

Agile

software development

1

What is Agile?

Issues with **traditional team management** methods

- Plan for up to a year of development, but circumstances and needs change
- Project managers have direct control over development
- Clients are not involved in process
- Strict long-duration waterfall structure, usually can't go back
(Ex: planning -> design -> programming)

Agile is a software development methodology to build a software **incrementally** using **short iterations** of 1 to 4 weeks so that the development is aligned with the changing business needs.

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

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

Important aspects

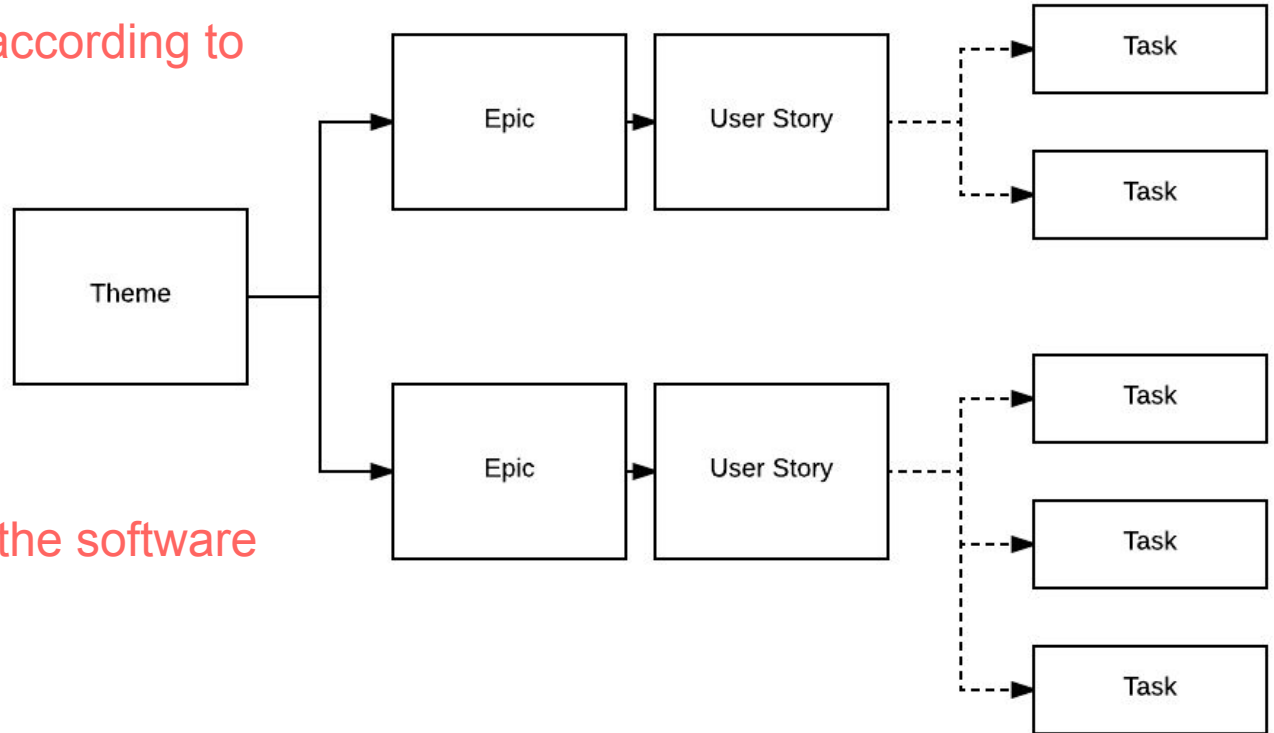
- Accept **change**
- Accept feedback
- **Repeat** what is difficult
- Emphasis on creation of **value**
- Team takes **responsibility** of the product
- Timeboxing of work into **sprints**

2

What does an
Agile workflow look like?

