ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

UNIVERSITY FOR INDUSTRY

COLLEGE OF ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

SOFTWARE COMPONENT DESIGN

AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

SECTION B

Name	ID
EDEN NEKER	ETS0395/13
ELSABETH ZELEKE	ETS0417/13
ETSUB TILAHUN	ETS0446/13
EYERUSLAEM BEZU	ETS0454/13
FIKREYOHANNES GIZACHEW	ETS0509/13

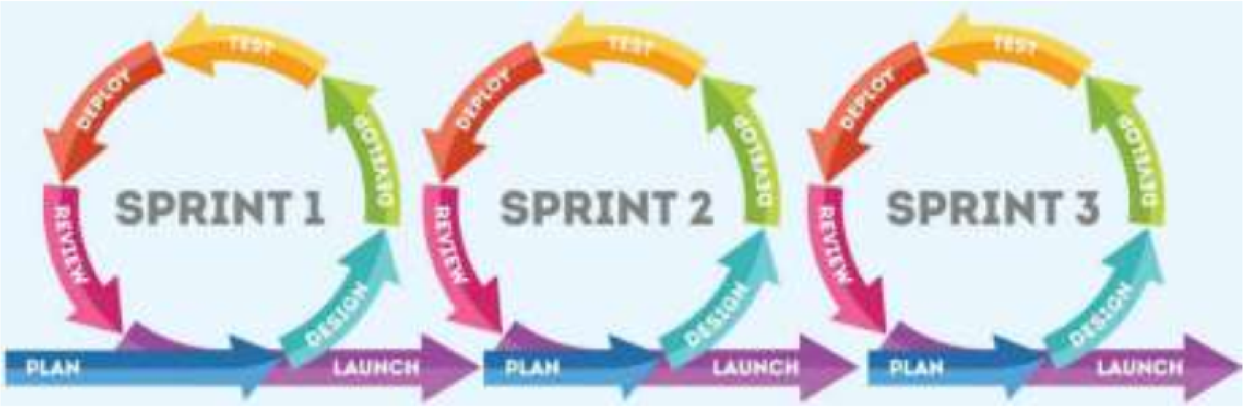
Submitted To: Mr. Gizatie Desalegn

Submission Date: Dec 18, 2024

Agile methodology

Agile methodology is a project management and software development approach that aims to be more effective, focuses on delivering smaller pieces of work regularly instead of one big launch, and allows teams to adapt to changes quickly and provide customer value faster. This methodology mainly emphasizes flexibility, collaboration, continuous improvement, rapid delivery of functional software, and customer-centricity through adaptive or change-driven life cycles. It involves iterative work cycles and incremental delivery of parts of the project, allowing for flexibility and customer feedback at each iteration.

Since the life cycle is repeated several times, the changes can be easily accommodated in the next iteration without disrupting the work. In addition to providing an opportunity to accommodate changes, each iteration also provides the opportunity to incorporate feedback and even change the scope of the project. It is also a hybrid of the iterative and incremental approaches that provides a solution to the challenges raised by the traditional approach by increasing the customer's value of the software after each iteration.



It is rooted in the Agile Manifesto, a set of principles introduced in 2001 to address the inefficiencies and rigidity of traditional development models like the waterfall methodology.

Key values of agile

The agile manifesto outlines four key values that guide agile practices. These are:

1. Individuals and Interactions Over Processes and Tools: This refers to focusing on empowering teams, communication, and collaboration rather than rigid adherence to tools and processes.
2. Working Software Over Comprehensive Documentation**:** This includes delivering usable software frequently rather than extensive documentation that might become obsolete.
3. Customer Collaboration Over Contract Negotiation: This refers to engaging customers regularly to ensure the software meets their needs, rather than relying solely on pre-defined contracts.
4. Responding to Change Over Following a Plan: This refers to adapting to changes quickly, even late in the development cycle, to provide value in a dynamic environment.

