

**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA**  
**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND**  
**INFORMATION TECHNOLOGY**

**PROJECT BASED LEARNING**



**CASE STUDY ANALYSIS**  
**CONSUMER FINANCE COMPANY**

**Submitted by:**

**Khushi Gupta (9921103141)**

**Rahi Agarwal (9921103145)**

**Course Name:** Agile Software Development Process

**Course Code:** 22B12CS414

**Program:** B.Tech CSE

4th year, 8th Semester

**April, 25**

# TABLE OF CONTENTS

<b>Introduction.....</b>	<b>2</b>
<b>Case Study:.....</b>	<b>2</b>
<b>1. Traditional Development Methodology.....</b>	<b>2</b>
1)Adapted Waterfall Implementation Stages:.....	2
2)Benefits of this Adapted Waterfall Approach (in this challenging context):.....	7
3)Challenges of this Adapted Waterfall Approach (Significant in this context):.....	7
4)Significant Drawbacks of Relying Primarily on Waterfall in this Scenario:.....	8
<b>2. SCRUM.....</b>	<b>9</b>
1)Scrum Implementation Phases for Regulatory System Reinvention.....	9
2)Overall Benefits of Using Scrum:.....	15
3)Overall Challenges of Using Scrum:.....	15
4)Overall Drawbacks of Using Scrum (Potential Pitfalls):.....	16
<b>Conclusion.....</b>	<b>16</b>
<b>REFERENCES.....</b>	<b>17</b>

## Introduction

This analysis explores and compares two distinct software development methodologies – a traditional, adapted Waterfall approach and the agile Scrum framework – to determine the most effective strategy for a company facing the complex task of reinventing its core systems. The evaluation will delve into the typical implementation stages of each methodology, along with their inherent benefits, challenges, and potential drawbacks. Ultimately, the aim is to discern which approach offers the most viable path forward for successful system modernization and ongoing operational efficiency in a dynamic environment.

## Case Study:

The consumer finance company faced challenges due to changing state and federal regulatory compliance requirements, resulting in the need to reinvent their custom-built storefront and home office systems. The IT and PMO teams were not equipped to handle the complexities of developing new systems, leading to schedule overruns, turnover of staff and technologies, and the need to restart projects multiple times.

## 1. Traditional Development Methodology

Given the history of failures, lack of expertise, and the complexity driven by regulatory changes, a heavily **adapted and risk-mitigated Waterfall model**, potentially with elements of iterative feedback can be used.

### 1)Adapted Waterfall Implementation Stages:

#### 1. Extensive and Phased Requirements Analysis:

- **Activities:**

- **Deep Dive Regulatory Analysis (Phase 1):** Engage external legal and compliance experts to conduct a thorough and phased analysis

of all current and anticipated state and federal regulatory compliance requirements. Document these exhaustively.

- **Business Process Mapping:** Map current storefront and home office processes in detail to understand how they interact with the existing systems and where regulatory changes will have the most impact.
  - **Stakeholder Workshops (Iterative Mini-Cycles within Requirements):** Conduct multiple workshops with storefront staff, home office personnel, IT, PMO, and compliance to validate understanding of requirements and identify potential conflicts or overlooked areas. Allow for mini-iterations within this phase to refine requirements based on feedback.
  - **Technical Feasibility Study (Initial Pass):** A preliminary assessment of the feasibility of implementing the requirements with the existing (or potentially new) technology landscape.
  - **Detailed Requirements Specification Document (Living Document):** Create a comprehensive document that is treated as a "living document," acknowledging the potential for changes as regulatory understanding evolves. Implement a strict change control process.
- **Output:** A detailed, phased Requirements Specification Document with sign-offs at each phase, business process maps, and a preliminary technical feasibility report.

