12 principles of Agile Manifesto

Outline

1. 1st category: Delivering Working Software
2. 2nd category: on flexible design and adapting to change
3. 3rd category: Principles about collaborative communication and organization

12 principles of Agile Manifesto

12 principles are to

- narraw down the four broad agile value statements
- elaborate on the values and give more specific guidelines to follow

12 principles in three categories

- Delivering Working Software
- Flexible Design and Adapting to Change
- Collaborative Communication and Organization

First category: Delivering Working Software

Working software is a key aspect of Agile

- focusing on "delivering working software" drives development
- Even in the early phases of development, each delivery of working software becomes a working prototype that will be improved upon in the next round of development.
- A working piece of software satisfies the client

-> let them see demonstration of their product throughout the entire process.

1st category: Delivering Working Software

First principle:

"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software".

- "valuable"

-> ask your client what they determine as valuable.

-> the features that your client values the most become the highest priority for your development team.

Second principle, on the frequency of deliveries:

"Deliver working software frequently, from a couple of weeks to a couple of months with a preference to the shorter timescale"

The more frequent your software delivery is,
the more opportunity your client has to provide input on development.

1st category: Delivering Working Software

Third principle:

"Working software is the primary measure of progress"

- the focuse of the team: working software, so is the progress measure
- "working" – completeness
  - a feature should only be considered complete when all the code has been written, tested and documented.
  - a product manager should make sure that the dev team is completing features instead of having many started, but few that are actually done.

-> but be change-friendly

1st category: Delivering Working Software

Which of the following do you think are good ways to make your software product more change friendly?

- A. Frequent client communication.
- B. Neat, commented source code.
- C. Continuously reviewing and improving your project.
- D. Updated, prioritized list of features.
- E. Development team that is open to change.

Second category: on flexible design and adapting to change

Change is inevitable

- plan your project in such a way that change is not detrimental.
- A good way to make your product more responsive to change is to have a *flexible design and an adaptable process*.

2nd category: on flexible design and adapting to change

4. Principle about changing requirements

"Welcome changing requirements, even late in development."

- Agile processes harness change for the customer's competitive advantage.
- Change will ultimately create a better product.
- It'll satisfy your client, and it'll give your product that competitive edge.

2nd category: on flexible design and adapting to change

5. Principle on good development

"Continuous attention to technical excellence and good design enhances agility."

- Having readable, simple code, and more flexible designs, will allow changes to be easily implemented.
- Having a good design can let you know what components are dependent on each other.

2nd category: on flexible design and adapting to change

6. Principle on sustainable development

Agile processes promote sustainable development:
»"The sponsors, developers, and users should be able to maintain a constant pace indefinitely."

Familiar with the unfortunate development push to get things completed just before a deadline?

- This inconsistent pace leads to burnt out development teams.
- This will have a negative impact on project success.

2nd category: on flexible design and adapting to change

7. Principle on simplicity

"Simplicity, the art of maximizing the amount of work not done, is essential."

- ✗ delivering less.
- ✓ delivering what is essential, and reducing unnecessary work.
  - writing less code and documentation
  - focusing on delivering a high impact product that is as simple as it can be.

2nd category: on flexible design and adapting to change

Quiz context:

Sam is one of the developers on your development team. He is obsessed with the Agile concept of simplicity.

- The software code he writes is very minimal.
- It does the work it is supposed to, but without any unnecessary or complicated code.
- He also doesn't add comments to document his code.
- Sam thinks that the team should only develop the essential features that make the product function, and they shouldn't waste their time on flashy features.
- He refuses to participate in elaborate documentation but will write notes on how he developed the feature, any key notes, and a brief training document for the end user.

2nd category: on flexible design and adapting to change

Which of his practices if any do you think actually follow the concept of simplicity from the Agile Manifesto?

- A. Minimal code that satisfies the desired functionality.
- B. no comments in the code.
- C. only developing essential features.
- D. essential documentation over elaborate documentation.

2nd category: on flexible design and adapting to change

Which of his practices if any do you think actually follow the concept of simplicity from the Agile Manifesto?

