

Learning Agile: Understanding Scrum, XP, Lean and Kanban

Andrew Stellman & Jennifer Greene

Definition

Agile - a set of methods and methodologies that help a team to think more effectively, work more efficiently, and make better decisions

- agile has to be a *mindset* shared by the whole team
- Scrum, XP, Kanban, and Lean are agile schools of thought

Agile Values

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is still value in the items of the right, we value the items on the left more.

Agile Principles (Delivering the Project)

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development.
Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Agile Principles (Communicating and Working Together)

4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Agile Principles (Project Execution)

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.

Agile Principles (Constantly Improving the Team and the Project)

- 10. Simplicity—the art of maximizing the amount of work not done—is essential.
- 11. The best architectures, requirements, and designs emerge from self-organizing teams.
- 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Scrum and Self Organizing Teams

- 3 roles: Product Owner, Scrum Master, team member
 - product owner works with team to dcreate product backlog
 - scrum master helps with roadblocks and facilitates sprint reviews and retrospectives
- Sprints
- Daily Scrum (3 questions)
 - should be and opportunity to 'inspect' others' work and catch issues early, change plans for day, and implement feedback look

- 5 Values
 - Courage
 - Commitment
 - Respect
 - Focus
 - Openness
- Incremental Development - breaking project into phases
- Iterative Development - understand value software delivers looking at exactly how it will be delivered and changing course if there's a way to deliver more value
- Collective Commitment - genuinely trying to make your software more useful

- Gold-plating - over-compensating by adding unrequested features
- User Stories
 - "As a <type of user>, I want to <specific action> so that <specific result desired>."
 - should have a title
 - conditions of satisfaction (done-done)
- Story points and project velocity (burndown chart)

XP and Embracing Change

Programming Practices

- TDD
- Pair Programming

Integration Practices

- 10-minute build
- Continuous Integration

Planning Practices

- Weekly Cycle
- Stories
- Quarterly Cycle
- Slack

Team Practices

- Sit together (balance between collaboration and distraction)
- Informative workspace (e.g. project boards)

TDD and Pair Programming are like diet and exercise. We all know we should be doing it, but it's all too easy to drop them when we get busy.

XP Values

- Communication
- Simplicity
- Feedback
- Courage
- Respect

XP, Simplicity, and Incremental Design

- part of XP's goal is to build software that can be extended and changed easily
- clever vs. simple solutions
- code smells (anti-patterns)
- refactoring - leave time to pay down on tech debt after each dev iteration
- fail-fast system
- avoid monolithic design

Holistic Practices

1. Incremental Design (last responsible moment)
2. Energized Work -> autonomy; sustainable pace; 40-hour work week
3. Whole Team -> decide on things together and trust decision once it's made

Emergence - complex behavior arising from simple systems

Lean, Eliminating Waste, and Seeing the Whole

Thinking tools:

- eliminating waste - remove work that doesn't contribute to the project
- amplify learning - use feedback to improve
- decide as late as possible - have the most information
- deliver as fast as possible - understand cost of delay
- empower the team - focused and effective environment; team of energized people
- build integrity in - software makes sense to others; coherent whole
- see the whole - see all work clearly, warts and all

Options thinking - knowing the difference between things you've committed to and things you have the right to do

Set-based development - having seeveral paths to take for alternatives

Magical thinking - manager believes that any task, no matter the size or complexity can be done by the team

Eliminating waste:

- partially done work
- extra processes
- extra features
- task switching
- waiting
- motion
- defects

See the whole

- need to have objective measures to track progress
 - lead time - time between feature request and delivery
- root cause - actual reason problem is happening
 - five "why's" exercise
- deliver as fast as possible

Kanban, Flow, and Constantly Improving

Core Practices:

- Visualize
- Limit WIP
- Manage Flow
- Make process policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

The Agile Coach

- someone who helps a group adopt agile
- answers the questions to make people feel comfortable with the changes
- Shuhari for mastering something
- Meet them where they are, not where you want them to be

John Wooden's 5 Principles of Coaching

- Industriousness
- Enthusiasm
- Conditioning
- Fundamentals
- Development of Team Spirit