Extreme Programming

Embrace Change

Introduction

Dr. Peter Lappo
www.smr.co.uk

Agenda

- Junior and Senior Programmer.
- Review of existing practice.
- How does XP improve current practice.
- Overview of XP.
- A new paradigm.

Classical Software Development

- Waterfall
  - Analysis
    - Design and Code
      - Test
        - Deployment
- Typical time from start to end 3-24 months.
- Plenty of reviews and checks in the process.
- Process analogous to building houses.

Pros

- Pros
  - Industry standard process.
  - Everyone understands the process.
  - Approved by standards bodies.
  - Plenty of books and training available.
  - Management feel they are in control.
  - The current paradigm.

Cons

- Cons
  - Projects are often late, cancelled or don't deliver what users want.
  - Status is hard to determine (99% complete syndrome).
  - Heavy weight in terms of documentation and people.
  - Project plans rarely reflect reality.
  - Software is built to "frozen" requirements.

Traditional Risk (Fear) Management

- Traditional processes control risk by,
  - defining a fixed process with intermediate products,
  - plan the project based on set of requirements,
  - enforce QA checks on intermediate products,
  - stop until intermediate products are correct.
- Traditional processes assume
  - you know what you want up front,
  - cost of change increases rapidly with time.

Consequences of Current Practice

- Escalating costs.
- Long lead times.
- Lack of flexibility.
- Not responsive to changing business needs

- Organisations must be agile in a changing and competitive world.

Some Options to Improve the Current Practice

- Become very heavy with the current process.
- Introduce modern methods and tools.
- Get ISO 9001 accreditation.
- Hire more QA staff to police the process.
- Hire more and better development staff.
- Beat each other up when we fail.

Alternatively Face Reality

- Virtually impossible to get the requirements right before writing code.
- "Frozen" requirements are often wrong.
- New requirements are often "discovered" during code and test.
- Things will change so,

"**Embrace Change**."

Kent Beck, Extreme Programming Explained

XP Risk Management

- XP controls risk by,
  - Short iterations focused on business priorities and working code.
  - Constant communication and feedback
  - High customer involvement.
  - Self directing teams.
  - Robust set of automated tests.
  - Quality software that is easy to change.
  - Light on documentation (but not too light).

XP Risk Management (con't)

- XP assumes
  - you don't know exactly what you want up front,
  - cost of change increases **slowly** with time.

Customer Benefits

- Working code in production earlier.
- Software that the business wants.
- Closure on features.
- Greater freedom to change at any time.
- Can stop the project at any time and still have a useful system.
- Robust set of automated tests for the entire life cycle.
- Accurate view of project status.

Developer Benefits

- Bigger role in planning and estimating.
- Job satisfaction due to greater responsibility.
- Encouraged to produce quality software.
- Greater sense of team work and involvement.
- Closure on requirements increases job satisfaction and reduces risk of cancelling the project.
- More fun!!

XP Values

- Feedback
- Communication
- Simplicity
- Courage

XP Practices

- The Planning Game.
- Small Releases.
- Metaphor.
- Simple Design.
- Testing.
- Refactoring.
- Pair Programming.
- Collective Ownership.
- Continuous Integration.
- 40-hour Week.
- On-site Customer.
- Coding Standards.

Coverage and Applicability

- XP covers most of the life cycle from requirements capture to code and testing.
- Excludes feasibility studies and deployment.

- Applicable to most types of small to medium sized (12 people).
- Especially projects with uncertain or fluid requirements.

Maturity

- XP has been practised to the *extreme* for several years.
- 9 books on XP including an article in the Economist.
- Several international conferences.
- Support via discussion groups.
- Training and mentoring available.

- In fact it is just best practice.

XP : A New Paradigm?

- XP is a quality and people centric process.
- Self directing teams working together.
- Less "political".
- Comparable to Volvo way of building cars.

- Potential rewards are high.
  - Customers with flexible software addressing business needs.
  - Happy and fulfilled workers.

Software Development as a Factory

- Current practice assumes software development is like a factory production process.
  - feed raw materials in (requirements)
  - out comes a product (software)

Software Development as a Factory Design Process

- Software Development as a Factory is an inappropriate analogy
- Developing software is actually more like the creative process of designing a factory.
- Production process analogy causes a lot of strain between management and developers.
  - We are late, lets hire more people.
  - Design has finished now lets write the code.

Agile Software Development

- Light weight methods similar to XP e.g.
  - Crystal
  - Scrum
  - DSDM
- Include agile software development in the scope of the group.

End