



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

COURSE NAME : 19CS302 AGILE SOFTWARE ENGINEERING

II YEAR /III SEMESTER

Unit 2- Agile Development

Topic 4: Extreme Programming





Brain Storming



1. Agile vs Traditional Development process.



What is Extreme Programming?



- XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.
- **eXtreme Programming (XP)** was conceived and developed to address the specific needs of software development by small teams in the face of vague and changing requirements.
- Extreme Programming is one of the Agile software development methodologies. It provides values and principles to guide the team behavior.



Conti...



- The team is expected to self-organize. Extreme Programming provides specific core practices where –
 - Each practice is simple and self-complete.
 - Combination of practices produces more complex and emergent behavior.



Conti...



Embrace Change:

- A key assumption of Extreme Programming is that the cost of changing a program can be held mostly constant over time.

This can be achieved with –

- Emphasis on continuous feedback from the customer
- Short iterations
- Design and redesign
- Coding and testing frequently
- Eliminating defects early, thus reducing costs
- Keeping the customer involved throughout the development
- Delivering working product to the customer



Extreme Programming in a Nutshell



Extreme Programming involves –

- Writing unit tests before programming and keeping all of the tests running at all times. The unit tests are automated and eliminates defects early, thus reducing the costs.
- Starting with a simple design just enough to code the features at hand and redesigning when required.
- Programming in pairs (called pair programming), with two programmers at one screen, taking turns to use the keyboard. While one of them is at the keyboard, the other constantly reviews and provides inputs.

