# **Agile Manifesto**

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

While there is value in the items on the right, we value the items on the left more.

## The 12 Principles of Agile

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a 3 couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of 6 conveying information to and within a development team is face-to-face conversation.

### Advancing the principles of Agile



#### Learn more at AgileAlliance.org

#### THE MANIFESTO AUTHORS

Kent Beck Mike Beedle Arie van Bennekum

Alistair Cockburn Ward Cunningham Martin Fowler

Robert C. Martin Steve Mellor Dave Thomas

James Grenning Jim Highsmith Andrew Hunt

Ron Jeffries Jon Kern Brian Marick

Ken Schwaber Jeff Sutherland

©2001-2019 The Agile Manifesto Authors. This declaration may be freely copied in any form, but only in its entirety through this notice.

Working software is the primary measure of progress.

Agile processes promote sustainable 8 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.



Simplicity - the art of maximizing the amount of work not done – is essential.



The best architectures, requirements, and designs emerge from self-organizing teams.



At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.