

Agile Development

Principles and Processes for
great software development



About me

- Scott Schimanski
- QA Manager at Healthwise, Inc.
- 18+ years in software industry as Developer, Project Manager and Tester
- Been involved in Scrum for 3+ years
- Blog: www.elegantcode.com
- Email: scott@elegantcode.com



<ELEGANT*CODE/>

What about you?



Overview

- ◉ What is Agile
- ◉ Iterative and Incremental Development
- ◉ Agile Practices
- ◉ Agile Methods (Scrum and XP)

What is Agile?

Agile Values

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Agile Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Agile Principles Continued

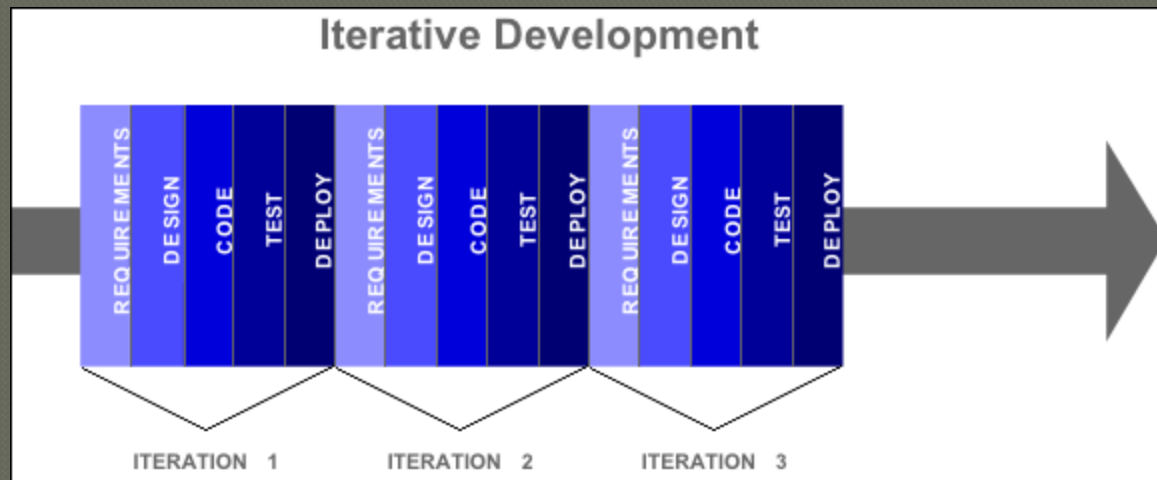
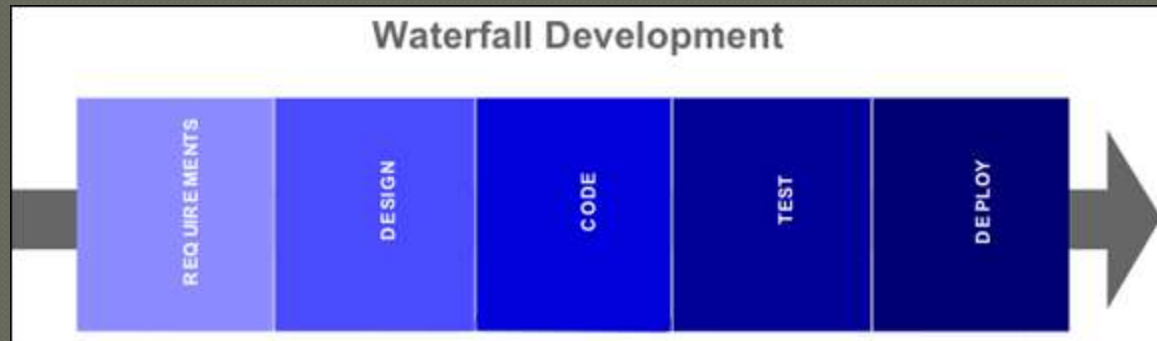
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Core Agile Tenants

