Overview of SDLC, Agile methodologies, Setting up the development environment

Lecture 1

Introduction

- Overview of Software Development Life Cycle (SDLC)
- Introduction to Agile Methodologies
- Setting Up the Development Environment

What is SDLC?

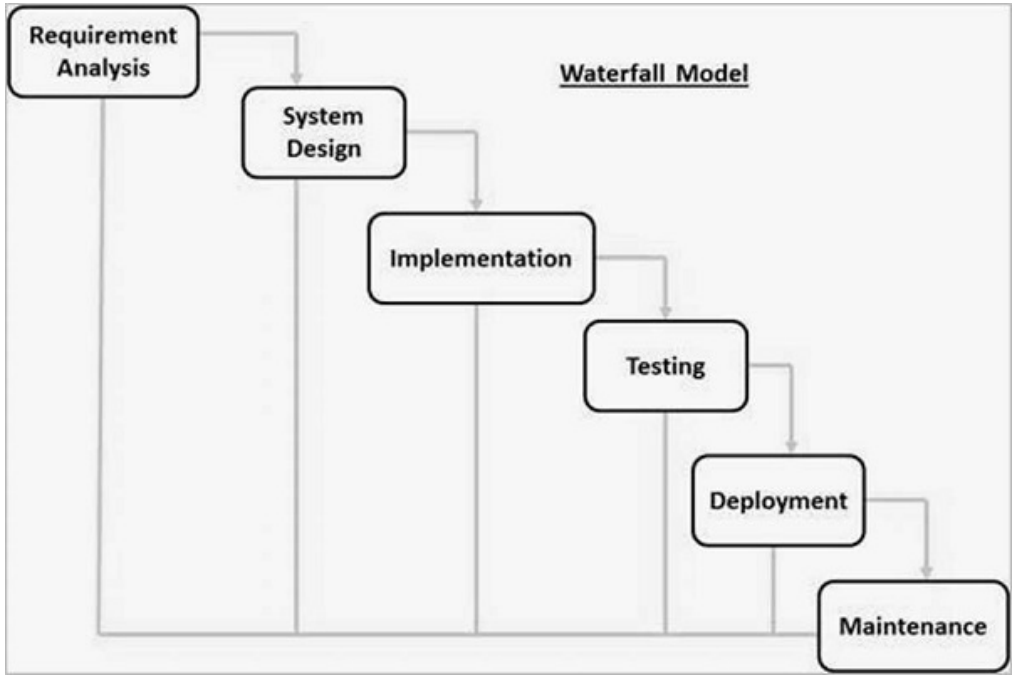
- Definition: The Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop, and test high-quality software.
- Phases of SDLC:
  - Requirement Analysis
  - System Design
  - Implementation (Coding)
  - Testing
  - Deployment
  - Maintenance

SDLC Models

- Common SDLC Models:
  - Waterfall Model
  - V-Model
  - Iterative Model
  - Spiral Model
  - Agile Model

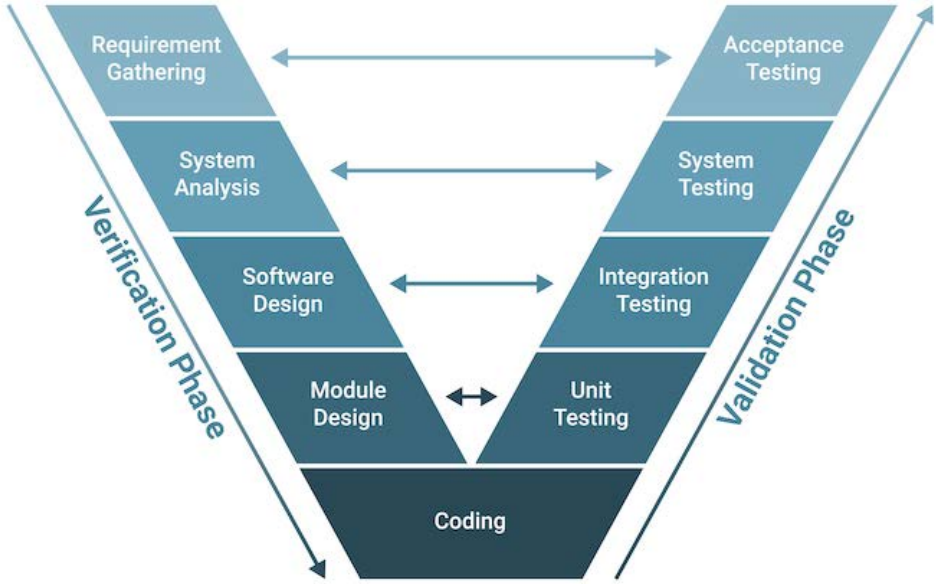
Waterfall Model

- Characteristics:
  - Sequential and linear approach.
  - Each phase must be completed before the next begins.
  - Phases: Requirements, Design, Implementation, Verification, Maintenance.
- Advantages:
  - Simple and easy to understand.
  - Well-defined stages and milestones.
  - Easy to manage due to its rigidity.
  - Suitable for projects with clear, stable requirements.
- Disadvantages:
  - Inflexible to changes once a phase is completed.
  - High risk and uncertainty.
  - Difficult to go back to any stage once it's completed.
  - Not suitable for complex and object-oriented projects.