Main Task Distribution

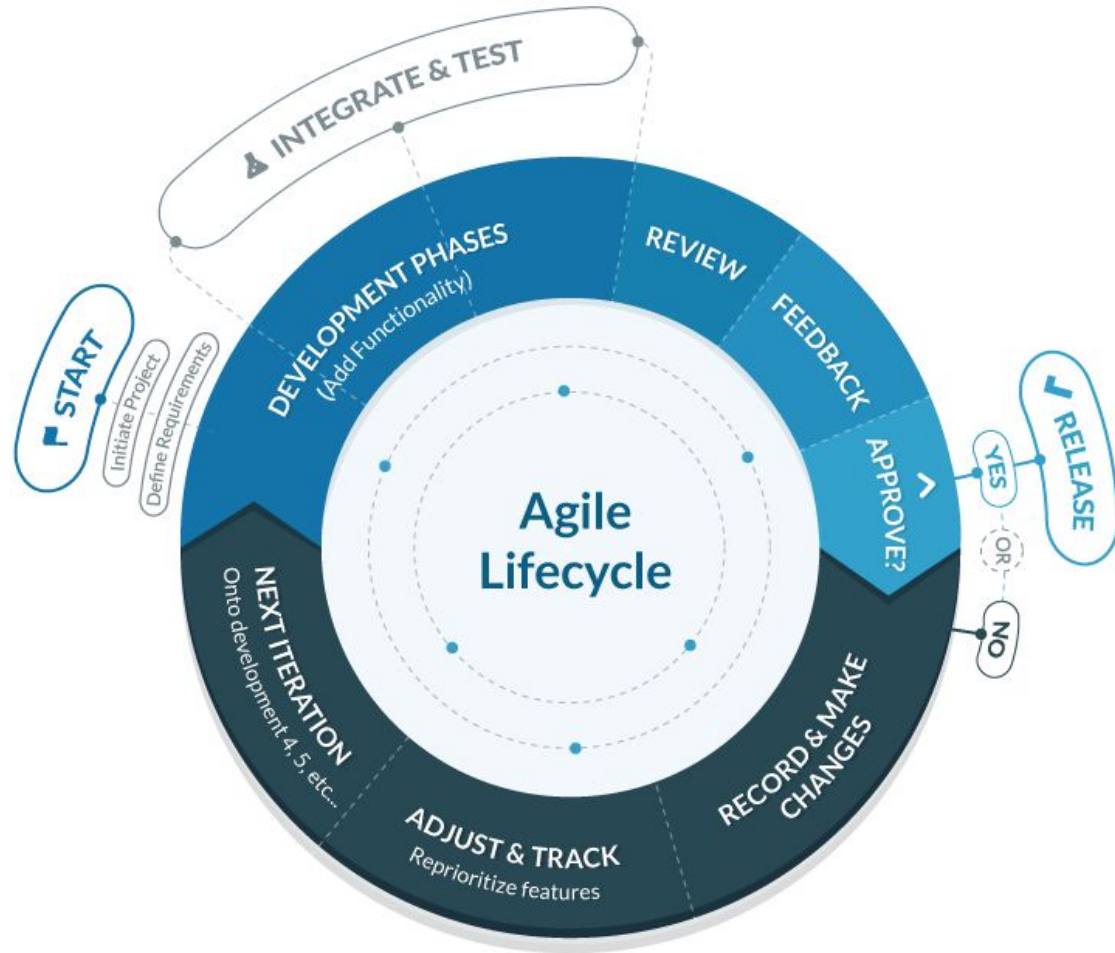
- Composed of several themes
- Tasks are sorted according to Epic & User Story



- Theme: aspect of the software

Useful Tools and Elements

- Define **requirements**
- Testing during development
- **Review** & commit
- Risk vs value?
- Next iteration



the SCRUM SOFTWARE DEVELOPMENT PROCESS



INPUTS FROM CUSTOMERS,
TEAM, MANAGERS & EXECS.



