International Journal of Multidisciplinary Research and Growth Evaluation.

# Waterfall to Agile: Transitioning SAP Projects for Greater Flexibility, Efficiency, and Testing Excellence

**Sireesha Perabathini**
Independent Researcher Illinois, USA

* Corresponding Author: **Sireesha Perabathini**

## Article Info

**ISSN (online):** 2582-7138
**Volume:** 05
**Issue:** 03
**May-June 2024**
**Received:** 01-05-2024
**Accepted:** 20-05-2024
**Page No:** 964-968

**Abstract**
SAP projects have traditionally been managed using the Waterfall methodology, which follows a linear, step-by-step approach to development and testing. However, many organizations are adopting agile methodologies as businesses demand faster delivery cycles and increased adaptability to change. Agile is based on iterative development, regular feedback, and team focus and is more customizable and efficient, especially for SAP implementations. In this paper, we describe the process and pros and cons of moving from Waterfall to Agile in SAP projects with a special focus on testing. It examines how agile frameworks (such as Scrum and Kanban) can be adapted for SAP projects, the integration of testing throughout the project lifecycle, and how quality assurance can be maintained in an Agile SAP environment.

**DOI:** https://doi.org/10.54660/.IJMRGE.2024.5.3.964-968

**Keywords:** Agile, Scrum, Kanban, Test-Driven Development (TDD) in SAP, Waterfall to Agile, Agile and Testing, Best Practices, Continuous Improvement

## 1. Introduction

SAP systems, including SAP ERP, SAP S/4HANA, and SAP Fiori, play a critical role in supporting enterprise-level business functions. Traditionally, SAP projects have adhered to the Waterfall methodology, where the process follows a rigid and sequential path—requirements gathering, design, implementation, testing, and deployment. However, this approach may lead to delayed feedback, extended testing periods, and longer business transformation cycles [4].

Agile processes, by contrast, are faster and more flexible and focus on continuous improvement, stakeholder engagement, and speed to market. With its emphasis on incremental development, Agile presents an opportunity for SAP projects to be more responsive to business needs and improve overall testing efficiency [1, 3].

### 1.1 Objectives of the paper

This paper aims to:

- Explore the primary challenges and benefits of transitioning SAP projects from Waterfall to Agile.
- Explore the role of testing in Agile SAP projects with a comparison to Waterfall.
- Provide strategies for adapting agile methodologies like Scrum and Kanban for SAP-specific needs.
- Discuss the role of continuous testing, automation, and quality assurance as a solution.
- Discuss case studies that demonstrate the successful implementation of Agile methodologies with a focus on the testing phase
- Waterfall to Agile: Discuss the Best Way to Work on the transition in SAP Projects.

## 2. Waterfall methodology in SAP projects

### A. Overview of waterfall in SAP projects

The waterfall model in SAP projects has a sequential, phase-based approach, which begins with the Requirements Gathering phase, where all business and technical requirements are gathered in advance. That leaves limited flexibility for changes later

In the process. Secondly, in the System Design phase, the system's architecture and configurations are planned in detail. This phase is followed by the Implementation phase, where the actual SAP system configurations and customizations are implemented according to the design documents. Testing is a separate and often lengthy phase where testers validate the entire system, including functional, integration, and regression testing. This phase typically occurs after implementation. Then finally, after the testing is successful, the system is rolled out, followed by ongoing support and maintenance. While the Waterfall model provides thorough documentation and extensive upfront planning, it has critical disadvantages in terms of flexibility and responsiveness to changes, particularly during the testing phase.

### B. Challenges of waterfall in SAP projects

The Waterfall model in SAP projects presents several challenges, including delayed feedback, as testing is performed only after the development phase, which often leads to the discovery of defects late in the process, resulting in higher costs for corrections. It does not have an adaptive change management system, so it's hard to scale to business requirements and scope once the design phase has begun. Additionally, testing is siloed from development and business engagement, which can lead to misalignment between the tested product and actual user needs. The long testing phase results in an extended time to market, delaying the delivery of the solution to the business. Stakeholder involvement is limited, typically only occurring during the requirements phase or at specific milestones, which means continuous feedback is often missed. Finally, the cost of defects is high, as issues discovered late in the process are more expensive to fix, especially when changes affect multiple areas of the system.

