Grenze International Journal of Engineering and Technology, June Issue

Agile Methodologies for Embedded Systems

Aayushi Shukla[1] and Manisha Agarwal[2]
[1-2]Banasthali Vidyapith, Department of Computer Science, India
Email: aayushishuklartm@gmail.com, manishaagarwal18@yahoo.co.in

***Abstract*— The software development process has been completely transformed by Agile Methodologies, as it emphasizes iterative progress, flexibility and continuous involvement of stakeholders. The application of agile principles in embedded systems, especially the ones governed by Real-Time Operating Systems (RTOS), remains difficult because these systems demand strict timing, hardware dependencies and real time constraints. The complexity of embedded systems is continuously increasing, as their use in various applications is constantly growing. This paper discusses how we can adapt Scrum, kanban, and Extreme Programming (XP) to build embedded real-time systems. The focus is mainly on shorter sprint cycles, hardware backlogs and continuous integration. To understand the feasibility of this approach, a case study, Kavach, the Indian Railways Automatic Protection System is presented to demonstrate how agile practices address large scale development challenges, safety requirements and enhance the development process of embedded real-time systems.**

***Index Terms*— Agile Methodologies, Embedded Systems, Real-Time Operating System (RTOS), Safety-Critical Systems, Scrum, Kanban, Extreme Programming (XP), Kavach (Indian ATP System).**

I. INTRODUCTION

Agile methods are a go to approach in software development because it fulfills the demand for adaptability, flexibility, continuous integration and rapid delivery. Though the agile principles have proven effective in application development, their adoption in the domain of embedded systems brings a set of unique challenges, especially those governed by real-time operating systems.

Embedded real-time systems are essential in domains such as industrial automation, automotive, aerospace and railway safety, where system failures result in significant risks. For timely execution and deterministic task scheduling, these systems rely on RTOS, making their development process rigid and well suited for waterfall or V-model approaches. The predictability and verification demands of real-time embedded environments appears at odds with the adaptive, flexible and iterative nature of agile principles.

In embedded projects, integration of agile becomes non-trivial in aspects such as, co-development of hardware and software, difficulty in testing embedded code independently of hardware and the need of extensive verification and validation. Despite these barriers, the application of agile practices to embedded and real-time software domains is growing interest over the recent years. This paper explores how Scrum, Kanban, and Extreme Programming(XP) are adapted for embedded real-time systems. The Kavach System is presented as a case study, highlighting adaptations, benefits and challenges.

The remaining paper is structured as follows: Section 2 presents a literature review of agile methodologies and their applications in embedded real-time systems.

Grenze ID: 01.GIJET.12.2.87

Section 3 discusses the proposed agile methodology and the application of proposed methodology in the development of the Railway ATP System. Section 4 summarizes this paper with key findings and future directions.

II. Literature Survey

A. Agile Methodologies

Agile development methods emerged as a reaction to the rigidity of traditional frameworks such as waterfall. The Agile Manifesto [1] emphasized adaptability, closer collaboration with stakeholders, and iterative delivery rather than strict sequential planning. The most widely used agile methods are Scrum, Kanban, and Extreme Programming (XP).

Scrum

Scrum breaks down product development into short, time-boxed sprints[Figure]. It involves the role of Product Owner and Scrum Master and practices like daily scrum, sprint analysis, and continuous feedback cycles [2]. The scrum cycle is shown in Fig. 1.

Kanban

Kanban focuses on visual boards for the representation of workflows and limits work in progress, helping teams maintain flow and support incremental, continuous delivery [3]. The kanban workflow is shown in Fig. 2.

Extreme Programming (XP)

Extreme Programming (XP) emphasized on engineering excellence, including test-driven development and continuous integration, to maintain high levels of code quality while remaining responsive to requirement changes[4]. The XP practices are shown in Fig. 3.

