eXtreme Programming

Ken Auer
kauer@rolemodelsoft.com

Roy Miller
rmiller@rolemodelsoft.com

RoleModel Software
Holly Springs, NC
919-557-6352
http://www.rolemodelsoft.com
The Joy of Software

- To change Hardware
  - Soldering Iron
  - Screwdrivers
  - Change PCB

- To change Software
  - Just change some symbols
The Reality of Software

- Lots of assumptions/dependencies
- The steep part of the curve stinks!
- How do we avoid life there?
BDUF? (Big Design Up Front)

- Design is critical, but why do it all up front?
- Have a picture of the horizon, but haven’t gotten any closer to it
Basic Misunderstanding

Documentation is not understanding
- One study of typical requirements documents: 15% complete, 7% correct, not cost effective to increase (Source: Elemer Magaziner)

Formality is not discipline
- What happens when you apply good rules at the wrong time
- Consistently signing a document is not the same as consistently reflecting on whether you are doing the best thing
- What if people “cheat”?

Process is not skill, especially when innovating!
- You can’t “code by number”

* Some content supplied by Jim Highsmith, “Adaptive Software Development”
The Wrong Model

¬ “Do It Right the First Time” sends wrong message and causes bad behavior
¬ We can’t
  ▲ be uncertain
  ▲ experiment
  ▲ learn from mistakes
  ▲ deviate from plan
¬ “Don’t worry about getting it right the first time, get it right the last time.”

Envision-Explore, not Plan-Do

* Some content supplied by Jim Highsmith, “Adaptive Software Development”
Relying on a plan can produce the product you intended, just not the product you need. It assumes you were smarter when you started and didn’t learn anything along the way.

* Some slide content supplied by Jim Highsmith, “Adaptive Software Development”
Words of Wisdom

Paul MacCready on Innovation and Competitive Advantage
A Better Approach

- Demands and depends on collective understanding, discipline, and skill
- Leverages learning
- Lets you go fast
- Lets business drive (really)
- Lets design/systems evolve over time, at relatively low cost
- Preserves option to change
- Reduces the total cost of ownership of systems
Dee Hock--former CEO Visa

“Simple, clear purpose and principles give rise to complex, intelligent behavior.”

“Complex rules and regulations give rise to simple, stupid behavior.”
What XP Is*

“Extreme Programming is a discipline of software development based on values of simplicity, communication, feedback, and courage. It works by bringing the whole team together in the presence of simple practices, with enough feedback to enable the team to see where they are and to tune the practices to their unique situation.”

* http://xprogramming.com/xpmag/whatisXP.htm
Extreme Programming

- One Team that shares core values and
  - works toward a sound business asset
  - knows where it is
  - leverages its learning
  - embraces three fundamental roles
    - "Developers" estimate cost
    - "Customers" decide priorities
    - "Management" facilitates relationship in a context

- Turns "best practice" knobs up to 10
The Four Values

- Simplicity
  - OAOO: Once And Only Once
  - TSTTCPW: The Simplest Thing That Could Possibly Work

- Communication: developers and business people

- Feedback: the earlier the better

- Courage: don’t be afraid to write code, change code, or throw away code
What the Values Mean

- Customer Value...
  - results vs. plausible deniability
- Collaboration...
  - innovation through group interaction vs. individual heroics
- Adaptation...
  - embracing change vs. “change control”
- Minimalism...
  - simplicity vs. “all things to all people”
Twelve Original Practices

- The Planning Game (Split Business & Technical)
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40 Hour Week
- On Site Customer
- Coding Standard
Revised(?) Practices

**Joint**
- Short Iterations
- Common Vocabulary
- Open Workspace
- Reflection

**Development**
- Test-first Programming
- Pair Programming
- Refactoring
- Collective Ownership
- Continuous Integration
- Just-in-time Design

**Customer**
- Story Telling
- Release Planning
- **Acceptance Tests**
- Frequent Releases

**Management**
- Appropriate Responsibility
- Diplomacy
- (Quarterly) Review
- Reality Checking
- Sustainable Pace
The Customer Drives
What’s up?
Will The Tests Pass?
Design is Crucial...
Do it Daily
I Heard It Through The Pairvine
Can You Take XP Seriously?

“I predict that Kent Beck and his XP movement will be as much a symbol of our times as Watts Humphry and the CMM were a symbol of the eighties and early nineties.”

Tom DeMarco
Cutter Report on Light Methodologies

“XP is not an option... it’s a requirement.”

Kevin Tyson
CTO, Netfolio
XP - The Big Picture

- Poor responsiveness to learning and poor practices both lead to steep curve
- BDUF doesn’t flatten the curve
  - Attempts to reduce how often you are bitten
  - Adds costly artifacts at expense of real progress
- XP helps push the curve down
  - Leverages learning
  - Move more confidently
  - Move faster for longer