## 3. Agile methodology in sap projects

### A. Overview of Agile in SAP Projects

Agile methodologies, such as Scrum and Kanban, deliver incremental value through short development cycles known as sprints (typically 2-4 weeks). Agile prioritizes collaboration, flexibility, and continuous feedback, making it a great fit for environments that require quick adaptation, such as SAP projects [2].

- **Scrum in SAP:**

Scrum, a widely used Agile framework, defines key roles including the Product Owner, who represents business stakeholders and manages the backlog of requirements; the Scrum Master, who makes sure that the team adheres to Agile principles and removes any roadblocks; and the Development Team, a cross-functional group responsible for completing tasks within each sprint. Scrum can be applied to develop, configure, and test SAP modules in SAP projects iteratively. Each sprint delivers a usable product increment, enabling continuous business involvement and feedback, which helps ensure that the system remains aligned with evolving business requirements [2].

- **Kanban in SAP:**

Kanban is an agile framework, which deals with the work-in-progress and ensures that the tasks are completed efficiently. Kanban is highly graphical, making it easier to follow the workflow of tasks, and ideal for environments where continuous delivery is required, such as the regular updates and enhancements seen in SAP applications.

- **Benefits of agile in SAP projects:**

Agile offers several benefits, such as faster delivery, as it promotes delivering working software in small increments so that business stakeholders can realize value more quickly. As it is flexible, it enables the incorporation of changes in requirements, even late in the development process, which is crucial in fast-paced business environments. Agile enables better collaboration by facilitating frequent communication between development, testing, and business teams leading to effective coordination between teams and effective decision making. Additionally, continuous testing is integrated throughout the development process, leading to higher quality and fewer defects in production.

- **Challenges of agile in SAP projects:**

Teams Transitioning from Waterfall to Agile can face cultural resistance, as teams familiar with Waterfall's structured approach may be reluctant to embrace a more flexible and iterative nature. Additionally, scaling Agile practices to large, cross-functional SAP teams can present challenges, especially when working across multiple SAP modules or integrating with legacy systems. Also, SAP-specific constraints (hard configurations, integrations, legacy systems, etc) may require more upfront preparation than Agile usually does, complicating the adoption of Agile in such environments.

- **Hybrid agile methodology:**

While traditional agile methodologies like Scrum and Kanban are commonly used in SAP implementations, Hybrid Agile has become an effective approach for large-scale, complex SAP projects. This approach combines agile sprints for development and testing with Waterfall's more structured planning and deployment phases. By integrating the strengths of both methodologies, organizations can achieve faster delivery and clearer milestones [5].

## 4. Testing in agile SAP Projects

One of the biggest differences between Agile and Waterfall is how testing is approached. In Agile, testing is an integral part of the SDLC and not a separate phase. This provides ongoing feedback and allows teams to address issues early in the process.

### A. Continuous testing and test automation

In Agile SAP projects, testing is performed in parallel with development, often as part of each sprint. Continuous testing means that each piece of functionality is validated as it's developed, reducing the chances of defects building up over time [6]. Continuous testing is typically combined with continuous integration and continuous delivery (CI/CD) pipelines, ensuring that tests are run automatically with each code change and providing rapid feedback to developers and stakeholders. This enables teams to address issues early and continuously refine the product.

Test automation can be very beneficial in agile projects as some tests in SAP projects are repetitive. Automated tests, such as regression and performance tests, are integrated into the CI/CD pipeline, ensuring that changes do not break existing functionality [7]. Tools like SAP TAO, Tricentis Tosca, UFT, and Worksoft Certify are commonly used to automate testing in SAP environments, allowing teams to run automated tests quickly and frequently during each sprint.

Automated tests not only help accelerate testing but also ensure that the testing process remains consistent, reducing human error and improving reliability.

**B. Test Driven Development (TDD) in SAP**

Test-Driven Development (TDD) is a cornerstone practice in agile methodologies that focuses on writing test cases before the actual development code is created. TDD is an industry-standard used in software development for the quality and maintainability of code, but its use on SAP projects is a unique challenge with huge scope for improvement as regards customization, configuration, and integration with business processes.

