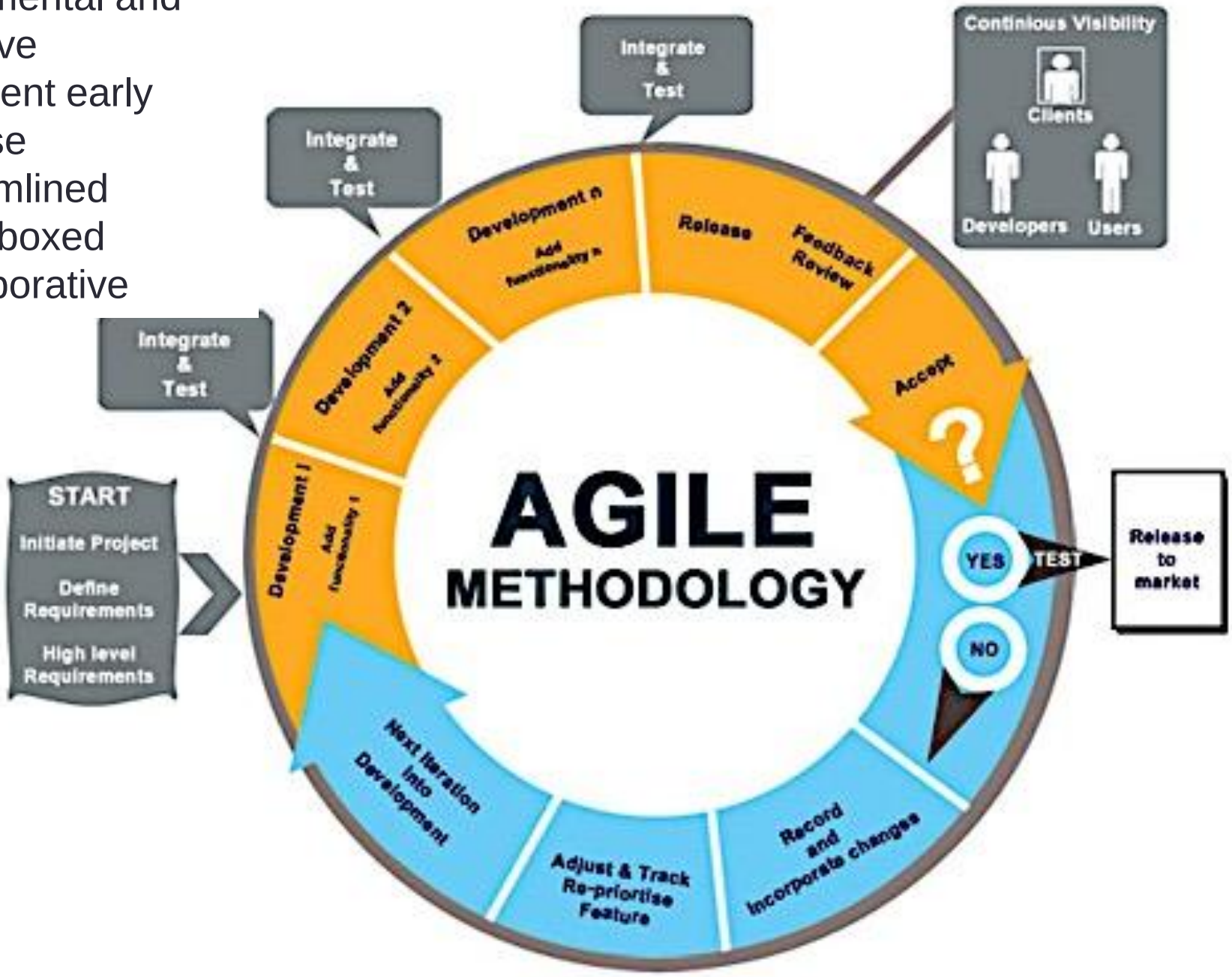
(11224) INTRODUCTION TO SOFTWARE ENGINEERING

