AGILE SOFTWARE DEVELOPMENT

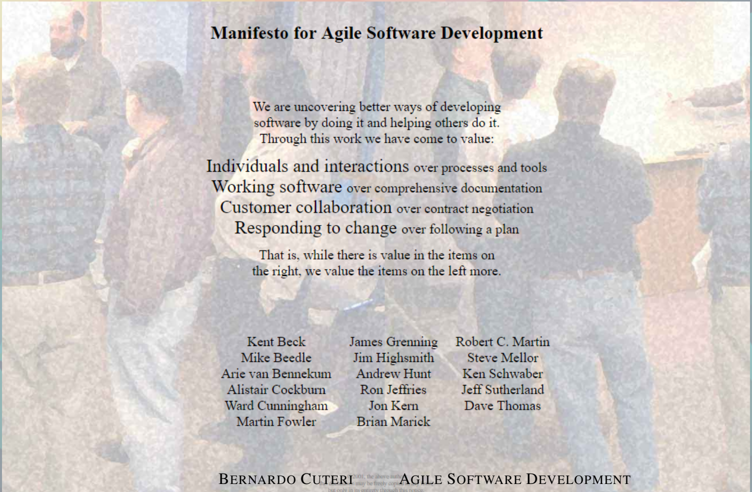
Bernardo Cuteri

What is Agile

- a group of software development methodologies based on iterative development
- requirements and solutions evolve through collaboration between self-organizing cross-functional teams
- encourages frequent inspection and adaptation
- intended to allow for rapid delivery of high-quality software in an iterative way
- Agile development refers to any development process that is aligned with the concepts of the Agile Manifesto

THE AGILE MANIFESTO

Agile has been popularized by the agile manifesto. The Manifesto was developed by a group fourteen leading figures in the software industry, and reflects their experience of what approaches do and do not work for software development.



Manifesto for Agile Software Development

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

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Principles behind the Agile manifesto

These are the principles of 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 timescale.
4. Business people and developers must work together daily throughout the project.

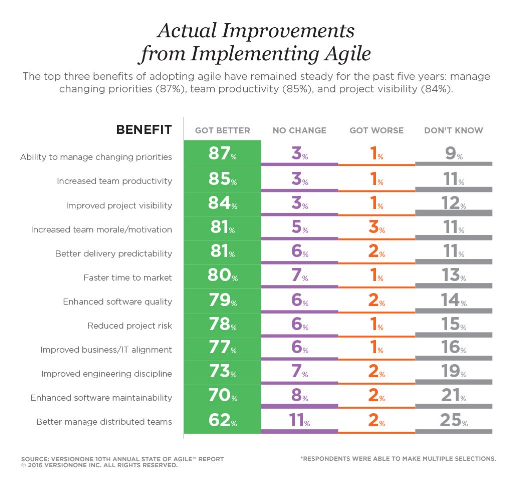
Principles behind the Agile manifesto (cont.)

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.
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.

PRINCIPLES BEHIND THE AGILE MANIFESTO (CONT.)

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.

Reported improvements from implementing Agile



Actual Improvements from Implementing Agile

The top three benefits of adopting agile have remained steady for the past five years: manage changing priorities (87%), team productivity (85%), and project visibility (84%).

BENEFIT	GOT BETTER	NO CHANGE	GOT WORSE	DON'T KNOW
Ability to manage changing priorities	87%	3%	1%	9%
Increased team productivity	85%	3%	1%	11%
Improved project visibility	84%	3%	1%	12%
Increased team morale/motivation	81%	5%	3%	11%
Better delivery predictability	81%	6%	2%	11%
Faster time to market	80%	7%	1%	13%
Enhanced software quality	79%	6%	2%	14%
Reduced project risk	78%	6%	1%	15%
Improved business/IT alignment	77%	6%	1%	16%
Improved engineering discipline	73%	7%	2%	19%
Enhanced software maintainability	70%	8%	2%	21%
Better manage distributed teams	62%	11%	2%	25%

