Extreme Programming

Duration: 2 Days
Available Languages: English German

Audience

Developers, Scrum Masters, Product Owners, Managers

Precondition

Participants need a basic understanding about software development in general. Being able to program is not necessary.

Goals

Learn the practices of Extreme Programming and how they benefit improving your software development process.

Contents

Extreme Programming is a development method described by Kent Beck, Ron Jeffries, Ward Cunningham and others in the late 1990ies. It is an Agile process framework like Scrum. Younger than Scrum and with its origins deeply rooted in software development, it is not a universal agile framework and can only be applied to software development. But in software development, it is regarded as one of the most successful process frameworks, because it closes the gaps that others leave around technical practices.

- History and Origins of Extreme Programming
- The Manifesto of Agile Software Development
- The Manifesto of Software Craftsmanship
- Context: Scrum, Agile, Kanban, Lean
- Risk, the four variables, cost of change, the two values of software
- Activities: Coding, Testing, Listening, Designing
- Values
    - # XP: Communication, Simplicity, Feedback, Courage, Respect
    - # Scrum: Focus, Openness, Respect, Courage, Commitment
- Feedback
    - # XP Feedback Loops
    - # XP Feedback Dimensions: System, Customer, Team
- User Stories and Backlogs
    - # User Story Template
    - # INVeST Stories
    - # SMART Tasks
    - # Acceptance Criteria
    - # BDD/ATDD overview

    # Spikes
- Process Evolution
    # Comparison with Scrum
    # Why Scrum is not enough
    # How XP turns the waterfall on its head
    # Test Automation Pyramid
    # DevOps Pyramid
- Roles
    # XP Coach
    # Onsite Customer
    # Developer
- Principles: Feedback, Assuming Simplicity, Embracing Change
- Ceremonies
    # Release
    # Iteration
    # Planning
    # Review
    # Retrospective
    # DSM
- Practices
    # Fine-scaled Feedback
        # Pair Programming
        # Planning Game
        # Test-Driven Development (TDD, BDD)
        # Whole Team
    # Continuous Process ("Automate All-the-things, Continuous All-the-things")
        # Continuous Refactoring, Design Improvement
        # Continuous Integration, Release, Delivery, Deployment
        # Small Releases
        # Automation
        # Trunk-based Development
    # Shared Understanding
        # Coding Standards
        # Collective Code Ownership
        # Simple Design
        # System Metaphor
    # Programmer Welfare
        # Sustainable Pace

Many people (including myself) consider XP to be the primary catalyst that got attention to agile methods, and superior to Scrum as a base for starting out in agile development. # Extreme Programming, Martin Fowler, Chief Scientist ThoughtWorks

Booking

Contact Siddhesh Nikude, +91-95-52572354, training@nelkinda.com