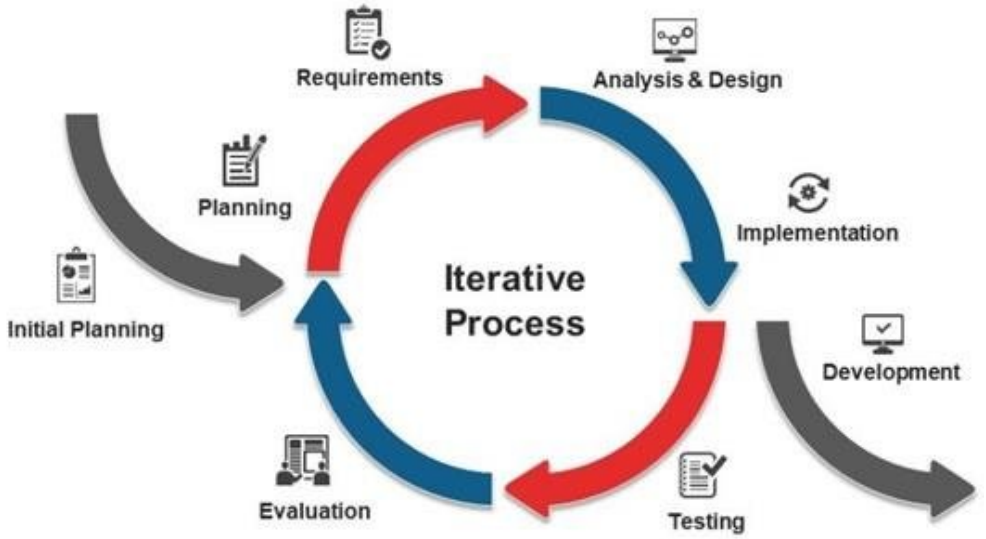
V-Model

- Characteristics:
  - Extension of the Waterfall model.
  - Each development stage is associated with a testing phase.
  - Phases: Requirements, System Design, Architecture Design, Module Design, Coding, Unit Testing, Integration Testing, System Testing, Acceptance Testing.
- Advantages:
  - Simple and easy to use.
  - Testing activities like planning and designing happen early.
  - Proactive defect tracking.
  - Works well for small to medium-sized projects with clear requirements.
- Disadvantages:
  - Very rigid and least flexible.
  - No early prototypes of the software are produced.
  - High risk and uncertainty.
  - Changes in requirements can be costly and time-consuming.



