AGILE AND DEVOPS

Understanding Modern Software Development Methodologies

Presented By:

Bishal Chapagain

OVERVIEW

- SOFTWARE DEVELOPMENT METHODOLOGIES
- WATERFALL METHODOLOGY & LIMITATIONS
- AGILE METHODOLOGY
- INTRODUCING DEVOPS
- AGILE AND DEVOPS
- CONCLUSION

SOFTWARE DEVELOPMENT METHODOLOGY

- A structured approach that guide how teams plan, build, test, and deliver software products.
- Address specific challenges in software development.

Evolution of Software Development Methodologies

1970s - 1990s
Waterfall
Linear, sequential approach

2000s
Agile
Iterative, collaborative approach

2010s - Present
DevOps
Integrated, automated approach

WATERFALL METHODOLOGY

1

Step-by-step linear development approach

2

Each phase must be completed before the next begins

3

Limited flexibility and late-stage feedback issues

LIMITATIONS

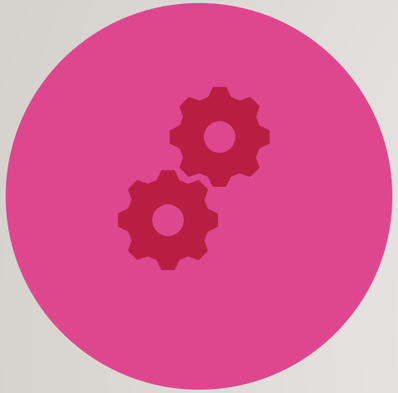
Limited Client Involvement

End users typically only see the product after completion

Changes late in the process are costly and difficult to implement

Not suitable for long-running projects with evolving requirements

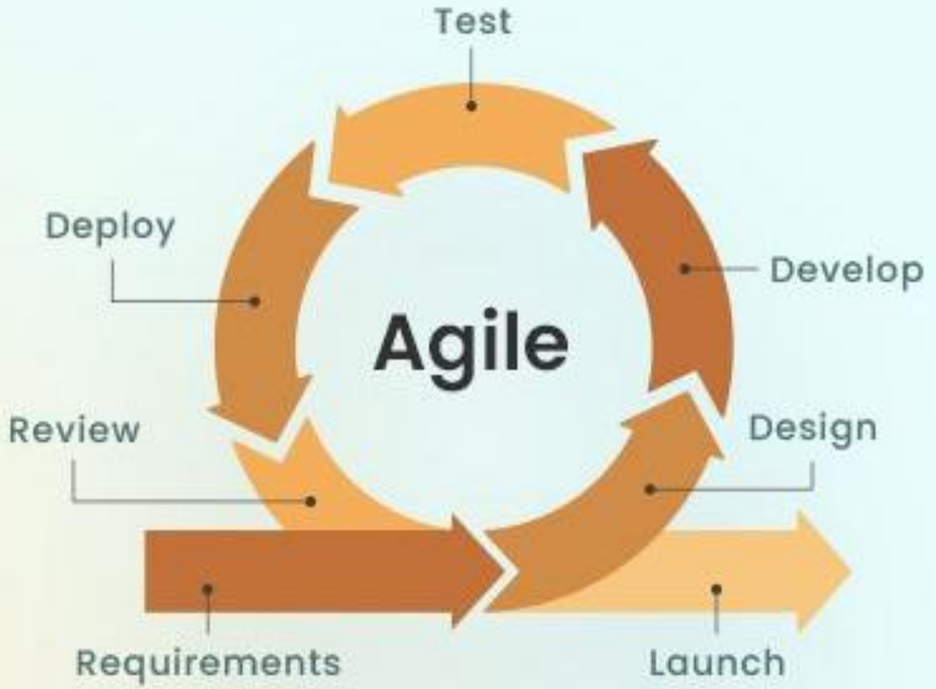
THE AGILE REVOLUTION



Iterative approach to
software development

Delivers software in small,
incremental cycles

Waterfall vs Agile:



AGILE : CORE FOCUS

A response to the limitations of waterfall model

Customer feedback

Adaptability

Continuous improvement

Collaboration

AGILE MANIFESTO : CORE VALUES

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 AGILE COMPONENTS

- User Story: A description of a feature from the end user's perspective.
- Sprint: A short, time-boxed period (1-4 weeks) for completing work.
- Daily Stand-up: A brief daily meeting for a team to discuss about work progress.
- Kanban Board: A visual tool for managing workflow.
- Product Backlog: A prioritized list of features and requirements.

SCRUM

The most popular application of agile methodology.

