Gordon College
College of Computer Studies
ITE315A/L, CSE316A/L - Project Management and Agile Methodologies

Project Management

As defined by Gartner:
Project management is "the application of knowledge, skills, tools and techniques to project activities to meet the project requirements".

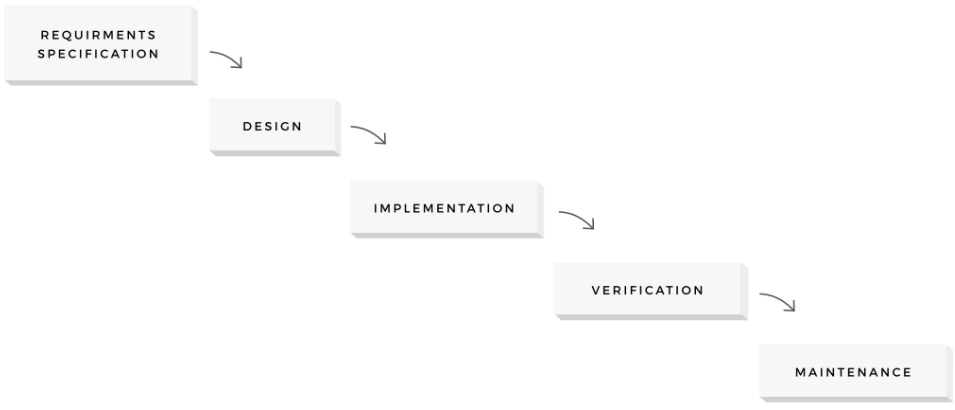
Project Management Phases

a typical project management process includes the following phases:

1. Initiation
2. Planning
3. Execution
4. Performance/Monitoring
5. Project close

Used as a roadmap to accomplish specific tasks, these phases define the project management lifecycle.

Traditional / Waterfall Model



1. Requirements Specification
2. Design
3. Implementation
4. Verification
5. Maintenance

Waterfall Model

Waterfall model has a strong emphasis on planning and specifications development: it is considered to take up to 40% of the project time and budget.
Another basic principle of this approach is a strict order of the project phases. A new project stage does not begin until the previous one is finished.

What is Agile?

All these words are associated with Agile:

- Framework
- Process
- Discipline
- Approach
- Plan
- Methodology
- Design
- Philosophy

Definition

Agile is an **approach** of building products or services by **EMPOWERING** and **TRUSTING** people, acknowledging **CHANGE AS NORM**, and promoting **CONSTANT FEEDBACK**.

Agile is a **PHILOSOPHY** that uses organizational models based on people, collaboration and shared values. Agile uses rolling wave planning; iterative and incremental delivery; rapid and flexible response to change; and open communication between teams, stakeholders, and customers.

Agile is a **MINDSET** established through **4 VALUES**, grounded by **12 PRINCIPLES** & manifested through many **DIFFERENT PRACTICES.**

4 Agile Values

1. Individuals & interactions over Processes & tools
2. Working software over Comprehensive documentation
3. Customer collaboration over Contract negotiation
4. Responding to change over Following a plan

That is, while there is value in the items on the right, we value the items on the left more.

12 Agile Principles

1. **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
    - **In Practice:** Instead of waiting months or years for a finished product, deliver working software in small, frequent increments (e.g., every two weeks). This provides value to the customer sooner and generates feedback early.

2. **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
    - **In Practice:** Change is not seen as a setback but as an opportunity to improve the product and ensure it better meets the user's current needs. Agile processes are built to adapt quickly to new information.

3. **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**
    - **In Practice:** This is the engine of agility. Short, time-boxed iterations (like Sprints in Scrum) create a regular rhythm of delivery, feedback, and adaptation.

4. **Business people and developers must work together daily throughout the project.**
    - **In Practice:** Breaks down the traditional "throw it over the wall" mentality. Constant collaboration (e.g., through a dedicated Product Owner) ensures the team is always building the right thing and misunderstandings are minimized.

5. **Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**
    - **In Practice:** Focus on people, not just processes. Management's role is to remove obstacles and empower teams, not micromanage them. Trust is a fundamental ingredient.

