Mawlana Bhashani Science and
Technology University

PRESENTATION

on

Extreme Programming(XP)

Presented by: Sanjida Akter Sampa
IT-20032(R/A)

Submitted by

Sanjida Akter Sampa

ID:IT-20032(R/A)

3rd year 2nd semester

Submitted To

Mr. Ziaur Rahman

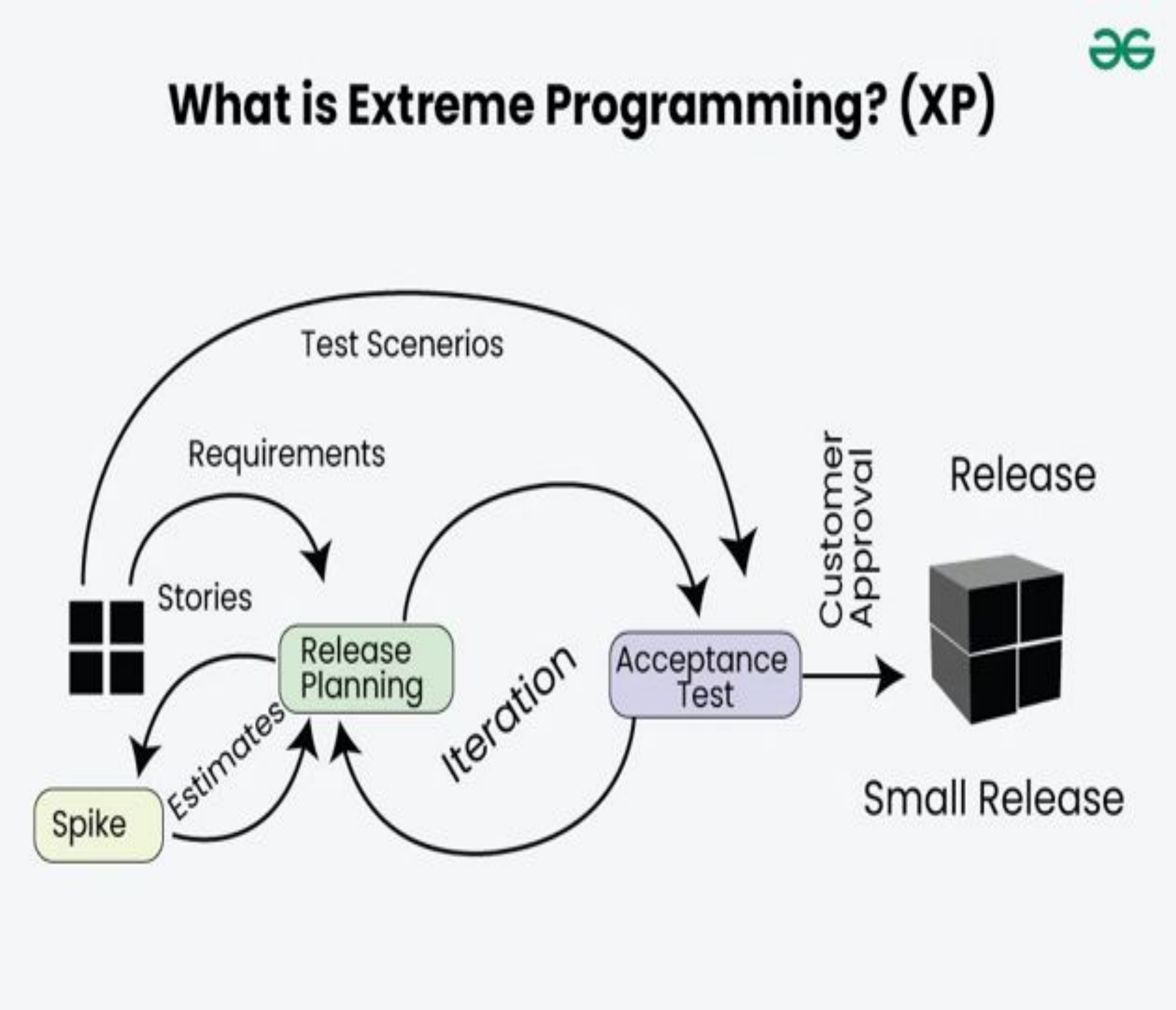
Assistant Professor

Dept. of ICT

Mawlana Bhashani Science and Technology University

EXTREME PROGRAMMING

- is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.
- XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop software.
- is one of the Agile software development methodologies. It provides values and principles to guide the team behavior.



