

A COMPARATIVE STUDY OF CHANGE MANAGEMENT PRACTICES IN TRADITIONAL AND AGILE PROJECT MANAGEMENT FRAMEWORKS

Uday Kumar Reddy Gangula

ukgangula@gmail.com

Abstract:

This research examines the differences in change management approaches between traditional and agile project management systems. The ability to handle changes in project environments has become the main factor that determines project success in today's dynamic business world. The research shows that the traditional Waterfall model and agile Scrum and Kanban frameworks handle risk differently because they approach change as a deviation to control delivery risk or as a value-creation opportunity to address market risk. The fundamental approach to change management between these two paradigms differs because traditional frameworks focus on risk control through plan and budget adherence, while agile frameworks use change to create value for market needs. The fundamental approach of these systems toward change management becomes evident through their operational structures. The Waterfall model implements a Change Control Board (CCB) to defend project baselines through strict procedures, yet Scrum and Kanban integrate adaptation into their workflows through Product Backlogs and continuous flow systems. The success of each method depends on specific project conditions, which include stable requirements and uncertain project environments. The current project management environment demands hybrid models because neither traditional nor agile approaches work best for all situations.

Index Terms: Project Management, Change Management, Change Control, Waterfall Model, Agile, Scrum, Kanban, Iterative Development

I. INTRODUCTION

Modern project management originated in the manufacturing and construction industries, as these fields required accurate planning methodologies to forecast outcomes [1]. The environment caused the creation of classic linear techniques, which were best for projects with clear needs. The main idea was that the entire project should be fully defined before work began, so that the workflow would be organized and easy to manage throughout the project.

The shift to software development and knowledge-based work in the late 20th century highlighted the limitations of traditional project management methods. The new kinds of initiatives had conditions and needs that were hard to forecast and changed quickly [2]. Under old approaches, the distribution of items resulted in products that initially met customer needs but did not address their current wants [1]. The "Manifesto for Agile Software Development" puts being able to adapt to following plans to the letter. The method changed the major source of project value from an afterthought.

The two paradigms are fundamentally different because they have different ideas about how to handle

change. The old way of thinking sees change as a threat to the stability of a project, which needs careful control to keep risks from happening. Agile models see change as a way to get feedback that can help you develop. The difference between these two strategies lies in their perception of change, which influences how they manage risks such as delivery and market risks. To select the most effective project management methods, it is essential to evaluate a range of options carefully. [3]

II. THE TRADITIONAL PARADIGM: A FOUNDATION IN PREDICTABILITY AND CONTROL

A. *The Waterfall Model: A Sequential Approach*

Dr. Winston W. Royce established the Waterfall model as a linear sequential project management method in 1970 [2]. The Waterfall model operates through sequential phases, which start only after the completion of the preceding stage [1].

Typical phases include:

1) *Requirements:*

- This process requires complete documentation of functional and non-functional requirements at a detailed level.
- The project team must establish all essential details about the scope, timeline, and resource allocation before starting work.

2) *Analysis:*

- The analysis process verifies that requirements meet both technical and business requirements for feasibility.
- System models serve as guidelines for designers during the design development process.

3) *Design:*

- Design requires developers to create both high-level architectural designs and complete component specifications. [4]
- Development requires a complete design document, which serves as a guide for building the system.

4) *Implementation (Coding):*

- Development involves converting design specifications into operational code through modular unit construction. [1]
- The system development process requires combining individual components into a full OS.

5) *Testing (Verification):*

- The complete system undergoes thorough testing to verify its compliance with the original requirements. [1]
- The formal QA process enables teams to detect and resolve system defects.

6) *Deployment (Maintenance):*

- The system delivery to users becomes possible after the verification process reaches its successful conclusion.
- The system requires ongoing maintenance activities and small post-deployment updates.