6. **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**
    - **In Practice:** While documentation has its place, direct communication is richer, faster, and less prone to misinterpretation. This principle emphasizes co-located teams or excellent virtual communication practices.

7. **Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**
    - **In Practice:** This is a direct rejection of "crunch time" and burnout. Sustainable pace (e.g., a 40-hour work week) ensures teams remain productive, creative, and healthy over the long term.

8. **Working software is the primary measure of progress.**
    - **In Practice:** Progress isn't measured by documents written, tasks completed, or hours worked. The only true measure of progress is a functional, tested product increment that delivers value.

9. **Continuous attention to technical excellence and good design enhances agility.**
    - **In Practice:** You can't go fast with bad code. Practices like refactoring, continuous integration, and Test-Driven Development (TDD) are essential. They keep the codebase clean and adaptable, making it easier to handle change.

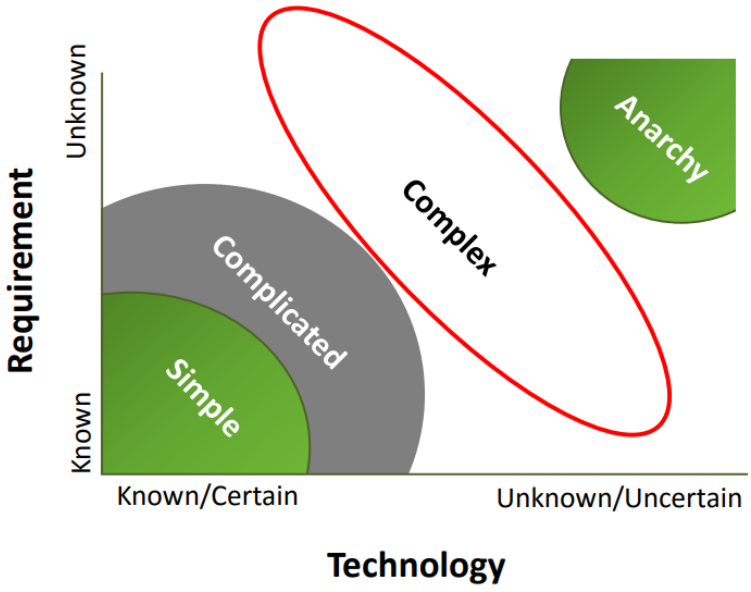
10. **Simplicity—the art of maximizing the amount of work not done—is essential.**
    - **In Practice:** Do the simplest thing that could possibly work. Avoid over-engineering and building features "just in case." This focuses effort on what delivers immediate value and reduces waste.

11. **The best architectures, requirements, and designs emerge from self-organizing teams.**
    - **In Practice:** Instead of being assigned tasks by a manager, the team (who has the technical expertise) decides how best to accomplish its work. This leverages the collective intelligence of the team and fosters ownership and innovation.

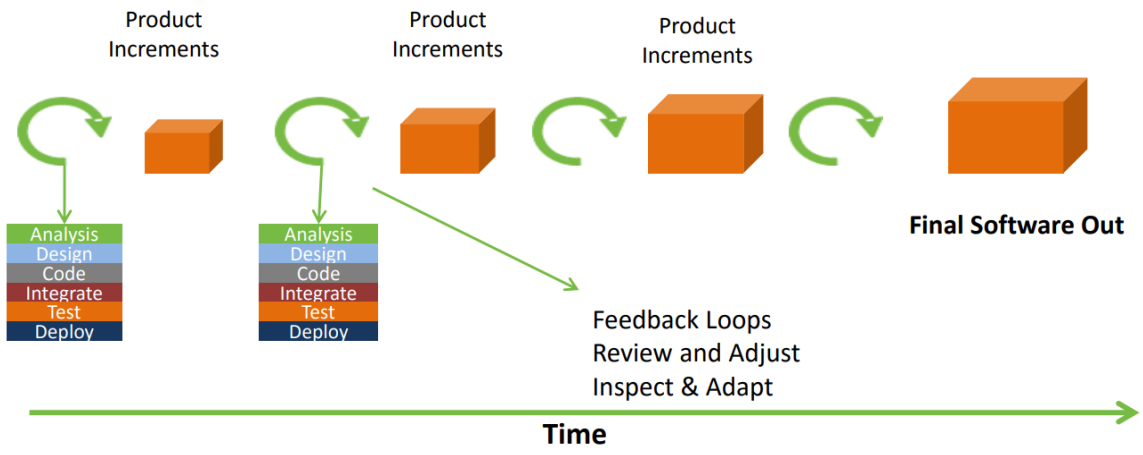
12. **At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.**
    - **In Practice:** The principle of continuous improvement. Through ceremonies like the Sprint Retrospective, the team regularly inspects its process and finds ways to improve, becoming more efficient and effective over time.

