**DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY**

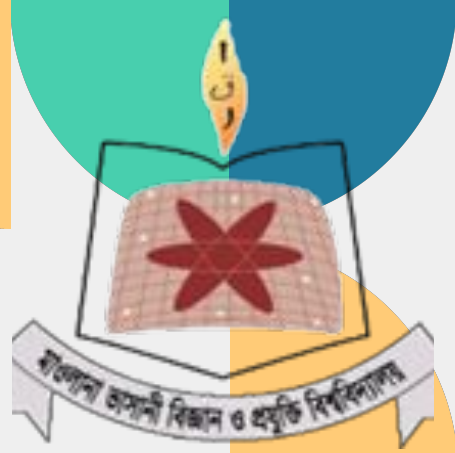**COURSE: SOFTWARE ENGINEERING**

# EXTREME PROGRAMMING

Presented By

S M Nasimul Hasan
IT-21026

Presented To

Ziaur Rahman
Associate Professor

# Objectives

Extreme Programming (XP)

Core Values of XP

Key Practices

Benefits of XP

Conclusion

# Introduction

What is Extreme Programming?

- Extreme Programming (XP) is an agile software development methodology.

- It was created by Kent Beck in 1998.

- XP emphasizes customer satisfaction, teamwork, and continuous improvement.

**Communication**: Effective communication among team members and stakeholders.

**Simplicity**: Keeping the design and code as simple as possible.

**Feedback**: Regular feedback from customers and team members.

**Courage**: Courage to make changes and take risks.

**Respect**: Mutual respect among team members.

# Core Values of XP

# Key Practices

**Test-Driven Development (TDD)**: Writing tests before writing code.

**Pair Programming**: Two developers work together at one workstation.

**Continuous Integration**: Integrating code changes frequently.

**Refactoring**: Improving the design of existing code without changing its functionality.

**Small Releases**: Releasing small, incremental updates to the software.

# Test-Driven Development (TDD)

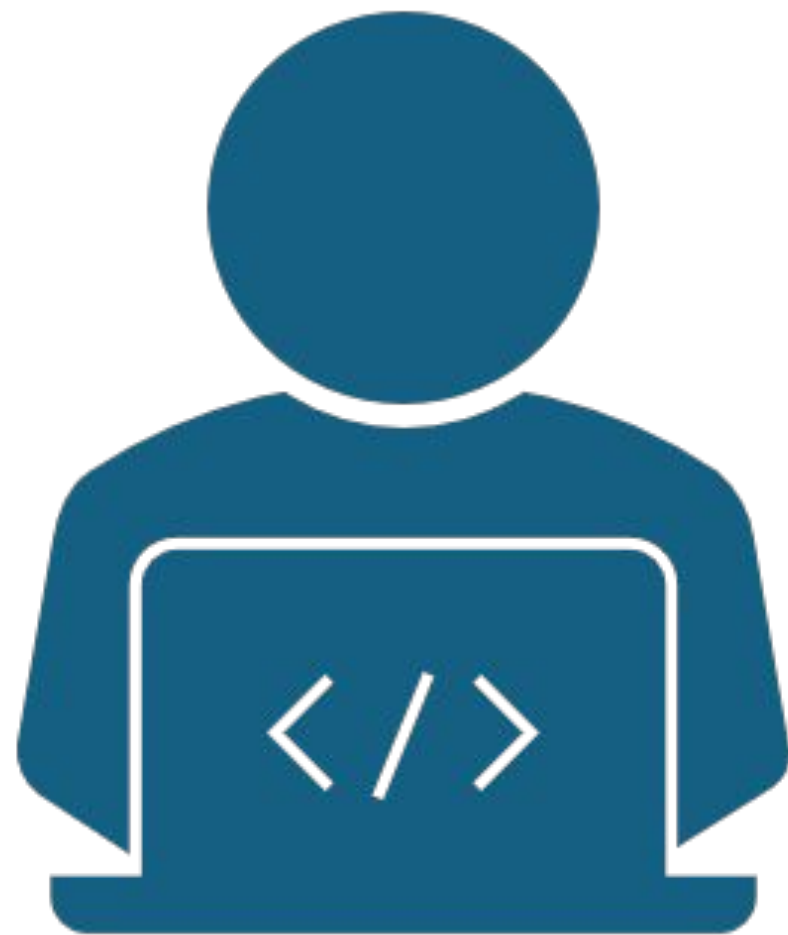 Write a test for a new feature before writing the code.

