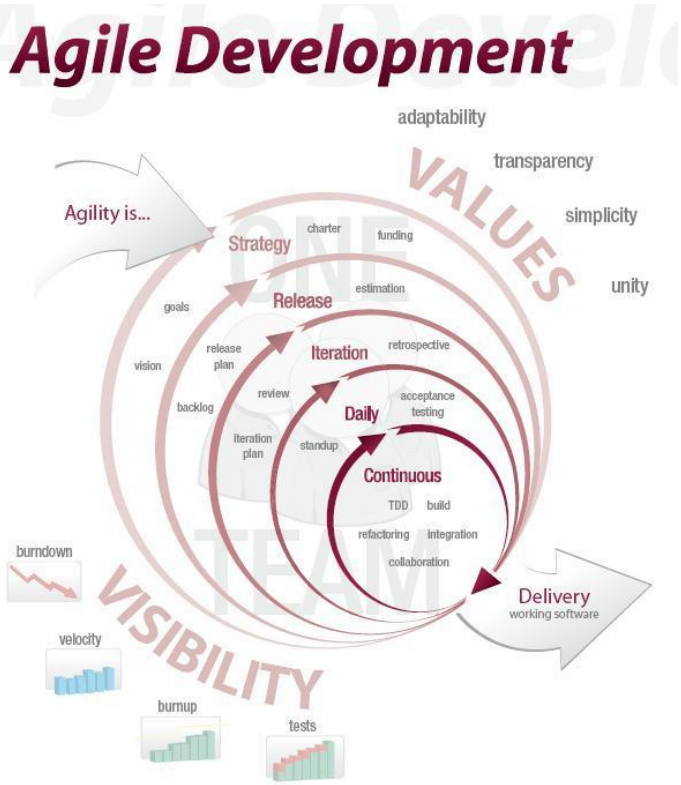
TruePlanning®
by PRICE® Systems

Applying Agile Practices to Improve Software Quality

Name: Arlene Minkiewicz
Chief Scientist
17000 Commerce Parkway
Mt. Laurel, NJ 08054
arlene.minkiewicz@pricesystems.com
Phone: 856 608-7222

Agenda

- Introduction
- Agile Development
- Software Quality
- Agile Practices Most Likely to Impact Software Quality
- Conclusions



Introduction

- **Many development organizations are revisiting how they build software because of**
  - Increasing frustration with software project failures
  - Fast paced software development
  - Rapidly changing business needs
- **Many are turning to agile development**
- **But agile means different things to different organizations**
  - Some say it improves software development productivity while others say it slows them down
  - Some say it improves the quality of the finished product while others claim it interferes with delivering quality
- **This paper focuses on helping business leaders determine if agile practices will help them meet their quality goals**

Agile Development

- **From the agile manifesto**

"*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 on the items on the right, we value the items on the left more"

Agile Development

- **Agile practices focus on:**
  - Simplicity
  - Customer focus
  - Shared responsibility
  - Close collaboration
  - Frequent and direct communication
- **Agile development is based on several values**
  - Communication
  - Simplicity
  - Feedback
  - Courage
  - Respect

Agile Development

- **There are many implementations of agile… Xtreme Programming, SCRUM, Lean Software Development, etc.**
- **Agile practices include:**
  - Test Driven Development
  - Simple Design
  - Pair Programming
  - Refactoring
  - Continuous integration
  - User Stories
  - Short Iterations

Software Quality

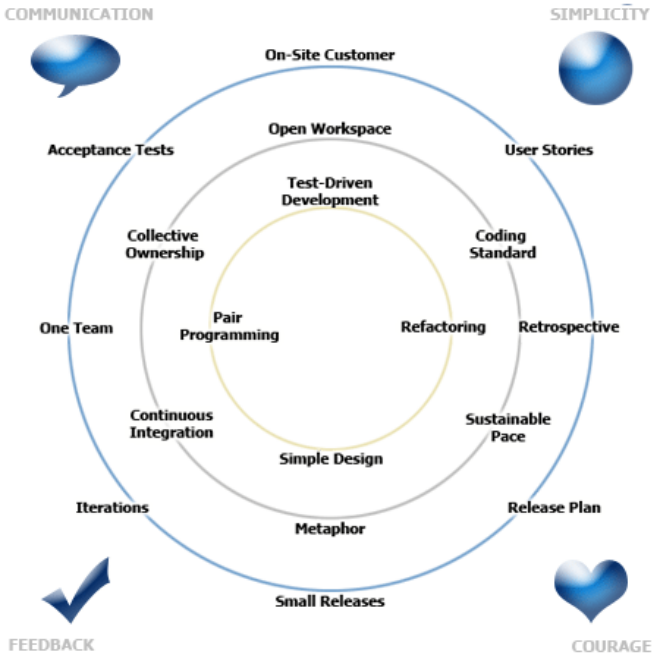
- **According to Wikipedia, software quality is….**

  "A measure of how well software is designed (quality of design) and how well the software conforms to the design (Quality of conformance"

- **This definition implies two aspects of software quality**
  - Building the right thing
  - Building it right
- **Various Agile practices exist to address each of these aspects of quality**