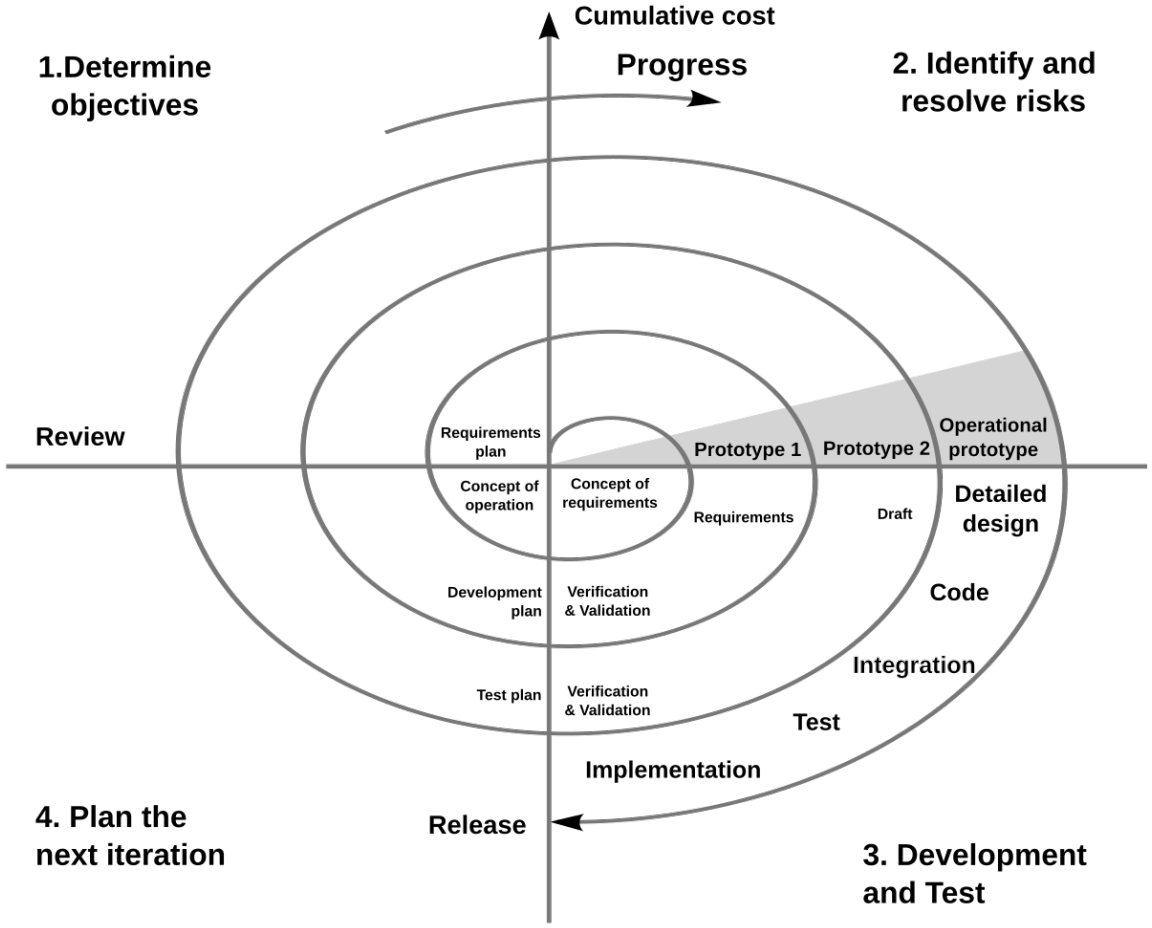
Iterative Model

- Characteristics:
  - Development starts with a simple implementation of a subset of the software requirements.
  - Iteratively enhances the evolving versions until the full system is implemented.
  - Phases: Planning, Analysis, Design, Implementation, Testing, Evaluation.
- Advantages:
  - Produces working software early in the lifecycle.
  - More flexible to changes in requirements.
  - Easier to test and debug during smaller iterations.
  - Low risk as risks are identified and resolved during each iteration.
- Disadvantages:
  - Requires good planning and design.
  - More resources may be required.
  - System architecture or design issues may arise because not all requirements are gathered upfront.
  - Each phase of an iteration is rigid with no overlaps.