Select a principle and think how we can apply in our traditional projects.

<u>Project Noise and Method Selection</u>

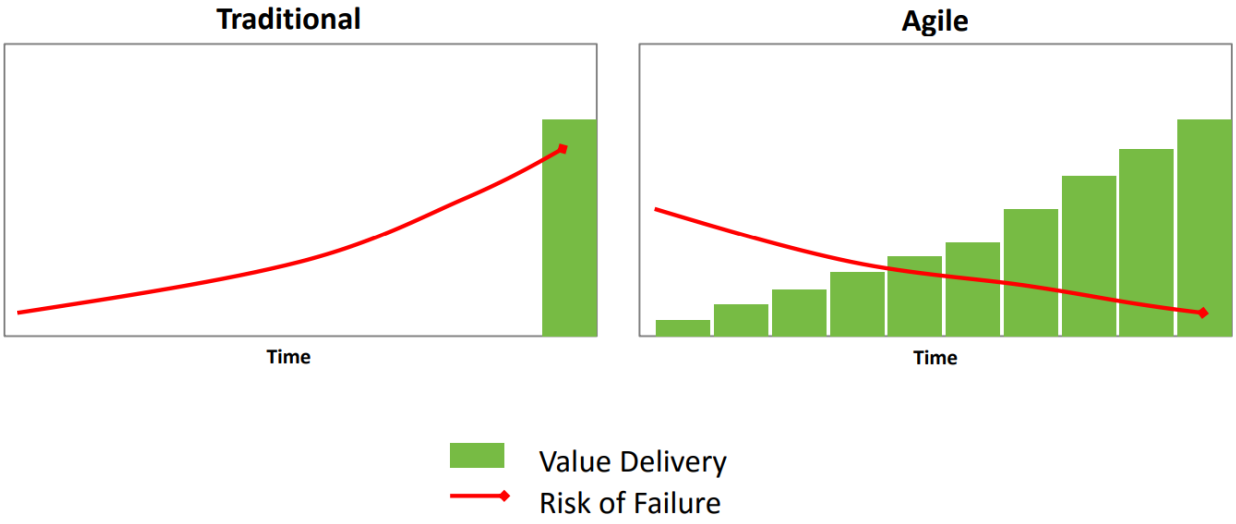


Agile Incremental Delivery



Agile adapts to frequent feedback by delivering working tested code.