XP Variables

- ► Cost

The amount of money to be spent. The resources (how many developers, equipment etc.) available for the project are directly related to this variable.

- ► Time

Determines when the system (release) should be done.

- ► Quality

The correctness of the system (as defined by the customer) and how well tested it will be.

- ► Scope

Describes what and how much will be done (functionality).

Change-driven process

► A **risk** is some "variable" that means a danger to the success of the project. Typical risks are for instance

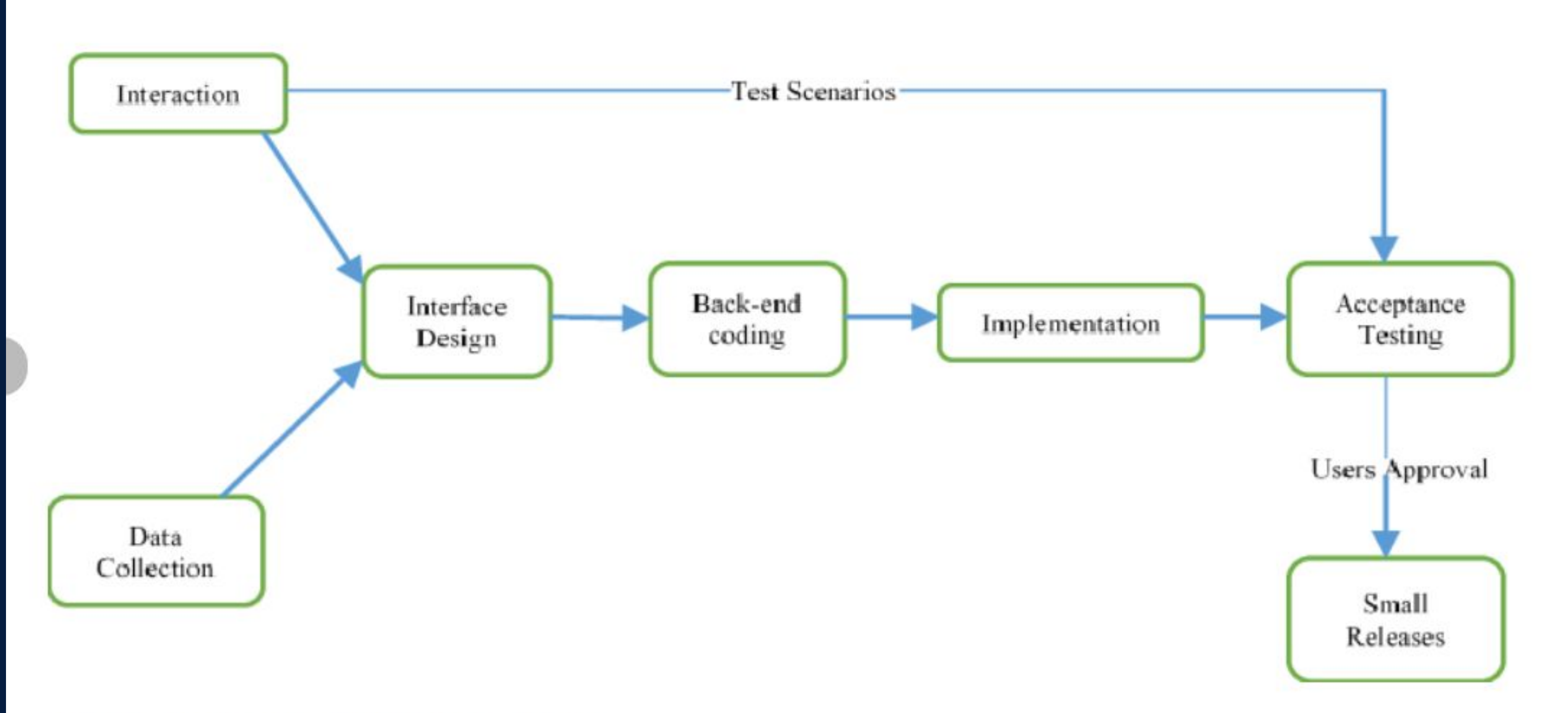
–Requirements change.