Especially in the case of SAP, where customizations and configurations are essential to align the system with business needs, TDD can help organizations ensure that all custom functionalities are developed with a clear focus on business requirements from the outset. This strategy offers a proactive way to make the SAP implementations, defects, and overall system stability better.

**1) Core Principles of Test-Driven Development (TDD):**

The primary principle behind TDD is straightforward: developers/Testers write test cases before writing the actual functional code. This approach is structured around the Red-Green-Refactor cycle, which is repeated in the following stages:

- **Red:** Write a test case that defines a function or behavior. Initially, the test will fail since the code hasn't been written yet.
- **Green:** Write the minimum code required to pass the test. This step is focused on creating just enough functionality to pass the test.
- **Refactor:** Clean up and optimize the code to make sure it is efficient and maintainable without modifying the system functionality.

TDD can also have the most impact on SAP, for example, making sure SAP settings and modifications support business objectives from the start of the project.

**2) Benefits of TDD:**

Alignment with Business Requirements- One of the most significant challenges in any SAP project is to make sure that customizations and configurations meet the evolving business requirements. Test-driven development (TDD) takes a business-driven approach by ensuring that the customizations meet the expected outcomes from the very beginning. For example, when developing custom SAP Fiori applications or configuring SAP S/4HANA, TDD ensures that functional requirements—such as user roles, data flows, and specific business logic—are automatically checked, validated, and refined in the earliest stages of the development cycle.

Better Test Coverage & Early Defect Detection- TDD facilitates full test coverage from the start, by making tests pre-written in development. It encourages developers to take into consideration the business requirements and edge cases more seriously, which creates better code and a better understanding of potential defects or inconsistencies sooner. In the SAP context, TDD can be applied across various areas, such as ensuring custom SAP Fiori apps meet user experience and functional specifications, writing test cases for custom ABAP code or reports to meet performance and correctness standards, and automating integration tests to confirm that customizations work seamlessly with existing SAP modules (e.g., Finance, Logistics, HR) and third-party applications.

**3) Challenges of TDD:**

TDD is a great approach. However, when used in SAP projects, organizations may face some issues. SAP customizations are often the major hurdle faced which often involves complex business logic. Testing such customizations effectively is very demanding because it requires complex knowledge of SAP and business processes. Additionally, there is a lack of mature, out-of-the-box tools for TDD in SAP environments, especially for non-ABAP development like SAP Fiori. While frameworks such as ABAP Unit exist for ABAP, creating similar test suites for SAP Fiori, UI5, or integrations can be more difficult and may require extra setup and configuration. Finally, adopting TDD in SAP projects necessitates a cultural shift. Developers accustomed to traditional testing methods may initially resist the practice of writing tests before code, making training and change management crucial for successful implementation.

**C. Agile test management tools**

Several tools can assist in test management and tracking in Agile SAP projects:

- JIRA with XRay: XRay integrates with JIRA to support test management in an agile environment, allowing teams to track test cases, executions, and defects within the same tool.
- Zephyr: Another popular test management tool that integrates with JIRA, providing Agile teams with test case creation, execution, and reporting features.
- HP ALM: Although traditionally used in Waterfall projects, HP ALM can be adapted to manage testing in agile environments.
- Azure DevOps: Azure DevOps integrated tools for planning, tracking, and automating tests, enabling teams to manage test cases, defects, and test execution within a unified platform while ensuring seamless collaboration across development and testing cycles.

**5. Case Studies: Agile and testing**

**A. Case Study 1: SAP implementation using scrum**

A large public sector IT Company in Denmark implemented Test-Driven Development (TDD) for their SAP implementation to address challenges such as long development cycles and defects in final deliverables. By writing test cases before code, they ensured that SAP customizations met business requirements from the start. TDD allowed for continuous testing, early detection of defects, and improved collaboration between developers, business stakeholders, and testers. The result was a 40% reduction in development time, higher-quality deliverables, and better alignment with public sector needs. TDD's iterative feedback process led to a more efficient, stable SAP solution, delivering faster and more reliable outcomes for the clients [6].