- ✓ A. Minimal code that satisfies the desired functionality.
- ✗ B. no comments in the code.
- ✗ C. only developing essential features.
- ✓ D. essential documentation over elaborate documentation.

Sam's view of simplicity vs what the manifesto suggests

- ✓ A. Minimal code that satisfies the desired functionality.
- Minimal code is a great way to focus on simplicity.
- ✗ B. no comments in the code.
- Comments in the code are useful to explain the rationale for a particular solution.

2nd category: on flexible design and adapting to change

Sam's view of simplicity vs what the manifesto suggests

✗ C. only developing essential features.

- The team should develop features not based on their own criteria of what is important, but what is going to satisfy the client. (If flashy features is what they want, then that's what they should get.)

✓ D. essential documentation over elaborate documentation.

- His documentation style is a great way to get essential information to the right people.
  - There may not be a need for comprehensive documentation.
  - Short and simple documentation might encourage people to actually read it.

3rd category about collaborative communication and organization

Principles about collaborative communication and organization

- Communication is essential for any team creating great software.
- Teams can be organized to facilitate better communication and drive great software development.

3rd category about collaborative communication and organization

Eighth principle on trust

"Build projects around motivated individuals. Give them the environment and support that they need, and trust them to get the job done."

- A product manager might need to take a step back.
- it is important to act as more of a facilitator for your team.

"Tell people where to go, but not how to get there, and you'll be amazed by the results. – George S. Patton"

3rd category about collaborative communication and organization

Ninth principle about self-organization

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

- Agile encourages teams to self-organize.
- As a team they decide how they want to organize the project.
    - assigning tasks
    - choosing tools to use.
    - ...

3rd category about collaborative communication and organization

Tenth principle about client-developer communication

"Business people and developers must work together daily throughout the project."

A product manager should act as

- a facilitator
- but never as a messenger between the two parties

3rd category about collaborative communication and organization

Eleventh principle about communication

"The most efficient and effective method for conveying information to and within a development team is face-to-face conversation."

To reduce miscommunication and to increase the work speed

- speak directly to a project team member – the most effective
- have frequent, short, in-person meetings for co-location team
- unable to have in-person communication, use communication tools such as video conferencing for global team

3rd category about collaborative communication and organization

The final principle about reviewing what's been done

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

- Your process is very much like the product.
- Tests and reviews should be made to improve it.

Imporve the process

"The important thing is not your process. The important thing is your process for improving your process. – Henrik Kniberg"

- Processes are very similar to writing code.
- A process that is perfect the first time is equivalent to a program that has no bugs when you first compile it.
- It's impossible.

3rd category about collaborative communication and organization

Quiz context

You are working for a global corporation that develops applications for airlines. Your latest project is managing the development of an entertainment app that replaces the television screens on airlines. Your development team is very experienced with development, as well as with Agile. They self-organize the team.

What does this self-organizing team actually look like? Check all that apply.

- A. there is no work for the product manager to do.
- B. they have agreed upon certain practices to follow.
- C. they've decided to self-assign as they complete tasks.
- D. they have appointed one leader who is in charge of the team.

3rd category about collaborative communication and organization

What does this self-organizing team actually look like? Check all that apply.

- A. there is no work for the product manager to do.
- ✓ B. they have agreed upon certain practices to follow.
- ✓ C. they've decided to self-assign as they complete tasks.
- D. they have appointed one leader who is in charge of the team.

3rd category about collaborative communication and organization

Explanation: B and C are great ways that teams can self-organize

✓ B. they have agreed upon certain practices to follow.

✓ C. they've decided to self-assign as they complete tasks.

- Self-organizing teams are supposed to encourage communication, teamwork, efficient development, motivation, and respect.
- Appointing someone as a leader does not put everyone at the same level of respect, and it does not encourage equal responsibility.
- A self-organizing team also does not mean that there is nothing for a manager to do.
- He needs to manage and coach team member in the practices chosen.

3rd category about collaborative communication and organization

Agile project vs non Agile one

- Both could go astray or even fail.
- The Agile project was failing early, and we were able to rescue it.
- With the non Agile project, we may not know the project was failing until we delivered.

More about Agile Manifesto: `agilemanifesto.org`.