Lecture 15: Extreme Programming

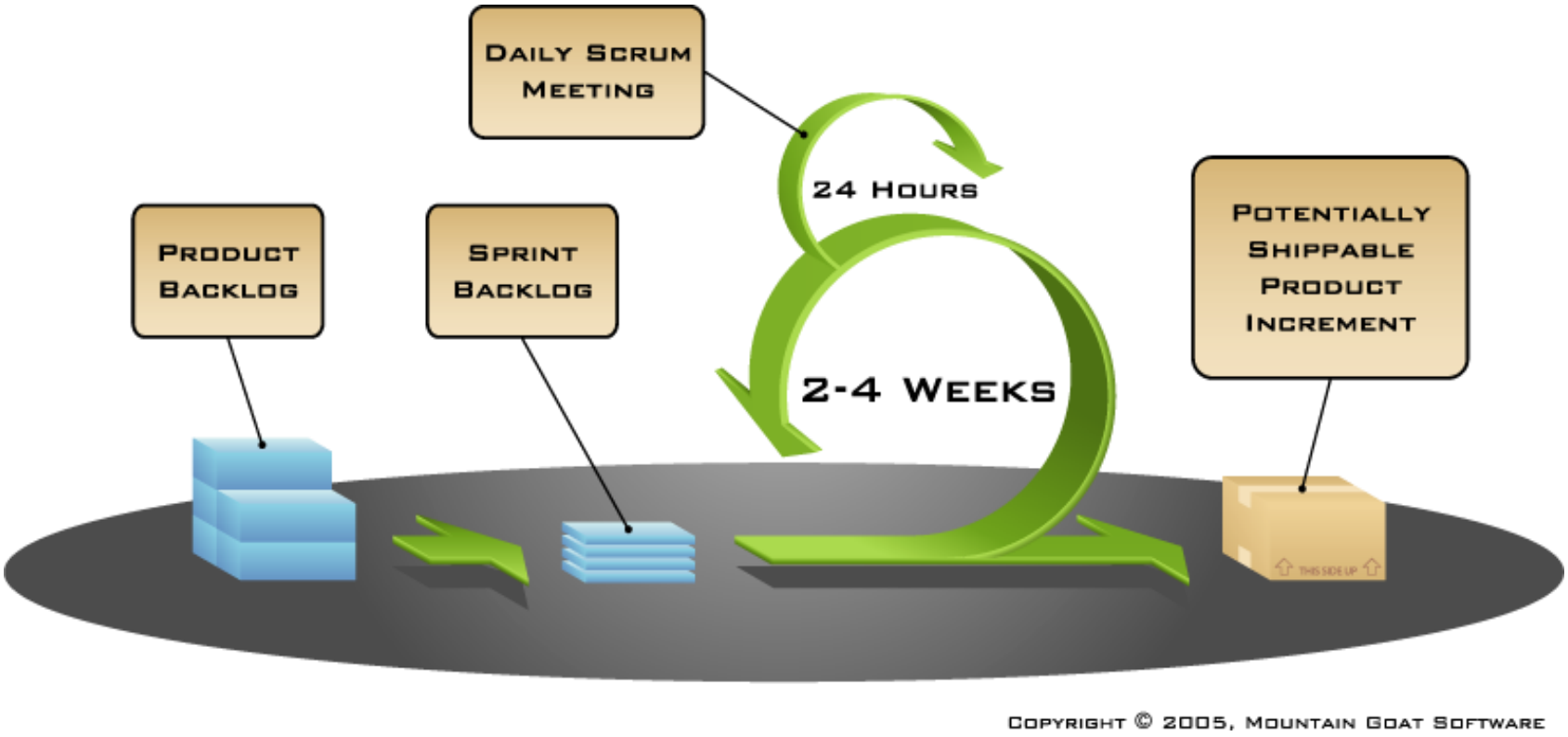
Shereen Fouad

- Incremental and Iterative
- Frequent early release
- Streamlined
- Time-boxed
- Collaborative



https://www.interaction-design.org/ux-daily/110/problems-associated-with-integrating-agile-methods-in-ucd-

How Scrum Works?



Backlog – prioritize list of tasks to be completed.

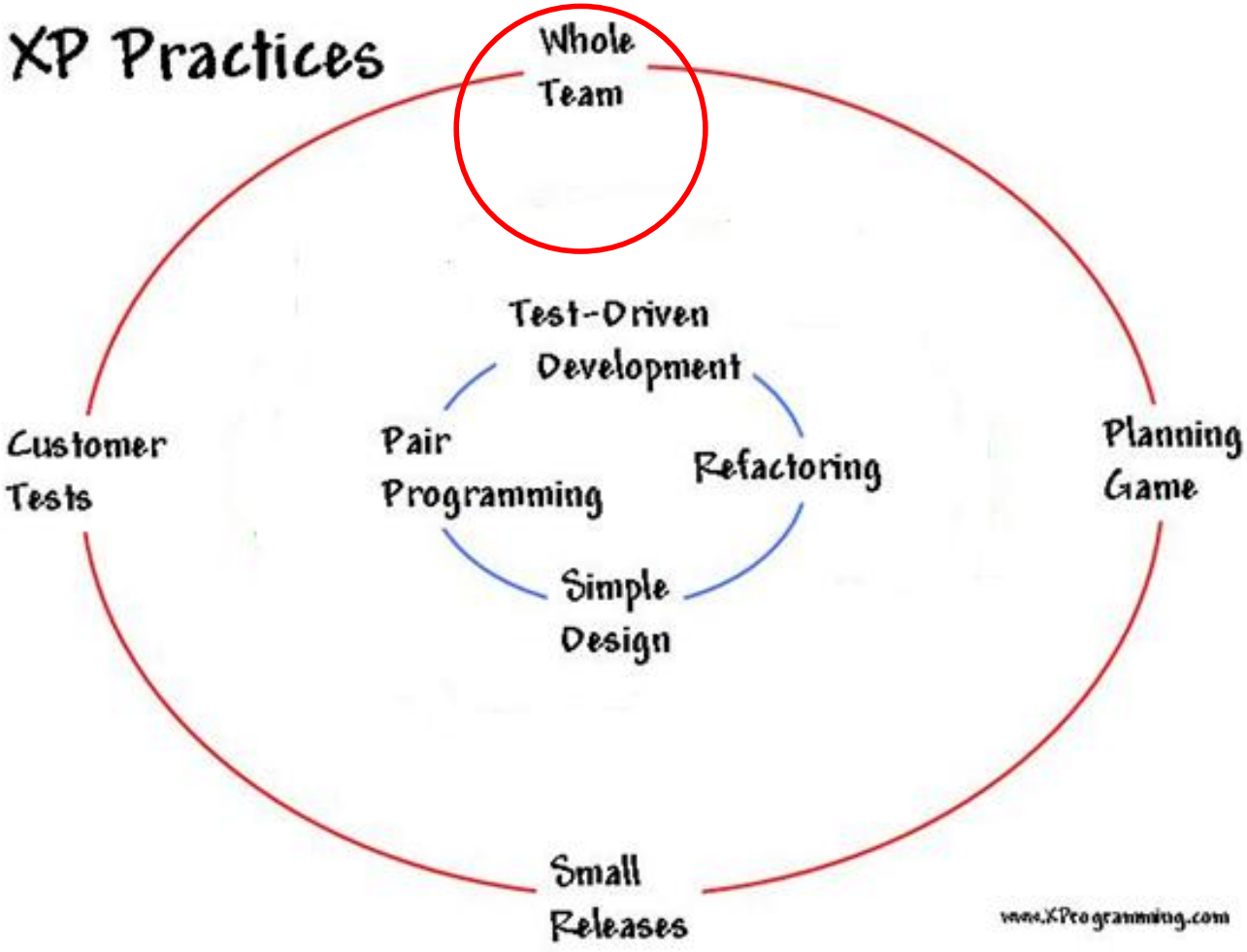
Scrum – Meeting to discuss sprint status. Attended by all and managed by ScrumMaster

Sprint – A few days of intense work to produce a product increment.

Extreme Programming (XP)

- The best known and most commonly employed agile process
- The process is cyclic, with a cycle usually taking 2 weeks.
- The planning and each iteration is based on User Stories.
- In many ways, user stories serve the same purpose as Use Cases in the Rational Unified Process, but they are different.