- Iterative and Incremental Delivery
- Time boxed Iterations
- Transparency
- Feedback
- Customer inclusion
- Self-directed teams

Iterative and Incremental Delivery

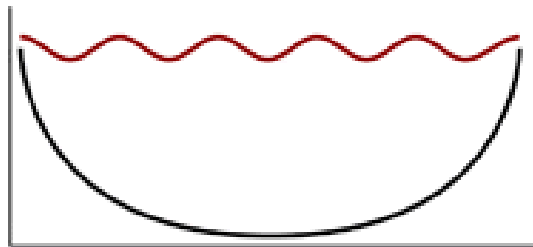
Iterative



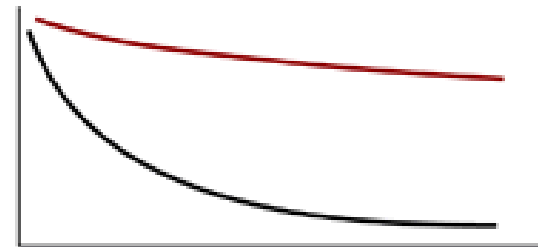
Agile vs Traditional

Agile Development Value Proposition

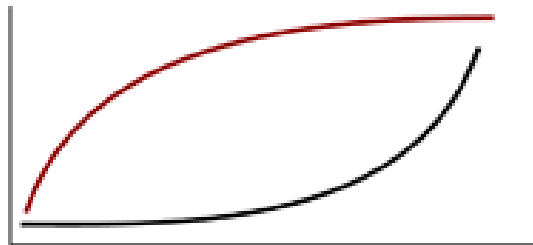
VISIBILITY



ADAPTABILITY



BUSINESS VALUE



RISK

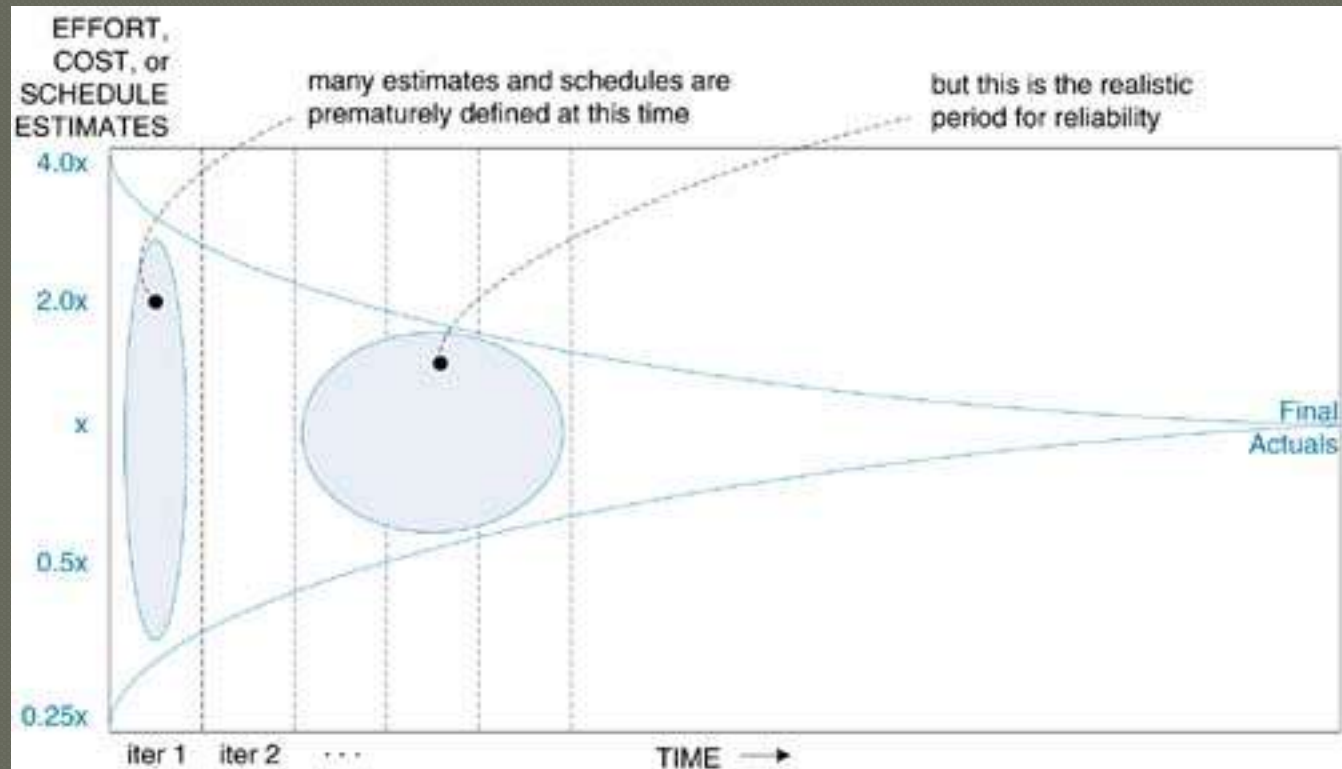


— AGILE DEVELOPMENT

— TRADITIONAL DEVELOPMENT

Copyright © 2004 - 2006, Version One, LLC.

Cone of uncertainty



Agile Principles in Practice

Time Boxed



Take a thin slice

- ◉ Vertical slice of the product
- ◉ Thin (small) piece of functionality



Must be done!

- Completed
- Verified
- Installable
- Shippable
- Meets customer needs
- Adds business value



