

Agile Manifesto

Traditional method did not allow developers to be responsive enough to market's dynamic nature.

Development teams started exploring alternatives to outdated methods.

They noticed some common themes that produce better results.

In Feb 2011, they shared their experiences, ideas, practices & suggest ways to improve.

- ① Agile Manifesto → focus on People, Comm's, Product, Flexibility
- ② Agile Principles
- ③ Agile Alliance.

Ref. to figures for manifesto & details.

- » **The Agile Manifesto:** An intentionally streamlined expression of core development values.
- » **The Agile Principles:** A set of 12 guiding concepts that support agile project teams in implementing agile techniques and staying on track.
- » **The Agile Alliance:** A community development organization focused on supporting individuals and organizations that are applying agile principles and practices.

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.

* Agile Manifesto Copyright © 2001: 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.
This declaration may be freely copied in any form, but only in its entirety through this notice.

Why agile values are so important:

- » **Individuals and interactions over processes and tools:** Why? Because research shows a 50 times increase in performance when we get individuals and interactions right. One of the ways we get this right is by collocating a development team with an empowered product owner,
- » **Working software over comprehensive documentation:** Why? Because failure to test for and correct defects during the sprint can take up to 24 times more effort and cost in the next sprint. And after the functionality is deployed to the market, if a production support team that wasn't involved in product development performs the testing and fixing, the cost is up to 100 times more.

- » **Customer collaboration over contract negotiation:** Why? Because a dedicated and accessible product owner can generate a fourfold increase in productivity by providing in-the-moment clarification to the development team, aligning customer priorities with the work being performed.
- » **Responding to change over following a plan:** Why? Because 64 percent of features developed under a waterfall model are rarely or never used (as discussed in Chapter 1). Starting with a plan is vital, but that is when we know the least. Agile teams don't plan less than waterfall teams — they plan as much or more. However, agile teams take a just-in-time approach, planning just enough when needed. Adaptation of the plan to the realities along the way is how agile teams deliver products that delight customers.

Four Values of the Agile Manifesto

① Value-1: Individuals & Interactions Over Processes & Tools

⇒ Managed & Controlled communications are often ambiguous, and time consuming and distract development teams from the work of creating a product.

ex. - emails, spreadsheets, documents.

⇒ Increased overhead cost & delays

⇒ Refer to Table 2-1

✓ TABLE 2-1

Individuals and Interactions versus Processes and Tools

	Individuals and Interactions Have High Value	Processes and Tools Have High Value
Pros	<ul style="list-style-type: none"> Communication is clear and effective. Communication is quick and efficient. Teamwork becomes strong as people work together. Development teams can self-organize. Development teams have more chances to innovate. Development teams can customize processes as necessary. Development team members can take personal ownership of the project. Development team members can have deeper job satisfaction. 	<ul style="list-style-type: none"> Processes are clear and can be easy to follow. Written records of communication exist.
Cons	<ul style="list-style-type: none"> Development team members must have the <i>capacity</i> to be involved, responsible, and innovative. People may need to let go of ego to work well as members of a team. 	<ul style="list-style-type: none"> People may over-rely on processes instead of finding the best ways to create good products. One process doesn't fit all teams — different people have different work styles. One process doesn't fit all projects. Communication can be ambiguous and time-consuming.



Refer to Table 2-1



We use processes & tools in Agile Development;
to ~~do~~.



Processes & tools in Agile are intentionally streamlined
& directly support product creation.



Agile environment is human centric & participatory.

② Value 2: Working Software Over Comprehensive Documentation.

⇒ Working software means the S/W meets "Definition of done"

⇒ S/W Developed, tested, integrated, documented.

⇒ 75% done with your Project?

⇒ Customer ran out of money, Can we have our 75% now?

⇒ On Traditional Project ⇒ No Working S/W; 75% Progress 40% done

⇒ On Agile Project ⇒ Working potentially shippable functionality for 75%
Requirements

Highest priority 75% requirements ⇒ Done

⇒ Ref. Table - 2-2 Examples of Traditional Project docs & their usefulness

TABLE 2-2

Identifying Useful Documentation

Document	Does the Document Support Product Development?	Is the Document Barely Sufficient or Gold-Plated?
Project schedule created with expensive project management software, complete with Gantt Chart.	No. Start-to-finish schedules with detailed tasks and dates tend to provide more than what is necessary for product development. Also, many of these details change before you develop future features.	Gold-plated. Although project managers might spend a lot of time creating and updating project schedules, project team members tend to want to know only key deliverable dates. Management often wants to know only whether the project is on time, ahead of schedule, or behind.
Requirements documentation.	Yes. All projects have requirements — details about product features and needs. Development teams need to know those needs to create a product.	Possibly gold-plated; should be barely sufficient. Requirements documents can easily grow to include unnecessary details. Agile approaches provide simple ways to describe product requirements.
Product technical specifications.	Yes. Documenting how you created a product can make future changes easier.	Possibly gold-plated; should be barely sufficient. Agile documentation includes just what it needs — development teams often don't have time for extra flourishes and are keen to minimize documentation.
Weekly status report.	No. Weekly status reports are for management purposes but do not assist product creation.	Gold-plated. Knowing project status is helpful, but traditional status reports contain outdated information and are much more burdensome than necessary.
Detailed project communication plan.	No. Although a contact list can be helpful, the details in many communication plans are useless to product development teams.	Gold-plated. Communication plans often end up being documents about documentation — an egregious example of busywork.

classmate
Date _____
Page _____

③ Value 3 Customer Collaboration over Contract Negotiations

- ⇒ Customer is not enemy
- ⇒ Ref. fig
- ⇒ Collaboration rather than confrontation.
- ⇒ Focus on negotiation discourages valuable customer input, and can create adversarial relationship between customer and project teams
- ⇒ In Agile, we experience a Partnership between customer & development team.
- ⇒ Discovery, questioning, learning & adjusting during course of project.

- » **Start of a project:** When the customer and the project manager — or another project team representative — negotiate contract details.
- » **Any time the scope changes during the project:** When the customer and the project manager negotiate changes to the contract.
- » **End of a project:** When the project team delivers a completed product to the customer. If the product doesn't meet the customer's expectations, the project manager and the customer negotiate additional changes to the contract.

Value ∇ : Responding to change over following a plan.

- Change is a valuable tool

- Rigorous change management procedures & budget structures make changes difficult.

Traditional project teams often blindly follow a plan,

Misses opportunities to create more valuable products.

Ref fig. Relationship between time, opportunity for change & of change.

Agile project accommodate change systematically.

- Change is predictable & manageable.

It increases project stability.