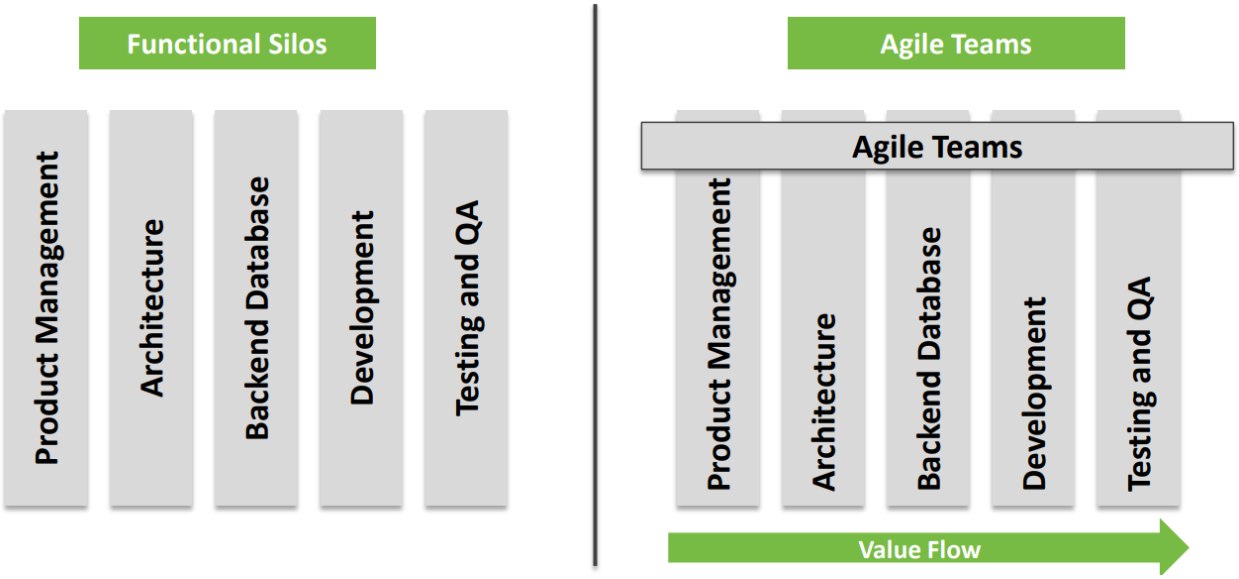
Traditional vs Agile Value Delivery



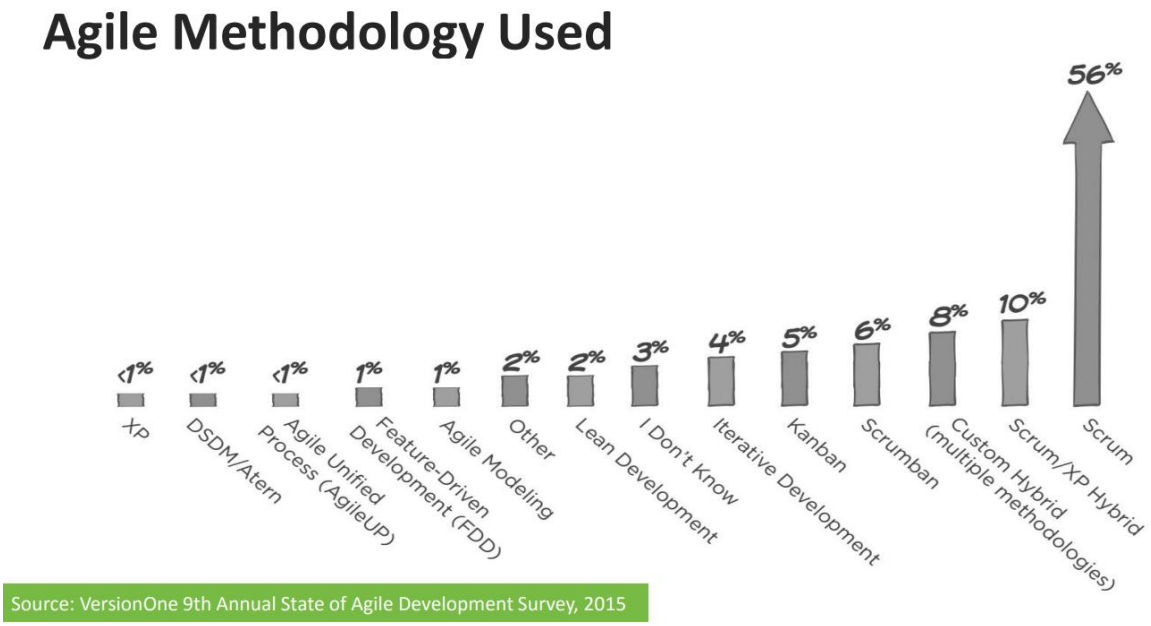
Traditional vs Agile Project Status Reporting

TRADITIONAL	AGILE
100% of the system	30% of the system
30% done	100% done
No testing yet	With known quality