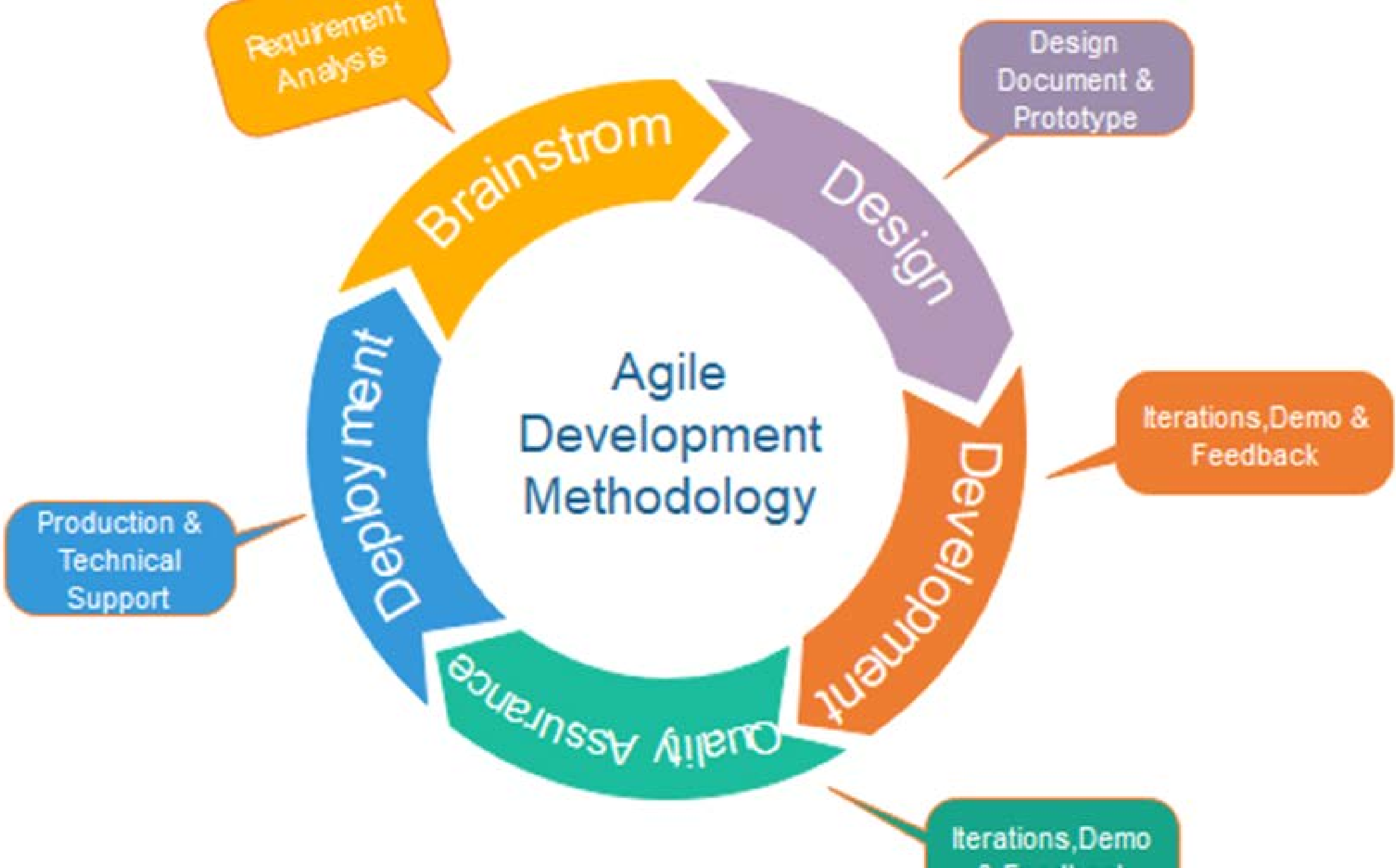
Spiral Model

- Characteristics:
  - Combines iterative development with systematic aspects of the Waterfall model.
  - Focuses on risk assessment and minimization.
  - Phases: Planning, Risk Analysis, Engineering, Evaluation.
- Advantages:
  - Risk management is a strong focus.
  - Flexibility in requirements.
  - Suitable for large, complex, and high-risk projects.
  - Customer feedback is integrated early and frequently.
- Disadvantages:
  - Can be expensive and time-consuming.
  - Requires expertise in risk assessment.
  - Complex to manage.
  - Not suitable for small projects.



Agile Model

- Characteristics:
  - Iterative and incremental approach.
  - Emphasizes flexibility, customer feedback, and rapid delivery.
  - Phases: Concept, Inception, Iteration/Construction, Release, Maintenance, Retirement.