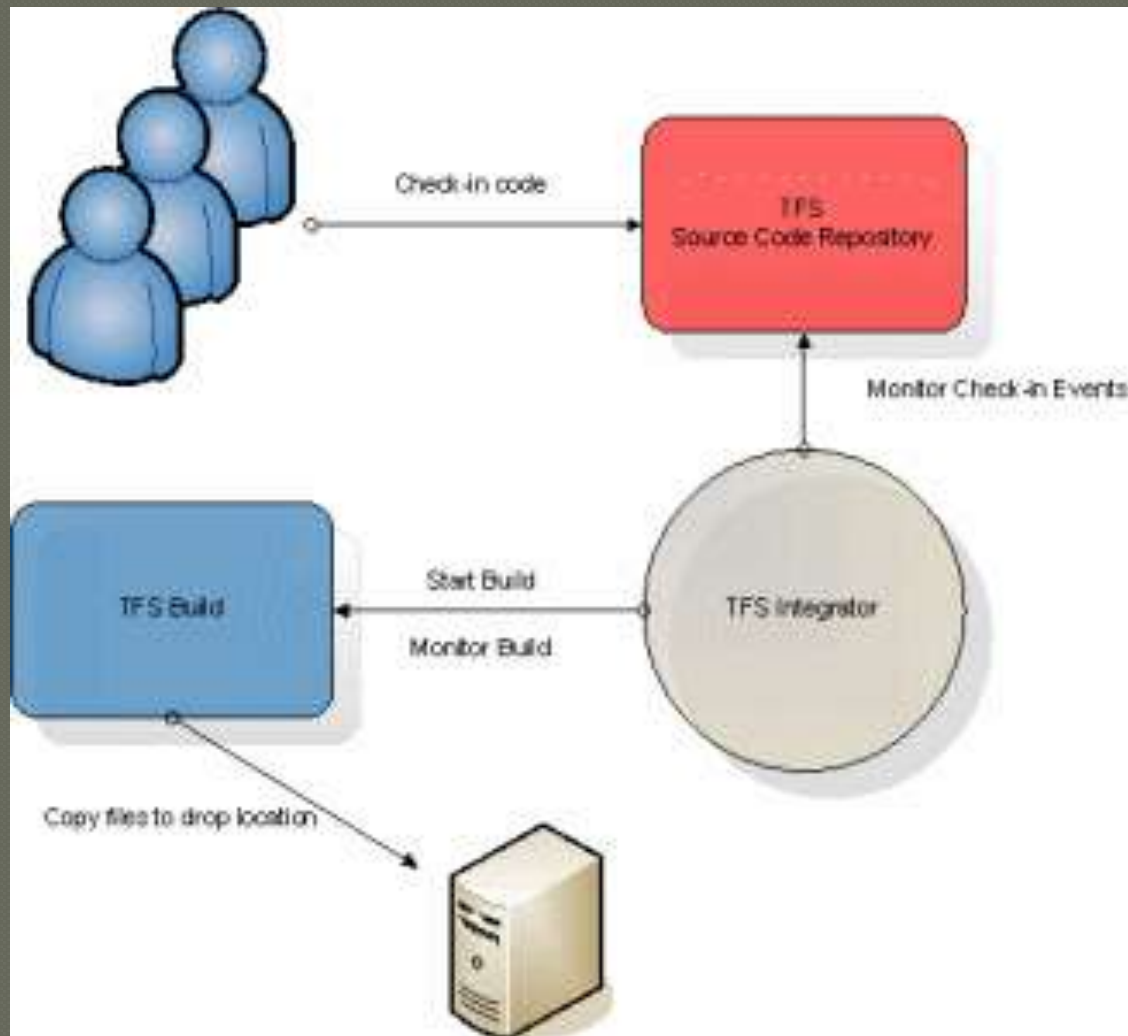
Feedback



Testing

- ◉ Testing occurs every step of the way
- ◉ Sometimes tests are written before the code is written
- ◉ Unit testing – Testing the code
- ◉ Acceptance tests – Testing the functionality

Continuous Integration



Keep your room (code) clean



Refactoring

- ◉ Cleaning up the codebase without changing functionality
- ◉ Code smells
- ◉ Refactoring patterns