- The users needs change
- The problem changes
- The market changes
- The requirements are imprecise or not fully understood

► XP explicitly deals with the first risk: "Embrace Change".

XP Roles

- ▶ Developer (also called Programmer by some teams)
  - developer should know how to program.
- ▶ Customer
  - who should know what to program
- ▶ Tracker
  - The job of the tracker is to gather whatever metrics are being tracked for the project – at least the Load Factor and the Functional Test scores.
- ▶ Coach
  - is the responsibility of everyone in the team.
- ▶ Tester
  - is really focused on the customer to help write test   and runs them.
- ▶ Consultant
  - is to provide knowledge.



Extreme Programming (XP) Methodology

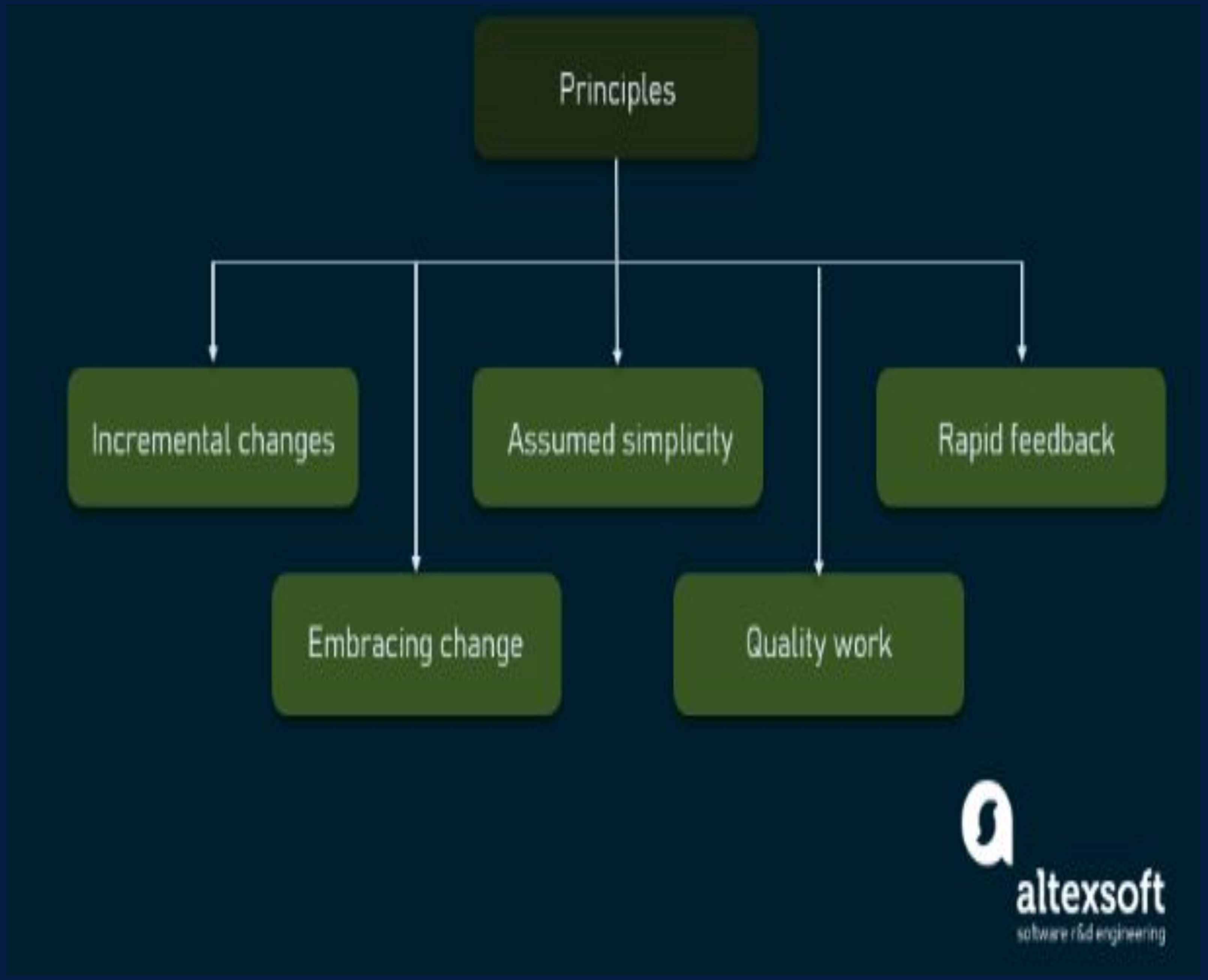
12 Practices of XP

- ▶ The Planning Game
  - The XP planning process. Business gets to specify what the system needs to. Development specifies how much each feature costs and what budget is available day/week/month.
- ▶ Small Releases
  - Every release should be as small as possible, containing the most valuable business requirements.
- ▶ Metaphor
  - is a simple evocative description of how the program works.
- ▶ Simple Design
  - The system should be designed as simply as possible at any given moment
- ▶ Testing
  - Programmers develop software by writing tests first, and then code that fulfills the requirements reflected in the tests. Customers provide acceptance tests that enable them to be certain that the features they need are provided.
- ▶ Refactoring
  - XP Team Refactor out any duplicate code generated in a coding session

- ▶ Pair Programming
