



ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

COLLEGE OF ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

Software Component Design

NAME	ID
1. Abel Negatu	ETS0033/13
2. Amen Tegenu	ETS0137/13
3. Betelhem Haile	ETS0232/13
4. Bethlehem Getachew	ETS0247/13
5. Bezawit Berihun	ETS0257/13
6. Bezawit Yehalem	ETS0264/13

Submitted to: Gizatie Desalegn

Submission Date: Dec 18,2024

Table of Contents

Introduction.....	3
Core Principles of Agile.....	3
Key Features of Agile.....	3
Phases of Agile Development.....	3
Importance of Agile.....	4
When to Use Agile.....	5
Advantages of Agile.....	5
Disadvantages of Agile.....	5
Popular Agile Frameworks.....	6
Conclusion.....	6

Agile Model

Introduction

The Agile model is a modern approach to software development that focuses on delivering value through iterative and incremental progress. It emphasizes collaboration, adaptability, and customer satisfaction, making it an ideal choice for dynamic and complex projects. Unlike traditional models, Agile allows teams to respond to changes quickly, ensuring that the final product aligns with user needs. By breaking work into smaller, manageable pieces called sprints, Agile promotes continuous delivery and improvement throughout the development cycle.

Core Principles of Agile

Agile is guided by the **Agile Manifesto**, which emphasizes:

1. **Individuals and Interactions** over processes and tools.
2. **Working Software** over comprehensive documentation.
3. **Customer Collaboration** over contract negotiation.
4. **Responding to Change** over following a plan.

Key Features of Agile

- **Iterative Development:** Work is divided into small, manageable increments called **sprints** (usually 1–4 weeks).
- **Customer Involvement:** Regular feedback ensures the product meets user needs.
- **Cross-functional Teams:** Developers, designers, and testers work together.
- **Adaptability:** Changes can be incorporated at any stage.
- **Focus on Quality:** Testing and reviews are integrated into the process.

Phases of Agile Development

Agile development is an iterative process that breaks down complex projects into manageable phases, ensuring adaptability and continuous improvement. Each phase contributes to maintaining a user-focused approach, allowing teams to deliver value incrementally. The following are the key phases of Agile development:

1. **Concept Phase**

The process begins by defining the project vision, goals, and user requirements. This phase establishes the foundation of the project by identifying the problem to solve and the

value to deliver. Stakeholders and team members collaborate to align on the project's scope and priorities.

2. **Iteration Planning**

Once the vision and goals are clear, the team divides the work into smaller, manageable units called sprints. During this phase, tasks are prioritized based on user stories, and a plan for the upcoming sprint is created. The goal is to ensure that the most critical and valuable features are addressed first.

3. **Design and Development Phase**

In this phase, the team works collaboratively to create prototypes, implement features, and conduct initial testing. Designers, developers, and testers interact closely to ensure the work aligns with user expectations and meets quality standards. The focus is on building working software incrementally.

4. **Review and Feedback Phase**

At the end of each sprint, the team presents deliverables to stakeholders in a sprint review. Stakeholders provide feedback, which the team uses to refine and improve the product. This phase ensures that the product evolves in line with user needs and business objectives.

5. **Release Phase**

Once a sprint's deliverables are polished and tested, they are released to users as a functional increment of the product. This phase provides an opportunity to gather real-world feedback, ensuring that the product delivers value and performs as expected.

6. **Repeat**

The cycle continues with subsequent sprints, iterating on the product until all objectives are achieved and the final product is completed.

Importance of Agile

Agile is significant in modern software development for several reasons:

- It ensures **customer-centric development** by incorporating regular feedback and adapting to changing requirements.
- Agile promotes **collaboration** across cross-functional teams, fostering better communication and transparency.
- It enables **faster delivery** through iterative sprints, allowing teams to provide working software frequently.
- Agile reduces the risk of project failure by focusing on **continuous testing and improvement** throughout the development cycle.

When to Use Agile

The Agile model is most effective in projects where requirements are likely to change or evolve, such as software development, product design, or innovation-driven industries. It is particularly suitable for projects with high levels of complexity or uncertainty, where rapid prototyping and regular feedback are critical. Agile is also ideal for teams that value collaboration and want to engage stakeholders actively in the development process. However, it may not be the best choice for projects with fixed requirements and timelines or for teams unaccustomed to iterative workflows.

Advantages of Agile

Agile offers numerous benefits, including:

- **Flexibility and Adaptability:** Teams can respond quickly to changes in requirements or market conditions.
- **Frequent Deliverables:** Regular releases ensure faster time-to-market and early user feedback.
- **Enhanced Collaboration:** Agile encourages teamwork and active communication among team members and stakeholders.
- **Improved Quality:** Continuous testing and iteration lead to high-quality products.
- **Customer Satisfaction:** Involving customers throughout the process ensures the final product meets their needs.

Disadvantages of Agile

Despite its strengths, Agile has some drawbacks:

- **High Dependency on Team Collaboration:** It requires a cohesive and communicative team, which can be challenging in remote or large teams.
- **Scope Creep:** Frequent changes can lead to uncontrolled growth in project scope.
- **Uncertain Timelines and Budgets:** The iterative nature can make it difficult to estimate project completion accurately.
- **Learning Curve:** Teams new to Agile may face challenges in adapting to its processes and principles.
- **Documentation Gaps:** Prioritizing working software can sometimes lead to insufficient documentation.

Popular Agile Frameworks

Agile is implemented through various frameworks, each designed to suit different project needs. Below are some of the most popular Agile frameworks:

- **Scrum:** a widely used framework focusing on short, time-boxed iterations called sprints (usually 1–4 weeks). It emphasizes roles like Product Owner, Scrum Master, and Development Team, as well as regular ceremonies such as daily stand-ups, sprint planning, and retrospectives.
- **Kanban:** a visual framework that uses a Kanban board to track tasks through columns like "To Do," "In Progress," and "Done." It emphasizes continuous delivery, limiting work in progress (WIP), and improving workflow efficiency.
- **Extreme Programming (XP):** this framework prioritizes high-quality software through practices like test-driven development (TDD), pair programming, and continuous integration. It focuses on technical excellence and frequent releases.
- **Lean Software Development:** derived from Lean manufacturing principles, this framework emphasizes waste reduction, faster delivery, and continuous improvement while delivering maximum value to the customer.
- **Feature-Driven Development (FDD):** a framework focused on building and delivering features in a short time. It combines model-driven and iterative approaches, making it ideal for large-scale projects.

Each framework offers unique advantages, allowing teams to choose the one that best fits their project's requirements and workflow style.

Conclusion

The Agile model is a transformative approach to software development, enabling teams to deliver high-quality products that align with user expectations. Its emphasis on flexibility, collaboration, and continuous improvement makes it well-suited for the fast-paced and ever-changing demands of modern projects. By adopting Agile, teams can navigate complexities, adapt to changes, and achieve meaningful results that drive success.