Constant improvement



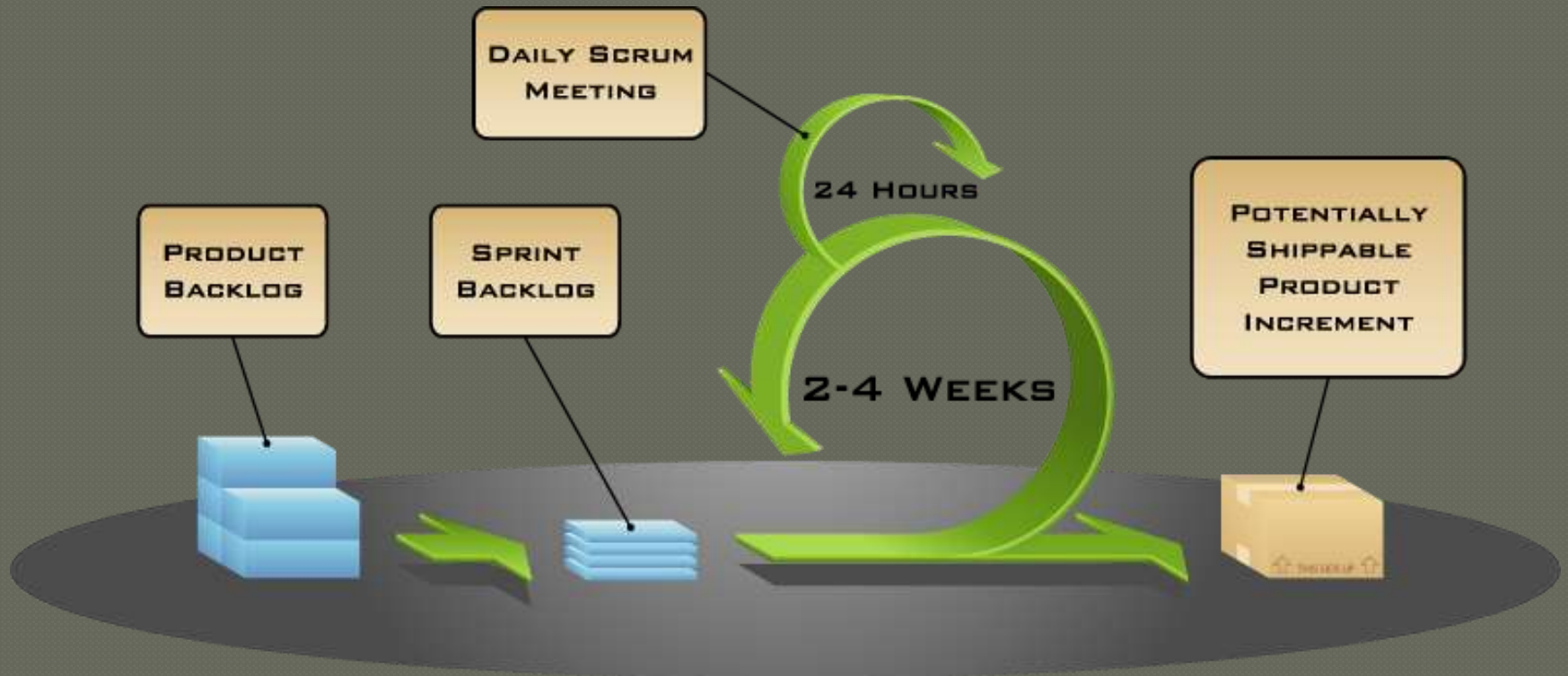
Agile Processes

Scrum and XP

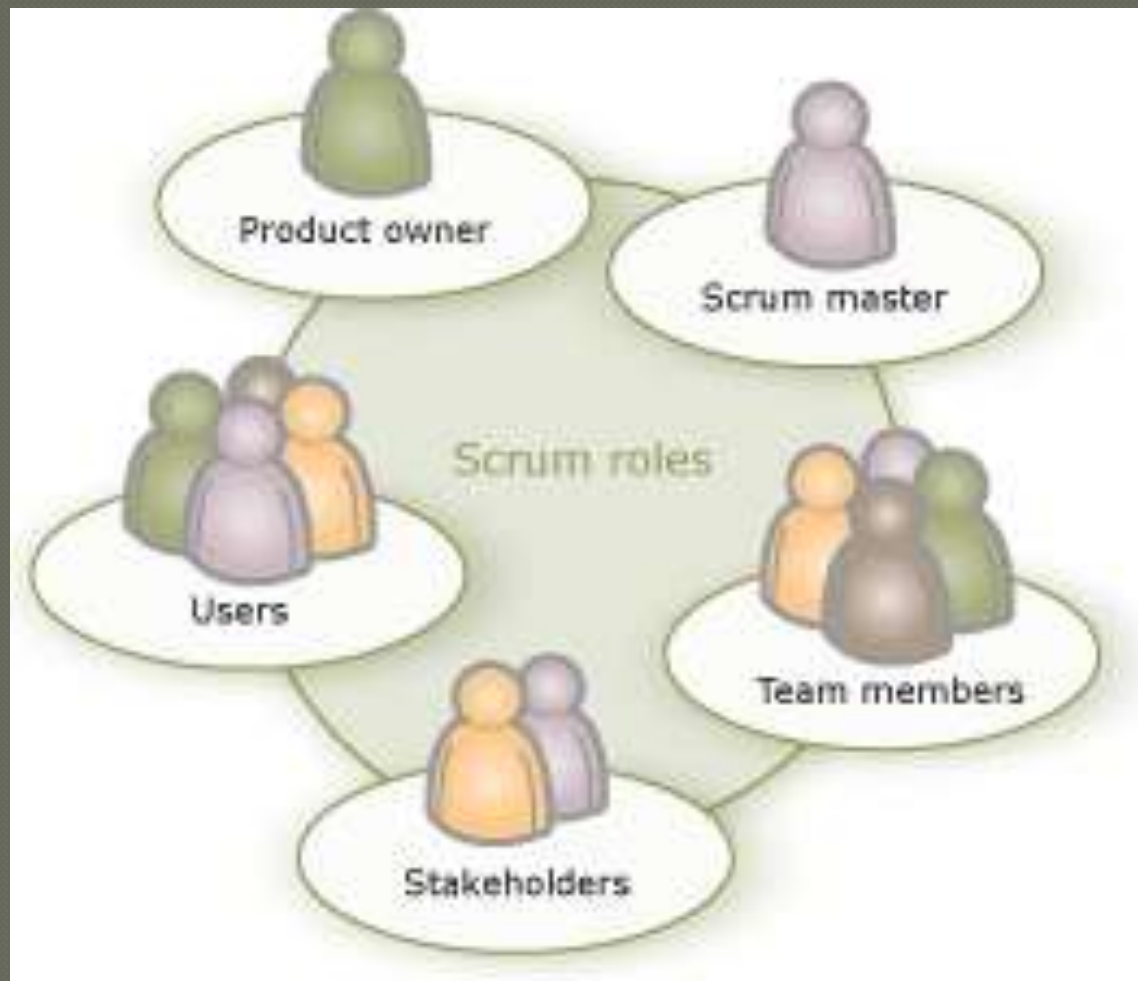
