SENG 31232 - Assignment 01

M.M.A.Prasadini De Silva

SE/2019/047

Content

1. Lean

Lean methodology is a business approach focused on delivering value to customers through continuous improvement and a culture of respect for people.

Software development aligns naturally with Lean principles due to similarities with manufacturing in the following ways:

1. It typically adheres to a structured process.

2. It involves clear conditions of acceptance for deliverables.

3. It ultimately yields tangible value, often in the form of functional software.

The success of applying Agile and Lean principles to software development has sparked interest in other departments and industries. Lean methodology, originally derived from manufacturing practices pioneered by Toyota, emphasizes the elimination of waste, continuous improvement, and delivering value to customers.


By embracing below core principles of lean methodology, companies across various industries can streamline operations, deliver higher value to customers, and achieve sustained growth through continuous improvement.

1. **Value**: Understand and prioritize customer needs to deliver products or services with optimal value and pricing.

2. **Value Stream**: Analyze and optimize the entire product life cycle to identify value-adding processes and eliminate waste.

3. **Flow**: Continuously assess and synchronize processes within the value stream to ensure smooth operations and timely deliveries.

4. **Pull**: Produce goods based on actual customer demand to enhance flexibility and reduce delivery times.

5. **Perfection**: Embrace continuous improvement, applying rigorous measurements and making incremental changes to enhance productivity and customer value.

2. Kanban

Kanban is a dynamic workflow management system widely used in agile software development teams to organize and improve their performance. The term "kanban" originates from Japanese, translating to "billboard/placard," representing the system's foundation of using cards. Each card on the kanban board signifies a work item, progressing through vertical columns that depict its status. The board stays in motion, with new features pulled from the backlog and added to in-progress work based on team capacity.

It's important to note that Kanban is not a methodology itself, but a powerful tool used by agile teams to consistently deliver high-value features. Many product and software development teams rely on kanban boards as a simple yet effective way to visualize work and enhance collaboration. Popular tools like Notebooks, Roadmaps, and Develop offer built-in kanban functionality, providing product teams and developers with kanban-style workflow boards to streamline their processes.
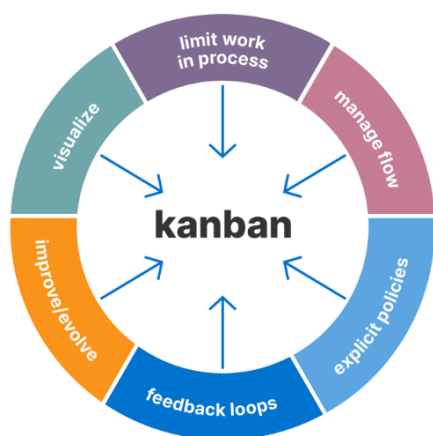
A kanban system is a defined workflow management method. By following these key steps, teams can effectively implement and utilize a Kanban workflow, enabling smoother collaboration, improved productivity, and better delivery of value to customers. Here are the essential steps to follow:

1. **Visualize the Workflow:** Map processes to a Kanban board (physical or virtual) based on current practices.

2. **Set WIP Limits:** Determine maximum cards allowed in each column to avoid overloading the team.

3. **Define Clear Policies:** Establish guidelines for backlog management and issue handling.

4. **Measure and Manage Flow:** Use metrics like cycle time and throughput to evaluate performance.

5. **Encourage Feedback and Improvement:** Empower team members to provide feedback and experiment with optimizations.
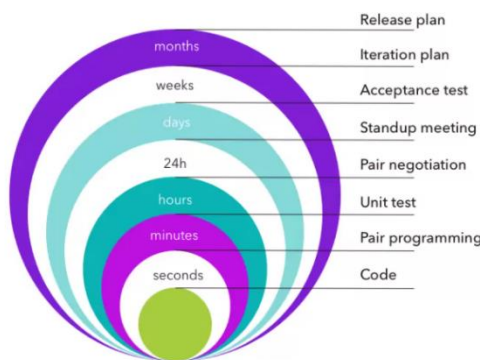
3. Extreme Programming (XP)

Extreme Programming (XP) is an Agile project management framework that prioritizes values, principles, and programming practices. Its primary objective is to empower small and medium-sized Agile teams to deliver top-notch software while being adaptable to evolving customer requirements. XP provides precise guidelines for developers, resulting in the delivery of high-quality software within the required timeframe.

Key characteristics of Extreme Programming include:

- Short development cycles, typically lasting one week.
- Focus on continuous delivery of small functionalities through practices like Test-Driven Development, Code Review, and Pair Programming.
- Core values such as simplicity, communication, feedback, commitment, and respect.
- Putting the product and developers at the center of the development process.
- Flexibility to adapt responsibilities and artifacts based on the team's needs.