Waterfall Model

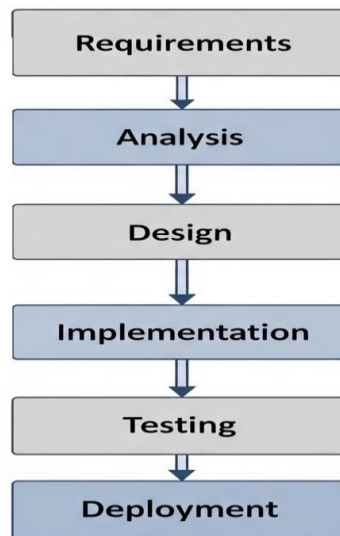


Fig. 1. The traditional Waterfall model, illustrating its rigid, sequential flow from requirements to deployment

B. The Philosophy of Change as Deviation

The Waterfall approach sees changes as possible risks that need to be dealt with. For the project to be successful, it is important to stick to the original scope, timeframe, and budget. Making modifications to the design and code in the late phases means undertaking a significant amount of expensive rework. The basic goal of this paradigm is to stop any changes except those that are absolutely necessary. The fundamental goal of change management in this method is to focus on control rather than flexibility.

C. The Formal Change Control Process

To enforce this control, a formal, auditable Change Control Process is used:

- 1) Change Initiation: The Change Request Form (CRF) serves as the starting point for proposed modifications, which need to include detailed information about reasons and projected advantages, together with an initial assessment of consequences.
- 2) Impact Analysis: The Project Manager leads a team of experts who assess how changes affect project scope, cost, timeline, quality, and resource allocation. [5]
- 3) Change Control Board (CCB): The CCB, which consists of sponsors, clients, and senior managers, reviews the CRF and analysis to make decisions about change approval, rejection, or postponement. [5] [6]
- 4) Implementation & Re-baselining: Project documents need updates for approved changes, which establish a new performance baseline for upcoming work. The Change Issues and Decision (CID) Log serves as a tool to track progress according to the project requirements. [5]

III. THE AGILE PARADIGM: A FOUNDATION IN ADAPTABILITY AND FEEDBACK

A. The Agile Manifesto: A New Philosophy

The agile paradigm emerged from the 2001 *Manifesto for Agile Software Development* [7], which presented four fundamental values that challenge the conventional plan-driven approach:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- The core value of agile change management focuses on adapting to change instead of following established plans.

Agile approaches embrace change as a vital learning experience, which enables organizations to deliver value to their customers [8]. Agile frameworks operate through continuous cycles of development because

they serve high-uncertainty environments better than following rigid initial plans.

B. Change Management in Scrum: Structured Flexibility

Scrum functions as a popular agile framework that provides organizations with a flexible method to handle complex project changes, particularly in software development [7].

1. *Core Concepts:* The Scrum framework divides work into defined Sprints, which last between 1 and 4 weeks [9], to produce deliverable product increments. The framework establishes three essential roles for its operation:

- The Product Owner serves to optimize product value and oversees the Product Backlog management.
- The Scrum Master maintains Scrum principles throughout the project.
- The Development Team operates independently to achieve the Sprint targets.

Product Backlog contains all desired features, while the Sprint Backlog shows the specific features chosen for a particular Sprint period [9].

2. *Product Backlog as the Locus of Change:* Scrum handles change through the continuous development of the Product Backlog. The Product Owner determines the order of priority for user stories, which represent stakeholder requests and team suggestions [9]. The process contains built-in change management capabilities, which eliminate the requirement for an independent committee.

3. *Sprint Integrity vs Backlog Fluidity:*

The Product Backlog remains flexible, but the Sprint Backlog becomes unchangeable after Sprint planning is complete. The scope becomes protected from changes during the middle of a Sprint after planning finishes. The team must postpone all new requests until the following Sprint cycle begins. The system provides teams with short-term stability through a fixed scope while maintaining flexibility for long-term adjustments, which helps them maintain focus on their work and stay aligned with stakeholder requirements.

C. Change Management in Kanban: Continuous Flow

The Toyota Production System gave rise to Kanban, a non-restrictive workflow management system for change control.

1. *Core Concepts:* The system focuses on three core elements, which include workflow visualization, WIP restriction, and continuous process enhancement [9]. The Kanban board serves as the primary tool, displaying work stages that include To-Do, In Progress, and Done. [7] The implementation of WIP limits helps organizations prevent bottlenecks while ensuring tasks reach completion before beginning new work.

