AGILE-PARADIGM SHIFT IN SDLC

Session - 8

Extreme Programming

Objectives

- Describe the basic concepts of Extreme Programming
- Describe the common practices in Extreme Programming
- Describe the different phases in Extreme Programming
- Describe the documents and artifacts used in Extreme Programming
- Describe the important roles in an Extreme Programming team
- List the advantages and disadvantages of Extreme Programming

Introduction [1-4]

Extreme Programming (XP) is one of the many popular Agile processes. It concentrates on software development rather than software project management.

- XP emphasizes teamwork, customer, developer, manager, and coordinator as a team and work together as equal partners in project development.

Introduction [2-4]

- The primary practices of XP are:

Sit together	• Development team sits and operates from one single open space.
Cross-functional team	• Team consists of everyone who is essential for the success of the project.
Informative workspace	• Informative charts are placed around the workspace to display the status and progress of the project.
Energized team	• Working overtime is not encouraged.

Introduction [3-4]

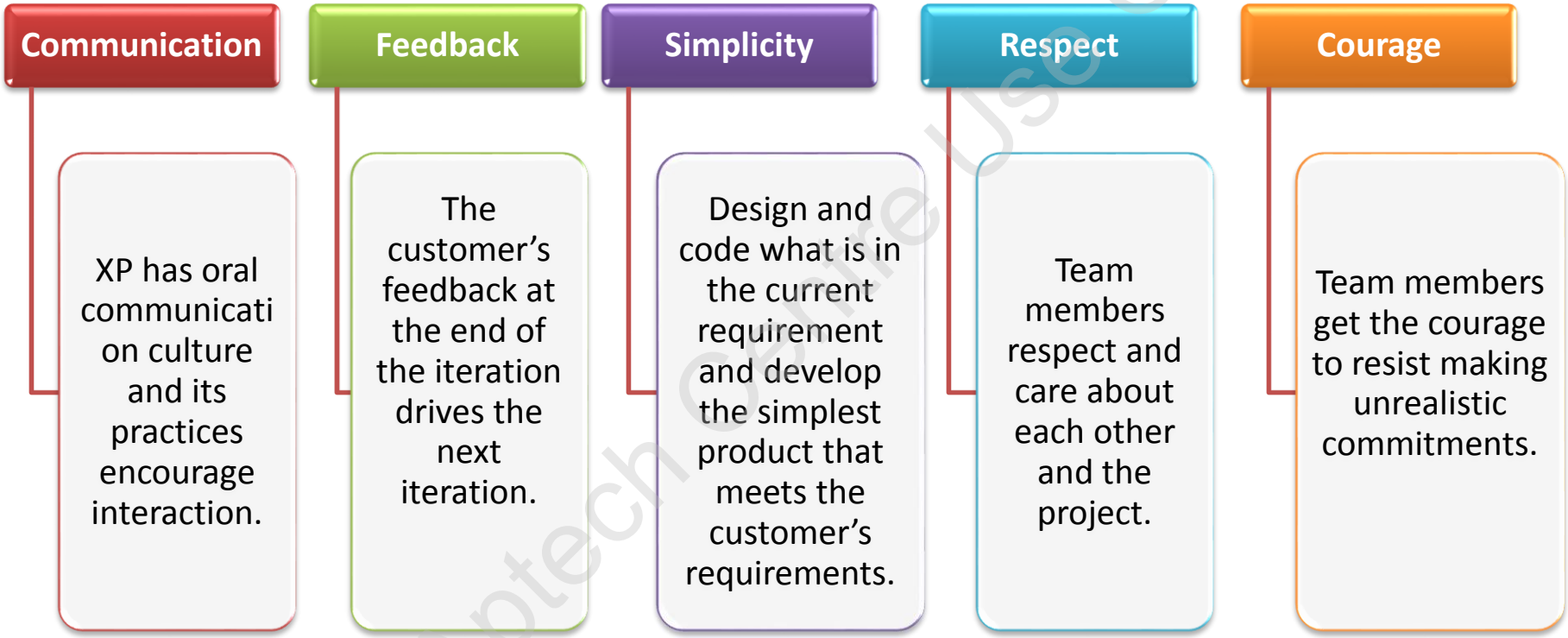
Pair programming	• Refers to the practice where two programmers work together, working on the same design of the production software.
User-stories	• Team writes short statements of customer-visible functionality required in the product.
Weekly cycle	• Weekly cycles are held to review the progress of the project. The feedback from them drives the next weekly cycle.
Quarterly cycle	• Team picks a theme for the quarter and the business value is delivered by the end of it.