XP Practices

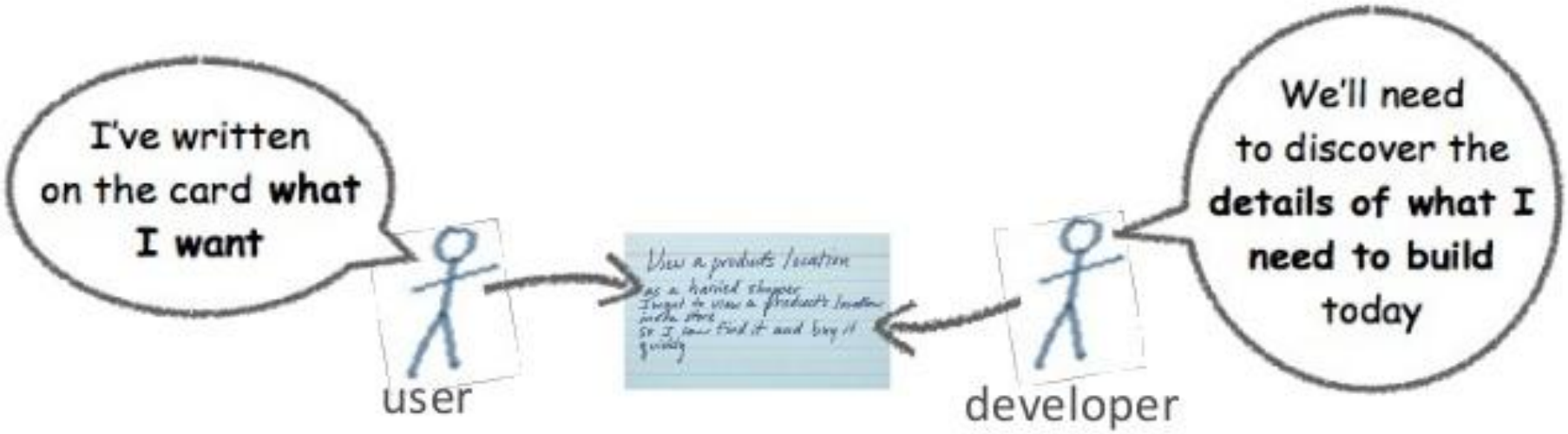


(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

XP Practices: Whole Team

- All contributors to an XP project are one team
- Must include a business representative--the ‘Customer’
  - Provides requirements
  - Sets priorities
  - Steers project
- Team members are programmers, testers, analysts, coach, manager

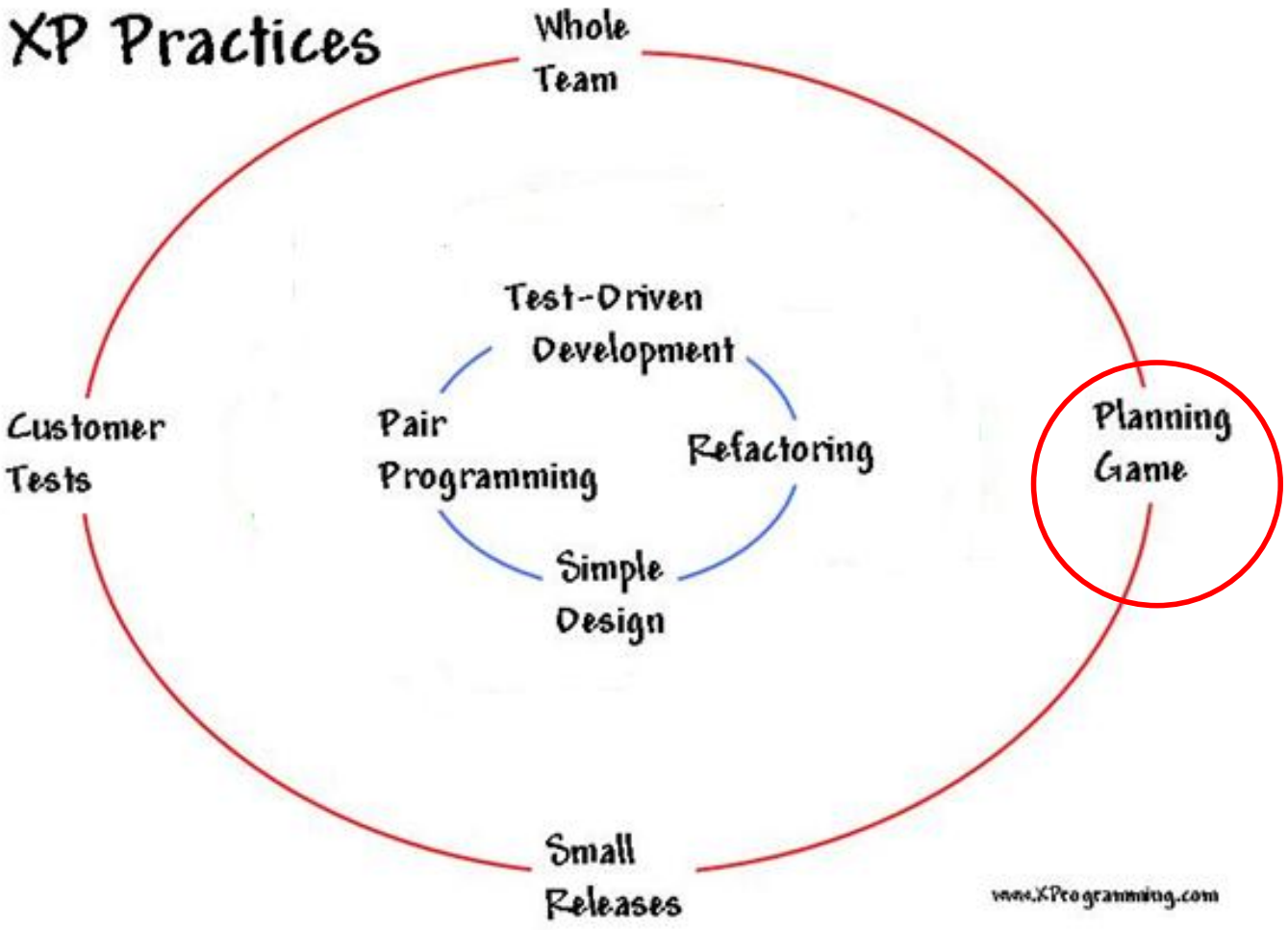
Stories facilitate a conversation with a goal of shared understanding



Collection of user stories

- NOT sufficient to serve as a functional requirements document.
- Form the basis for planning,
- Can be used to estimate how long it will take to implement that piece of functionality.
- Annotated with some indication of the importance of that piece of functionality (how urgent it is to implement it)

XP Practices



(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

XP Practices: Planning Game

- Two key questions in software development:
  - **Release Plan**
  - Predict what will be accomplished by the due date
  - **Iteration Plan**
  - Determine what to do next
- Need is to steer the project
- Exact prediction (which is difficult) is not necessary