2. *Continuous and On-Demand Change:* The Kanban system allows organizations to make changes whenever needed because it does not use time-boxed cycles like Scrum does [9]. The system allows teams to add new work items to the queue, which they can pull into their work when they have available capacity for dynamic priority management. The flexible nature of Kanban makes it suitable for fast-changing environments such as IT operations and support teams because it supports immediate reaction times.

Scrum vs. Kanban: Scrum implements fixed cycles for change management, which creates a balance between short-term forecasting and flexibility. The Kanban system provides ongoing priority management and fast response capabilities, which maximize workflow efficiency above scheduling consistency. Organizations choose between managing change through established patterns or immediate reaction methods based on their specific needs.

IV. COMPARATIVE ANALYSIS OF CHANGE MANAGEMENT PRACTICES

The fundamental differences between traditional and agile change management emerge from opposing approaches, which produce distinct organizational processes and roles and final results. The section uses multiple dimensions to create a structured analysis that demonstrates the fundamental distinctions between

these approaches.

A. Philosophy and Attitude Towards Change

- **Traditional (Waterfall):** The enterprise operates its business on a system that focuses on keeping control while lowering possible hazards. The project team sees any change from the original plan as a severe danger to success because they define project success as sticking to the original scope, timeline, and budget. The basic goal of change management is to control and limit these changes from the original plan.
- **Agile (Scrum & Kanban):** The organization bases its operations on a philosophy that focuses on adapting to change while achieving maximum value. The process requires accepting changes because they bring necessary benefits to the project. The project team uses changes to gain feedback and learn while directing the project toward better results. The organization focuses on flexible adaptation to customer requirements instead of maintaining strict adherence to the original plan.

B. Process and Mechanism

- **Traditional (Waterfall):** The mechanism functions through a structured system external to the organization and adheres to official procedures. The process requires extensive documentation through Change Request Forms (CRFs) and detailed impact assessments, and Change Control Board (CCB) formal approval. The system functions as a strict obstacle that blocks all changes except those deemed essential. [5]
- **Agile (Scrum & Kanban):** The system operates with minimal complexity because it exists within the core operational flow. Scrum teams maintain their Product Backlog through continuous grooming led by the Product Owner and priority adjustments. Kanban teams use WIP limits and explicit policies to manage their work queue while performing fluid re-prioritization. The system operates with minimal resistance to allow quick responses.

C. Timing and Cost of Change

- **Traditional (Waterfall):** The project lifecycle shows an exponential rise in change expenses. The cost of implementing changes remains affordable during requirements definition, but becomes unaffordable and disruptive when modifications occur after implementation and testing, cause they affect all previous completed phases.
- **Agile (Scrum & Kanban):** Agile development operates through iterative cycles, which maintain affordable change costs at a consistent level across the entire project duration. The process of working in small increments and obtaining regular feedback enables teams to detect changes early, which prevents major disruptions from occurring. The method reduces the probability of performing expensive rework during the final stages of development. [10]

D. Decision-Making Authority and Roles

- **Traditional (Waterfall):** The system operates with centralized authority, which follows a hierarchical structure. The CCB serves as the last decision-making authority for change approval or rejection because it operates independently from daily development team activities. The Project Manager functions as an information manager who connects the CCB to the development team through their role as process facilitator and gatekeeper. [5]
- **Agile (Scrum & Kanban):** The system distributes its authority through specific roles that operate independently from each other. The Product Owner who works directly with the team maintains complete authority to decide product requirements and their order of importance in Scrum. The Kanban system operates without a designated change authority because team members share responsibility for the board while following established workflow rules, which receive business-driven priority updates. [7]

E. Stakeholder and Customer Involvement

- **Traditional (Waterfall):** The process requires stakeholders to participate through scheduled formal

interactions, which mainly consist of transactions. Stakeholders take part in two main stages of the process: requirement definition and approval at the start, and user acceptance testing at the end. The stakeholders participate in scheduled status meetings and CCB process activities, but their involvement remains restricted during the project duration. [1]

- **Agile (Scrum & Kanban):** The process requires ongoing stakeholder participation, which combines teamwork with essential contributions to achieve project goals. The customer or their representative who functions as Product Owner maintains active participation as an essential team member throughout the project. The customer provides ongoing feedback to clarify requirements, which helps determine work priorities that directly influence product development.