## 2. High-Level and Modular Design:

- **Activities:**
  - **Architectural Design:** Define the high-level architecture of the new systems, focusing on modularity and clear separation of concerns to ease future changes and maintenance.
  - **Data Modeling:** Develop a comprehensive data model that adheres to regulatory data retention and security requirements.
  - **Interface Design (Wireframing/Mockups - Iterative Feedback):** Create wireframes and basic mockups of key user interfaces for both storefront and home office systems. Obtain early and frequent feedback from end-users.
  - **Technology Stack Selection (with External Consultation):** Carefully evaluate and select the technology stack, potentially involving external consultants with expertise in building compliant financial systems.
  - **Integration Planning:** Detail how the new systems will integrate with existing infrastructure.
- **Output:** High-Level Design Documents, Data Model, UI Wireframes/Mockups with feedback logs, Technology Stack Justification, and Integration Plan.

### 3. Phased and Incremental Implementation (Construction):

- **Activities:**

- **Modular Development:** Develop the system in independent, functional modules. Prioritize modules based on the urgency of regulatory compliance.
- **Strict Coding Standards and Documentation:** Enforce rigorous coding standards and comprehensive documentation due to past staff turnover.
- **Regular Internal Builds and Integration Testing:** Frequent internal builds and integration testing of completed modules to identify issues early.
- **Dedicated QA and Testing Team (Potentially Augmented):** Establish a strong QA team, possibly augmented with external testing expertise, to ensure thorough testing of each module against requirements and regulatory guidelines.

- **Output:** Developed and internally tested software modules.

### 4. Rigorous and Independent Testing:

- **Activities:**

- **Unit Testing:** Developers test individual components.
- **Integration Testing:** Testing the interaction between modules.
- **System Testing:** Testing the entire integrated system against functional and non-functional requirements.
- **User Acceptance Testing (UAT) with Business and Compliance:** Involve end-users (storefront and home office staff) and compliance

personnel heavily in UAT to ensure the system meets their needs and regulatory obligations.

- **Independent Verification and Validation (IV&V):** Consider engaging an independent third-party to review the system for compliance and quality.
- **Output:** Comprehensive Test Reports, UAT Sign-offs, and IV&V Assessment.

## 5. Phased Deployment and Rollout:

- **Activities:**
  - **Pilot Deployment:** Deploy the new systems to a small group of storefronts or home office users for real-world testing and feedback.
  - **Gradual Rollout:** Based on the success of the pilot, gradually roll out the new systems to the remaining locations and users.
  - **Comprehensive Training and Support:** Provide thorough training to all users and establish robust support mechanisms.
  - **Data Migration and Cutover Planning:** Carefully plan and execute data migration from the old systems to the new ones.
- **Output:** Deployed and Operational New Systems, Training Materials, and Support Procedures.

## 6. Maintenance and Compliance Monitoring:

- **Activities:**
  - Ongoing monitoring of the systems for performance and stability.
  - Regular maintenance and bug fixes.

- Continuous monitoring of regulatory changes and proactive updates to the systems to maintain compliance.
- Establish a robust change management process for any future modifications.
- **Output:** Maintained and Compliant Systems.

## 2)Benefits of this Adapted Waterfall Approach (in this challenging context):

- **Structured Documentation for Compliance:** The Waterfall emphasis on documentation can be beneficial for demonstrating adherence to stringent regulatory requirements.
- **Clear Phases for Oversight:** Defined stages might provide a sense of control and allow for formal reviews and approvals, which can be important for risk mitigation.
- **Potentially Easier for Inexperienced Teams to Follow (Initially):** A linear approach might seem less daunting for teams lacking agile experience, although this can be deceptive.

## 3)Challenges of this Adapted Waterfall Approach (Significant in this context):

- **Inflexibility to Evolving Regulations:** The biggest challenge is the inherent inflexibility of Waterfall when dealing with *changing* regulatory requirements. Any significant regulatory shift during the later stages can lead to costly rework and schedule overruns, repeating past failures.
- **Late Feedback and Potential for Misalignment:** Despite the attempts at early feedback, the core of Waterfall relies on upfront requirements. If the understanding of regulations or user needs is incomplete or evolves, significant rework might be needed late in the project.



- **Risk of "Big Bang" Deployment:** Even with phased rollout, the underlying development is often sequential, increasing the risk of major issues being discovered only during integration or testing.
- **Difficulty in Adapting to Team Turnover:** The linear nature and potential for large, undelivered chunks of work can be severely impacted by staff turnover, as knowledge transfer can be difficult.
- **Lack of Early Value Delivery:** Users only see the working system late in the process, delaying the realization of business value and the opportunity for course correction based on real usage.