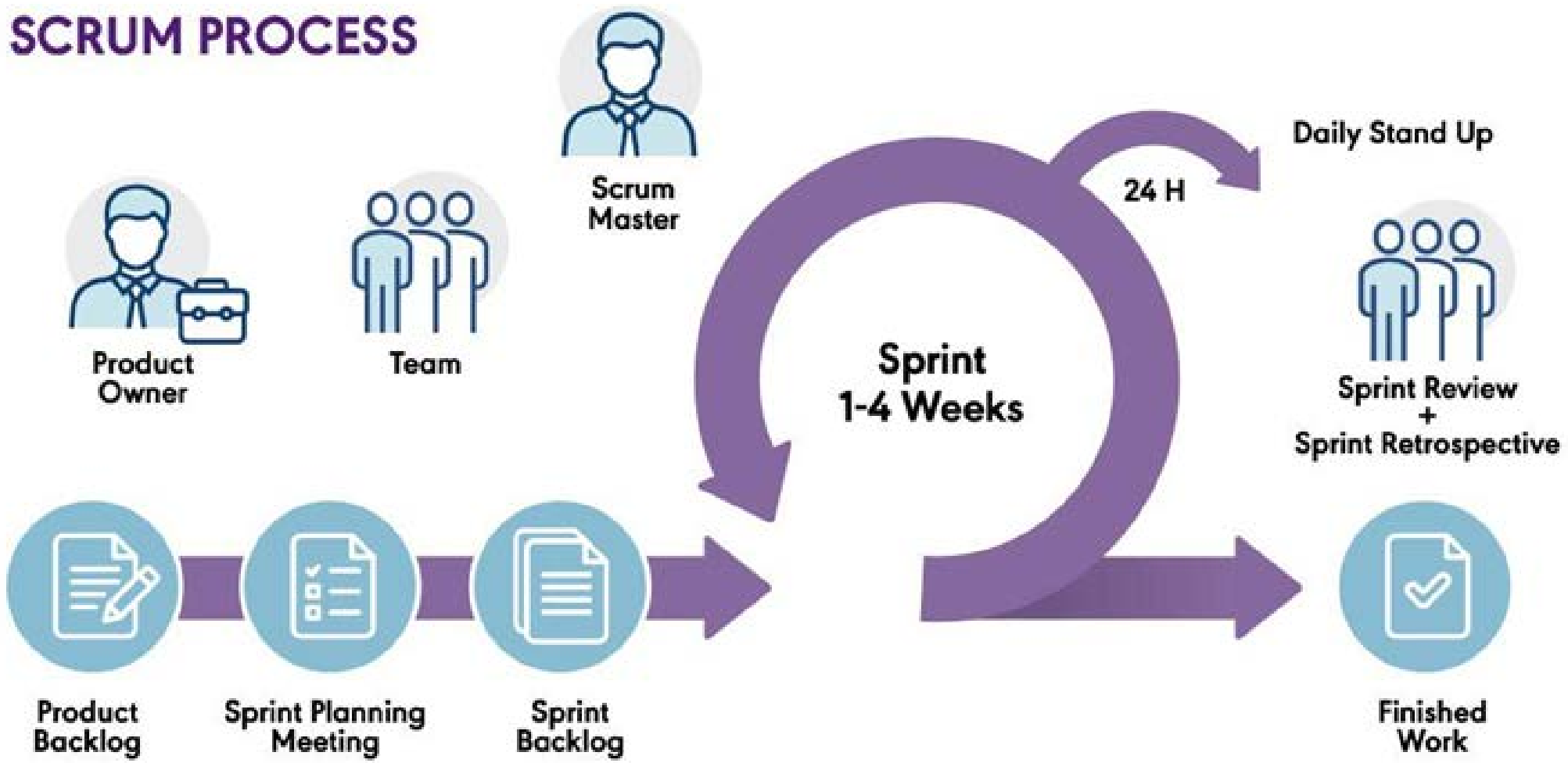

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Iterations,Demo & Feedback
Development
Iterations,Demo
Quality Assurance
Production & Technical Support
Deployment
Agile Development Methodology

Agile ...

- Advantages:
  - Highly flexible and adaptable to changes.
  - Continuous customer feedback.
  - Faster delivery of functional software.
  - Encourages collaboration and communication.
- Disadvantages:
  - Requires experienced and skilled team members.
  - Can be difficult to predict effort and time.
  - Less emphasis on documentation.
  - Can lead to scope creep due to changing requirements.

Scrum

- **Introduction:**
  - Scrum is a popular Agile framework used for managing complex software development projects.
  - It emphasizes iterative progress through sprints, which are time-boxed periods typically lasting 2-4 weeks.


SCRUM PROCESS
Scrum Master
Daily Stand Up
24 H
Product Owner
Team
Sprint 1-4 Weeks
Sprint Review + Sprint Retrospective
Product Backlog
Sprint Planning Meeting
Sprint Backlog
Finished Work