F. Documentation and Artifacts

- **Traditional (Waterfall):** The entire process depends on complete formal documentation systems. The method relies on five essential artifacts, which include the detailed requirements specification, project management plan, formal CRFs, impact analysis reports, and CID (Change, Issues, Decision) Log for request tracking.

- **Agile (Scrum & Kanban):** The process depends on flexible visual documentation that adapts to changing needs. The primary focus of this approach centers on delivering functional software instead of creating extensive documentation. The Scrum framework maintains its Product Backlog as its primary artifact while using Kanban boards to display work status and user stories that receive detailed explanation through conversations before developers start work. [11]

The table below presents essential comparison points about change management approaches through a clear summary of their main distinctions.

TABLE I- COMPARATIVE ANALYSIS OF CHANGE MANAGEMENT CHARACTERISTICS

Characteristic	Waterfall	Agile (Scrum)	Agile (Kanban)
Change Philosophy	Change = risk, minimize; focus on plan adherence	Change = value, adapt per Sprint	Change = flow, optimize throughput
Mechanism	Formal change control (CRF, CCB)	Product backlog prioritization	Continuous backlog re-prioritization
Key Artifact	CRF, project plan, logs	Product & Sprint backlogs	Kanban board
Decision Authority	Centralized (CCB)	Product Owner	Team (with policies)
Cost of Change	Grows exponentially	Low, stable via iterations	Low, stable via continuous flow
Stakeholder Role	Formal, phase-gated	Continuous collaboration & reviews	Continuous input for prioritization
Pace of Change	Slow, major events	Rhythmic (per Sprint)	Continuous, anytime

V. DISCUSSION: IMPLICATIONS AND CONTEXTUAL SUITABILITY

Organizations must choose between traditional and agile change management methods based on their company goals, project needs, and cultural values. How organizations choose between these two methods affects how they manage project risks, which in turn impacts their outcomes.

The two main risks that projects face are delivery risks, which arise from not meeting scope requirements or running out of time or money, and market risks, which stem from producing goods that fail to meet

modern consumer needs. The Waterfall model reduces delivery risks by employing strict change management rules that maintain the project's original scope, timeline, and budget. The strategy puts projects at greater risk in the market since it takes longer to develop and doesn't allow for change, which means that the products don't meet current market needs.

Agile frameworks reduce market risks through their approach of continuous development and ongoing feedback collection. The delivery risk increases in Agile because the framework accepts changes to project scope and plans while it adapts to new requirements. The main difference between Waterfall and Agile lies in their approach to risk management, where Waterfall focuses on maintaining plan consistency and Agile focuses on delivering correct products.

A. Impact on Project Outcomes

- **Risk Management:** Traditional models protect against scope expansion and budget problems through early validation of assumptions, which minimizes market dissimilarities.
- **Stakeholder Satisfaction:** The iterative approach of Agile results in better user satisfaction because the product development process adapts to changing requirements. The delivery of products according to specifications in Waterfall development does not guarantee user satisfaction because it fails to meet their needs.
- **Team Morale:** The self-organization structure of Agile enables team members to take ownership through decentralized decision-making processes, which include Scrum Product Owners and Kanban boards. The top-down change control process of Waterfall through CCBs restricts team members from making independent decisions.

B. Context is King: Selecting the Right Framework

No single approach works best for every situation. The selection of a framework depends on the specific characteristics of the project.

- **When Traditional Excels:** The Waterfall approach delivers its best results in stable projects with low uncertainty and fixed requirements and regulatory needs, which apply to construction, manufacturing, and hardware development, and become expensive when changes occur late in the process. [1]
- **When Agile Excels:** Agile methodology performs best in situations with unpredictable, fast-paced operations and shifting requirements, which include software development, R&D, and innovation, as well as marketing activities that require quick responses and flexibility.