XP Iteration Planning

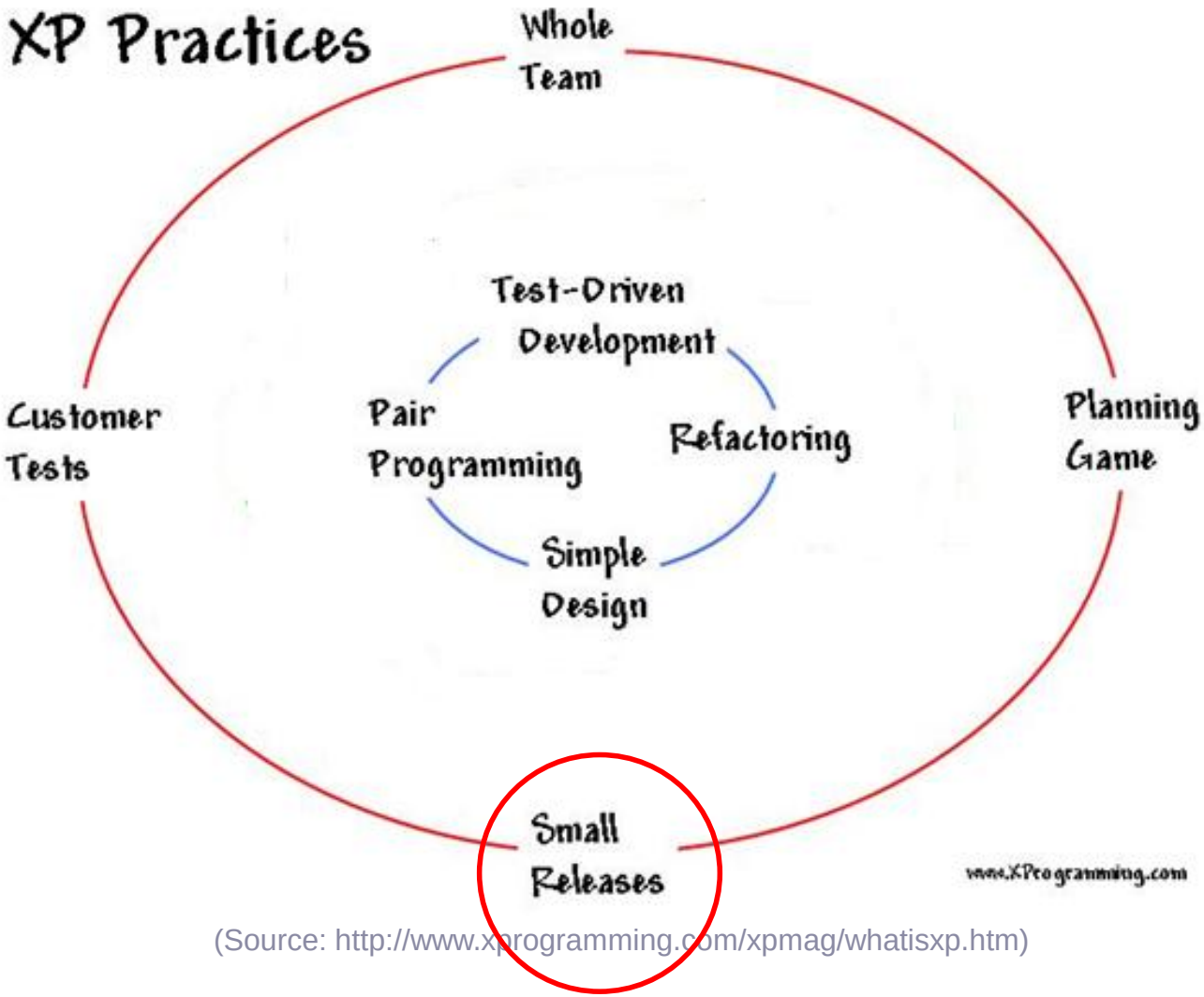
- Choose user stories for this cycle and failed acceptance tests from the previous release.
- Break them to tasks
- Tasks are written on index cards by programmers, NOT by customers
- Developers take on individual tasks and estimate how long to implement them.
- Test and integrate software
- Release the software
- Do user acceptance testing and evaluate

Responsibilities and Collaboration (CRC)

- Short
- Non-procedural descriptions of how classes fit into a system design.
- Make it possible for a whole team to think about and discuss the object oriented design of a system without getting mired in details of code.

Class Name: Student	
Superclasses: User	
Subclasses: Postgrad, Undergrad	
Responsibilities	Collaborators
ID number Get Mark transcript Get Attendance record	Marks, Modules AttendanceHistory