Contribution to Agile

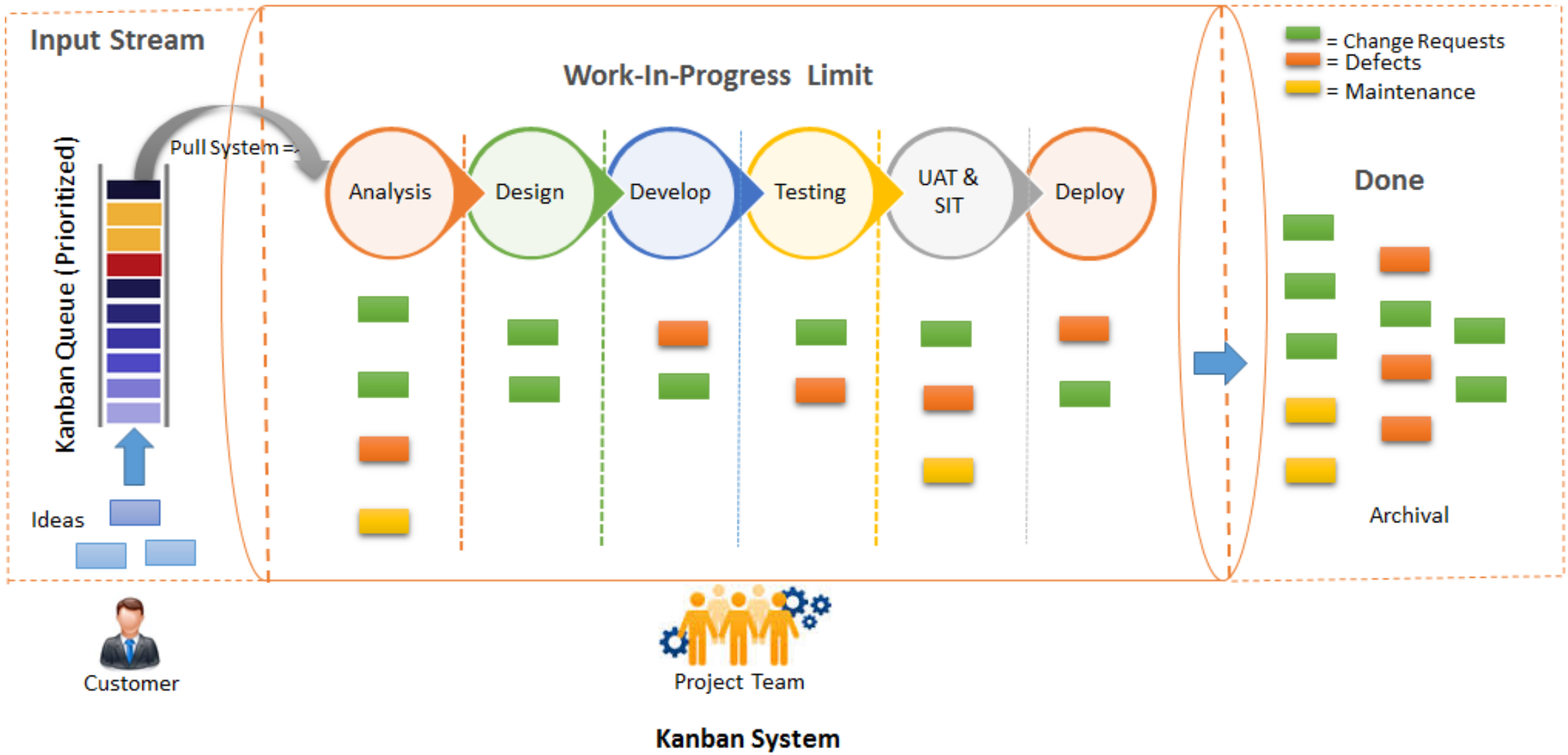
- Scrum provides a structured approach to Agile, promoting transparency, inspection, and adaptation.
- It encourages continuous improvement through regular feedback loops and retrospectives.

Scrum - **Unique Features**

- **Roles:**
  - **Product Owner:** Represents the stakeholders and is responsible for defining the features of the product.
  - **Scrum Master:** Facilitates the Scrum process, removes impediments, and ensures the team follows Scrum practices.
  - **Development Team:** A cross-functional group responsible for delivering the product increment.
- **Events:**
  - **Sprint Planning:** The team plans the work to be completed in the upcoming sprint.
  - **Daily Scrum:** A short daily meeting to synchronize activities and plan for the next 24 hours.
  - **Sprint Review:** The team demonstrates the work completed during the sprint to stakeholders.
  - **Sprint Retrospective:** The team reflects on the sprint and identifies improvements for the next sprint.
- **Artifacts:**
  - **Product Backlog:** A prioritized list of features, enhancements, and bug fixes for the product.
  - **Sprint Backlog:** A list of tasks to be completed during the sprint.
  - **Increment:** The sum of all the product backlog items completed during a sprint.

Kanban

- Introduction: Kanban is a visual workflow management method that helps teams visualize their work, limit work-in-progress, and maximize efficiency.



