UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Agile Model

Submitted by:
Luminario, Venice Lou Gabrielle M.
Gabuyo, Ivan Love
Calica, Ljay
Enverzo, Kyle Andrey
Aquino, Jester
Elpedes, Glen Jeorge
Tan, Charles Dominic

Instructor:
Engr. Maria Rizette H. Sayo

December 13, 2025

I.

- Provide usable software in short iterations so customers get value early and continuously.
- Adapt to changing requirements even late in development without disrupting the project.
- Involve customers regularly to ensure the product meets real user needs.
- Identify and resolve technical and requirement risks early through small, incremental releases.
- Promote close cooperation between developers, testers, and stakeholders.
- Maintain quality through continuous testing, integration, and feedback.
- Make progress, issues, and priorities visible through regular reviews and updates.
- Focus on high-priority features to maximize business value in minimal time.

II.

1.Scrum

Uses short iterations called *sprints*, daily stand-up meetings, sprint planning, review, and retrospective to deliver incremental software.

2.Extreme Programming (XP)

Focuses on technical excellence using practices like pair programming, test-driven development (TDD), continuous integration, and frequent releases.

3.Kanban

Visual workflow management using boards to limit work in progress and improve flow efficiency.

4.Lean Software Development

Eliminates waste, improves efficiency, and delivers only what adds value to the customer.

5.Feature-Driven Development (FDD)

Develops software by designing and building features in short, structured iterations.

6.Dynamic Systems Development Method (DSDM)

Emphasizes user involvement, timeboxing, and frequent delivery of functional components.

7.Crystal Methods

A family of Agile methods tailored to team size and project criticality, focusing on communication and simplicity.

8.Adaptive Software Development (ASD)

Focuses on continuous adaptation through speculation, collaboration, and learning.

III.

The Agile model in software engineering is an iterative and incremental approach focusing on flexibility, collaboration, and rapid delivery of working software in small cycles (sprints). Instead of building everything at once, teams deliver functional features in short bursts (1-4 weeks), constantly gathering customer feedback and adapting to changing requirements, which contrasts with traditional waterfall methods. This process emphasizes teamwork, continuous improvement, and delivering business value quickly, ensuring higher quality and user satisfaction.

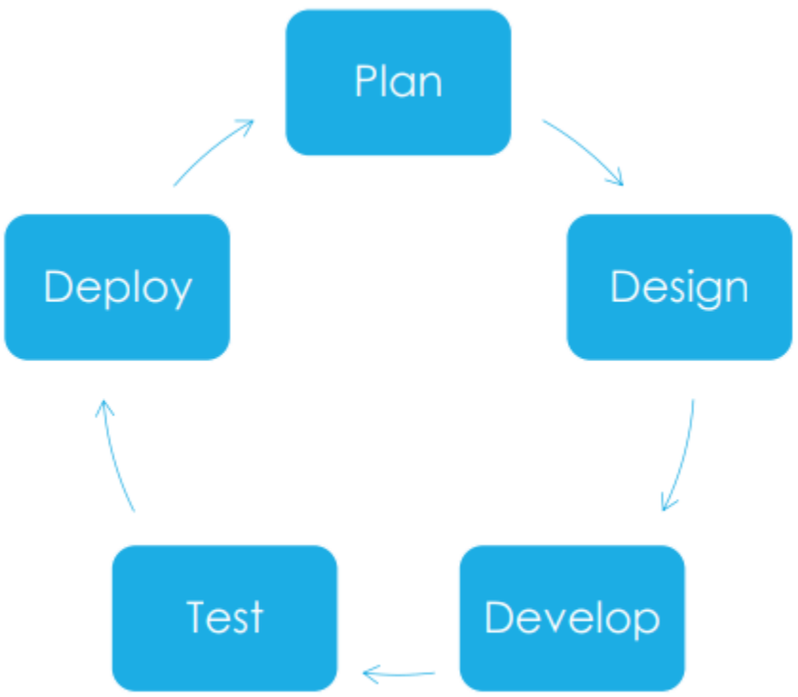

Figure 1 - Software Development Cycle

Key characteristics

Iterative & Incremental: Develop software in small, manageable chunks, with each release adding more features.

Short Cycles (Sprints): Work is broken down into short, time-boxed iterations (usually 2-4 weeks).

Collaboration: Close cooperation between developers, testers, and stakeholders (customers/users).

Adaptability: Embraces changing requirements and adjusts plans frequently.

Working Software: Focuses on delivering functional software at the end of each iteration.

Self-Organizing Teams: Empowers cross- functional teams to manage their own work.

Benefits

Faster Time-to-Market: Quick releases of valuable features.

Higher Quality: Continuous testing catches issues early.

Increased Flexibility: Easily accommodate changes in requirements.

Better Customer Satisfaction: Frequent delivery and feedback loops align with user needs.

Advantages of the Agile Model

- Working through Pair programming produces well-written compact programs which have fewer errors as compared to programmers working alone.
- It reduces the total development time of the whole project.
- Agile development emphasizes face-to-face communication among team members, leading to better collaboration and understanding of project goals.

- Customer representatives get the idea of updated software products after each iteration. So, it is easy for him to change any requirement if needed.
- Agile development puts the customer at the center of the development process, ensuring that the product meets their needs.

Disadvantages of the Spiral Model

- The lack of formal documents creates confusion and important decisions taken during different phases can be misinterpreted at any time by different team members.
- It is not suitable for handling complex dependencies.
- The agile model depends highly on customer interactions so if the customer is not clear, then the development team can be driven in the wrong direction.
- Agile development models often involve working in short sprints, which can make it difficult to plan and forecast project timelines and deliverables. This can lead to delays in the project and can make it difficult to accurately estimate the costs and resources needed for the project.
- Agile development models require a high degree of expertise from team members, as they need to be able to adapt to changing requirements and work in an iterative environment. This can be challenging for teams that are not experienced in agile development practices and can lead to delays and difficulties in the project.
- Due to the absence of proper documentation, when the project completes and the developers are assigned to another project, maintenance of the developed project can become a problem.

References

Agile Development Models - Software Engineering - GeeksforGeeks
https://www.geeksforgeeks.org/software-engineering/software-engineering-agile-development-models/