PRODUCT OWNER



THE TEAM



PRODUCT
BACKLOG



SPRINT
PLANNING
MEETING



SPRINT
BACKLOG



SPRINT MASTER



DAILY STAND UP
MEETING



SPRINT REVIEW



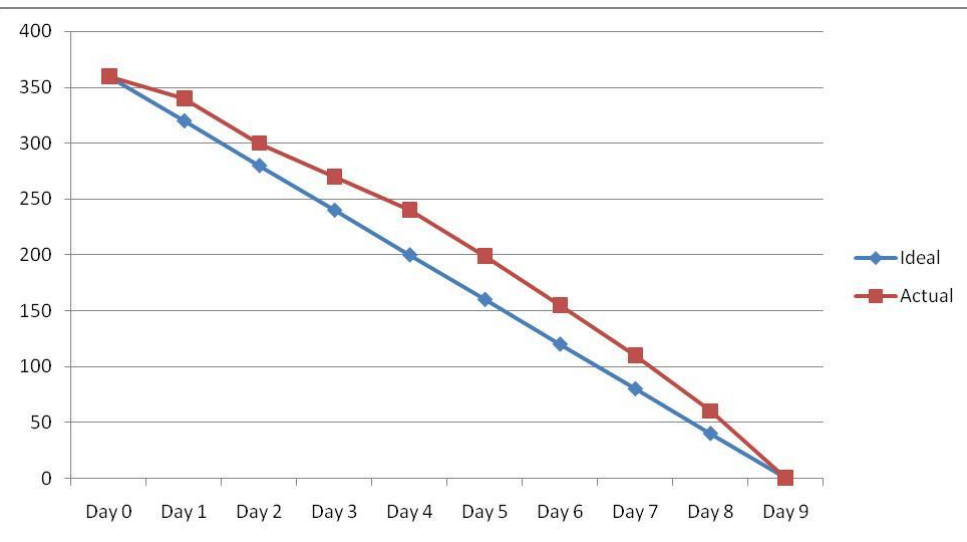
FINISHED WORK



