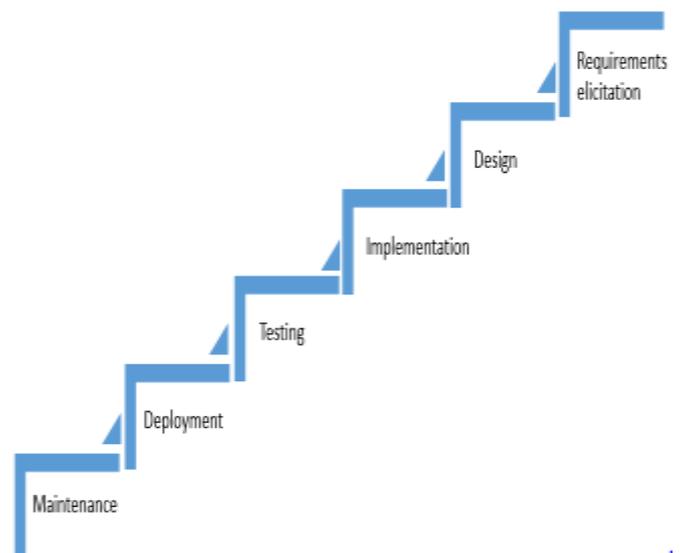


Fig-2 Waterfall model.

2.1 Advantages of the sequential approach

- Plain, easy to understand and straight-forward approach.
- It is document-intensive so every step, decision, research, etc. are well documented and documents can be utilized in further alike projects.
- As there are well define deliveries at the end of each phase, it's easy to measure development.
- Project is broken into discrete and clearly understandable phases so project management is straightforward and smooth.
- Client involvement is minimal so no time is spent waiting for the consent of the client at each stage.
- It is best for miniature projects where requirements are clear up front.
- Well suited for a project that is expected to flow steadily downwards and where extremely structured programs are needed [4].
- Team members can operate on other projects before their phase begins or after their phase ends, returning to the project as needed.
- Waterfall model has a clear and rigid structure so it does not demand any special training for project managers or employees. It has an easy learning curve.

2.2 Disadvantages of the sequential approach

- It is expensive to revert to the previous phase.
- If one phase is delayed, subsequent phases are also delayed, affecting the deadline.
- As there is no room for additional requirements, if a client wishes to append a new feature when the project is in the implementation or testing phase it's unfeasible to do so.
- Not suitable for a large and complicated project where requirements are not well defined.
- It involves a high-risk factor.
- Ultimate working software is available only at the end of the process and as no prototype is used, the client doesn't have a clear picture of the product until the end.

- A trivial error in any previous phase can cause a huge problem for subsequent phases as each phase depends on the prior phase.
- Testing is delayed which can negatively influence the project.
- Client engagement is minimal which can let to confusion between team and client.
- Errors can escape the process and be found in later stages [5].

3. Iterative Project Management approach

In the iterative approach, the software development is cut in a series of repeated cycle or loops called iterations. The first four phases of SDLC: Requirements elicitation, Design, Implementation, Testing occurs in repeated loops. According to the PMBOK Guide Sixth Edition [6]: "Iterative Life Cycle is a project life cycle where the project scope is generally determined early in the project life cycle, but time and cost estimates are routinely modified as the project team's understanding of the product increases." With each iteration, new features are introduced or old once are revised as demanded. At the end of each iteration, a new version of the software is produced. The last stage of the cycle involves client collaboration where a modified version of product is reviewed by the client. Feedbacks given by client are shared with developers and any changes suggested by the client are incorporated in upcoming cycles. It is possible that business strategy may change or risks may emerge, in such cases strategic realignment of plan and goal can be done in the further iteration. Iterations take place until the software is evolved in a fully developed system as per the requirements.

The most suitable example of methodology which uses iterative approach is agile. According to the 12th Annual State of Agile report, the Iterative Planning was the second and iterations review was fourth most-employed Agile Technique in 2018 [7].

Craig Larman [8] explains that each iteration consists of the analysis of the plan, the Design, its Code and the Test together called as ADTC. The ADTC wheel is more technically referred to as the PDCA (Plan, Design, Check, and Adjust). This PDCA cycle is implemented on each iteration separately in the following manner [9]:

- P (Plan) – Iteration Planning

The team collaborates to outline the objectives, scope and risk assessment for the forthcoming iterations. If this is not the first iteration, the reviews and feedback of the client are discussed to plan accordingly. The team comes up with schedule estimates and also classifies the required resources. It also summarizes the work done and determines the Iteration Goals.

- D (Design) – Iteration Execution

In this stage, the product is revised to add new functionalities or enhance the older one. The system goes through SDLC phases of design, implementation, and testing. At the end of this stage, a modified version of the product is produced.

- C (Check) – Iteration Review

This stage is an essential one in an iteration as client is involved. The modified version of the product is presented to the client for review. Client reviews product to verify if all criteria and standards are met and gives feedback accordingly.

- A (Adjust) – Iteration Retrospect

At the end of each iteration, the team analyses the entire iteration to find what all went good, new problems are identified if any and which area needs to improve. The team works on taking the lessons from the current iteration as input to the upcoming iteration.

- Continuous interaction between team and client benefits team to have a clear picture of the client's vision with no room for any guesswork.

- Daily stand-ups and feedbacks from client enable the team to know whether the project is on the right track.

- As software is reviewed at the end of each iteration and it is redefined accordingly in the next iteration, it helps in making high-quality software with fewer or no defects.