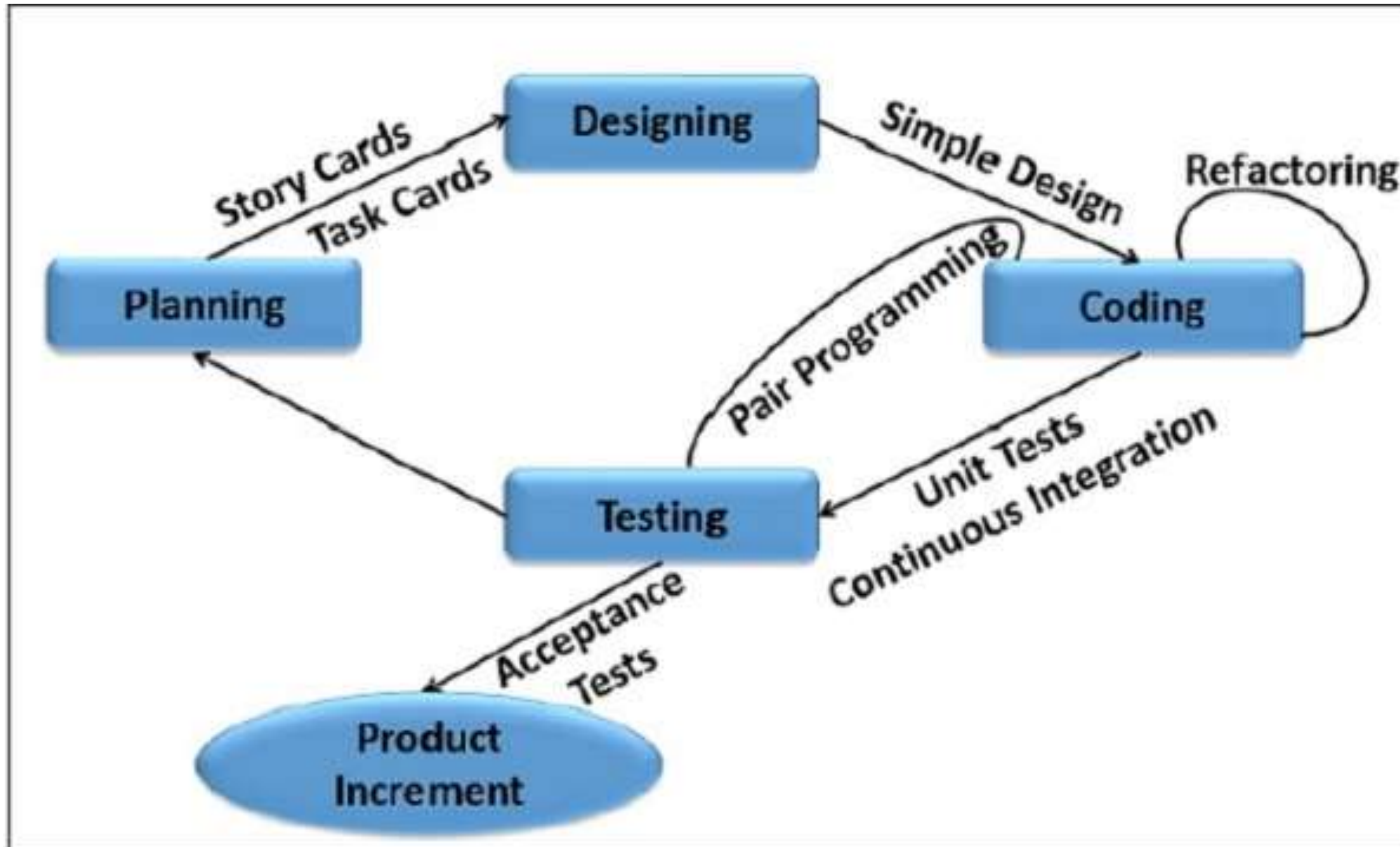


Extreme Programming in a Nutshell



- Integrating and testing the whole system several times a day.
- Putting a minimal working system into the production quickly and upgrading it whenever required.
- Keeping the customer involved all the time and obtaining constant feedback.

Iterating facilitates the accommodating changes as the software evolves with the changing requirements.