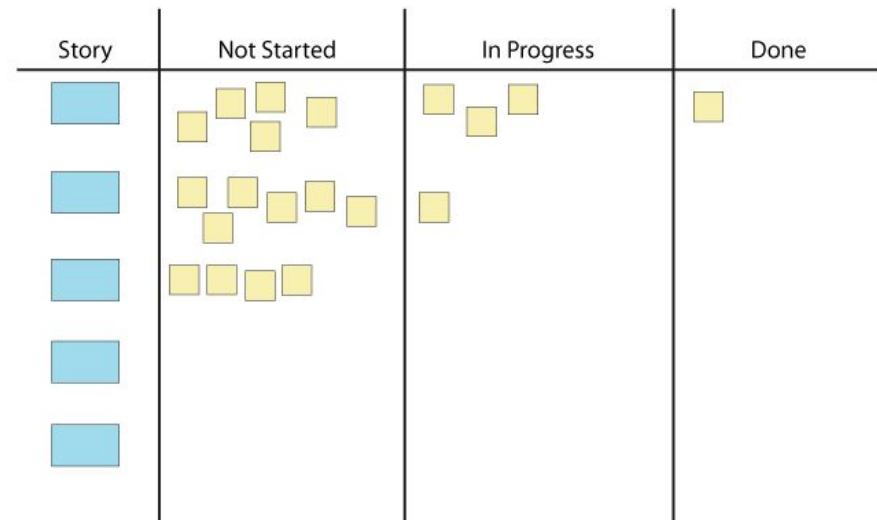
SPRINT
RETROSPECTIVE

Tools: Charts

Burn-down Chart



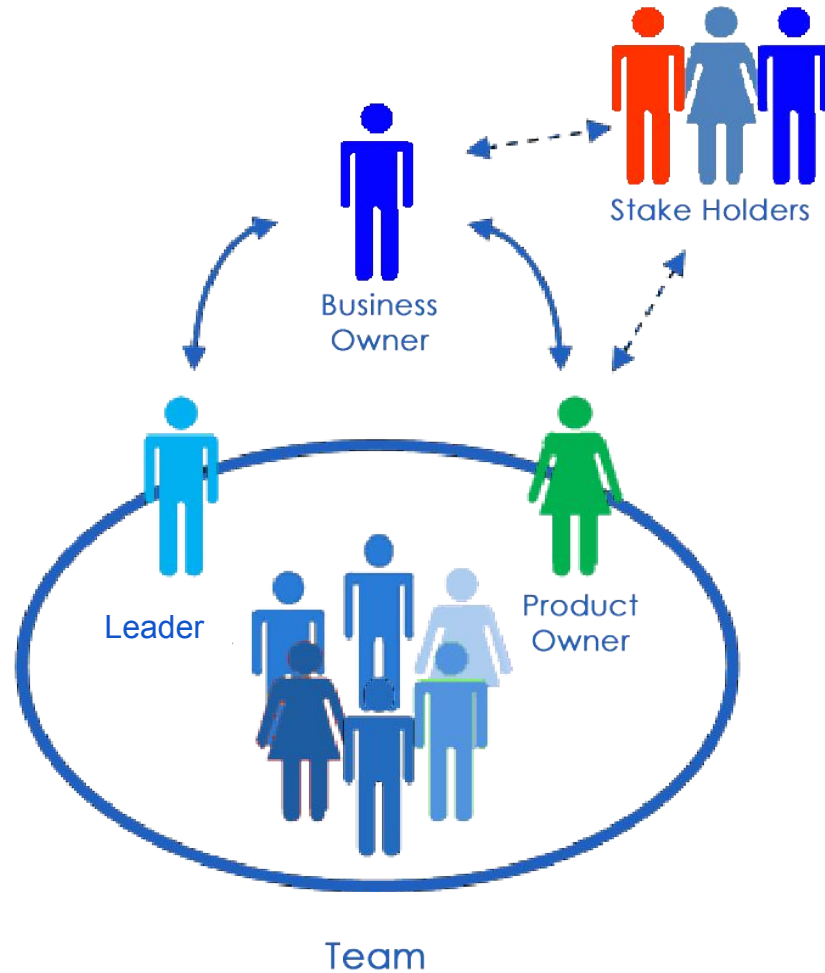
SCRUM Board



3

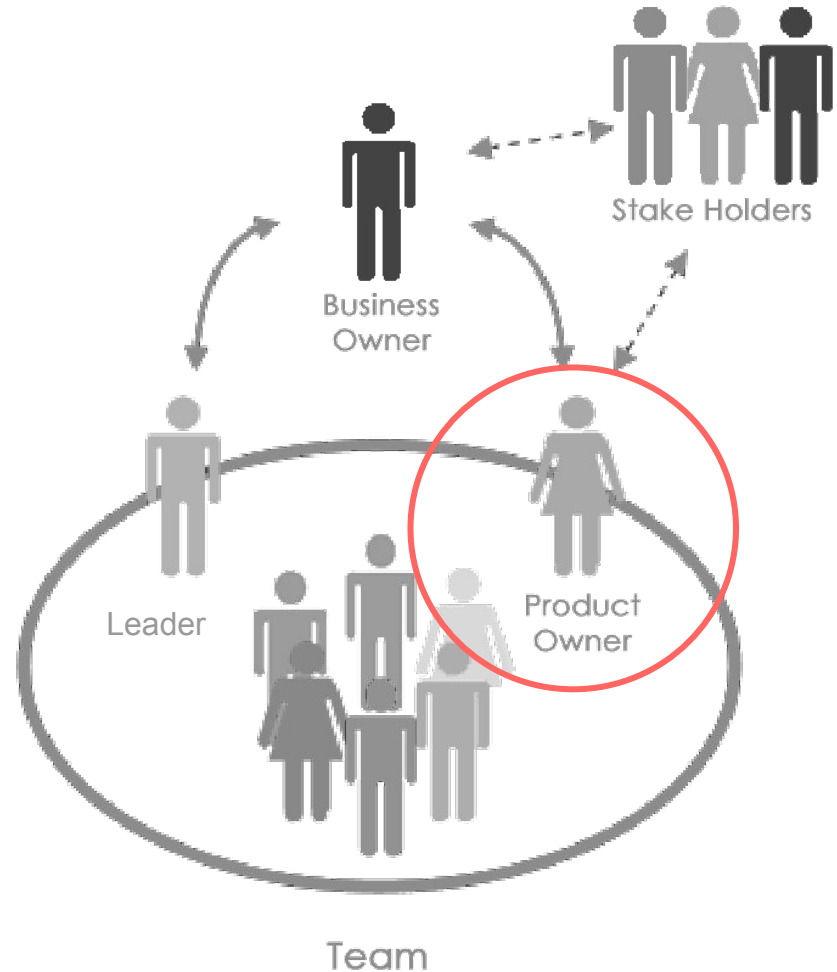
What is the hierarchy
of an Agile project?