**B. Case Study 2: SAP S/4 HANA implementation using agile**

A multinational food and beverage company in the US implemented a Hybrid Agile approach for their SAP S/4HANA migration to address the complexity of a global, multi-phase project. The hybrid method combined Waterfall for initial planning, requirements gathering, and high-level design, with Agile sprints for development, customization, and testing. This approach allowed the company to deliver incremental functionalities quickly while maintaining control over critical timelines and integration with legacy systems. Agile sprints enabled faster feedback and adaptability to business needs, while Waterfall enabled smooth deployment. The result was faster delivery, improved collaboration, and a seamless transition to SAP S/4HANA, meeting the business goals and reducing project risks.

## 6. Best practices for transitioning from waterfall to agile in SAP projects

### A. Establish clear objectives for transition:

Before transitioning to Agile, organizations should clearly define the goals and expectations of the change to be in alignment with business outcomes and address specific challenges experienced with Waterfall [4]. These objectives may include increasing responsiveness to business needs, improving collaboration between business, development, and testing teams, shortening the delivery cycle and time to market, and enhancing testing efficiency while reducing defects in production. By establishing clear objectives, organizations can better guide the transition to Agile, ensuring that its practices are adopted purposefully and with clarity.

### B. Gain executive and stakeholder buy-in:

Waterfall to Agile transition is to be agreed upon with all stakeholders in the organization, particularly by the executive level and key stakeholders [5]. Without this buy-in, teams may face resistance to adopting agile practices, and the transition may falter. The benefits of Agile, such as faster time to market, stakeholder engagement, and increased flexibility, should be clearly communicated to management to earn their support.

Involvement of Business teams during the agile process, giving frequent feedback, and analyzing priorities keeps the project in sync with business priorities. Intense partnerships between executives and stakeholders will keep things running smoothly and lay the foundation for success.

### C. Start with a pilot project:

A common best practice when transitioning from Waterfall to Agile is to start with a small, manageable pilot project. This approach allows organizations to experiment with agile practices on a smaller scale before committing to full-scale transition. The pilot project should consist of a cross-functional team and apply agile frameworks such as Scrum or Kanban.

The pilot should focus on a specific SAP module or area, and its success can serve as a learning opportunity for scaling Agile across the organization. Key metrics, such as time to market, defect rates, and stakeholder satisfaction, should be tracked to assess the effectiveness of the agile practices.

### D. Provide adequate training and resources:

A major challenge in migrating to Agile is the insufficient understanding of its principles and practices. For a successful transition, Organizations need to train their employees on Agile, Scrum, and Agile testing methods. This training should teach the key concepts of Agile, such as iterative development, continuous feedback, and collaboration. It should also explain the Scrum framework, including key roles, ceremonies, and artifacts like the Product Owner, Scrum Master, Sprint Planning, and Daily Standups. Additionally, the training should focus on Agile testing practices, such as test-driven development (TDD), continuous testing, and automation in the context of SAP projects. Teams should also be trained on agile tools like Jira, Azure DevOps, or Trello, which support project management and testing. Well-trained teams will be more confident in adopting the new approach, reducing confusion and frustration during the transition.

### E. Foster a collaborative and agile culture:

The biggest change in Waterfall to Agile is the change in the organization's culture. While Waterfall is monolithic and hierarchical, Agile emphasizes collaboration, communication, and empowerment. For a successful transition, organizations must foster a culture that embraces agile values. Encourage regular communication among business stakeholders, developers, testers, and project managers. Sharing project status, challenges, and progress openly with all team members will improve accountability in the team. It is very important to allow teams to make decisions, take ownership of their tasks, and continuously improve their processes, as this is important for maintaining morale and achieving high-quality outcomes. Encourage teams to cultivate a mindset of continuous learning and improvement, as Agile is an evolving process that requires teams to adjust their practices based on feedback and results.

### F. Implement agile gradually across teams:

Rather than transitioning the entire organization at once, it is often beneficial to introduce Agile incrementally across various teams. This approach can prevent overwhelm and resistance and allows teams to learn and grow with the methodology. Organizations should begin by introducing Agile to smaller, cross-functional teams that can work on SAP projects with fewer dependencies.

Gradual implementation allows teams to refine agile practices over time, adjusting and fine-tuning workflows before scaling to other parts of the organization. For larger SAP implementations, teams working on different modules can adopt Agile separately, with coordination between teams happening in regular joint sessions to ensure alignment.

