1

Acceptance Criteria

Define boundaries & conditions which a story is “done”

How does a team know when a story is done?

A story in and of itself does provide clarity of purpose of a unit of work. However, it may not have clarity of how a story is tested to make sure that the work completed meets the story format.

Acceptance criteria focuses on the end result of a story, not just the intent of a story. Clear criteria defines the boundaries and conditions under which a user story is fulfilled and helps team members to understand what's included and what's excluded from the scope of the user story.

When concisely written. criteria allows for accurate planning and estimation and help teams avoid ambiguity about when "done" is "done" because it is independently testable.

> "If we write and review the acceptance criteria before implementation begins, we're more likely to capture the customer intent rather than the development reality."
>
> **Steve Povilaitis**
> Agile Coach

Testable Criteria

Acceptance criteria serve as a basis for use cases and test cases that ensure you achieve business goals and produce *"bug-free"* outcomes checking if a solution works as expected. A bug is an unwanted result.

Acceptance criteria is to have between 1-3 criteria per user story. If there are more, then a team should consider splitting the story into two stories. It is recommended to use the **Given/When/Then** format.

User	As a logged-out user
Goal	I want to be able to sign in to a website
Rationale	So that I can find acess my personal profile

Given	I'm a logged-out system user and I'm on the Sign-In page
When	I fill in the "Username" and "Password" fields with my authentication credentials and I click the Sign-In button
Then	the system signs me in

Keep in mind

Acceptance criteria should highlight specific POEMS criteria based on observable behaviors and actions that lead to a specific feedback and outcomes.

- People : What are they specifically doing?
- Objects : What is the object doing?
- Environment : What signals is the environment giving people or objects?
- Messages : What specific verbal, written, or non-verbal messages are being delivered?
- Systems : What is the system doing that provides feedback that something is completed?

Product Manager will branch their repository for team to generate content

Acceptance Criteria for the group project will go through a series of forks in order to demonstrate common use of branching a repository to make edits.

There will be as many source files in the branch as there are stories. Edited files by each team member is then pushed to a new branch created in the fork. This is reviewed by the whole team during a pull request and changes are added or reverted into the original branch and file. The edits, once merged, are now permanent and show to all team members.

Key Terms

Source File The original copy of any file that will end up being edited within the forking process.

Branch A sub category (folder, location) of a repository, where edits made are stored but not applied to the source files.

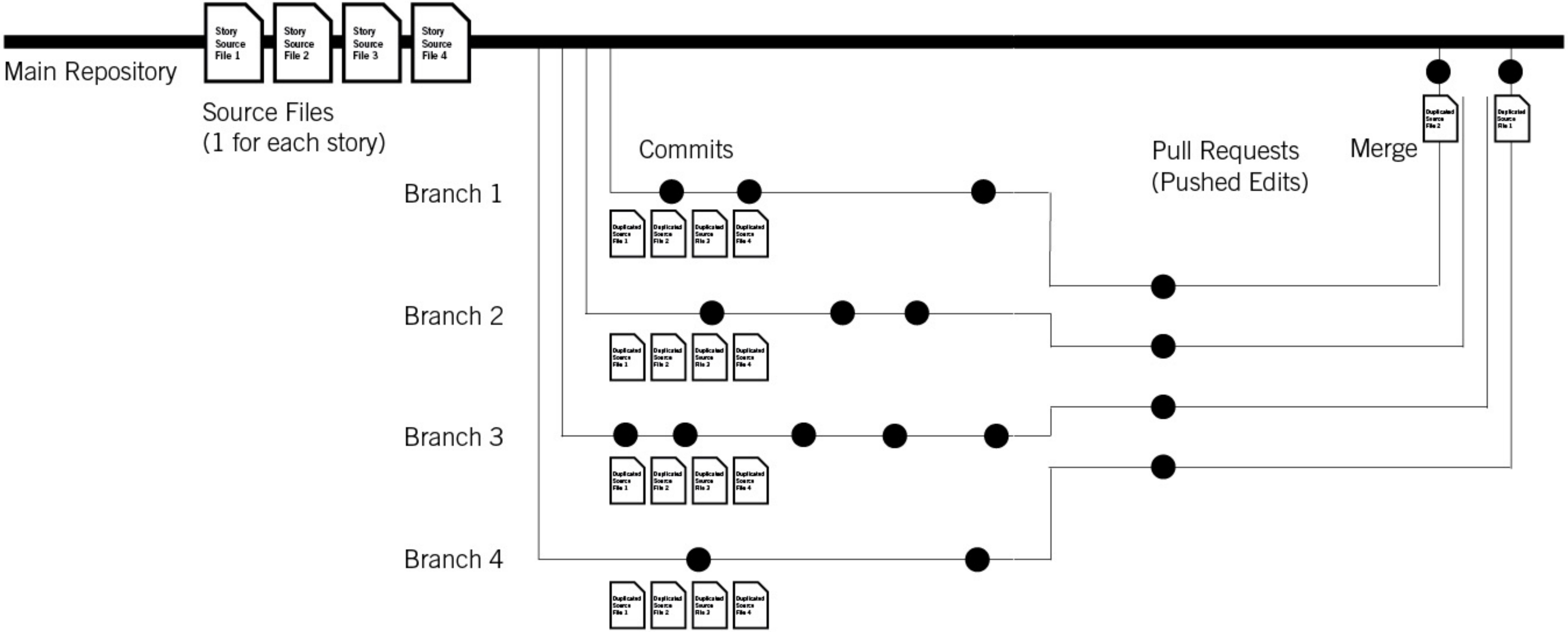
Staging Organizing and tagging edits as a group that is ready to be placed in a repository branch.

Pushing Applying staged changes to a repository branch for others to view and work with.

Pull Request A request for file edits on a chosen branch to be reviewed against source files for review.

Merge The act of taking file edits from a chosen branch and then applying them to the original source files, finalizing the changes.

Assignment at a glance



Assignment Steps

Preparing Source Files

- The product manager creates a new file in the standard master branch of their team repository. This file will be named for the story being defined "[team name] acceptance criteria [story number]"

Creating a Branch

- Each team member in will create a new branch called "[team member name] Edits [Year + Month]"

Adding Commits

- Each team member will add their version of story# [story number] to the file created in step 1, saving these edits in their branch.
- The edits from step 4 are staged and pushed to the branch created in step 3 and a pull request is made to the Team Lead.

Opening & Submiting a Pull Request

- The Team lead provides the pull request and reviews changes made.
- Changes can be discussed and altered using github's threaded discussion and alert tools.

Merging & Deploying

- Once the team lead has chosen a branch with the edits they prefer, they will then merge that branch into the master source file, finalizing only the changes from the chosen branch.
- Once the story has been finalized and agreed upon by the team, the info from the story is then copied into Github's actual story tracking system.
- This entire process is completed at least once per story within their acceptance criteria.