Scrum



The Scrum Process



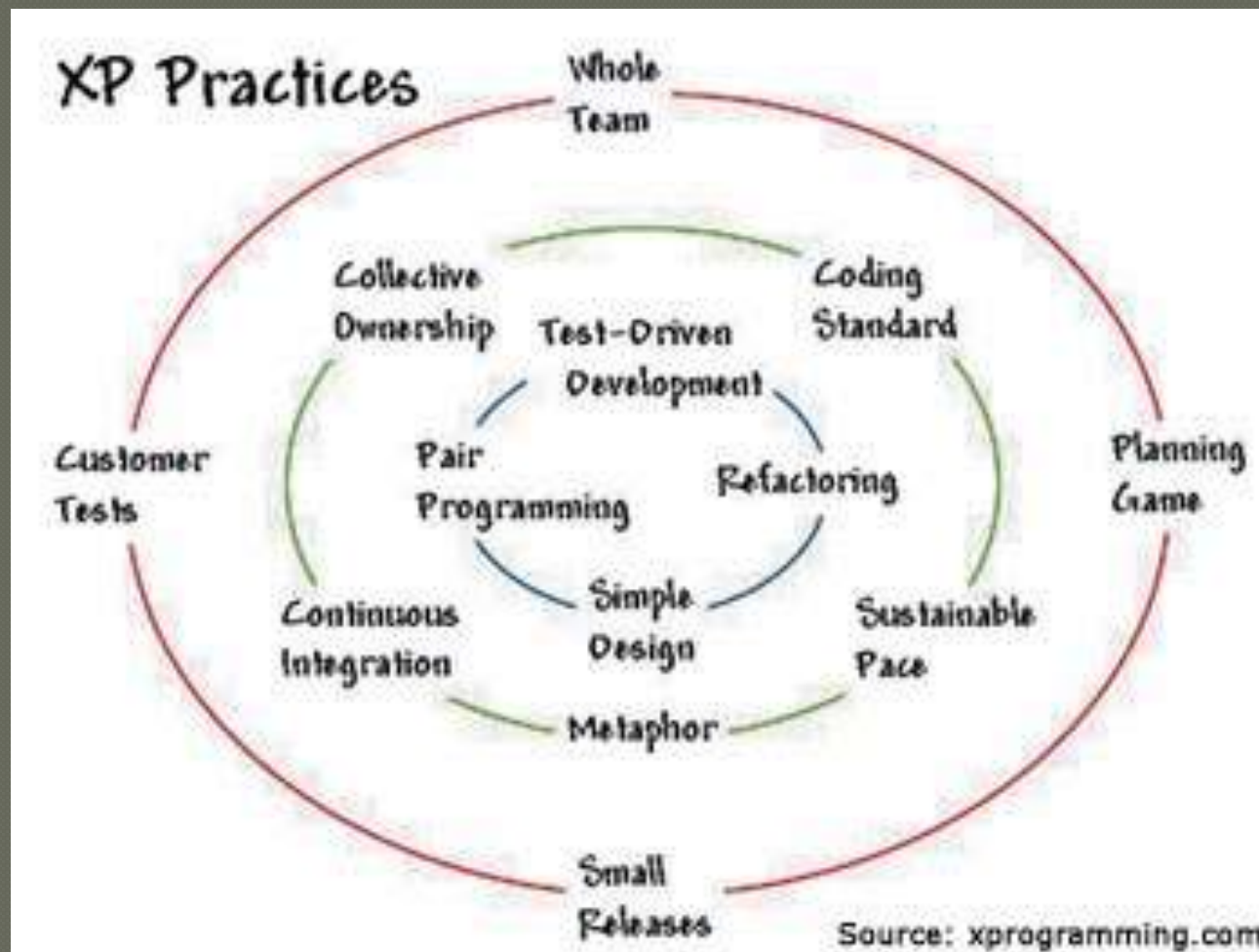
Scrum Roles



eXtreme Programming (XP)



XP Practices



XP Values