Introduction [4-4]

Slack	• Lower priority tasks are planned so they can be dropped if the team lags in their work.
Ten-minute build	• Project structure is designed in order to allow frequent testing and tests are run within ten minutes.
Test-first programming	• The user story is considered fulfilled when it passes the acceptance test. Automated unit tests are written using Test-Driven Development (TDD).
Continuous integration	• Programmers check in to the completed code and the associated tests several times a day.
Incremental design	• System is designed incrementally rather than designing the whole system in advance.

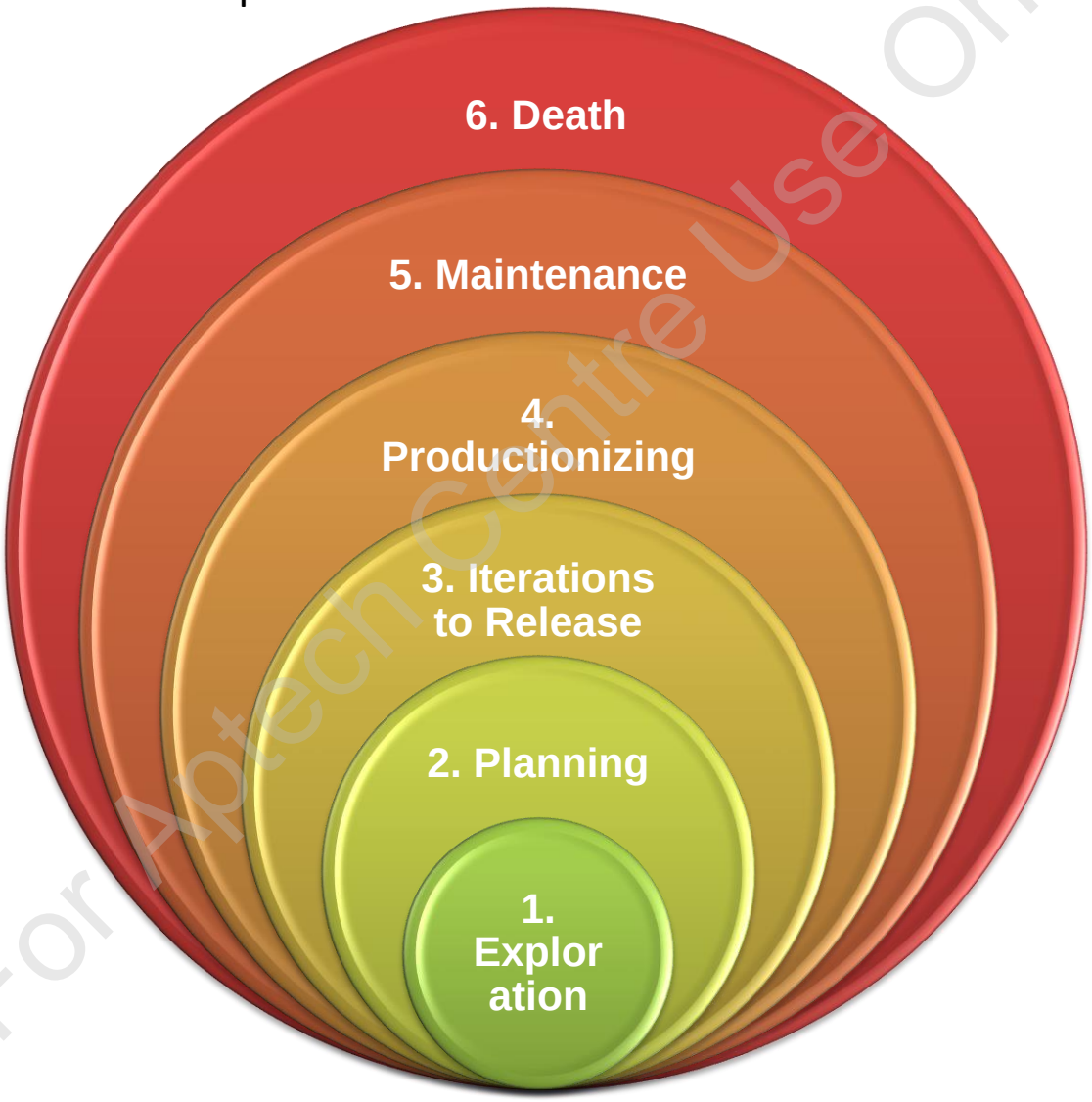
Values of XP

- XP methodology is based on:



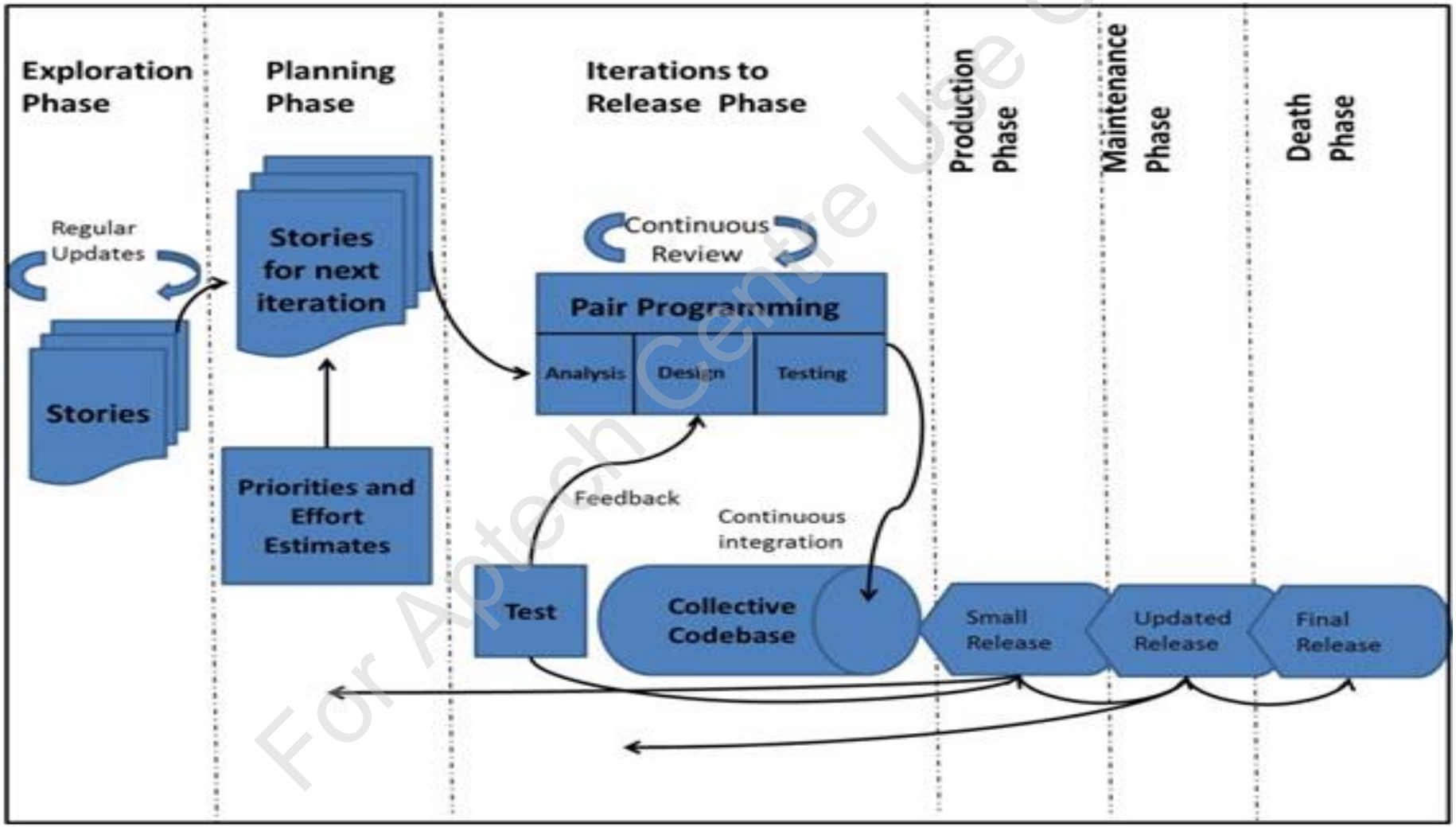
XP Process Activities [1-2]

- Life cycle of XP consists of phases.



XP Process Activities [2-2]

- Figure shows the different phases in XP.