  - All production code is written by two programmers sitting at one machine, with one keyboard and one mouse.
- ▶ Collective Ownership
  - The entire team takes responsibility for the whole of the system.
- ▶ Continuous Integration
  - Code is integrated and tested many times a day, one set of changes at a time.
- ▶ 40-Hour week
  - It emphasizes on the limited number of hours of work per week for every team member.
- ▶ On-site customer
  - Development team has continuous access to the customer who will actually be using the system.
- ▶ Coding Standards
  - Developers write all code in accordance with the rules emphasizing:
  - Communication through the code.
  - The least amount of work possible.

5 Principle of XP

- ▶ Rapid Feedback
  - is to get the feedback, understand it, and put the learning back into the system as quickly as possible.
- ▶ Assume Simplicity
  - to treat every problem as if it can be solved with simplicity.
- ▶ Incremental Change
  - solve problems with a series of small changes
- ▶ Embracing Change
  - The best strategy is the one that preserves the most options while actually solving your most pressing problem.
- ▶ Quality work
  - emphasize test-first programming



Benefits of XP

- Improved Code Quality
- Faster Adaptation to Changes
- Enhanced Collaboration
- Customer Satisfaction
- Reduced Project Risks
- Continuous Improvement
- Increased Developer Productivity
- Better Team Morale

Challenges in Extreme Programming (XP)

- High Communication Demands
- Requires Strong Team Discipline
- Intense Customer Involvement
- Not Applicable for all Teams
- Time-Consuming Practices
- Testing can be Significantly Overhead
- High Cost in Training and Transition

Thank You