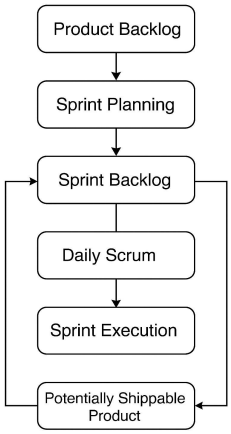

Figure 1. Scrum Cycle

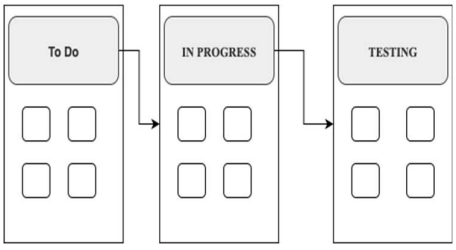

Figure 2. Kanban Workflow

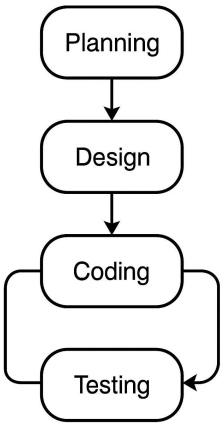

Figure 3. XP Practices

B. Embedded Systems and RTOS

Embedded systems combine specialized hardware with dedicated software to perform targeted functions. In many cases, they rely on a Real-Time Operating System (RTOS) to guarantee reliable execution of tasks within strict timing constraints [5]. Designing software in such contexts is difficult because it often depends on physical hardware for testing, faces uncertain integration schedules, and must satisfy rigorous assurance requirements, particularly in domains where safety is critical [6]. These factors require tailored practices and tools that align with real time requirements and makes direct application of agile difficult.

C. Agile in Embedded and Real Time Systems

Researchers and practitioners are showing a great interest in adoption of agile for embedded system development [12], [13]. In safety-critical projects, the adoption of Agile requires a balance between adaptability and strict verification mechanisms [7]. Researchers have proposed hybrid approaches, for example, blending Scrum with elements of the V-model to retain essential documentation practices [8]. Even in hardware-dependent contexts, Agile techniques have been shown to improve efficiency and productivity [9].

Studies exploring agile with embedded systems are mostly confined to sectors with high budgets and dedicated infrastructure, like medical, automotive and aerospace. The agile workflow for embedded real time systems is shown in Fig. 4.

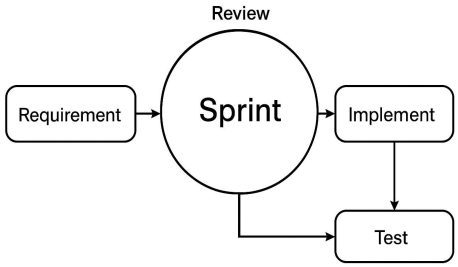

Figure 4. Agile Workflow for Embedded Real-Time Systems

III. Proposed Agile Methodology

Agile Methods offer adaptability, flexibility and continuous delivery which is against the constraints of real-time embedded systems [10]. However, we can achieve real-time performance and safety-critical requirements, by tailoring practices from Scrum, Kanban, and XP [11]. This approach ensures that agile methods align with the deterministic demands of real-time embedded environments [14]. The proposed framework is shown in Fig. 5.

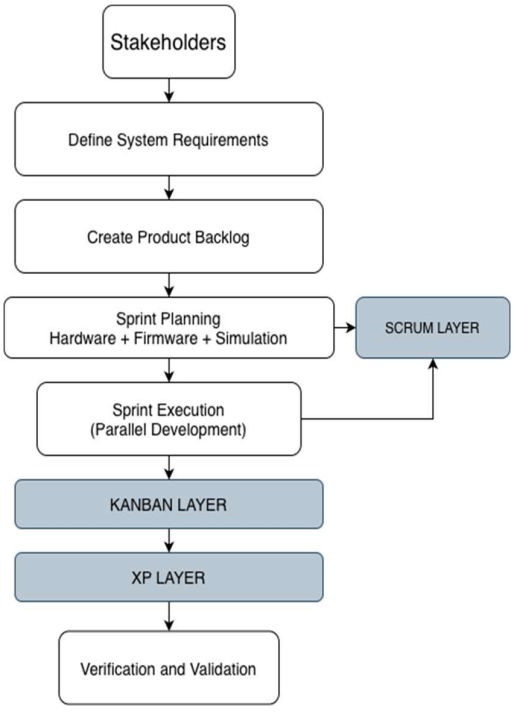

Figure 5. Proposed Agile Framework integrating Scrum, Kanban, and Extreme Programming(XP) for Embedded Real-Time Systems

