

# Writing a Product/Sprint Backlog

Sergio Peignier

## 1 Product Backlog

Single authoritative source for team's tasks. It contains features, requirements, and activities needed to deliver the product.

- Evolving document (elements can be added / removed through the project)
- Guide for the team to know the tasks to be completed
- Owned and adjusted by / with consent of the product owner. Product owner gathers input and feedback regarding the product, in order to decide how to maximize its value. Gathers requests to modify a product (new features, replace old features, remove features, and fix issues)
- External changes (new market opportunities, competitor threats, feedback from customers ...) may lead the team to adapt the backlog to incorporate new knowledge: the backlog is never finished
- Prioritized list of product requirements, i.e., : Features, Bug Fixes, non-functional requirements (what should be done to get a valid product)
  - Top of list: very specific
  - Bottom: vague
- Prioritize the items considering: risk, value, dependencies, size, and development effort.
- Item description
  - Description: Important details regarding the item
  - Value: for the client or the team
  - Order : depend on value and estimate
  - Estimate: effort for the team to achieve the task. Estimated **relatively** w.r.t other tasks (T-shirt sizes scale or Fibonacci scale). To estimate, each member can give an estimate, and average the estimates (keep initial estimate private to avoid anchoring bias)
- Formats: **user stories**, use cases, hypothesis driven

## User stories

- Short and simple descriptions of a feature from the users' perspective (user experience centered).
- As a **User Role** I want this **Action** so I can get this **Benefit**
- Imagine full personas (give them names, stories), put yourself in the users' shoes
- Each User story should be:
  - Independent (from other stories)
  - Negotiable
  - Valuable
  - Estimable (amount of work can be estimated)
  - Small (fit in a sprint, otherwise break it. Can be big at the beginning)
  - Testable (a test can be written to ensure it fits the requirement)
- **Epic**: group of user stories
- **Acceptance criteria**: checklist to ensure the user story is achieved
- **Backlog refinement** : keep Backlog described, estimated and prioritized. **When?**: depend on the team

## 2 Sprint Backlog

Subset of the product backlog to be completed

- List of work the team must address during the next sprint.
- Built progressively by the team while selecting product backlog items in priority
- Keep in mind the past performance to assess the teams' capacity.
- Could be broken down into developers' tasks (team members sign up for tasks, according to the backlog priority and their own skills and capacity → self-organization).
- Property of the developers: no additional work can be added except by the team.
- Adapt the product backlog and re-prioritize its items if necessary, accordingly to the sprint backlog.

- Scrum master: Team's member that aims at protect the team from external perturbations, helps to solve non-technical problems within the team, ensures part of the communication with the client (with the product owner), and has no hierarchical power on the team.
- Sprint duration: depend on the requierements change speed (if changes each week → sprints each week), estimate the amount of time needed to program the backlog, estimate the amount of time validation will take,
- **burndown chart**: measures time against the amount of work done and the amount of work remaining.
- **Velocity**: Average number of points a team burns down during a single Sprint

**Possible tools:**

- Asana
- Jira
- Trello