By repeating iterative processes, Extreme Programming enables teams to deliver high-quality software while effectively adapting to changing requirements and fostering continuous improvement throughout the development lifecycle. Extreme Programming follows a set of 5 iterative processes:

1. **Planning:** The client presents requirements as User Stories, and the team creates a release schedule with iterations to deliver functionality incrementally. "Spike" solutions may be used for uncertain tasks, breaking them down into smaller user stories.

2. **Designing:** Part of the planning process, this step can be emphasized separately. It aligns with XP's value of simplicity, introducing logic and structure while avoiding unnecessary complexity.

3. **Implementation:** Writing code using XP practices like code standards, Pair Programming, and Continuous Integration.

4. **Testing:** A foundational pillar of XP, encompassing unit tests, automated tests, and customer acceptance tests to verify system compliance with initial requirements.

5. **Listening:** Ongoing communication and feedback between clients and project managers, describing business logic and its value.

4. Rational Unified Process (RUP)

The Rational Unified Process (RUP), also known as the Unified Process Model, is a web-enabled and object-oriented software development process. In object-oriented software engineering, visual models organized around objects are utilized, and analysis, design, and implementation are conducted iteratively. Developed by IBM in 2003, RUP was one of the earliest widely recognized iterative methods.

RUP is considered an agile methodology due to its flexible and adaptive approach, allowing teams to adjust and repeat cycles until meeting requirements and objectives. Its main goal is to construct high-quality software within predictable budgets and efficient timeframes.

As an iterative software development framework, RUP methodology involves a life cycle that divides each project into four distinct phases.

1. **Inception:** The team defines the project's structure, core concept, and viability based on costs, resources, and objectives.

2. **Elaboration:** Assessing requirements, designing, and creating a project plan while addressing significant risks. This phase marks the decision to proceed with development.

3. **Construction:** Building all product components, integrating them, and focusing on resource management to optimize costs, schedules, and quality.

4. **Transition:** The product is released to clients, and the team handles bug fixes, user difficulties, and completion of postponed features.

5. Six Sigma

Six Sigma, developed by a scientist at Motorola in the 1980s, is a data-driven methodology for process improvement and quality control. Practitioners use statistics, financial analysis, and project management to identify and correct mistakes or defects in existing processes, leading to improved business functionality.

The Six Sigma method follows five phases known as DMAIC: Define, Measure, Analyze, Improve, and Control. By utilizing statistical analysis and a disciplined approach, Six Sigma helps businesses significantly reduce defects and enhance process efficiency across various industries and organizations.

The process is DMAIC. This is a step-by-step approach to completing a project. These steps include:

1. **Define:** Clearly define the problem by creating documents like problem and goal statements, process maps, and requirements. The project charter serves as a high-level map of the process.

2. **Measure:** Determine the current performance of the process to understand the magnitude of the problem. This step establishes a baseline and provides data for measurable outcomes in the project charter.

3. **Analyze:** Identify the root causes of the problem through data collection and analysis. Utilize techniques like brainstorming and the five ways to uncover the underlying issues before proposing solutions.

4. **Improve:** Mitigate the problem's causes and enhance project outcomes. Brainstorm potential solutions, select the most practical one, and develop an improved process map. Also, decide on how to measure the improvement.

5. **Control:** Ensure the sustained improvement of the process. Implement a monitoring plan to continue measuring improvements and maintain the gains achieved. The control chart allows for further analysis and adaptation for continued enhancement.

References

1. Indeed: "Career Advice: Lean Methodology" - (Publisher: Indeed) - URL: https://www.indeed.com/career-advice/career-development/lean-methodology

2. Aha!: "What Is Kanban? A Quick Introduction to Kanban for Agile Project Management" - (Publisher: Aha!) - URL: https://www.aha.io/roadmapping/guide/agile/what-is-kanban

3. Nearshore-IT: "Extreme Programming Complete Guide" - (Publisher: Nearshore-IT) - URL: https://www.nearshore-it.eu/articles/project-management-leadership/extreme-programming-complete-guide/

4. Study.com: "Rational Unified Process Methodology: Examples" - (Publisher: Study.com) - URL: https://study.com/learn/lesson/rational-unified-process-methodology-examples.html

5. Project Skills Mentor: "How RUP Works" - (Publisher: Project Skills Mentor) - URL: https://projectskillsmentor.com/project-skills/how-rup-works

6. Think Thyme: "Six Sigma - A Complete Guide" - (Publisher: Think Thyme) - URL: https://thinkthyme.com/project-management/six-sigma

7. Investopedia: "Six Sigma" - (Publisher: Investopedia) - URL: https://www.investopedia.com/terms/s/six-sigma.asp