Agile Practices Likely to Impact Quality

- **Pair Programming**
- **Test Driven Development**
- **Continuous Integration**
- **Short Iterations**



Pair Programming

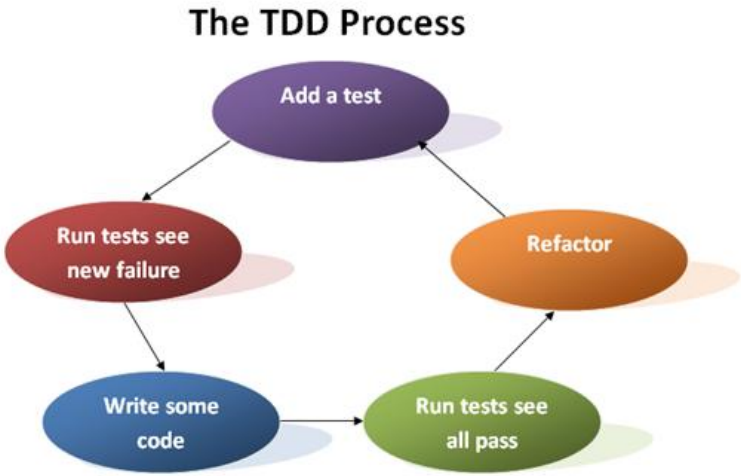
- **Two programmers complete single programming task**
  - One computer
  - One driver
  - One navigator
  - Roles change often
  - Pairings change often within the team

Pair Programming

- **Several studies have shown that pair programming has a positive impact on quality of code.**
  - With pair programming, number of post development tests that passed increased by 15% [5]
  - Experiment at University of Utah shows an average of 14% increase in passed tests on programs using [6]
- **Quality improvements attributed to pair programming are focused 'building it right'**

Test Driven Development

- **No code is written for a feature until the tests for that feature are written**
  - Originally tests fail since no code has been written
  - User story and interview with stakeholders leads to understanding of feature
  - Automated test is generated
  - Just enough code to make it pass
  - Once it passes, refactoring occurs to make it cleaner, simpler, using test to ensure it continues to pass

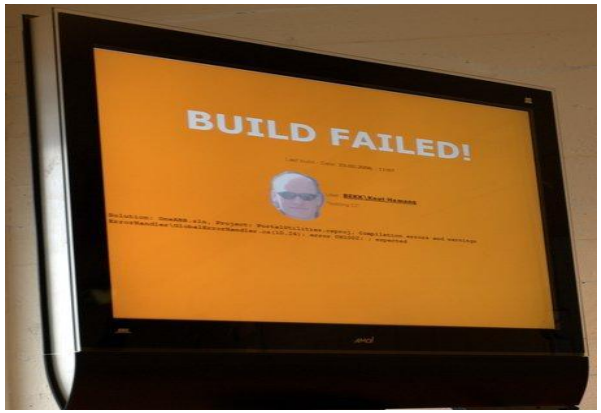


Test Driven Development

- **Tests conducted at Microsoft showed 2.5x and 4.2x defect rate decreases between projects of similar size and scope, one using TDD and one without [7]**
- **An article in IEEE Software documents 18 studies across the industry with 10 documenting improved quality with TDD, 7 with inconclusive results and only one with a quality decrease [8]**
- **TDD, properly implemented should help 'build it right'**
  - If customers also write tests, TDD can also help 'build the right thing'

Continuous Integration

- **Changes in code base are continuously integrated into an operational system**
  - Automated process integrated with automated test suite
  - Real time feedback to the developer when code change results in bad system behavior
  - When the build breaks, solving the problem becomes a top priority of the team
  - The more frequent integrations the easier it is to isolate the reason for the failure



Continuous Integration

- **While there are no studies focused specifically on continuous integration and quality**
  - One expert reports observing projects that use continuous integration have dramatically less bugs in production [10]
  - Steve McConnell suggests that a benefit of frequent builds is reduced risk of low quality
- **While the actual process of continuous integration is focused on 'building it right',**
  - The 'real time' availability of new and emerging features sets the stage for customer stakeholders to also ensure that the right thing is built

Short Iterations

- **Generally a ‘release’ of the software is made every couple weeks.**
  - Team transcends the ‘waterfall’ activities within this short time – focused on getting a few capabilities into the release
  - Customer stakeholders have the chance to review the software as capability emerges and redirect or reprioritize as necessary
  - Functional testing can be done as features are completed
- **While there’s no scientific or controlled evidence of increased quality with short iterations, properly executed with customer participation, this practice should result in ‘building the right thing’ that will result in customer satisfaction**

Conclusions

- **Agile development can be shown (in many cases) to …**
  - Decrease defects
  - Increase customer satisfaction
- **There is also research that supports the opposite conclusion**
- **The disparity stems from the fact that not all agile implementations are the same**
- **Some agile practices, or combination of agile practices, are more likely to result in quality improvements than others.**
- **All forms of agile encourage direct, frequent communication among the team and with the customer…**
  - Certainly an ingredient for customer satisfaction!