Kanban System

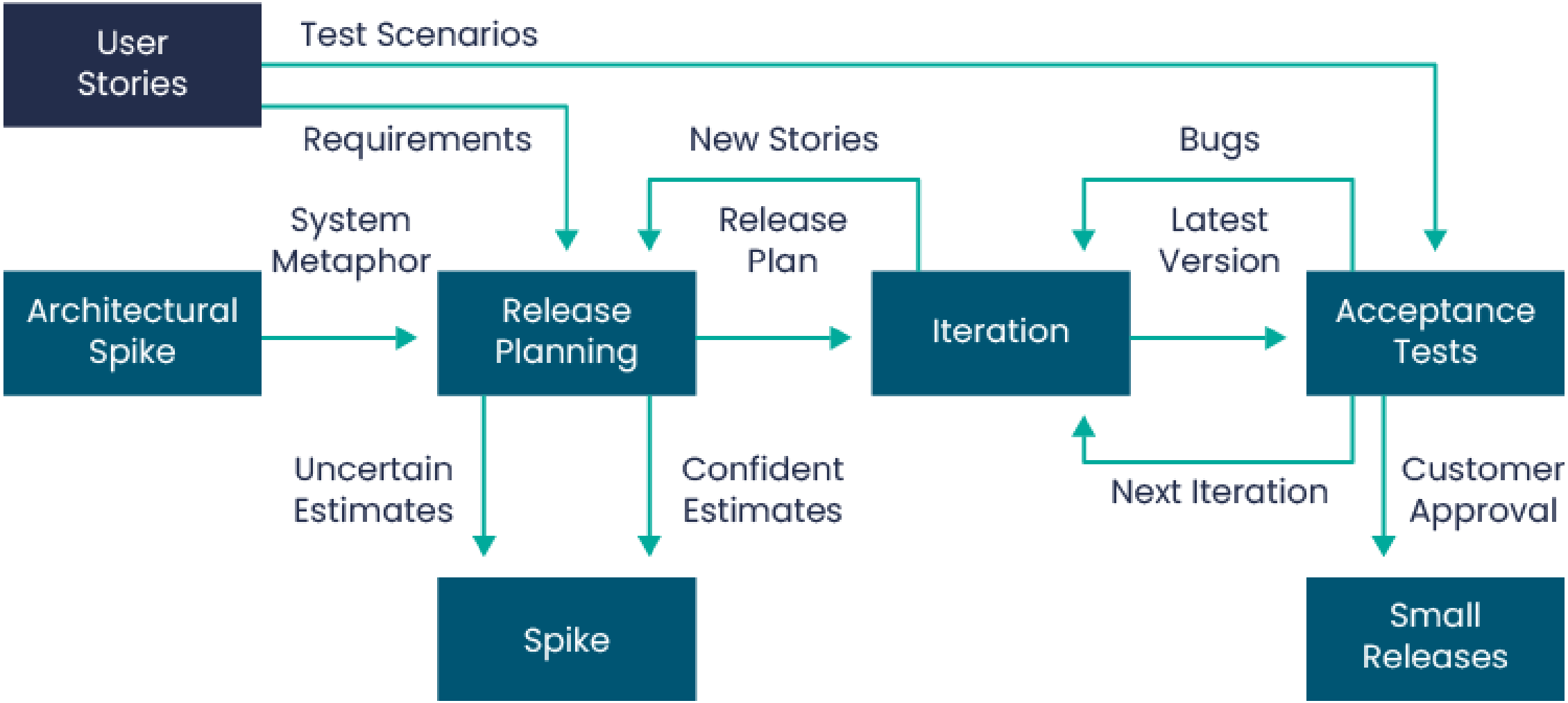
Kanban ....

- Unique Features:
  - Visual Board: A board with columns representing different stages of the workflow (e.g., To Do, In Progress, Done).
  - Work-in-Progress (WIP) Limits: Limits on the number of tasks that can be in each stage to prevent bottlenecks.
  - Continuous Delivery: Focuses on delivering small, incremental changes continuously.
- Contribution to Agile:
  - Kanban enhances visibility and transparency of the workflow.
  - It helps teams manage flow and improve processes incrementally.
  - Suitable for teams that need flexibility and continuous delivery without fixed iterations.

Extreme Programming (XP)

- Introduction: Extreme Programming (XP) is an Agile methodology that emphasizes technical excellence and customer satisfaction.