A. Tailored Agile Approach for Embedded Real-Time Systems

Scrum Adaptation

Scrum is the most widely used agile methodologies. It's designed to manage complex product development through incremental and iterative cycles.

- Sprints are reduced to 1-2 weeks to reflect hardware software co-development cycles.
- Scrum Roles - Product Owner, Scrum Master, and Development Team, are redefined with technical leads assuming dual responsibilities for hardware and software interfaces.

- To avoid dependencies, backlog items are split into hardware, firmware, and testing tasks.
- Scrum guarantees continuous validation and feedback after every sprint.

The scrum layer is shown in Fig. 6.

Kanban Integration

Kanban is a workflow management emphasizing on continuous delivery and limiting work in progress.

- Introducing Kanban boards for the visualization of hardware availability, testing bottlenecks and verification queues
- Work in Progress limits, ensures that system integration does not outspace simulation or safety checks.

The kanban layer is shown in Fig. 7.

XP Practices for Quality and Safety

Extreme Programming focuses on enhancing the software quality and responsiveness to the changing environments. It also ensures that the code for critical and safety functions are maintainable and reliable.

- Incorporate pair programming for safety-critical code to improve quality.
- Use test-driven development for software modules wherever hardware emulation is feasible.
- Emphasize continuous integration pipelines with hardware-in-the-loop (HIL) testing setups.

The XP layer is shown in Fig. 8.

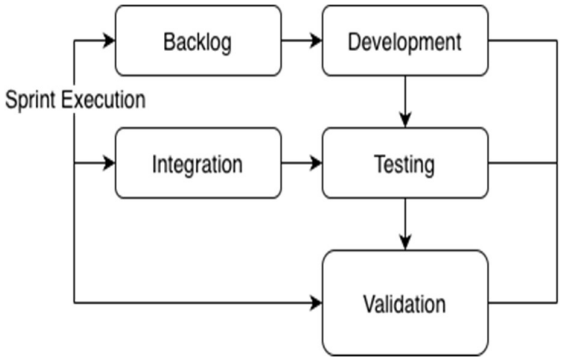

Figure 6. Scrum Layer

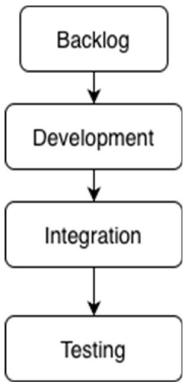

Figure 7. Kanban Layer

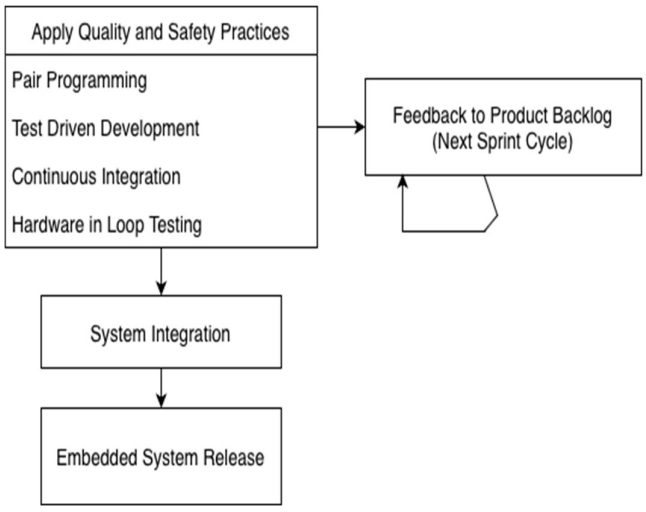

Figure 8. XP Layer

Documentation and Compliance

Each sprint ends with documentation artifacts that are traceable and satisfies regulatory audits and verification checkpoints. Use of automated traceability tools and version control ensures compliance without halting agility. The efficiency improvement across sprints is shown in Fig. 10.