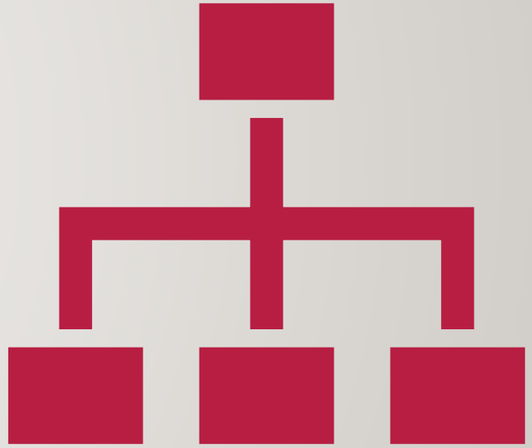
87% of teams who use agile use scrum.

Focuses on breaking tasks down and completing them in sprints.

SCRUM : TEAM ROLES

Scrum Master

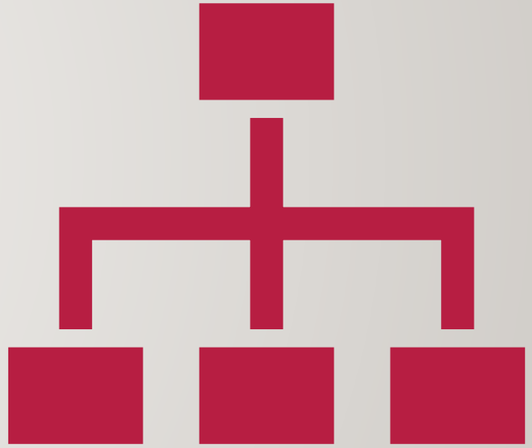
- Facilitates the Scrum process.
- Ensures rules and rituals are followed.
- Coaches the team.



SCRUM : TEAM ROLES

Product Owner

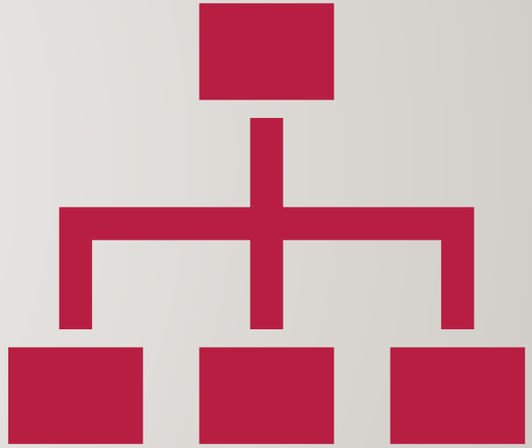
- Represents the voice of the customer.
- Defines and prioritizes the product backlog



SCRUM : TEAM ROLES

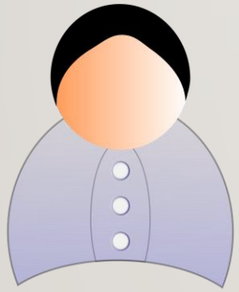
Development Team:

- Cross-functional team that delivers the product.
- Responsible for the actual work of creating the product.
- Includes developers, testers, designers, etc.



DEVOPS: BRIDGING THE GAP

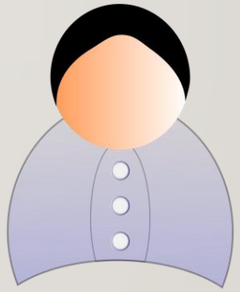
- The business requires change.
- Change is the root cause of most outages.



Dev

versus

Ops



It's not my code, it's your machine.

It's your code, not my machine.

PRINCIPLE OF DEVOPS: THE THREE WAYS

1. Flow

   Optimize the flow of work from development to operations to the customer

2. Feedback

   Shorten and amplify feedback loops to enable faster correction of problems.

3. Continual Learning and Experimentation

   Foster a culture of experimentation, learning from failure, and continuous improvement.

KEY DEVOPS PRACTICES

Continuous Integration (CI): Automating the integration of code changes
Continuous Delivery (CD): Automating the release of software

Automation: Automating manual tasks in the delivery pipeline.

Infrastructure as Code (IaC): Managing infrastructure through code.

Monitoring and Logging: Gaining visibility into system performance and behavior.

DEVOPS VS AGILE

DevOps is based on the ideas of Agile.

Agile focuses primarily on development.

DevOps extends Agile to the entire delivery lifecycle.

DevOps automates and optimizes the delivery pipeline to enable Agile's speed and flexibility

DEVOPS AND AGILE MYTHS

DevOps is just automation.

DevOps replaces Agile.

DevOps is only for startups.

DevOps is about culture, mindset, and collaboration.

DevOps extends and enhances Agile.

DevOps can be adopted by organizations of all sizes.

CHALLENGES IN ADOPTION

- Organizational culture and resistance to change.
- Lack of automation skills and tools.
- Siloed teams and communication gaps.
- Lack of leadership buy-in and support.

ANY QUESTIONS ?