Principles of Agile

The Agile Manifesto is supported by 12 principles that help to implement the key values mentioned above. These are:

1. **Customer Satisfaction:** This indicates that the highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. **Changing requirements:** This refers to welcoming changing requirements, even late in development, to ensure the customer's competitive advantage.
3. **Frequent (time-based) Delivery:** This indicates delivering working software frequently; it encourages the regular release of functional software increments in short iterations, which enables faster feedback and adaptation to changing requirements.
4. **Collaboration:** This indicates business people and developers must work together daily throughout the project. There should be communication and collaboration between stakeholders and the development team regularly to ensure transparency.
5. **Team Motivation and Support:** This indicates building projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. **Face-to-face conversation:** This indicates the most efficient and effective method of conveying information to a development team is face-to-face conversation.
7. **Measure progress:** This refers to having working software as the primary measure of progress.

8. **Sustainable development:** This indicates agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. **Technical excellence:** This indicates that integrating continuous attention to technical excellence and good design enhances agility to maintain high standards of technical craft.
10. **Simplicity:** This concentrates on the most valuable features and tasks and avoids unnecessary complexity.
11. **A self-organizing team:** This emphasizes that the best architecture, requirements, and designs that emerge from self-organizing teams.
12. **Regular feedback and adjustment:** This indicates that at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Key characteristics of Agile

The key characteristics that distinguish agile software development methodology from others include:

- Iterative Development: Software is built incrementally in small, manageable cycles.
- Incremental Delivery: Deliver functional parts of the software frequently, adding features progressively.
- Customer Feedback: Involve stakeholders regularly to gather feedback and improve the product.
- Collaboration: Teams work closely across disciplines (developers, testers, designers, product owners) with open communication.
- Flexibility: Easily adapt to evolving requirements and changing priorities.
- Focus on Quality: Continuous testing, peer reviews, and integration practices are embedded in the process.

Common Agile Frameworks

Sample frameworks that implement agile principles include:

1. Scrum: This focuses on team collaboration and iterative development. The key roles in this methodology include the product owner, who defines the product backlog and prioritizes tasks, and the scrum master, who ensures agile principles are followed and

removes blockers and development teams who are responsible for completing tasks in each sprint. The core elements that define this process include sprints (time-boxed iterations, typically 2-4 weeks), daily Scrum (standup meetings to discuss progress and impediments), and sprint review and retrospective (reflect on the sprint and improve processes).

2. Kanban: This focuses on visualizing work and optimizing flow. The core elements of this methodology include a kanban board that tracks work-in-progress limits by restricting the number of tasks in each stage and continuous delivery.
3. Extreme Programming (XP): This focuses on high-quality software and technical excellence.The core practices in this methodology include test-driven development (TDD), pair programming, and continuous integration.
4. Lean: This focuses on eliminating waste and maximizing customer value, focusing on the principles of optimizing the whole process, empowering teams, delivering fast, and improving continuously.

Advantages of Agile

- Improved Flexibility: Respond quickly to changing requirements.
- Faster Time-to-Market: Deliver usable features sooner with incremental releases.
- Enhanced Collaboration: Encourages communication between stakeholders and developers.
- Higher Quality: Continuous testing and feedback improve software reliability.
- Customer Satisfaction: Regular delivery and iterative feedback to meet user needs.
- Reduced Risk: Small, frequent releases make it easier to detect and fix issues early.

Challenges of Agile

- Requires Discipline: Teams need a strong commitment to Agile principles.
- Unclear Deliverables: Lack of a detailed upfront plan can cause confusion.
- Dependency on Team Skill: Success relies on the competence of the team.
- Scalability Issues: Managing Agile at Scale.
- Stakeholder Availability: ensuring frequent involvement of stakeholders.

Reference

Schwalbe, K. (2019). Information Technology Project Management (9th ed.). Cengage Learning.

Project Management Institute. (2017). A guide to the Project Management Body of Knowledge (PMBOK guide) (6th ed.). Project Management Institute.