 Run the test and see it fail.

 Write the code to make the test pass.

 Refactor the code while ensuring the test still passes.

# Pair Programming

Two developers work together on the same code.

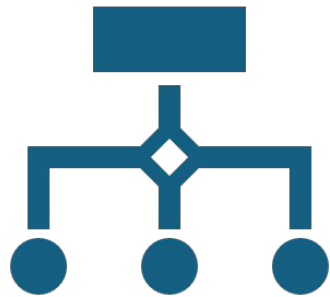One writes the code (Driver) while the other reviews it (Navigator).

Roles are switched frequently.

Benefits: Improved code quality, knowledge sharing, and reduced errors.

# Continuous Integration



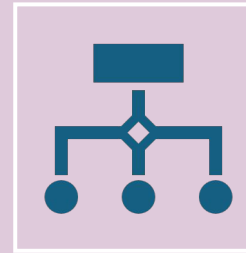Integrate code changes into the main branch frequently.



Automated tests are run to ensure the new code does not break existing functionality.



Benefits: Early detection of integration issues, faster feedback, and improved collaboration.

# Refactoring

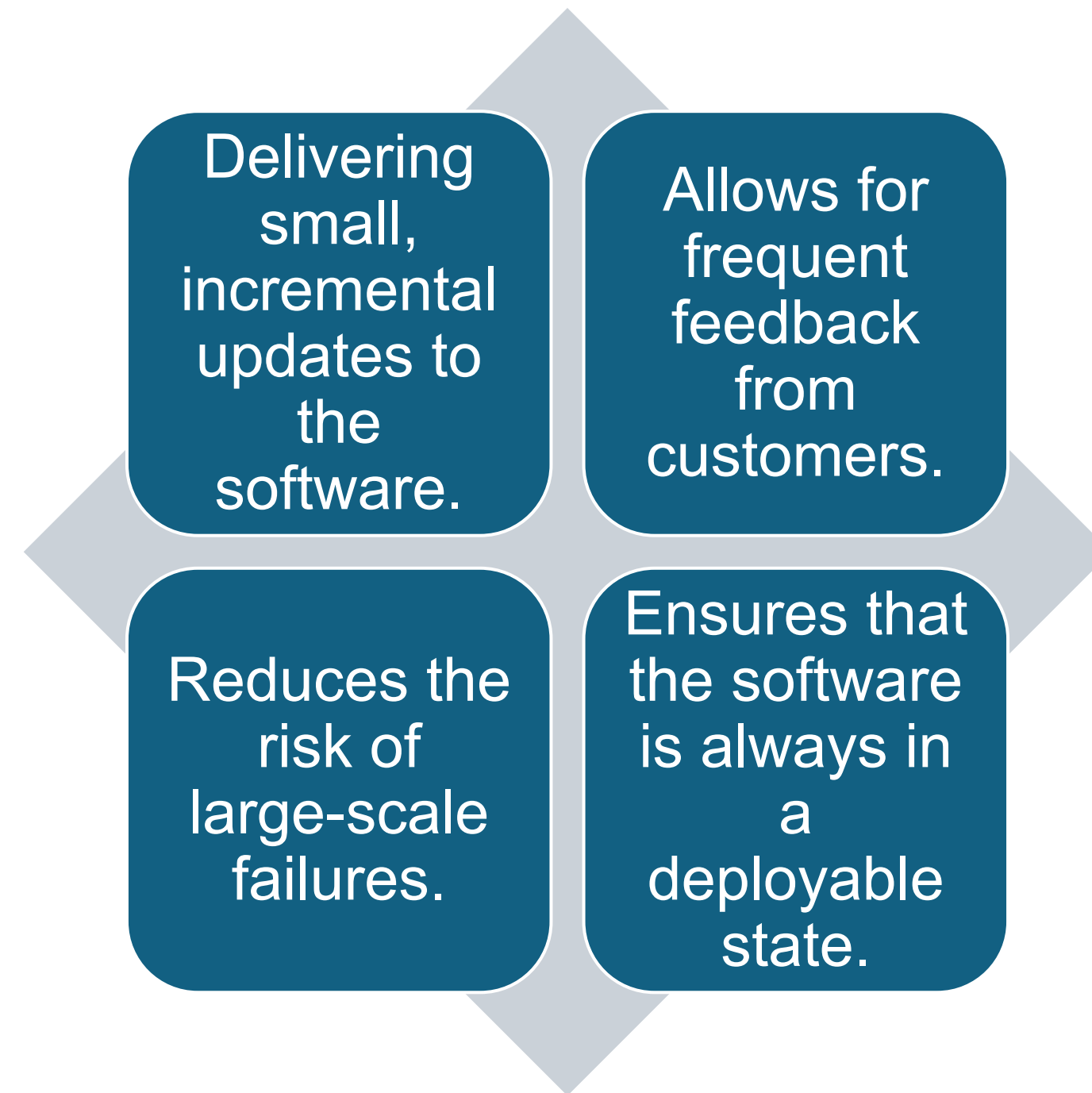The process of improving the structure and readability of code without changing its behavior.