Operating Model of Agile Team

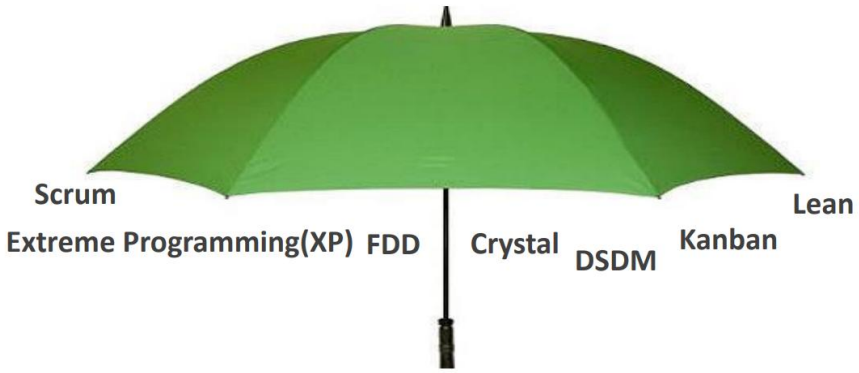


Various Agile Methods

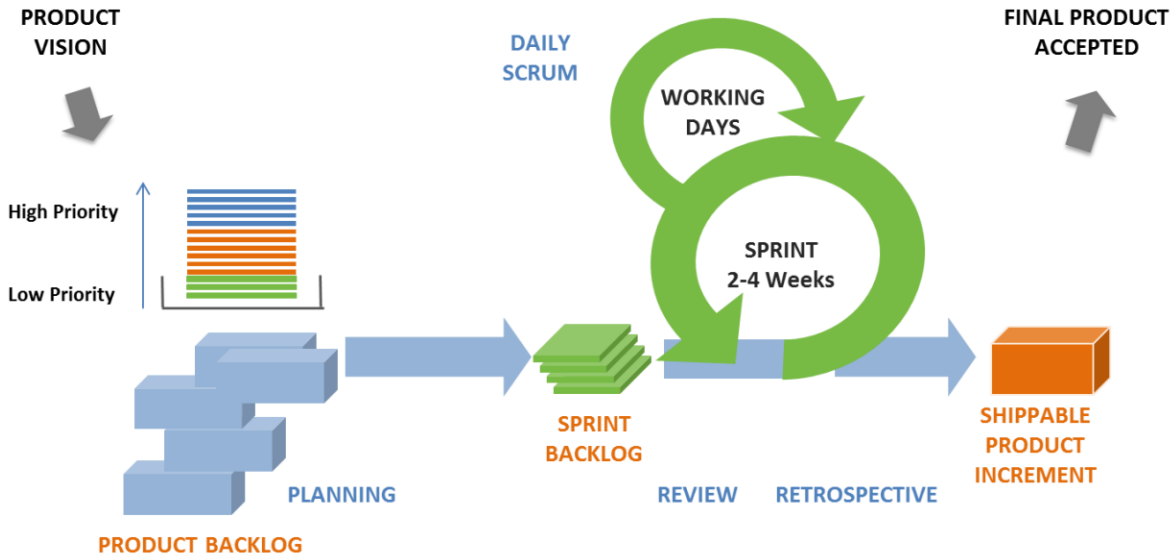


Source: VersionOne 9th Annual State of Agile Development Survey, 2015

Agile Umbrella



Scrum Process



Scrum Roles
Product Owner, Scrum Master and Development Team

Scrum Ceremonies
Product Backlog, Sprint Backlog and Increment

Artifacts
Sprint Planning, Daily Scrum, review, Backlog grooming /refinement, release planning and retrospective

Scrum Positives and Negatives

Positives

- The most common agile method
- Easy to understand and adapt
- Low barrier of entry
- Provide high level mechanics for complex work involving knowledge creation, and collaboration

Negatives

- No engineering practice defined
- Easy to follow path

Daily Scrum / Stand-up

- The daily stand-up is for and about the team and its commitments. In this meeting, the team checks in on how their work is progressing in the sprint, adjusts plans and gets assistance with removing impediments.
- Every day, same time, same place, same people. This provides a regular rhythm and cadence on everyone's calendar. The meeting lasts no more than 15 minutes.

Daily Scrum / Daily Stand-up Simulation

- What I did YESTERDAY?
- What am I planning to do TODAY?
- IMPEDIMENTS – If Any?

References:

https://www.altexsoft.com/media/2016/04/Agile-Project-Management-Best-Practices-and-Methodologies-Whitepaper.pdf
https://www.360pmo.com/docs/download/Agile%20Fundamentals%20for%20Project%20Managers.pdf