- Any new changes in requirement by the client are welcome as the iterative model can incorporated changes in the upcoming iteration.

- Iterative model enables parallel development

3.2 Disadvantages of the iterative approach

- As minimum time is spend on initial planning, there is possibility that an unpredictable issue may rise in later stages of project.

- As client's engagement is more, it can put increased pressure on developers.

- Lack of documentation

- Client involvement is an essential part. If the client is not actively participating, it can negatively affect the project.

4. Difference between sequential and iterative approaches

These iterations take place until a fully functional software is ready for release.

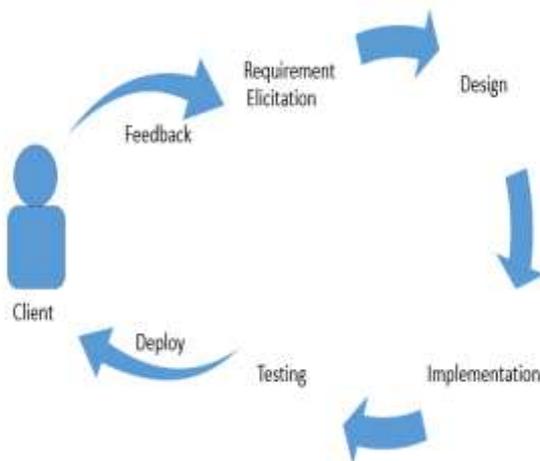


Fig -3 Iterative Approach.

3.1 Advantage of the iterative approach

- The iterative model facilitates the continuous delivery of a functioning model to the client at the end of each iteration.
- Client collaborates with developer and tester throughout the process which lets to clear understanding between them about the product as well as transparency is maintained between them.

Sequential approach	Iterative approach
Requirements and scope are specified upfront at the beginning for the entire process.	Requirement and scope are defined for the upcoming iteration.
It has a rigid structure	It has a flexible structure.
Welcoming the changed requirements in later phases is expensive.	Changes are accepted and applied in forthcoming iterations.
Customer engagement is minimal. Customers are only involved in the beginning phase of the process and during project delivery.	Customers are involved throughout the process. At the end of each iteration, the product is reviewed by the client.
Preferred for the small and medium-sized project which can be completed in one go.	Preferred for the vast and complex project which are needed to be completed in installments.

Table-1 Difference between Sequential and iterative approach.

Project is one-time delivered only at the end of the process.	Continues delivery at the end of each iteration.
Sequential is more document-driven approach.	Documentation is comparatively given less importance.
Risk of failure is more as result may not meet the customer's expectations.	Risk of failure is less as customers can suggest changes at the end of each iteration. So product build is as per customer's input and expectation.
Scope of the project is known beforehand	Scope of project is not known in advance.

5. When to use what

5.1 Sequential approach

- When the scope of the project is known in advance.
- When requirements are well defined and not subjected to changes.
- Small and medium-sized project.
- When the client is not available, or not actively committed to the process.
- There is no need of prototype and client needs the entire product in one time delivery.
- Where product development mainly consists of adding limited functionalities to an existing set of functionality [2].
- When management or client expects intensive-documentation.

5.2 Iterative approach

- When the scope of the project is unknown beforehand.
- When client doesn't have a clear vision of product and requirements are expected to change.
- Complex and large projects where is it not possible to develop product in one go.
- When the client is ready to actively be part of the process.
- A quick prototype with only certain functionalities is needed before the final product is available [2].

- When employees are well trained in using the iterative approach.
- When detailed documentation with precision is not expected.

6. CONCLUSION

Iterative and Sequential are both essential project management approaches. Both have their pros and cons and are suited for specific types of projects. To wrap up, the sequential approach can be applied to a small or medium-sized project where requirements are not subjected to change and client is not part of development process. The iterative approach is best suited for a project where scope and requirements are not well defined and in scenario where client is agreed to actively participate in process.

REFERENCES

- [1] Hans van Vliet, "Software Engineering: Principles and Practice" Third Edition, Wiley, 2007.
- [2] Vaishnavi Kannan, Smita Jhajharia, Dr. Seema Verma, "Agile vs waterfall: A Comparative Analysis", IJSETR, Volume 3, Issue 10, October 2014.
- [3] <http://www.softwaretestingclass.com/software-testing-life-cycle-stlc/>.
- [4] Adetokunbo A.A. Adenowo, Basirat A. Adenowo, "Software Engineering Methodologies: A Review of the Waterfall Model and Object Oriented Approach", International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013.
- [5] Eduardo Málaga Chocano, "A Comparative Study of Iterative Prototyping vs. Waterfall Process Applied To Small and Medium Sized Software Projects", April 2004.
- [6] A Guide to the Project Management Body of Knowledge (PMBOK Guide), Sixth Edition, by Project Management Institute, 2017.
- [7] 12th Annual State of Agile Report.
(<https://explore.versionone.com/state-of-agile/12th-annual-state-of-agile-report-overview>)
- [8] Craig Larman, "Agile and Iterative Development: A manager's Guide", Addison-Wesley Professional, 2003.
- [9] K.A.Chandrakanth, "PLAN DO CHECK ACT (PDCA) IMPROVING QUALITY THROUGH AGILE ACCOUNTABILITY".

BIOGRAPHIES



Aishwarya Vijay Patil
B.Tech,
Information Technology,
Rajarambapu institute of
technology, affiliated to Shivaji
University, Maharashtra, India