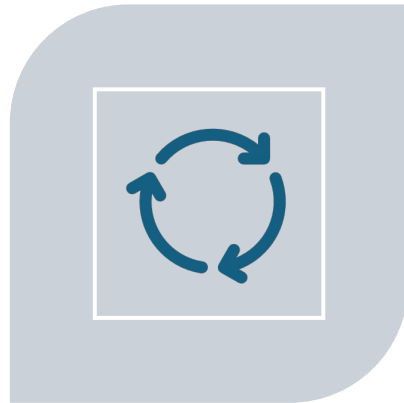Helps in maintaining clean and efficient code.

Regular refactoring prevents code decay and reduces technical debt.

# Small Releases

Delivering small, incremental updates to the software.

Allows for frequent feedback from customers.

Reduces the risk of large-scale failures.

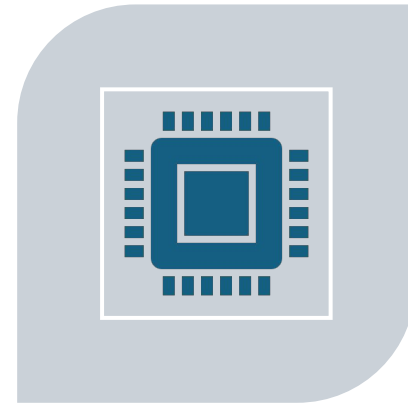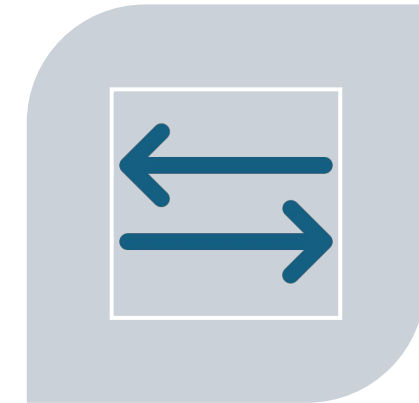Ensures that the software is always in a deployable state.

# Benefits of XP

HIGH CUSTOMER SATISFACTION DUE TO FREQUENT RELEASES AND FEEDBACK.

IMPROVED CODE QUALITY THROUGH PRACTICES LIKE TDD AND PAIR PROGRAMMING.

ENHANCED TEAM COLLABORATION AND COMMUNICATION.

FLEXIBILITY TO ADAPT TO CHANGING REQUIREMENTS.

# Conclusion

Extreme Programming (XP) is a powerful agile methodology that focuses on customer satisfaction, teamwork, and continuous improvement.

By following XP practices, teams can deliver high-quality software quickly and efficiently.

Thank You