

Release Plan Template – CSE 115a – Software Engineering

The team needs to capture the result of their release planning meeting in a document with the structure outlined below. This document (as well as other project documents should be made accessible to the team's TA (e.g., via shared Google docs or drive).

There is no further submission process. All project documents are created for the benefit of your project and team, not the entertainment of the teaching staff.

You may in addition use web-based agile tools if you so choose.

- **Heading:** Document name ("Release Plan"), product name, team name, release name, release date, revision number & revision date.
- **High level goals:** A description of the top-level goals for the release. Examples include, for a game: "Be able to play one complete level (but with limitations xx, yy, & zz)," "Have all controller capabilities implemented," "Be able to create levels using a level design tool;" or for the Osric system: "Be able to handle service requests for new and existing customers with access to requests by managers and technicians." These high-level goals may map to a single user story, but more typically will map to multiple user stories.
- **User stories defining the scope of the release:** A listing of all the user stories that are needed to implement the high-level goals. Each user story must have a level of effort estimate in story points. Each user story must be sized to fit within a single sprint. Each user story must be assigned to one of the sprints within the development period (usually 4 two-week sprints in a quarter-length course; 3 one-week sprints in a five-week summer course).

Either list the user stories in priority order within each sprint or indicate the priority of each user story explicitly.

Recall that a user story should take the form, "As a {user role}, I want {goal} [so that {reason}]". User stories should meet the "INVEST" criteria (independent, negotiable, valuable, estimatable, sized appropriately, and testable).

It is a good idea to identify each user story by a unique label that allows the user story to be referenced across different tools and documents.

- **The complete list of user stories** will take the form of:
 - Sprint 1
 - {priority} User story 1.1 [story points]
 - {priority} User story 1.2 [story points]
 - ...
 - {priority} User story 1.N1 [story points]
 - Sprint 2
 - {priority} User story 2.1 [story points]
 - {priority} User story 2.2 [story points]
 - ...
 - {priority} User story 2.N2 [story points]

- ... and so on for subsequent sprints.

Note: “User story x.y” is a “meta name”; use some more descriptive label instead.

- **Sanity check your release plan.**

Is the plan within the team’s capacity? Given what you know about your team’s capabilities at this point, is the the total amount of work doable (add up the story points for all user stories and compare with the team’s capacity).

Is the work distribution across sprints reasonable? Did you allow for time spent on infrastructure tasks and spikes? Holidays? Midterms?

- **Product backlog:** A listing of all high-level goals and user stories that were discussed in the release planning meeting, but which did not make it into the release at this point. User story priorities may change in the course of the project and therefore the PO may decide to downgrade some user stories currently in the release plan and promote some user stories currently in the backlog. The release plan and product backlog should be revisited and updated after each sprint.

The product backlog remaining at the end of the last sprint can serve as the starting point for a subsequent release.

Initial Presentation: The release plan will be the basis for your team’s initial presentation.