XP Practices

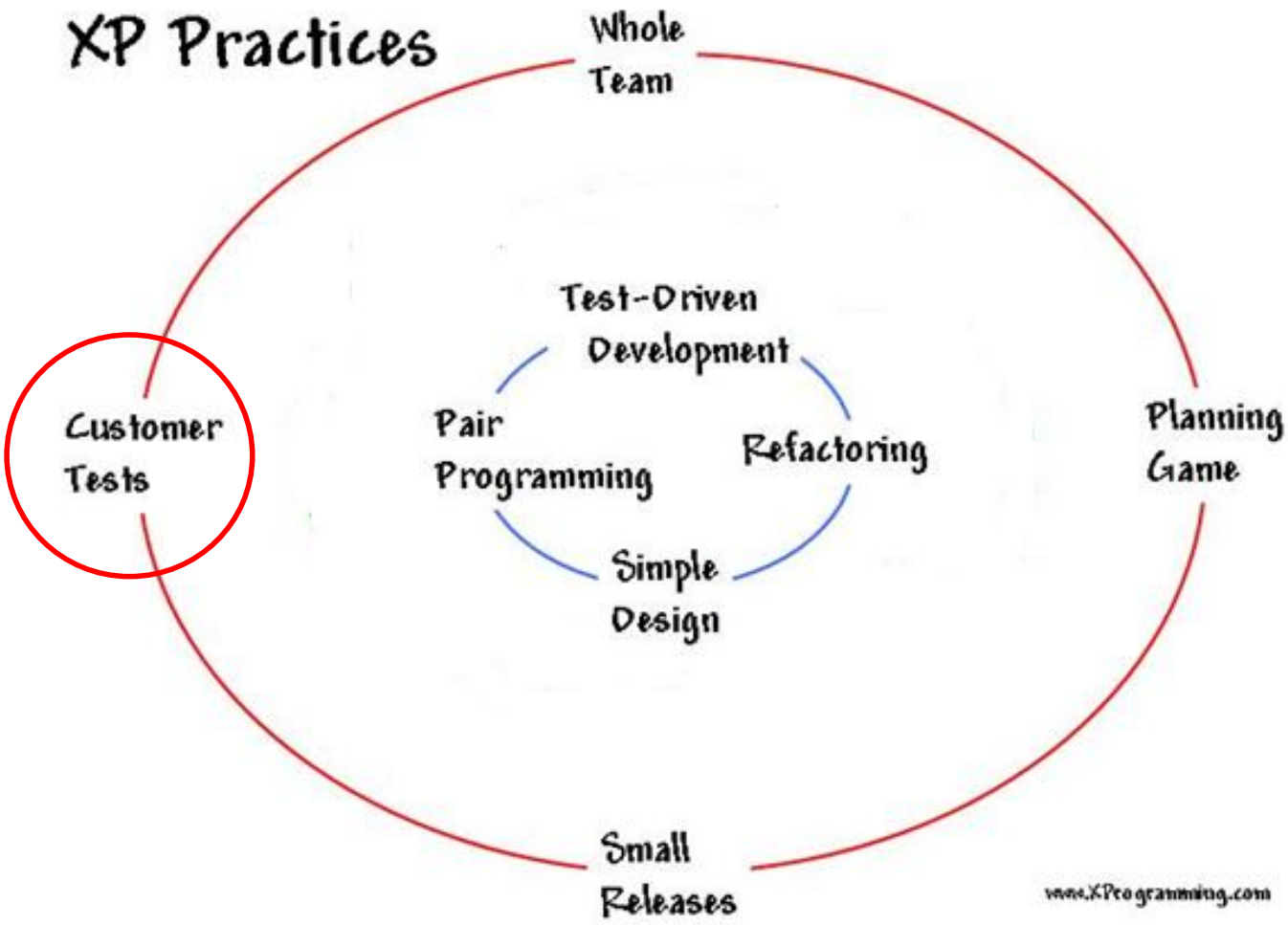


(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

XP Practices: Small Releases

- Team releases running, tested software every iteration
- Releases are small and functional
- The Customer can evaluate or in turn, release to end users, and provide feedback
- Important thing is that the software is visible and given to the Customer at the end of every iteration
- Release deadlines are never allowed to slip.

XP Practices



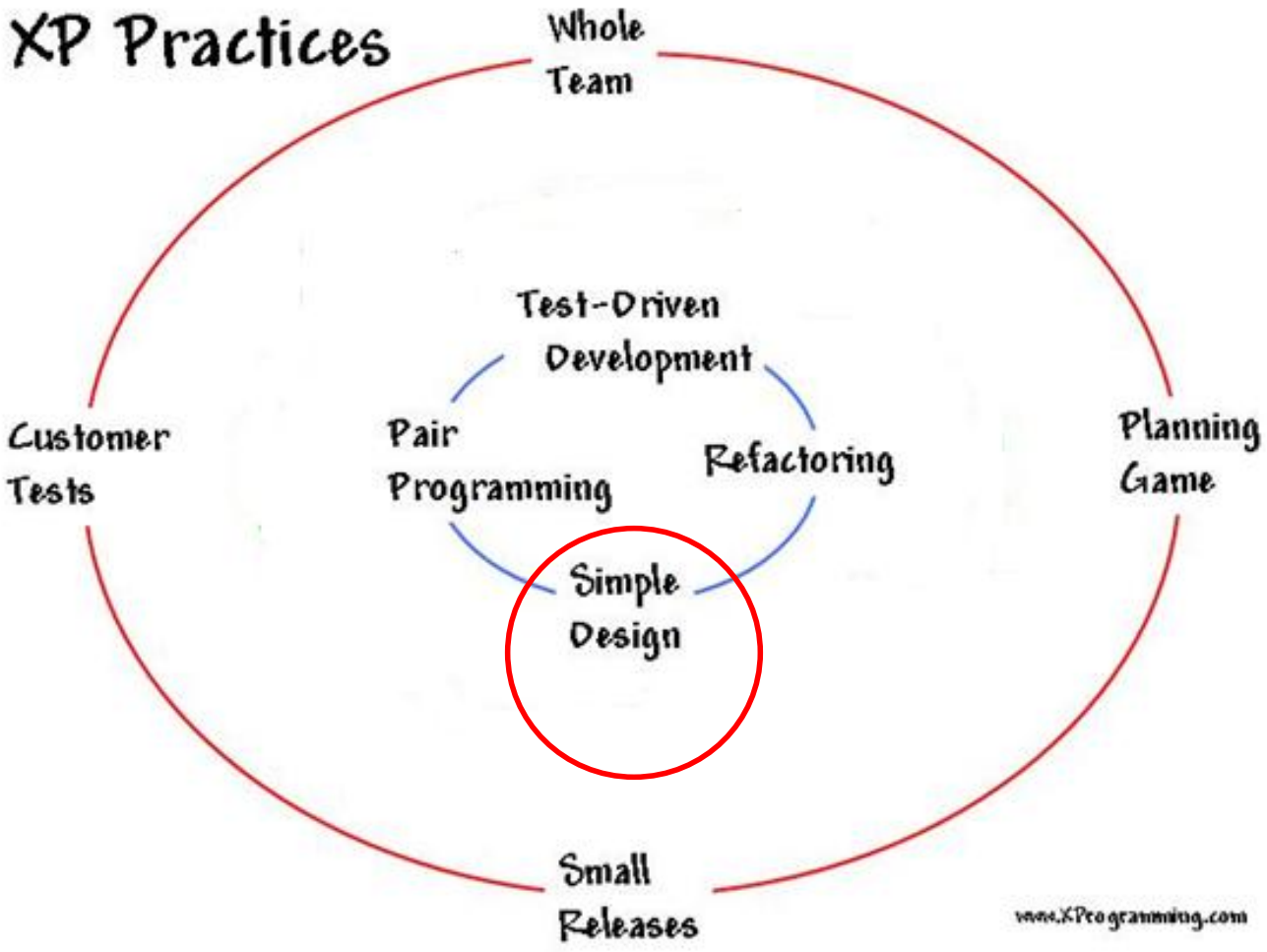
(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

XP Practices: Customer Tests

- The Customer defines one or more automated acceptance tests for a feature
- Team builds these tests to verify that a feature is implemented correctly
- Once the test runs, the team ensures that it keeps running correctly thereafter