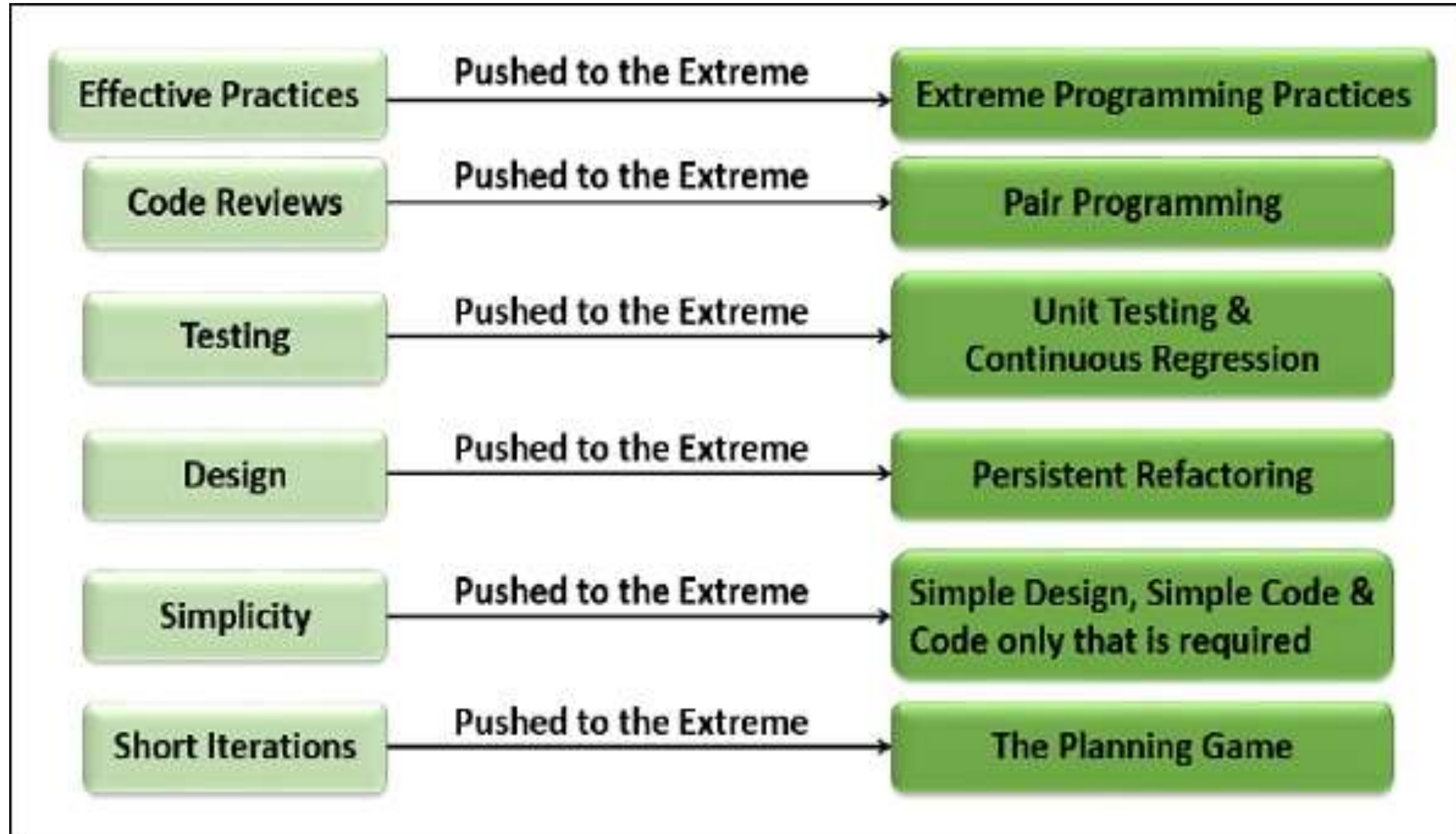


Why is it called “Extreme?”



Extreme Programming takes the effective principles and practices to extreme levels.

- Code reviews are effective as the code is reviewed all the time.
- Testing is effective as there is continuous regression and testing.
- Design is effective as everybody needs to do refactoring daily.
- Integration testing is important as integrate and test several times a day.
- Short iterations are effective as the planning game for release planning and iteration planning.





History of Extreme Programming



- Kent Beck, Ward Cunningham and Ron Jeffries formulated extreme Programming in 1999. The other contributors are Robert Martin and Martin Fowler.
- In Mid-80s, Kent Beck and Ward Cunningham initiated Pair Programming at Tektronix. In the 80s and 90s, Smalltalk Culture produced Refactoring, Continuous Integration, constant testing, and close customer involvement. This culture was later generalized to the other environments.
- In the Early 90s, Core Values were developed within the Patterns Community, Hillside Group. In 1995, Kent summarized these in Smalltalk Best Practices, and in 1996, Ward summarized it in episodes.



- In 1996, Kent added unit testing and metaphor at Hewitt. In 1996, Kent had taken the Chrysler C3 project, to which Ron Jeffries was added as a coach. The practices were refined on C3 and published on Wiki.
- Scrum practices were incorporated and adapted as the planning game. In 1999, Kent published his book, 'Extreme Programming Explained'. In the same year, Fowler published his book, Refactoring.
- Extreme Programming has been evolving since then, and the evolution continues through today.



Success in Industry



The success of projects, which follow Extreme Programming practices, is due to –

- Rapid development.
- Immediate responsiveness to the customer's changing requirements.
- Focus on low defect rates.
- System returning constant and consistent value to the customer.
- High customer satisfaction.
- Reduced costs.
- Team cohesion and employee satisfaction.



Extreme Programming Advantages



Extreme Programming solves the following problems often faced in the software development projects –

Slipped schedules – and achievable development cycles ensure timely deliveries.

Cancelled projects – Focus on continuous customer involvement ensures transparency with the customer and immediate resolution of any issues.

Costs incurred in changes – Extensive and ongoing testing makes sure the changes do not break the existing functionality. A running working system always ensures sufficient time for accommodating changes such that the current operations are not affected.



Production and post-delivery defects: Emphasis is on – the unit tests to detect and fix the defects early.

Misunderstanding the business and/or domain – Making the customer a part of the team ensures constant communication and clarifications.

Business changes – Changes are considered to be inevitable and are accommodated at any point of time.

Staff turnover – Intensive team collaboration ensures enthusiasm and good will. Cohesion of multi-disciplines fosters the team spirit.



Assessment 1



1. Compare and contrast Agile vs Traditional Software Development process.

Ans : _____





References



1. Roger S. Pressman, Software engineering- A practitioner's Approach, 10th Edition, McGraw-Hill, 2017.
2. Ken Schawber, Mike "Agile Software Development with Scrum" Pearson Education, 2nd Edition, 2015.

Thank You