C. Hybrid Approaches

The combination of stable and volatile project domains leads organizations to implement Water-Scrum-Fall hybrid models for their projects. The approach begins with Waterfall planning and architecture before moving to Agile development iterations and finishes with structured deployment methods. The implementation of hybrid approaches demonstrates experienced change management practices that unite project forecasting with flexibility.

CONCLUSION

The comparative study shows that traditional and agile project management systems operate from opposing principles regarding how to handle change. The Waterfall model emerged from a time when physical production and predictable outcomes were dominant because it approaches change as an element to be controlled. The Change Control Process of the Waterfall model reflects its philosophy by using formal procedures with high resistance to change, which protect the initial plan through detailed documentation and centralized authority. Agile frameworks Scrum and Kanban operate from a different perspective than traditional methods because they recognize change as a fundamental factor for achieving success in software development. The Scrum prioritized backlog and Kanban continuous flow operate as integrated

mechanisms that welcome change as a permanent learning opportunity to create more value.

The evaluation of their individual approaches demonstrates distinct trade-offs between them. The traditional method provides stable control and predictability, yet restricts flexibility and response speed, which works well for steady environments. Agile methodology delivers outstanding flexibility and customer engagement, but requires sacrificing initial project clarity, which benefits projects operating in unpredictable settings. The fundamental philosophical distinction between these approaches leads to all process differences, decision-making authority, cost management, and stakeholder participation.

Organizations must choose change management practices based on the specific requirements of their situation rather than seeking a single superior approach. Organizations that achieve success through effective project management use their ability to determine project environmental characteristics to pick the optimal framework or combination of frameworks for their specific needs.

REFERENCES:

- [1] Geekbot, “What is the Traditional Waterfall Model and Why does it Fail?” *Geekbot Blog*, Apr. 05, 2019. <https://geekbot.com/blog/what-is-the-traditional-waterfall-model-and-why-does-it-fail/>
- [2] H. J. M. Ruël, T. Bondarouk, and S. Smink, “The waterfall approach and requirement uncertainty,” *International Journal of Information Technology Project Management*, vol. 1, no. 2, pp. 43–60, Apr. 2010, doi: 10.4018/jitpm.2010040103.
- [3] Y. Grushka-Cockayne, V. Holzmann, H. Weisz, and D. Zitter, Eds., *A new hybrid approach for selecting a project management methodology*. Project Management Institute, 2015. [Online]. Available: <https://www.pmi.org/learning/library/consistent-approach-provides-high-performance-9889>
- [4] A. Gallagher, J. Dunleavy, and P. Reeves, “IBM Developer,” Apr. 23, 2019. <https://developer.ibm.com/articles/waterfall-model-advantages-disadvantages/>
- [5] B. Roseke, “Guide to Project Change Management,” *ProjectEngineer*, Oct. 22, 2019. <https://www.projectengineer.net/guide-to-project-change-management/>
- [6] O. Nazarenko, “Agile and waterfall change Management Process | MindK,” *Web and Mobile App Development Company — MindK.com*. <https://www.mindk.com/blog/change-control-process/>
- [7] “Agile vs Scrum vs Kanban – Compare project management methodologies,” *Studio by UXPin*. <https://www.uxpin.com/studio/blog/agile-vs-scrum-vs-kanban/>
- [8] R. Lynn, “The history of Agile,” *Planview*. <https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/history-of-agile/>
- [9] M. Rehkopf, “Kanban vs. Scrum: Which Agile are you?” *Atlassian*. <https://www.atlassian.com/agile/kanban/kanban-vs-scrum>
- [10] T. Brown, “Scrum vs. kanban: Which agile framework is better?” *Opensource.com*, Aug. 08, 2019. <https://opensource.com/article/19/8/scrum-vs-kanban>
- [11] “A close up to agile project management Scrum vs. Kanban.” <https://www.stackfield.com/blog/scrum-versus-kanban-79>