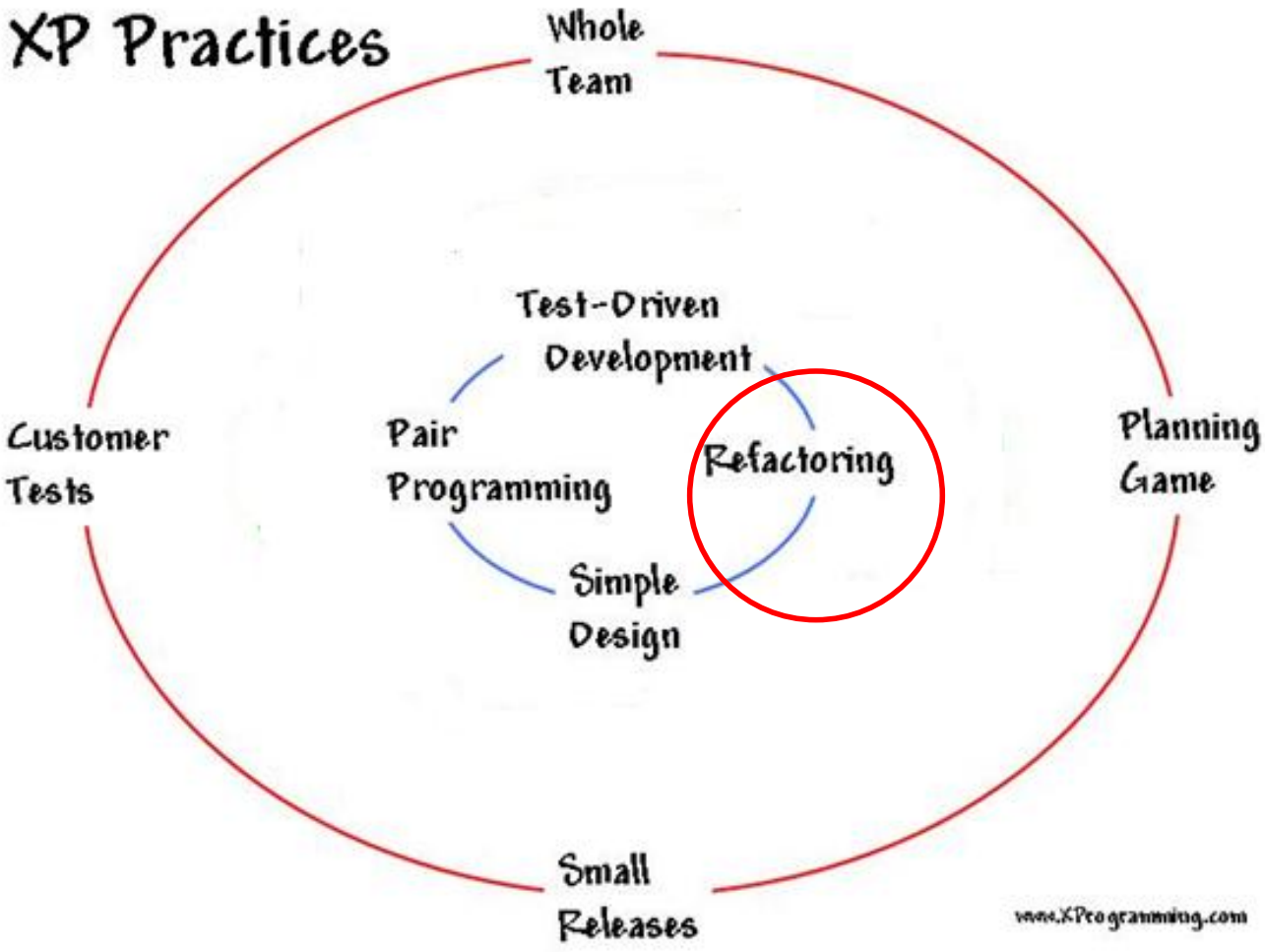
XP Practices

Keep the software simple and the design suited to current functionality



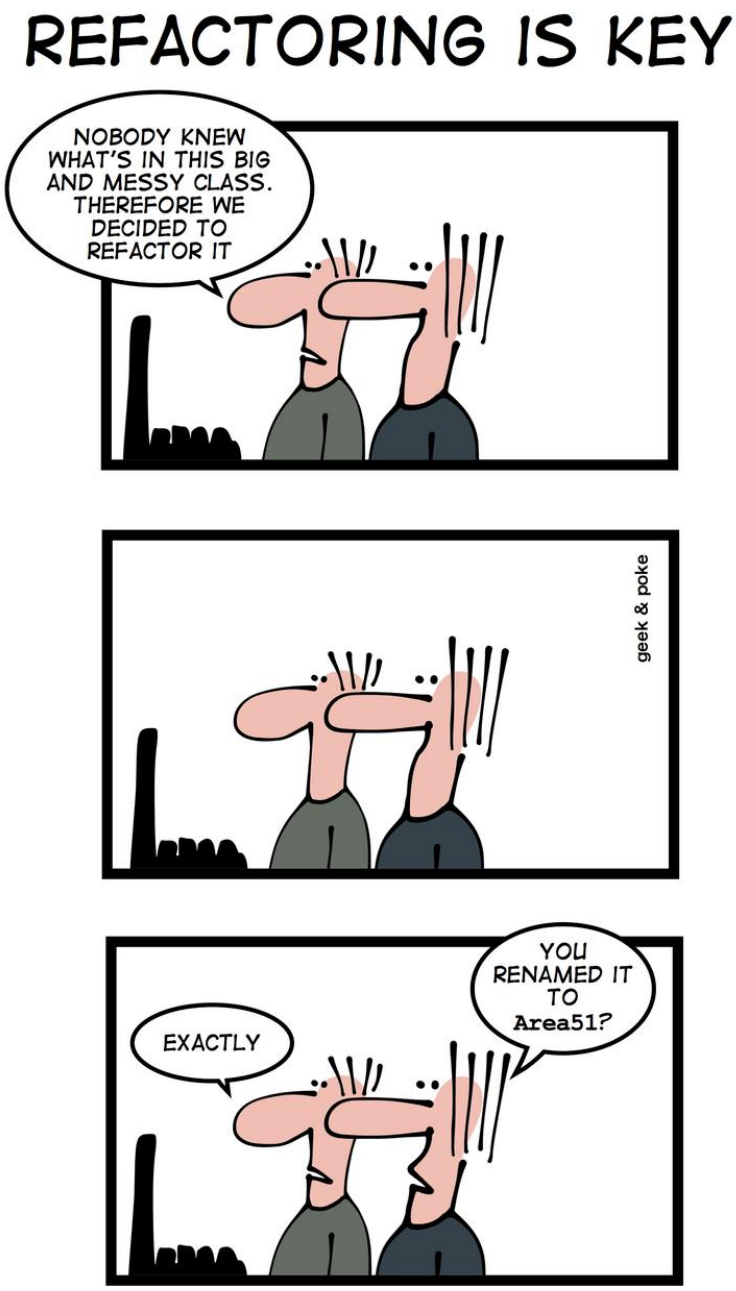
(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

XP Practices



(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

What is refactoring?



Refactoring

- It is the process of identifying low quality sections in the code base and rewriting them so that they become high quality.
- Changing how the system does something but not what is done
- No new functionality is added during a refactoring.

Example?