#### 4) Significant Drawbacks of Relying Primarily on Waterfall in this Scenario:

- **High Likelihood of Repeating Past Failures:** Given the history of schedule overruns and restarts, the inflexibility of Waterfall is a major risk factor in a context of changing regulations.
- **Increased Turnover Risk:** The lack of early wins and the potential for frustration with a rigid process can exacerbate staff turnover, especially if the team feels ill-equipped.
- **Potential for Massive Rework and Budget Overruns:** If significant regulatory changes occur mid-development, the need to backtrack and redesign can lead to substantial cost and time overruns, exceeding the company's capacity.
- **Lower Quality and Compliance Risks:** Delayed feedback and the potential for misinterpretations of complex regulations in the initial phases can lead to systems that are not fully compliant or user-friendly.
- **Missed Opportunities for Early Learning and Adaptation:** The lack of iterative cycles prevents the team from learning from early implementations and adapting the system based on real-world feedback and evolving understanding.

## 2. SCRUM

### 1) Scrum Implementation Phases for Regulatory System Reinvention

#### Phase 1 - Regulatory-Driven Product Backlog Emergence

- **Detailed Explanation:** This initial phase centers on creating a comprehensive and prioritized list of everything that needs to be built to reinvent the storefront and home office systems while ensuring full regulatory compliance. It's an ongoing process, not a one-time event.
  - **Activities:**
    - **In-Depth Regulatory Workshops:** Facilitated sessions involving legal experts, compliance officers, business stakeholders (storefront managers, home office department heads), IT architects, and potentially experienced Scrum Masters. The goal is to dissect each state and federal regulation relevant to the systems.
    - **User Story Mapping with Compliance Overlay:** Mapping out user journeys for various roles (loan officers, underwriters, customer service reps, etc.) and then overlaying specific regulatory requirements onto each step of the journey. This ensures that compliance is considered within the user experience.
    - **Technical Requirements Definition for Compliance:** Identifying the underlying technical needs to support regulatory compliance, such as data encryption, audit trails, reporting capabilities, data retention policies, and security protocols.
    - **Risk Assessment and Prioritization:** Evaluating the risk and impact of non-compliance for each regulatory requirement. The Product Owner, in collaboration with compliance and business stakeholders, prioritizes the backlog based on this risk, business value, and technical dependencies.

- **Initial Estimation by the Development Team:** The Development Team provides preliminary estimates (e.g., story points) for the effort involved in implementing each backlog item, including the work needed for compliance.
- **Definition of "Ready":** Establishing clear criteria that a Product Backlog Item (PBI) must meet before it can be selected for a Sprint (e.g., clear acceptance criteria, regulatory references, initial design considerations).
- **Focus:** Building a shared understanding of the regulatory landscape and translating it into a tangible, prioritized backlog that guides the development effort.

## Phase 2 - Short, Compliance-Focused Sprints

- **Detailed Explanation:** This phase involves the iterative execution of short work cycles (Sprints) with a specific focus on delivering working software that addresses the highest-priority regulatory requirements.
  - **Activities:**
    - **Sprint Planning Meeting:** The Scrum Team collaboratively selects a small number of high-priority PBIs from the top of the Product Backlog that they can realistically deliver within the Sprint timeframe. The Sprint Goal is defined, clearly articulating the compliance-related objective for the Sprint.
    - **Task Breakdown:** The Development Team breaks down the selected PBIs into smaller, actionable tasks and estimates the effort required for each.
    - **Sprint Backlog Creation:** The tasks and PBIs committed to for the Sprint are organized into the Sprint Backlog.
    - **Commitment and Transparency:** The team commits to the Sprint Goal and makes their plan visible.

- **Focus:** Delivering tangible, tested software increments that demonstrably meet specific regulatory requirements within a short timeframe, allowing for early validation and adaptation.

