

# How to implement Scrum

and FAQ

# Phases of implementing Scrum

## Phases I: Follow the workflow of Scrum strictly

- Scrum is simple, so don't omit any part of Scrum to make it simpler.
- Forget what you have known.
- Support from the managerial is necessary.

## Phase II: Adjust according to your situation

- Find the iteration cycle for your team.
- Find the development for your team.
- Create the release rhythm for your product.

# Transformation From Traditional to Scrum

- From waterfall to iterative development
- Fix iteration cycle and no arbitrary change to stories
- From project manager to Scrum Master
- From product manager to Product Owner
- From project members to team members
- From requirement to user story
- The team decompose stories to tasks
- From task assigning to task claiming
- From Gantt Chart to Burndown Chart
- From individual assessment to team performance assessment

# Scrum Meetings

- Sprint Planning (Part 1): prioritize stories based on I/O.
- Sprint Planning (Part 2): decompose stories and claim tasks.
- Daily Scrum: time-fixed briefing and no problem solving.
- Sprint Review: demo and feedback.
- Sprint Retrospective: improve Scrum practice step by step.

# Scrum Practice

- Pair programming
- Coding standards
- Source code management
- Code review
- Daily commit
- Cross test
- Code refactoring
- Sharing meeting
- Simple design
- Automation test
- Framework

# Scrum FAQ

# Questions

- How to write a user story?
- Is a prototype necessary?
- Is a detailed design necessary?
- How to decide the priority of user stories?
- Can I add stories to a sprint?
- What if no acceptance criteria is defined for stories?
- What if not involved in development?

# How to write a user story?

- As a (Role), what needs to be done(Goal), so that (Value or Reason).
- Define acceptance criteria.
- Follow *INVEST* rules,
  - *Independent*, a story should not be dependent on other stories.
  - *Negotiable*, a story in Scrum is not a contract in Waterfall. Story shouldn't be too detailed and developers can offer some advice.
  - *Valuable*, a story should represent the value for users.
  - *Estimable*, a story should be estimated on its tasks.
  - *Sized right*, a story should be minimized and done in one Sprint (two weeks).
  - *Testable*, a story should have an interface to test and be automated tested. Each task should do JUnit Test.



# Is a prototype necessary?

- A prototype is not a must in Scrum.
- A prototype can be a supplement to user story, for it is more intuitive.

# Is a detailed design necessary?

- The answer is no.
- The detailed design and the product specifications should be decomposed into user stories.
- Reasons:
  - It can't be prioritized.
  - It can't be tracked separately.
  - It limits the Dev Team.

# Can I add stories to a Sprint?

- The big boss came up with some idea.
- So the PO tells the Scrum Master that the boss wants the idea to be done. Then the story is added to the Sprint. Or the PO talked to the developers directly, and they developed the feature without talking to the Scrum Master.
- Scrum Master should say, “No, please wait for my plan.”.

# How to define the acceptance criteria for a story?

- Usually the acceptance criteria has to be defined for a few important use cases.
- For example, the acceptance criteria of user login is
  - Enter a right username and password, you can log in.
  - Enter a wrong username and password, an error message should be prompted.

# What if not involved in the development?

- If you are not doing any of the development, and your idea is against the Dev team, you should
  - communicate with the Dev team in time, and find out the problem, and solve it asap.
  - Once a feature is done, verify it immediately.

# The Role of the Scrum Master

# From managing to serving

- From a manager to a server
- Change your attitude
- From centralization to decentralization
- Let the team make decisions and tolerate their mistakes

# How to do KPI?

- Agile team is an entity.
- Assess the team as a whole.
- Assess individual within the team.



# Self-improvement

- When you are not needed anymore, you, as the Scrum Master, are great.
- Then what can the Scrum Master do? You can,
  - Be a Scum Master Trainer
  - Lead other teams
  - Be an architect
  - Be a Product Owner
  - Be part of the Dev team

# The Role of the Dev Team

# the right number of team members

- If too many members, it will take too much time even doing daily Scrum.
- If too few, no bigger features can be done in a Sprint.
- 5-9 members is ideal.
- Scrum Master and Product Owner are not team members.

# a variety of skills and roles

- Backend and frontend developers
- Testers
- DBA
- All skills required to finish this sprint

# from task assigning to task claiming

- In the traditional project management, the project manager decomposes requirements and assigns tasks to the team.
- The Scrum team break down stories and claim tasks.
- Claim tasks that you want to do.

# improve in each Sprint

- improve you and your team after each Sprint
- Continuous improvement
- Find the best development practice for your team

# self-managing

- The Dev team should be self-managing
- Decentralized
- Team member should be independent and self-autonomous

# no gliding

- One developer added one more feature which might be cool but not within the plan.
- The feature can bring unexpected problems, and even serious consequences.
- If any ideas, submit it as a story and plan it in the follow Sprint.



# about documentation

- It is not mean that no documentation is necessary in Scrum.
- Various documents are needed, e.g. database design documents, API interface documents, installation documents, etc.

# update the Burndown

- If not updated daily, the actual line indicating in a Burndown Chart will go horizontally.
- Estimate the man-hour of the remaining task you are responsible for every day.

# update the Burndown

- If not updated daily, the actual line indicating in a Burndown Chart will go horizontally.
- Estimate the man-hour of the remaining task you are responsible for every day.

# Scrum Meetings

# no long meetings

- Product planning and Sprint planning should be done in one day.
- Scrum Master is in charge of the meeting agenda.
- Scrum Master and Product Owner should prepare for the planning meetings before it starts.

# Daily Scrum is not for problem solving

- Standup meeting is for communication, information update within the team, and risk detecting in time.
- Standup meeting is not problem-solving meeting.
- The problem mentioned at the meeting should be addressed after the meeting.

# Sprint irrelevant persons

- The story of a pig and a chicken
- Sprint irrelevant person can join the meeting without talking, e.g. PO.
- Speech Prop

# Sprint Review Meeting

- Sprint Review is a very good opportunity to raise the team morale. It is also to enhance the sense of achievement.
- It is also a very good opportunity to demonstrate the product and get feedback.
- It is also to show that the team follow the rules and is trustworthy.



# Sprint Review Meeting

- Sprint Review is a very good opportunity to raise the team morale. It is also to enhance the sense of achievement.
- It is also a very good opportunity to demonstrate the product and get feedback.
- It is also to show that the team follow the rules and is trustworthy.

# Sprint Retrospective Meeting

- Continuous improvement
- Plan
- Take actions

# Thank you!

CONTACT US

4006 8899 23 / 0532-8689 3032

[www.zentao.pm](http://www.zentao.pm)