Example

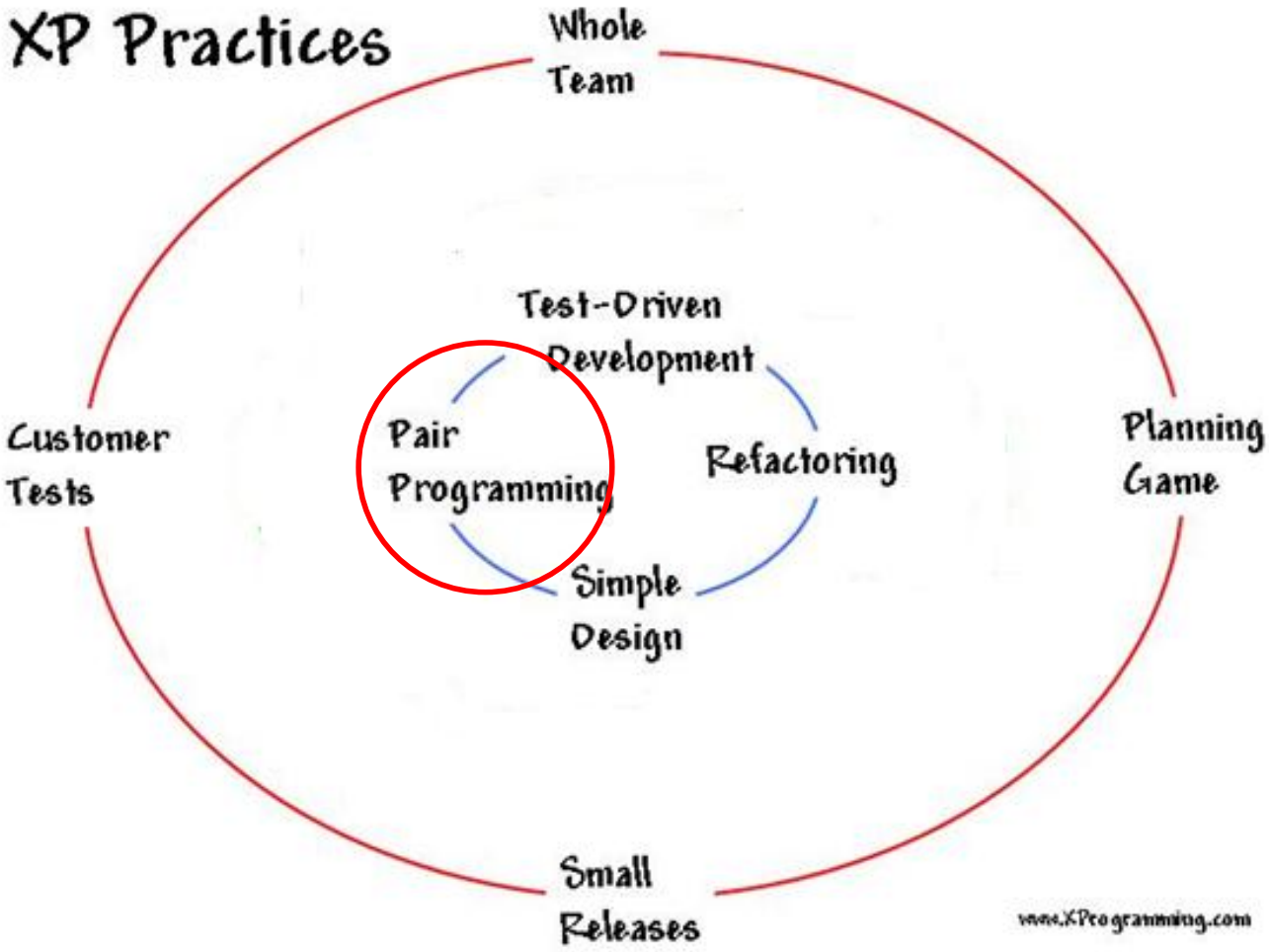
- Simple refactoring:
  - May be as small and simple as a name change for a variable, or
  - splitting a complicated method into multiple, better structured methods.
- More complicated refactoring:
  - identifying commonality in a set of classes and extracting an inheritance hierarchy from it to reduce duplication of code.

WHY Refactoring?

WHY Refactoring?

- Continual changes to the software as new features are implemented has a tendency to degrade the quality of code,
- The code base becomes more and more difficult and expensive to maintain over the lifetime of the system after delivery until, finally, it becomes cheaper to replace it entirely.
- Extreme programming follows a policy of aggressive refactoring for this reason.

XP Practices



(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

XP Practices: Pair Programming

All production software is built by two programmers

Why Pair Programming?

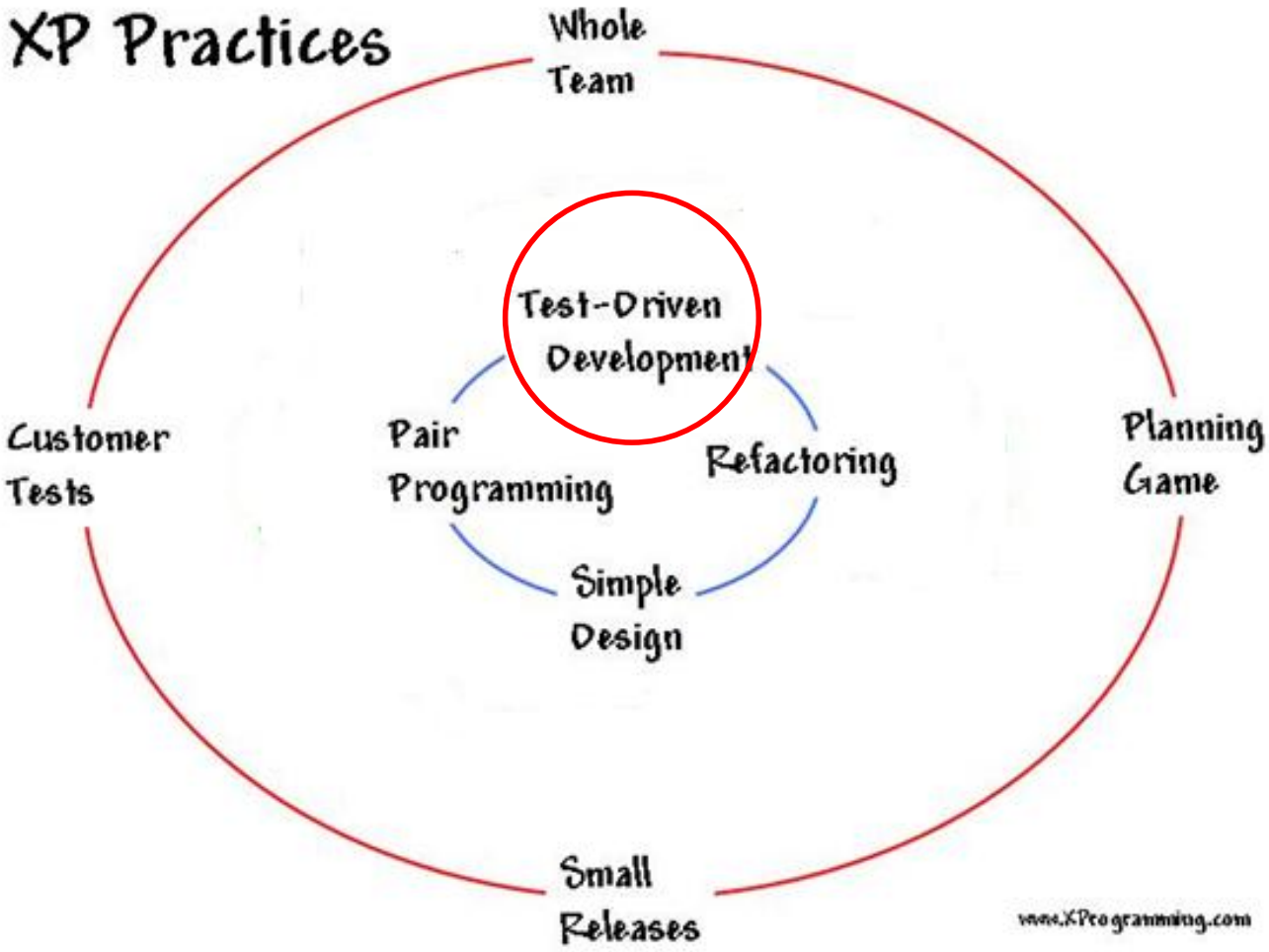
Why Pair Programming?