- Communication
- Simplicity
- Feedback
- Courage

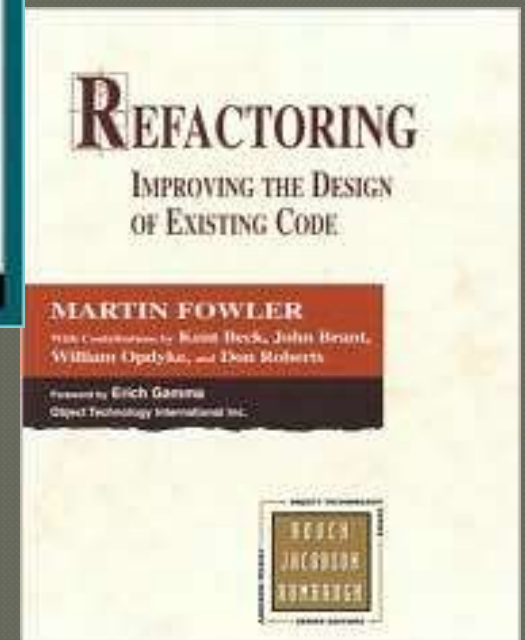
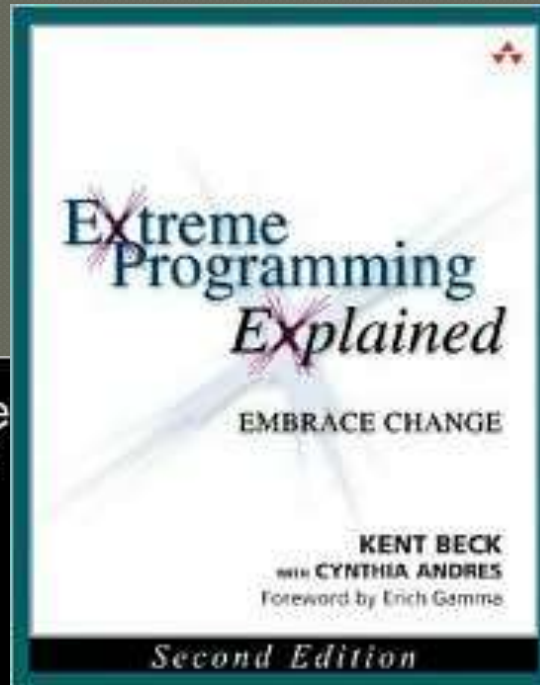
Some References



Agile Software Development with Scrum

red
yellow
green
blue
red
blue
yellow
green
blue

Ken Schwaber ■■■ Mike Beedle



Questions?