Hierarchy



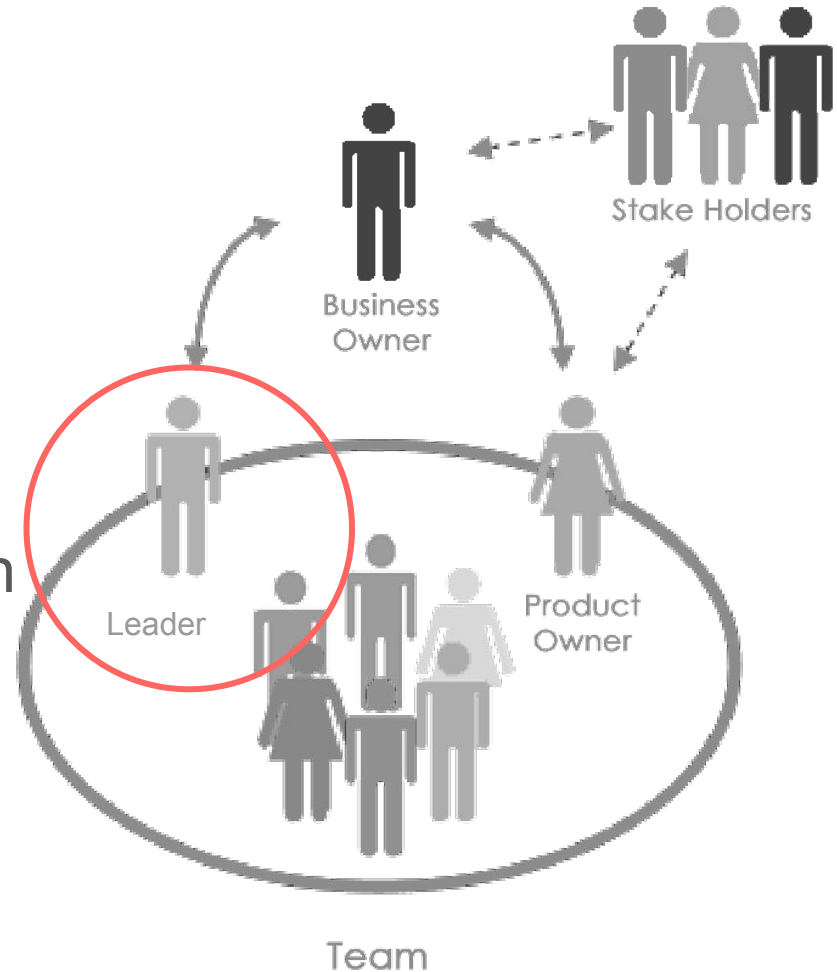
Product owner

- Set the **priorities**
- Define the **requirements** and Product Backlog
- **Represents** the **client**
- Prioritizes and prepares **user stories** for readiness