### **Phase 3 - Daily Transparency and Impediment Removal**

- **Detailed Explanation:** This phase involves a brief daily meeting to ensure the Development Team stays synchronized, identifies any roadblocks, and maintains focus on the Sprint Goal, particularly concerning compliance deliverables.
  - **Activities:**
    - **Daily Scrum (Stand-up):** Each Development Team member answers three key questions: What did I do yesterday towards the Sprint Goal (related to compliance)? What will I do today towards the Sprint Goal (related to compliance)? What impediments are preventing me from achieving the Sprint Goal (potentially compliance-related)?
    - **Scrum Master Facilitation:** The Scrum Master ensures the meeting stays within the timebox, facilitates communication, and takes ownership of removing any identified impediments that are hindering the team's progress on compliance tasks.
  - **Focus:** Maintaining daily visibility into progress on regulatory features, quickly addressing any issues that could delay compliance, and fostering a collaborative environment.

### **Phase 4 - Iterative Development with Built-in Quality and Compliance**

- **Detailed Explanation:** This is where the Development Team actively builds the software increments, with a strong emphasis on integrating quality assurance and compliance checks throughout the development lifecycle.

- **Activities:**
  - **Collaborative Development:** Developers work together, sharing knowledge and best practices, particularly around implementing complex regulatory logic.
  - **Test-Driven Development (TDD) or Behavior-Driven Development (BDD):** Writing automated tests (including those that verify compliance rules) before or during development to ensure the software meets requirements and regulatory obligations.
  - **Pair Programming and Code Reviews:** Encouraging collaboration and knowledge sharing, especially for critical compliance-related code, to ensure accuracy and adherence to standards.
  - **Continuous Integration and Continuous Delivery (CI/CD):** Implementing practices to frequently integrate code changes and automate the delivery pipeline, allowing for early and frequent testing of compliance features in integrated environments.
- **Focus:** Building high-quality, compliant software incrementally, with testing and verification embedded in the development process.

## **Phase 5 - Frequent Compliance Validation and Adaptation**

- **Detailed Explanation:** At the end of each Sprint, the team demonstrates the completed software increment to the Product Owner and key stakeholders, including compliance experts, to gather feedback and ensure the system meets regulatory expectations.
  - **Activities:**
    - **Sprint Review Meeting:** The Development Team showcases the working software increment. The Product Owner reviews it against the Sprint Goal and acceptance criteria, specifically focusing on whether compliance requirements have been met.

- **Compliance Stakeholder Feedback:** Compliance officers actively participate, providing feedback on whether the implemented features adhere to regulations and identifying any potential gaps or areas for improvement.
- **Product Backlog Refinement:** Based on the feedback received, the Product Owner updates the Product Backlog, potentially reprioritizing items or adding new ones to address evolving regulatory needs or identified issues.
- **Focus:** Regularly validating the implemented software against regulatory requirements, gathering crucial feedback, and adapting the plan based on this feedback to ensure ongoing compliance.

## Phase 6 - Empirical Process Improvement for Team Effectiveness

- **Detailed Explanation:** The Scrum Team reflects on the past Sprint to identify what worked well, what could be improved, and how to enhance their processes for future Sprints, with a specific focus on improving their ability to deliver compliant software efficiently and address past challenges.
  - **Activities:**
    - **Sprint Retrospective Meeting:** The entire Scrum Team (including the Scrum Master and Product Owner) participates in a facilitated discussion to identify areas for improvement in their processes, tools, and collaboration, particularly as they relate to building compliant systems and overcoming past issues like turnover and schedule delays.
    - **Action Planning:** The team identifies specific, actionable steps they will take in the next Sprint to implement the agreed-upon improvements.

- **Focus:** Continuously learning and adapting the team's processes to enhance their effectiveness in delivering compliant software and addressing the root causes of past project failures.