- All production code is reviewed by at least one other programmer
- Pairing also communicates knowledge throughout the team
- The cooperation leads to better understanding,
- Better initial approaches and less backtracking and rewriting.
- Fewer errors are made and less time is therefore needed to find and fix bugs.
- Significant support for refactoring

Why Pair Programming?

- An individual programmer become a bottleneck in a project
- If that programmer leaves then the project can be delayed

XP Practices



(Source: http://www.xprogramming.com/xpmag/whatisxp.htm)

Testing

- Incremental test development
- Customer involvement in testing
- Automated test frameworks
- Test Driven Development

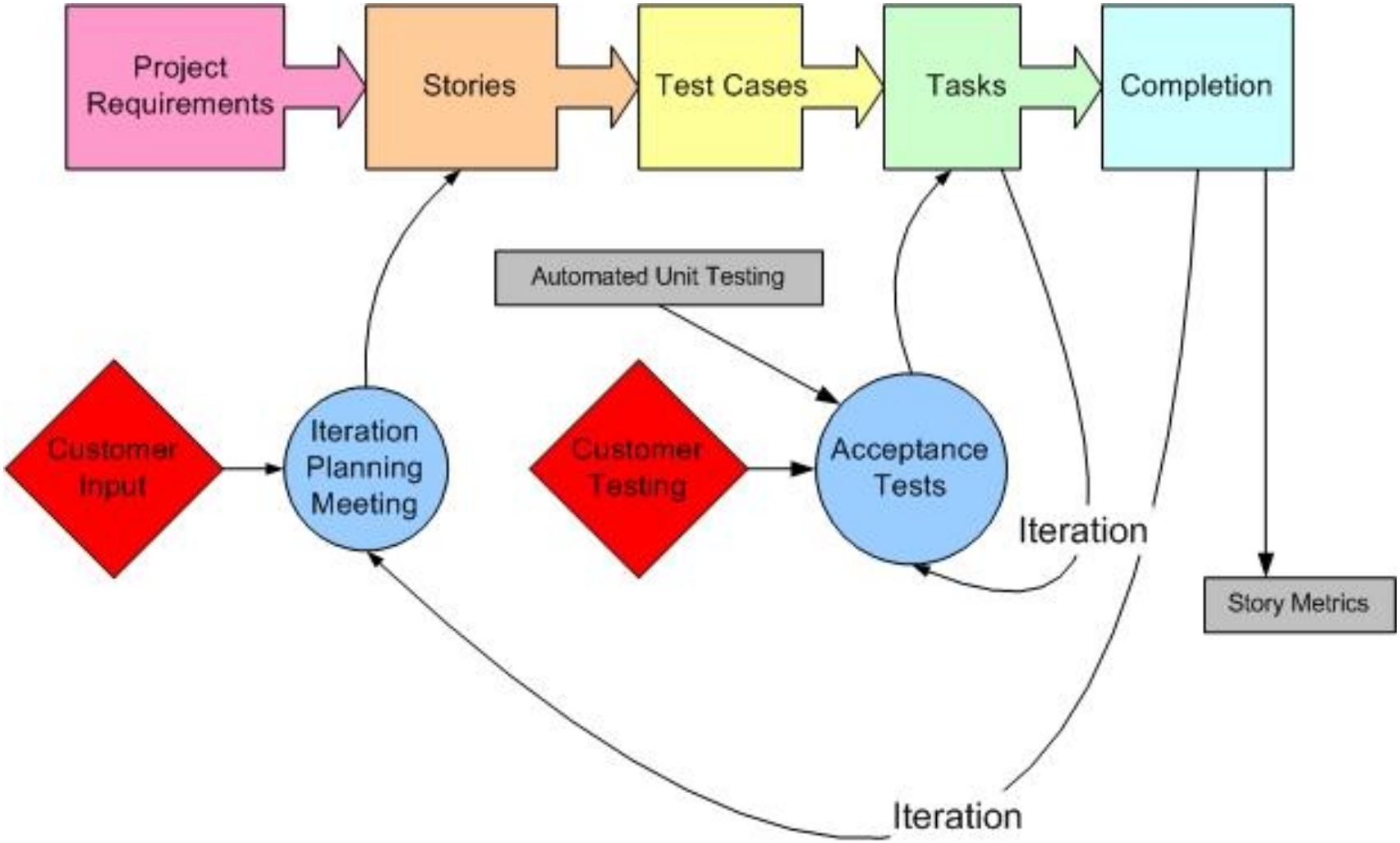
Why Test Driven Development in XP?

Why Test Driven Development in XP?

- In a plan driven approach, we would use the detailed specifications to design our tests so that, if the system passed the tests, we could be reasonably sure that it meets the specifications

- XP is an incremental development approach, we do not have the detailed requirements and specifications that we would expect to have in a plan-driven approach

Test Driven Development in XP



References

- A number of slides in this talk is based on:
  - Alan P. Sexton hand-outs (Introduction to Software Engineering. The University of Birmingham. Spring Semester 2014)

Thank YOU ☺