CALIFORNIA STATE UNIVERSITY
FULLERTON™

COLLEGE OF ENGINEERING
AND COMPUTER SCIENCE

# Advanced Software Process

## The Agile Process

*Overview*

Dave Garcia-Gomez
Faculty / Lecturer
Department of Computer Science

# Contents

- Key Practices of Agile Process
- The Agile Manifesto
- Principles behind the Agile Manifesto
- Agile Development Methods

# Key Practices of Agile Process

- Iterative development
- Risk-driven and client-driven
- Time-boxing
- Evolutionary development
- Adaptive planning
- Incremental delivery
- Evolutionary delivery
- …

# Iterative Development

Iterative development is an approach to building software (or anything) in which the overall lifecycle is composed of several iterations in sequence.

- Each iteration is a self-contained mini-project composed of activities such as requirements analysis, design, programming, and test.
- The goal for the end of an iteration is an iteration release, a stable, integrated and tested partially complete system.

http://guide.agilealliance.org/guide/iterative.html

# Risk-Driven and Client-Driven Iterative Planning

ID methods promote a combination of risk-driven and client-driven priorities.

Risk-driven iterative development chooses the riskiest, most difficult elements for the early iterations.

Client-driven iterative development implies that the choice of features for the next iteration comes from the client – whatever they perceive as the highest business value to them.

# Timeboxed Iterative Development

All the modern ID methods (including Scrum, XP, and so forth) either require or strongly advise timeboxing the iterations.

Fixed time for each iteration

**Timebox**: A timebox is a previously agreed period of time during which a person or a team works steadily towards completion of some goal. Rather than allow work to continue until the goal is reached, and evaluating the time taken, the timebox approach consists of stopping work when the time limit is reached and evaluating what was accomplished. Timeboxes can be used at varying time scales. Time scales ranging from one day to several months have been used. The critical rule of timeboxed work is that work should stop at the end of the timebox, and review progress: has the goal been met, or partially met if it included multiple tasks?

Source: http://guide.agilealliance.org/guide/timebox.html

# Evolutionary and Adaptive Development

Evolutionary iterative development implies that the requirements, plan, estimates, and solution evolve or are refined over the course of the iterations, rather than fully defined and "frozen" in a major up-front specification effort before the development iterations begin.

Evolutionary methods are consistent with the pattern of unpredictable discovery and change in new product development.

# Evolutionary and Adaptive Development

Adaptive development implies that elements adapt in response to feedback from prior work – feedback from users, tests, developers, and so on.

The intent is the same as evolutionary development, but the name suggests more strongly the feedback-response mechanism in evolution.

# Incremental Delivery

Incremental delivery is the practice of repeatedly delivering a system into production (or the marketplace) in a series of expanding capabilities.

The practice is promoted by IID and Agile methods.

Incremental deliveries are often between three and twelve months.

http://guide.agilealliance.org/guide/incremental.html

# Evolutionary Delivery

Evolutionary delivery is a refinement of the practice of incremental delivery in which there is a vigorous attempt to capture feedback regarding the installed product, and use this to guide the next delivery.

# Agile Development

Agile development methods apply timeboxed iterative and evolutionary development, adaptive planning, promote evolutionary delivery, and include other values and practices that encourage agility

– rapid and flexible response to change.

# The Agile Manifesto

- 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 value in the items on the right, we value the items on the left more.

# Principles behind the Agile Manifesto

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 time scale.
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.

# Principles behind the Agile Manifesto

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.
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.

# Agile Development Methods

- Scrum
- XP
- Evo
- Crystal
- Agile Modeling
- (Agile) Unified Process (Agile UP)
- Dynamic Solutions Delivery Model (DSDM)
- Feature-Driven Development (FDD)
- Lean Development
- …

# References


- Craig Larman. Agile & Iterative Development: A Manager's Guide. Addison-Wesley, Pearson Education, 2004. (11th Printing, Aug. 2009) (ISBN-10: 0-13-111155-8, ISBN-13: 978-0-13-111155-4)
- Agile Alliance: http://www.agilealliance.org/
  - http://www.agilemanifesto.org/
  - http://www.agilemanifesto.org/principles.html
  - http://guide.agilealliance.org/subway.html
- Agile Modeling and RUP – Scott Ambler: http://www.agilemodeling.com/
- Agile, XP Method – Martin Fowler: http://www.martinfowler.com/
- Agile, XP Method – Robert Martin: http://www.objectmentor.com/
- Extreme Programming – Don Well: http://www.extremeprogramming.org/
- Extreme Programming – Ron Jeffries: http://xprogramming.com/index.php
- Agile Development – Artern Marchenko : http://agilesoftwaredevelopment.com/