XP Document and Artifacts

User story cards	• Contain short requirement descriptions and commitments for further conversation between the customer and the developer.
Task list	• Listing of the tasks for the user stories to be developed for an iteration.
CRC cards	• Responsibilities and collaborators of classes are identified during a design transformation session.
Customer acceptance tests	• Textual descriptions and automated test cases developed.
Visible wall graphs	• Depict the number of stories completed and the test cases passed.

Roles in an XP Team

- Manager
- Coach
- Tracker
- Programmer
- Tester
- Customer

Advantages of XP

- Robustness
- Resilience
- Cost Savings
- Lesser Risks
- Employee Satisfaction

Disadvantages of XP

- Difficult to apply in a practical scenario.
- Code centric and not design centric.
- Does not measure of plan the quality aspect of development.
- Large and complex software projects are hard to design using XP practices.
- Too much refactoring is a waste of time.
- Difficult for testers to find errors as XP programs are not well-structured.

Case Study [1-8]

- **WDSGlobal** is a leading global provider of knowledge-based services to mobile operators, manufacturers, application providers, and sales channels. In 2004, **WDSGlobal** brought together three independent development regions to form a development team that operates around the clock.

- This case study describes the challenges faced by the team in this environment, the lessons learned, and how issues such as global continuous integration, conflicting priorities, and cultural differences were resolved across regions.

Case Study [2-8]

Background

- **WDSGlobal** has ten years of experience in the mobile market place.
- All the efforts that are put in the development environment is dedicated to the refinement of **WDSGlobal's** mobile configuration platform, which is used on the Sony Ericsson, Nokia, Siemens, and Vodafone Websites.
- As the mobile industry innovates frequently, the demand for **WDSGlobal** services has reached a markable level, placing a heavy burden on the development teams.

Case Study [3-8]

One Global Team

- The company created three regions – the Americas, Asia, and Europe, in order to meet the requirements of its customers across the globe.
- Till the end of 2003, Web-based tools were developed and deployed using XP and Java, as Application Programming Interfaces (APIs) by the UK team.
- In each region, a non-XP team uses these internal APIs to deliver to localized requirements with the help of non-Java technologies.
- At the beginning of 2004, it was proposed to combine all the development teams into one global team which uses XP and Java.
- This global team was to share the same code base to minimize duplication and reduce maintenance costs.

Case Study [4-8]

Hurdles

- Time Zone
- Cultural Differences
- Technical Background

Case Study [5-8]

Solution to Problem

- The global team found it difficult to maintain a common process vision and values.
- Everyone in the team had to trust each other as equal team members in order to share the same code base and system.
- In a non-distributed XP team, when new ideas and technologies are introduced, the team comes to an agreement before the team proceeds.
- With the virtually combined, yet distributed team, it was tough to introduce new technologies or architectural changes.

Case Study [6-8]

Solution to Problem

- Some of the practices followed are:
  - Daily Handovers
  - Face-to-Face Communication
  - Shared Common Environment
  - Remote Pairing
  - Round-the-World Program
  - Shared Stories
  - Putting Out Fires
  - Coach-to-Coach-Level Communication
  - XP Principles and Practices

Case Study [7-8]

Outcomes of Adopted Solution

- Balancing the Teams
- Introducing Process Changes
- Introducing Innovation
- Allowing Process Flexibility
- Outside Forces
- Including the Business
- Incorporating QA

Case Study [8-8]

Conclusion

- After a year, the company considers the transition to the globally distributed XP team to be successful.
- The team has proved that XP works for a globally scattered group developing around-the-clock with a shared codebase.
- With distributed XP, the company enjoys the advantages of an Agile process: the developers have direct contact with regional customers.
- Business is able to accommodate changes in the local and global markets while producing and delivering quality products.

Summary

- XP concentrates on the development rather than managerial aspects of software projects. It stresses customer satisfaction and teamwork.
- In XP, every individual contributor to the project is an indispensable part of the whole team.
- XP is based on five values namely, Communication, Simplicity, Feedback, Courage, and Respect.
- The different phases in XP are namely, Exploration Phase, Planning Phase, Iterations to Release Phase, Productionizing Phase, Maintenance Phase, and Death Phase.
- The important documents and artifacts in XP include: User story cards, Task list, CRC cards, Customer acceptance tests, and Visible wall graphs.
- The important roles in an XP team are Manager, Coach, Programmer, Tester, and Customer.
- The advantage of XP processes are its robustness, resilience, cost savings, lesser risks, and employee satisfaction.