B. Application of Agile Methodology in Kavach Development

Kavach, India's Automatic Train Protection (ATP) system, is a critical, RTOS based embedded system developed to prevent train collisions and enhance railway safety. This development had to meet strict real-time constraints, regulatory safety requirements, and large scale deployment challenges. The key features include:

- Collision Avoidance: Preventing rear-end and head-on collision by maintaining safe separation.
- Signal Passing: Preventing trains from crossing red signals by automatically applying brakes if the driver fails to respond.
- Monitoring:

Overspeed Monitoring - Monitor train speed and apply automatic brakes whenever speed exceeds limit.
Centralized Monitoring - For real time monitoring of train operations communication between control centers is performed.
The system overview of kavach is shown in Fig. 9.

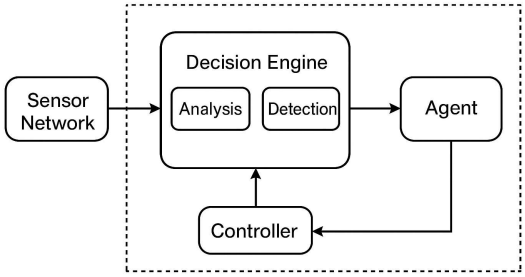

Figure 9. System Overview of Kavach

Challenges in Kavach Development

- Hardware dependencies for signaling and braking systems.
- Synchronization across locomotives, and centralized control systems.
- Safety compliance under SIL (Safety Integrity Level) guidelines.
- Geographic diversity and environmental constraints such as dust, vibration, high temperature.

Agile Implementation

- Scrum sprints were tailored to align with hardware module releases, enabling progressive rollout of features such as auto-braking, signal acknowledgement and loco to loco communication.
- Kanban boards track readiness across different zones and hardware vendors.
- XP practices such as versioned firmware releases and automated testing improves software stability and tracebility.
- Sprint reviews and daily stand ups helps in aligning development teams, private vendors, Indian Railways engineers.

The agile based kavach development workflow is shown in Fig. 11.
Observed Benefits

- Faster iterations in firmware testing through simulation environments.
- Improved cross team communication through stand ups and shared artifacts.
- Early detection in integration issues leading to reduction in regression bugs.
- Pilot deployment in railway corridors with enhanced safety features.

Application of agile practices in a sensitive manner, demonstrates the feasibility of agile in high assurance real time embedded systems within the Indian public infrastructure landscape.
Comparative analysis of traditional and proposed agile framework. This is shown in Table I.

TABLE I. TRADITIONAL VS PROPOSED AGILE FRAMEWORK

Parameter	Traditional Model	Proposed Agile Framework
Development	In sequential phases	In prints of 1-2 weeks
Integration	Late or at the end of the cycle	Continuous
Testing	After Integration	Continuous
Adaptibility	Rigid, cost of change is high	Highly adaptable
Documentation	Heavy and beforehand	Sprint-based
Deployment Time	Longer	Iterative
Integrated Communication	Isolated or non-effective	Daily stand-ups