## **Phase 7 - Value-Driven and Compliance-Gated Releases**

- **Detailed Explanation:** Releases of the new storefront and home office systems are planned and executed based on delivering tangible business value while ensuring that critical regulatory milestones have been met and compliance is not compromised.
  - **Activities:**
    - **Release Planning:** The Product Owner, in collaboration with business and compliance stakeholders, defines release goals based on delivering a meaningful set of features and achieving key compliance milestones.
    - **Release Readiness Assessment:** Before each release, thorough testing and validation are conducted to ensure that the software meets all relevant regulatory requirements and quality standards.
    - **Deployment and Rollout:** The new systems are deployed to the production environment in a controlled manner, potentially using phased rollouts to minimize risk.
    - **Post-Release Monitoring and Support:** Ongoing monitoring and support are provided to ensure the stability and compliance of the deployed systems.
  - **Focus:** Delivering valuable, compliant software to end-users in a controlled and iterative manner, ensuring that regulatory obligations are met at each release.

## 2)Overall Benefits of Using Scrum:

- **Enhanced Adaptability:** Easily incorporates changing regulations through iterative cycles and flexible backlog.
- **Improved Compliance:** Frequent reviews with compliance stakeholders ensure ongoing alignment.
- **Increased Transparency:** Provides clear visibility into progress and potential risks.
- **Empowered Teams:** Fosters collaboration, ownership, and potentially reduces staff turnover.
- **Faster Delivery:** Short Sprints deliver working, compliant software increments early.
- **Better Alignment:** Ensures the system meets the needs of business, IT, and compliance.
- **Continuous Improvement:** Regular retrospectives help optimize processes for compliant delivery.

## 3)Overall Challenges of Using Scrum:

- **Strong Product Owner Needed:** Requires a Product Owner with deep business and regulatory expertise.
- **Mindset Shift Required:** Transitioning to Agile principles can be challenging for traditional teams.
- **Documentation for Compliance:** Balancing agility with the need for thorough regulatory documentation.
- **Managing Scope Creep:** The dynamic regulatory environment can lead to scope management difficulties.
- **Building Collaboration:** Requires effective communication and teamwork across different departments.



#### 4)Overall Drawbacks of Using Scrum (Potential Pitfalls):

- **Risk of Non-Compliance:** Without a knowledgeable Product Owner, regulatory requirements might be missed.
- **Ineffective Implementation ("Agile Fall"):** Superficial adoption of Scrum can lead to limited benefits.
- **Difficulty in Long-Term Planning:** Short Sprints might not easily accommodate long-term regulatory roadmaps.
- **Increased Communication Overhead:** Frequent meetings can become inefficient if not managed well.
- **Potential for Team Burnout:** Unsustainable Sprint pace can lead to stress and turnover.
- **Risk of Technical Debt:** Focus on speed over quality can create long-term compliance issues.

## Conclusion

While an adapted Waterfall approach might offer a semblance of familiarity, its fundamental limitations in handling change make it a less suitable choice for this consumer finance company facing evolving regulatory demands and a history of project failures. The risk of repeating past mistakes is substantial.

Scrum, on the other hand, with its iterative nature and emphasis on continuous feedback and adaptation, offers a more promising path to success. Its ability to incorporate changing requirements quickly and involve stakeholders throughout the development process aligns well with the challenges presented. However, the successful adoption of Scrum will require a strong commitment to Agile principles, effective leadership, and a willingness to embrace a new way of working. The company would likely need to invest in training and potentially external coaching to facilitate this transition.

## REFERENCES

1. W. W. Royce, "Managing the development of large software systems," in *Proceedings of IEEE WESCON*, vol. 26, pp. 1-9, 1970.
2. K. Schwaber and J. Sutherland, *The Scrum Guide*. Scrum.org and ScrumInc, 2017.
3. M. Cohn, *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional, 2009.
4. R. Pressman and B. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. McGraw Hill, 2020.
5. C. Larman and B. Vodde, *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional, 2009.
6. J. Highsmith, *Agile Project Management: Creating Innovative Products*, 2nd ed. Addison-Wesley Professional, 2009.
7. S. Sommerville, *Software Engineering*, 10th ed. Pearson Education, 2016.
8. M. Fowler and K. Beck, *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 1999.
9. IEEE Std 12207-2017, *IEEE Standard for Systems and Software Engineering — Software Life Cycle Processes*. IEEE, 2017.
10. A. Cockburn, *Agile Software Development: The Cooperative Game*, 2nd ed. Addison-Wesley Professional, 2007.