Extreme Programming (XP) Methodology

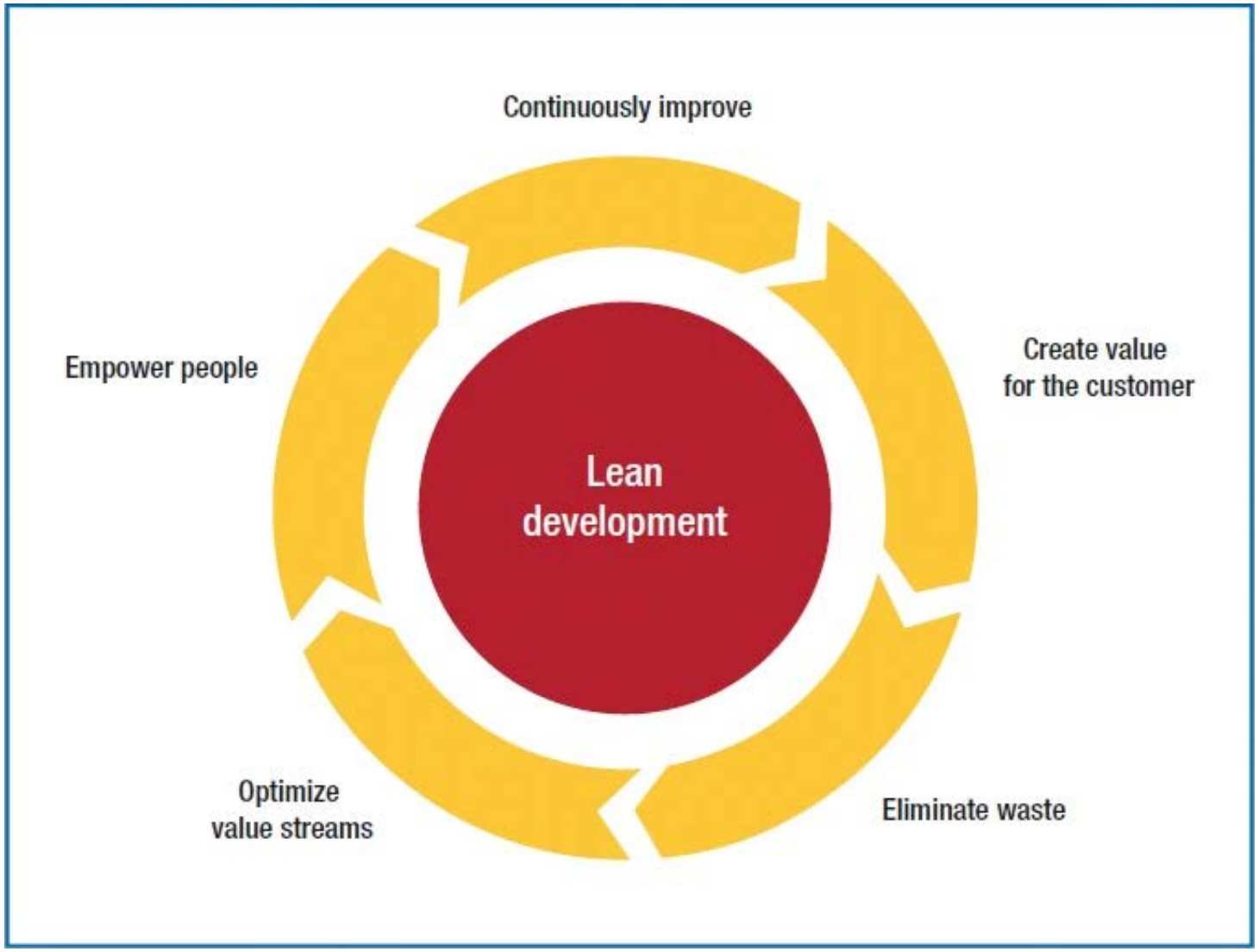


XP ...

- **Unique Features**
  - Pair Programming: Two developers work together at one workstation, improving code quality and knowledge sharing.
  - Test-Driven Development (TDD): Writing tests before code to ensure functionality and reduce bugs.
  - Continuous Integration: Frequently integrating code changes into a shared repository to detect issues early.
  - Refactoring: Continuously improving the codebase without changing its functionality.
- **Contribution to Agile**
  - XP promotes high-quality code and rapid feedback through practices like TDD and continuous integration.
  - It encourages close collaboration with customers to ensure the product meets their needs.

Lean Development

- Introduction: Lean Development is an Agile methodology inspired by Lean manufacturing principles, focusing on delivering value and eliminating waste.



Lean ...

- Unique Features:
  - Value Stream Mapping: Identifying and optimizing the flow of value through the development process.
  - Eliminating Waste: Removing activities that do not add value to the customer.
  - Continuous Improvement: Regularly assessing and improving processes.
- Contribution to Agile:
  - Lean Development emphasizes efficiency and value delivery.
  - It encourages teams to focus on what matters most to the customer and continuously improve their processes.

Integrated Development Environments (IDEs)

- Visual Studio Code (VS Code):
  - Lightweight and powerful source code editor.
  - Supports extensions for various programming languages and frameworks.
  - Integrated terminal and Git support.
- IntelliJ IDEA:
  - Comprehensive IDE for Java and other languages.
  - Advanced code navigation, refactoring, and debugging tools.
  - Built-in support for version control systems.
- **Step-by-Step Guide to Setting Up a Development Environment:** SkillReactor[1]

Version Control Systems

- Git:
  - Distributed version control system for tracking changes in source code.
  - Allows multiple developers to work on a project simultaneously.
- GitHub:
  - Web-based platform for version control and collaboration.
  - Hosts Git repositories and provides tools for issue tracking, code review, and project management.

Package Managers

- npm (Node Package Manager):
  - Default package manager for Node.js.
  - Manages dependencies for JavaScript projects.
- yarn:
  - Alternative to npm with a focus on speed and reliability.
  - Manages project dependencies and ensures consistent installations.

Build Tools

- Webpack:
  - Module bundler for JavaScript applications.
  - Transforms and bundles resources like JavaScript, CSS, and HTML.
- Gulp:
  - Task runner for automating repetitive tasks in development.
  - Uses a code-over-configuration approach for defining tasks.