SOURCE: VERSIONONE 10TH ANNUAL STATE OF AGILE™ REPORT

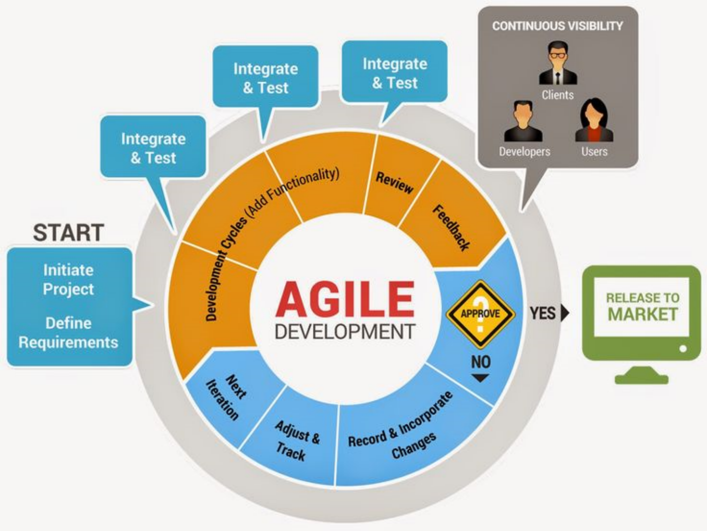
*RESPONDENTS WERE ABLE TO MAKE MULTIPLE SELECTIONS.

VersionOne releases 10th Annual State of Agile Survey Results (April 2016)

What Agile is NOT

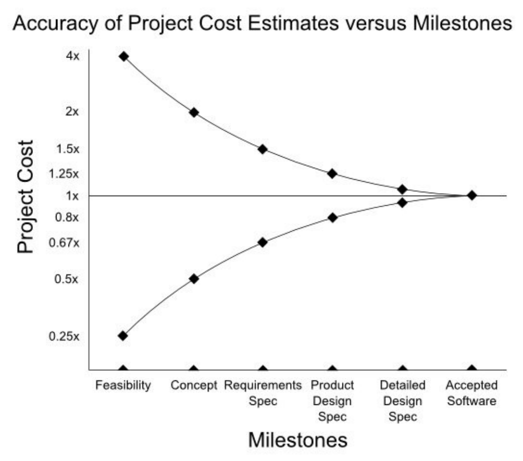
- it is NOT a silver bullet
- it is NOT for every team
- it is NOT a programming style
- it is NOT the only development methodology teams use
- it is NOT just waterfall, extreme programming, coding standards or scrum alone, it is a combination of these (and more) methods together.

AGILE DEVELOPMENT CYCLE



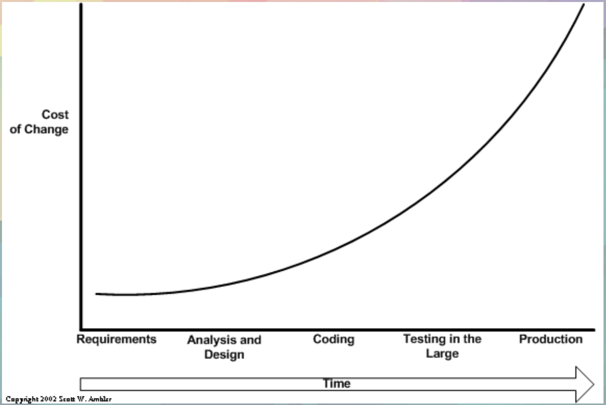
PROBLEMS MADE EASIER WITH AGILE

Estimates created early in the project are subject of high degree of error



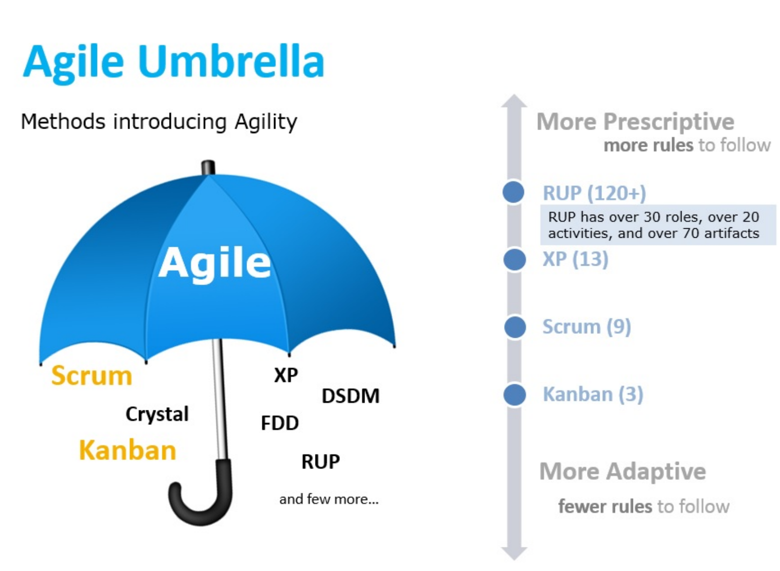
Problems made easier with Agile (cont.)

