Migration to Agile

ThoughtWorks

Overview

- Characteristics of software and software development
- The heuristics for building software
- How to go Agile
- XP values – one way to leverage the characteristics
- XP practices – concrete actions for an Agile process.
- Review and wrap up.

Software Characteristics

- Intangible
- Mutable
- Very difficult to define and design
- Easy to construct and run – raw machine power
- Extensively testable
- Can be deployed with partial functionality
- Development technology is changing rapidly

Heuristics for building software

- Automate everything that is repetitive

- Evolve as per needs, explore

- Build in small increments

- Test aggressively

- Get frequent feedback from actual use

- Implement in a minimal, simple way

- Avoid duplication – functionality / process

Agile values

- Communication
- Simplicity
- Feedback
- Courage

Before We Start

- People

- Team structure

- Environment

- Implementation platform

- Enabling tools (repository / build / test harness / deploy / issue tracking)

How to Implement Agile practices

1. Start small, watch benefit
2. Spot the pain
3. Bring in practices to relieve the pain
4. When it is no longer the biggest pain, repeat 2

Agile practices - i

- The planning game
- Small releases
- Metaphor
- Simple design
- Testing
- Refactoring

Agile practices - ii

- Pair programming
- Collective ownership
- Continuous integration
- 40-hour week
- On-site customer
- Coding standards

Agile Activities

- Bidding
- Scoping
- Startup
- Delivery
- Maintenance
- Enhancements

State of 'Flow'

- What is it like being 'agile'?
- Agile certification
- Investment for going agile

Recap

- Get the principles right, practices will follow.

- Aligns with the reality of development.

- A low risk path to Agile.

Thank You

Anand Joglekar

ThoughtWorks Technologies

Email: ajoglekar@thoughtworks.com

Cross dependencies in practices

- Testing ➔ simple design ➔ continuous integration
- Simple design ➔ Metaphor ➔ Refactoring ➔ Testing
- Coding standards ➔ Pair programming ➔ collective ownership
- Planning game ➔ Onsite customer
- Collective ownership ➔ Frequent releases ➔ Continuous integration