Deep Agile
Blending Scrum and Extreme Programming

Jeff Sutherland

Ron Jeffries

Separation of XP and Scrum Methods

- * Largely Historical
- * XP chose to write more down
- * XP “programmer” focus
- * Successful Scrum Teams use XP-like Practices

Need for “Technical” Practices

- Interative Planning and Delivery
- Feature Focus
  - ➔
- Incremental Design
- Frequent Testing

XP Practices not Purely Technical

- User Stories
- Cards
- Informative Workspace
  - Big Visible Charts
  - Tracking Boards
- Planning Game
- Small Releases
- … etc

Developer Focus

- Low in Scrum
  - (discovery)
- Higher in XP
  - (starting point)

XP's Project Management Practices

- Compatible with / Analogous to Scrum

- Small Releases
- Whole Team
- Planning Game
  - Iteration Planning
  - Release Planning
- …

XP Practices

- What are they?

- Their role in Scrum … or any project.

XP Practice

* Simple Design

* Must proceed incrementally
* Must deliver features right away
    * ➔
* Begin with a simple design

XP Practice

* Pair Programming

* Design is changing …

* How do we know what it is?

XP Practice

- Test-Driven Development

- System must work all the time;
  - Must ship every week or two;

- How do we know it isn't broken?
- Test it!

XP Practice

- Refactoring
- Simple Design cannot last!
  - Are we doomed???
- Design must improve …
  - Refactoring is the technique
    - (Refactoring: Martin Fowler)

XP Practice

- **Continuous Integration**

- **We’ll be integrating at least every two weeks;**
- **Integration is always hell;**

- **Get good at it!**

XP Practice

- **Shared Code**
- **“Any pair can improve any code at any time”**
  - **-- Kent Beck**
- **Faster progress**
- **More eyes**
- **Higher Quality**

XP Practice

- Coding Standard
- All eyes on all code
- Make it look alike
- Don’t go crazy – just make it look alike

XP Practice

- Whole Team

- All the people you need
- When you need them

- Technical Resources
- Business Resources

XP Practice

* User Stories

* Card

* Conversation

* Confirmation

XP Practice

- Automated Customer Tests
- How often do you need to test this?
- How often do you want to test it by hand anyway???

XP Practice

* Informative Workspace

* Tracking Boards
* Big Visible Charts

XP Practice

* Energized Work

* Work vigorously
* Rest effectively

* Be done at the end of every day!

XP Practice

* Sustainable Pace

* If you go too fast, you’ll go off the road!

* Do not push for more stories!!

XP Practice

- Slack

- Have items on the list that can be dropped
- You can always ask for more
- Something will come up

XP Practice

- Weekly Cycle

XP Practice

- **Quarterly Cycle**

XP Practice

- Code and Tests
- These are the only documents that count
- Generate the rest!

XP Practice

- Team Continuity

- Software is a work of the mind
- Why throw that away?

XP Practice

- Single Code Base
- Anything else is waste!
- Learn to do it right
- Generate old versions if you must – Don't save them

XP Practice

* Daily Deployment

* Go home clean!
* Deliver value instantly!

XP Practice

* Incremental Design

* (Discussed – Refactoring)

XP Practice

- **Sit Together**

- **And talk with each other!**

- **Would you rather have a hug from your Mom, or a nice note?**

XP Practice

- **Ten Minute Build**
- **Yes, we really mean it**

XP Practice

* Real Customer Involvement

* Don’t guess – find out.

* Set their expectations
* Help them learn too

XP Practice

- Incremental Deployment

XP Practice

- Shrinking Teams

- As need declines …
- People move on

XP Practice

- Root Cause Analysis
- “Ask Why Five Times”

XP Practice

- … and many many more …
- The richness of software development …
- And of how to do it …
- Cannot be plumbed by any list!

Forces and Responses

* These practices are not random,
* Nor are they mandatory
* But they are needed
* And there are reasons why

Forces and Responses

* **Teamwork**
  * **Team Room**
    * **2X Productivity!!**
  * **Don’t Specialize**
  * **Don’t Hand Off**
  * **Pair Programming**
  * **Shared Code**

Forces and Responses

- Business Value
  - Small Stories
  - Avoid Tasks
  - Automated Customer Acceptance Tests

Automated Customer Acceptance Tests

- Communcation
- Learning
- Confidence

- Repetition!

Forces and Responses

- Continuous Delivery
  - Grow the Design
    - Refactoring
  - Continuous Integration
  - Ten Minute Build
  - Shared Code
  - Energized Work
    - Sustainable Pace

Forces and Responses

- Predictability
  - Get Done = Done
    - Tests
      - Test-Driven Development
    - Stories vs Tasks

XP and Scrum and XP and …

- Fundamental Message

- Focus on what your team needs
- Focus on Running Tested Features
  - Done =Done

- Not “Is it Scrum?”
- Not “Is it XP?”
- “Is it working for us?”