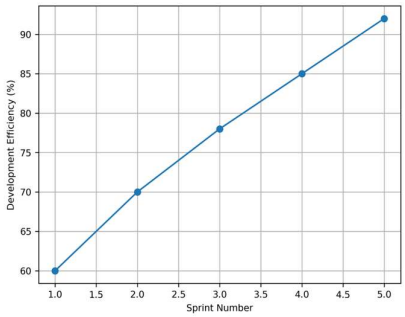

Figure 10. Efficiency improvements across sprints

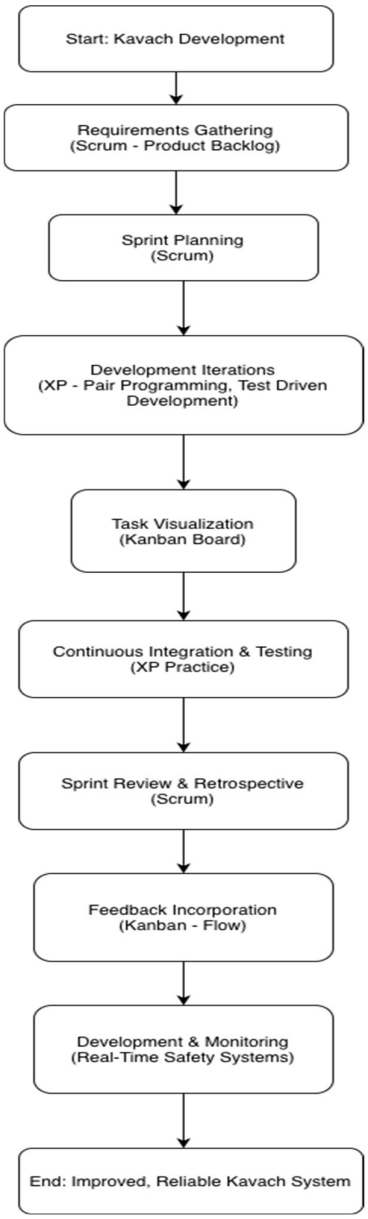

Figure 11. Agile-Based Kavach Development Workflow

IV. Conclusion

Agile methodologies are designed for iterative and flexible development and can be successfully adapted to the demands of embedded real time systems. Development teams can enhance productivity, ensure software quality and maintain compliance with safety and timing constraints. The application of Kavach illustrates how agile can be implemented in critical projects. This study opens up directions for further exploration such as agile metrics in RTOS environments, toolchain optimization, scalability and safety certification alignment. By continuing research and experimentation in this direction, agile practices can become an integral part of high assurance, embedded system development in both public and private sectors.

References

[1] K. Beck et al., "Manifesto for Agile Software Development," 2001.
[2] K. Schwaber and J. Sutherland, The Scrum Guide, Scrum.org, 2020.
[3] D. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, 2010.
[4] K. Beck, Extreme Programming Explained: Embrace Change, 2nd ed., Addison-Wesley, 2004.
[5] W. Wolf, Computers as Components: Principles of Embedded Computing System Design, 3rd ed., Morgan Kaufmann, 2012.
[6] B. Selic, "The Pragmatics of Model-Driven Development," IEEE Software, Sept. 2003.
[7] L. Damm and L. Lundberg, "Agile development of safety-critical software," in Proc. Int. Conf. Computer Safety, Reliability and Security, 2006.
[8] K. Raatikainen, M. Pikkarainen, and J. Maenpaa, "Agile development in embedded systems: A literature review," in Proc. 2011 Agile Conf., IEEE.
[9] H. Munir, P. Runeson, and K. Petersen, "Considering rigor and relevance when evaluating test-driven development: A systematic review," 2014.
[10] A. Qumer and B. Henderson-Sellers, "A framework to support the evaluation, adoption and improvement of agile methods in practice," J. Syst. Softw., vol. 81, no. 11, pp. 1899–1919, 2008.
[11] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," VTT Publications, no. 478, 2002.
[12] N. Bricout, F. Perin, and P. Berruet, "Towards an agile-based methodology for the development of embedded systems," in Proc. IEEE Int. Conf. Emerging Technologies & Factory Automation (ETFA), 2010, pp. 1–8.
[13] P. Diebold, J. Eckert, E. Ostberg, and M. Wagner, "What do practitioners think? An exploratory study on agile methods in embedded systems development," in Proc. IEEE 46th Hawaii Int. Conf. System Sciences (HICSS), 2013, pp. 4635–4644.
[14] R. Banerjee and S. Roy, "Applying agile methodologies to embedded systems development," in Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), 2014, pp. 1138–1143.