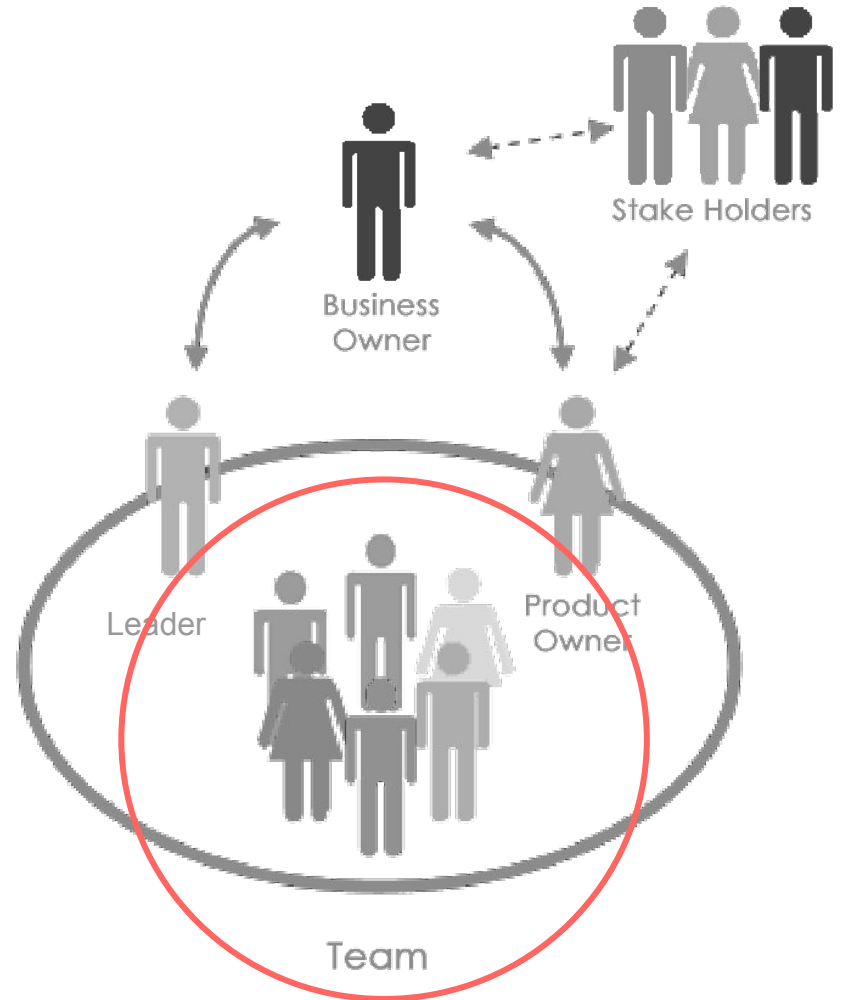
Leader

- Oversee the process
- Removes **impediments** to success.
- **Remind** the team of mission and value statements



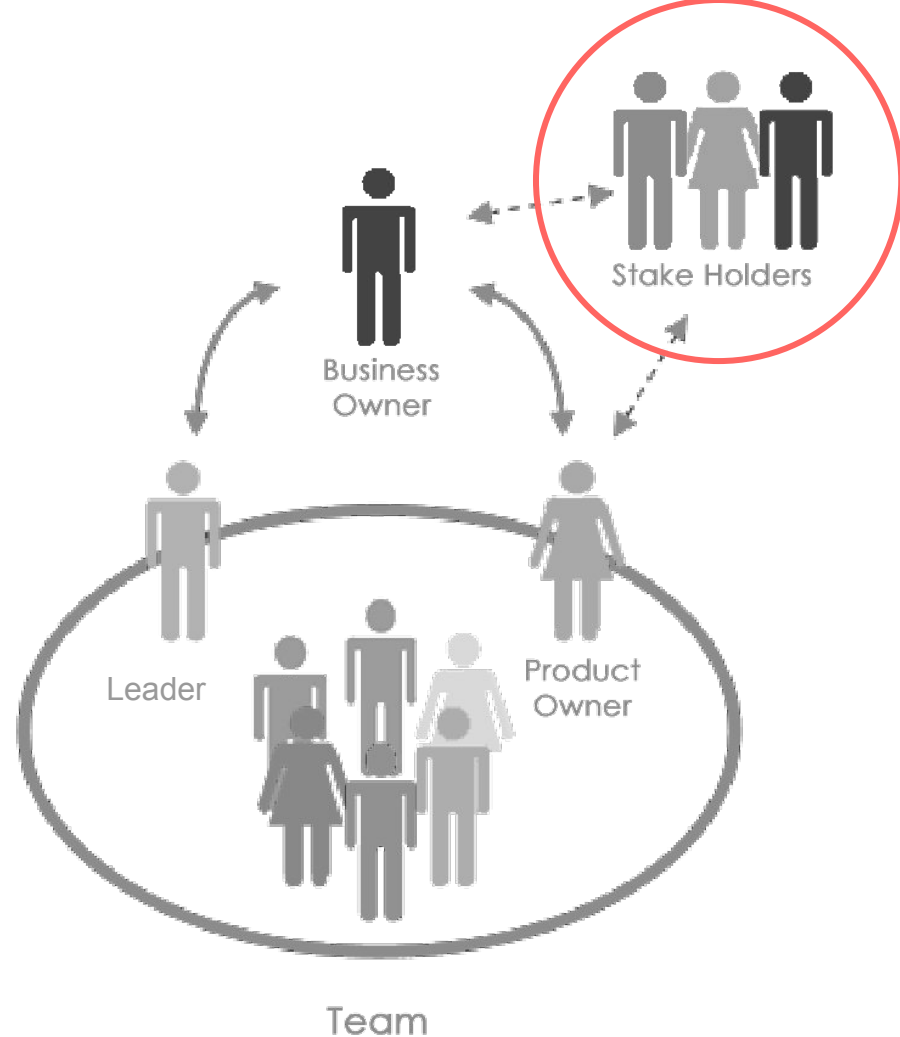
Agile team

- **Build** the product
- Work **together** to make the best of their talents
- **Adapt** quickly to new requirements



Stakeholders

- Set the right **requirements** with interested parties
- Get daily **feedback**
- **Fund** the project



4

What are the criticisms? Why is it relevant for us?

Main criticisms

- Hard to implement in large teams
- Misleading terminology
- Remove sense of accomplishment
- Generate anxiety every iteration

Relevance

Why should we use an agile development process instead of a waterfall method?

- Adaptability over predictivity
- Iterative over waterfall
- Code over documentation