The cost of changes increases as the product progresses from requirements to production



Copyright 2002 Scott W. Ambler

AGILE IMPLEMENTATIONS



Agile Umbrella

Methods introducing Agility

More Prescriptive

more rules to follow

RUP (120+)

RUP has over 30 roles, over 20 activities, and over 70 artifacts

XP (13)

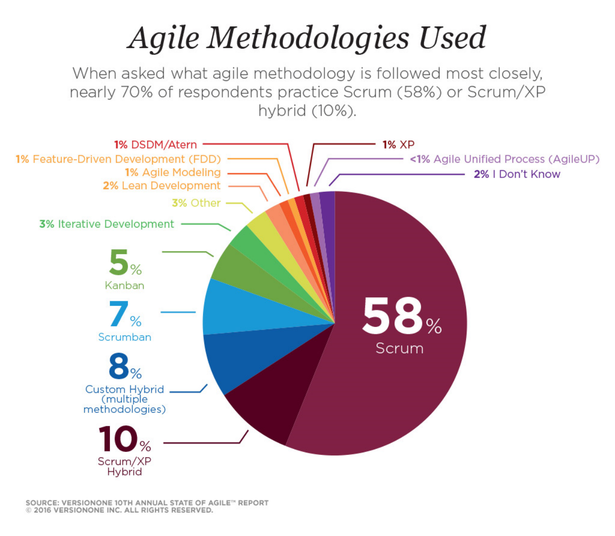
Scrum (9)

Kanban (3)

More Adaptive

fewer rules to follow

AGILE IMPLEMENTATIONS



Agile Methodologies Used

When asked what agile methodology is followed most closely, nearly 70% of respondents practice Scrum (58%) or Scrum/XP hybrid (10%).

SOURCE: VERSIONONE 10TH ANNUAL STATE OF AGILE™ REPORT
© 2016 VERSIONONE INC. ALL RIGHTS RESERVED.

VersionOne releases 10th Annual State of Agile Survey Results (April 2016)

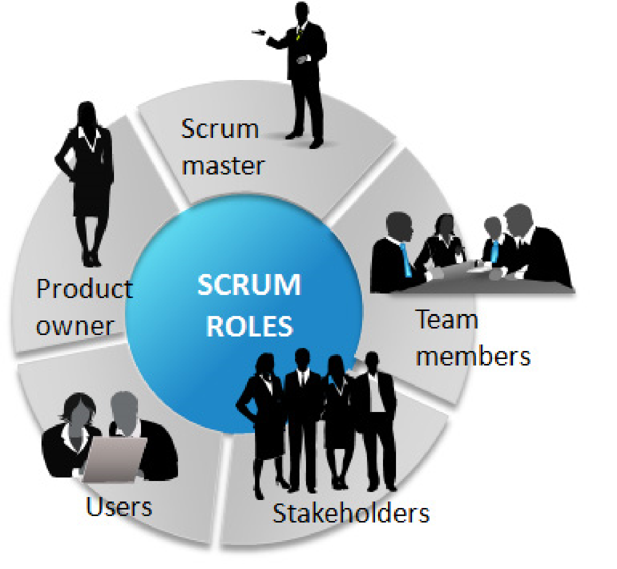
SCRUM: AN IMPLEMENTATION OF AGILE DEVELOPMENT

- A **light-weight** agile process tool
- Split the organization into small, **cross-functional**, **self-organizing** teams
- There are three main organization roles: **the developer team**, **the scrum master** and **the product owner**
- Split your work into a list of small, concrete deliverables. Sort the list by priority and estimate the relative effort of each item

Scrum: an implementation of Agile development (cont.)

- Split the time into short fixed-length iterations/**sprints** (usually 2-4 weeks), with potentially shippable code demonstration after each iteration
- Optimize the release plan and update priorities in collaboration with the customer
- Optimize the process by having **retrospective** after each iteration

SCRUM ROLES


SCRUM ROLES
Scrum master
Team members
Stakeholders
Users
Product owner

SCRUM IN A NUTSHELL

- **The product owner** creates a list of features to be implemented: **the backlog**.
- Features are split and described in terms of **user stories** and **epics**(sets of US).
- The scrum team **estimates** the work associated with each story.
- Features are ranked in order of importance.
- Most important features are moved to the **Sprint** (2/3 weeks)
- **Every day** there is a stand-up meeting of about 15 minutes (**the scrum meeting**) to discuss three things:
  - What did you do yesterday?
  - What will you do today?
  - Is there any obstacle?
- The **scrum master** helps in solving problems of the scrum team.
- When the Sprint ends, work is presented to PO and/or stakeholers **Sprint review**
- The sprint process is discussed and improved for the next iteration **Sprint retrospective**