### G. Adapt agile practices to SAP specific needs:

Agile processes like Scrum or Kanban might need to be tailored to the SAP projects *specifically*. SAP implementations are not just software projects but have complex system configurations, *third-party* integrations, and legacy system *integrations* that have their own unique issues that demand some modification to standard agile methodology. Longer sprint times might be required*; for example*, some SAP settings or configurations require a longer development time. Short sprint periods (e.g., 2-4 weeks) aren't always possible, and sprints may need to be prolonged or adjusted depending on *the complexity of the* task. *Additionally, i*f you're working on an SAP project and have legacy applications involved, the integration can be especially tough in an agile environment. Teams should also prepare for longer integration time and testing frequently for seamless SAP/legacy interoperability. And last but certainly not least, since SAP projects are so large and arduous, *the product backlog should be* very well- prioritized. Business stakeholders must review and prioritize backlog items regularly, so the team is focused on the most important functionality first.

### H. Use agile metrics to track progress and identify issues:

Agile metrics help to continuously monitor the progress and provide valuable insights into the project's performance by identifying potential bottlenecks and improvement opportunities. Key Agile metrics include velocity, which measures the amount of work completed by the team in a sprint, often represented in story points or work items; cycle time, which tracks the time it takes to complete a task or

Feature from start to finish; and defect density, which measures the number of defects identified during the sprint in relation to the amount of work completed. Other important metrics are sprint burndown (a visual representation of the work remaining vs time left in the sprint), and lead time, which measures the total time from when a backlog item is created to its completion and delivery. By using these metrics, agile teams can evaluate their performance, pinpoint areas for improvement, and continuously refine their practices.

**I. Review and adjust after each iteration:**
Continuous improvement is one of the fundamentals of Agile. Teams should conduct Sprint Retrospectives at the end of each sprint, looking at what worked, what wasn't working, and how it can be improved for the next sprint. Such retrospectives create an attitude of learning and adjusting which is necessary for long-term s Waterfall to Agile success. If teams implement what they learned from the retrospective, they can make changes to agile practices to suit the SAP project's specific needs and be more successful in the long run.

## 7. Conclusion
When choosing the right methodology for a project, there is no one-size-fits-all solution. Organizations should take into consideration the complexity of the project, customer requirements, the availability and skills of the team, and the level of uncertainty and change in the environment. Agile is better suited for projects that are dynamic, uncertain, and customer-oriented with unclear requirements and where feedback and learning are essential for success. Conversely, Waterfall is more suitable for projects that are static, predictable, and contract-oriented with well-defined requirements where quality and compliance are critical for success.
Transitioning from Waterfall to Agile in SAP projects is a major change that needs careful planning, well-defined goals, and willingness to embrace cultural change. While the transition may be challenging, the benefits of agile such as quick delivery, more flexibility, better collaboration, and improved testing—all of which can greatly enhance SAP implementations and create more value for the business.

## 8. References

1. Beck K, *et al*. Manifesto for Agile Software Development. Agile Alliance. 2001.
2. Schwaber K, Beedle M. Agile Software Development with Scrum. Prentice Hall. 2002.
3. Poppendieck T, Poppendieck M. Lean Software Development: An Agile Toolkit. Addison-Wesley. 2003.
4. Stepanov DY. Using waterfall, iterative and spiral models in ERP system implementation projects under uncertainty. Journal of Physics: Conference Series. 2021;2142:012016.
5. Ramachandran R. Best practices for agile project management in ERP implementations. ResearchGate. 2023. [Online]. Available: https://www.researchgate.net/profile/Ramya-Ramachandran-6/publication/387534405_Best_Practices_For_Agile_Project_Management_In_ERP_Implementations/links/67732968894c552085363275/Best-Practices-For-Agile-Project-Management-In-ERP-Implementations.pdf.
6. Beck K. Test Driven Development: By Example. Addison-Wesley. 2003.
7. Nadukuru S, Antara F, Chopra P, Renuka A, Goel O, Shrivastav EA. Agile methodologies in global SAP implementations: A case study approach. International Research Journal of Modernization in Engineering Technology and Science (IRJMETS). 2021. [Online]. Available: https://www.doi.org/10.56726/IRJMETS17272.