

Source Code Management

Training

# Source Code Management

The training will be based on the scheduling app development of the VacYOU robotic vacuum cleaner. It starts from the perspective of the development lead. He will create tasks and put them on the Kanban Board. Swim lanes by developer.

Roles:

* Development Lead
* Developer

Content

Development Lead:

* Create a repository
* Create Releases and Sprints
* Create-, estimate-, assign to sprint, prioritize 3 tasks.

Developer:

* Install git
* Review Tasks in Sprint Planner and takes the highest priority one
* Create fork
* Clone repository to local machine
* Task Status: in progress
* Write Source Code
* Commit Source Code with reference to the task
* Push to his fork repository
* Send a pull request

Development Lead:

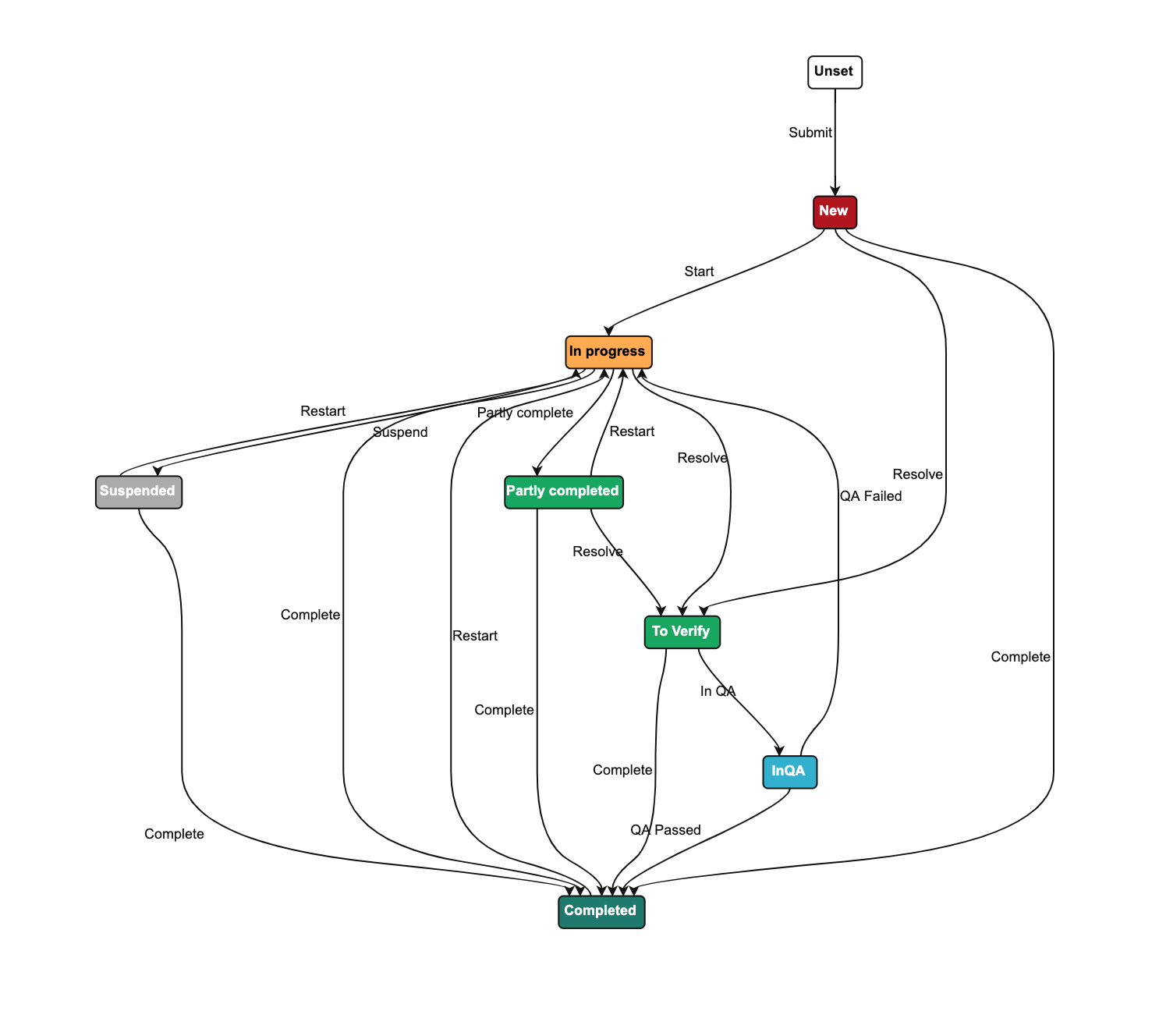
* Complete a Source Code review
* Merge pull request
* Track development progress

## Workshop: Create Releases and Sprints

Source code management is tightly integrated into the project management process. Committed source code changes will update tasks.

1. Create a new project based on „Default” template
2. Create Product Release 1(4 weeks)
3. Add two sprints (2 weeks each): Sprint 1 and Sprint 2.

## Workshop: Task Management



1. Create 5 tasks. Assign Story Points

Design Scheduler Screen on Xcode for iOS

Add button to UI to start Scheduler screen

Integrate calendar module into the Scheduler screen

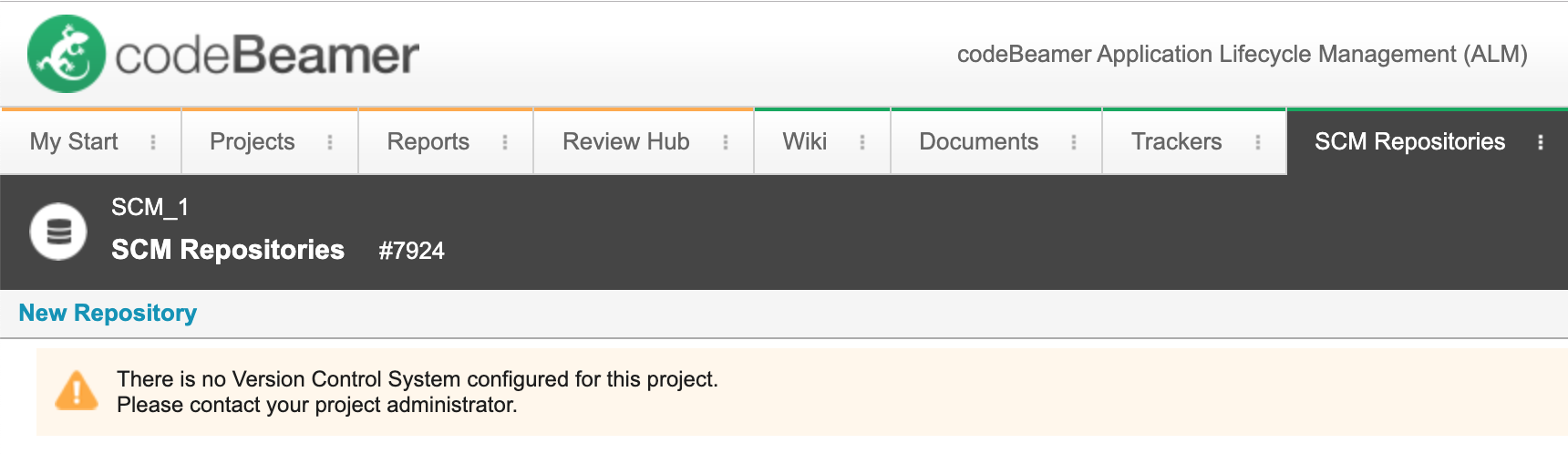
Add „Add cleaning session” button to schedule a cleaning program.

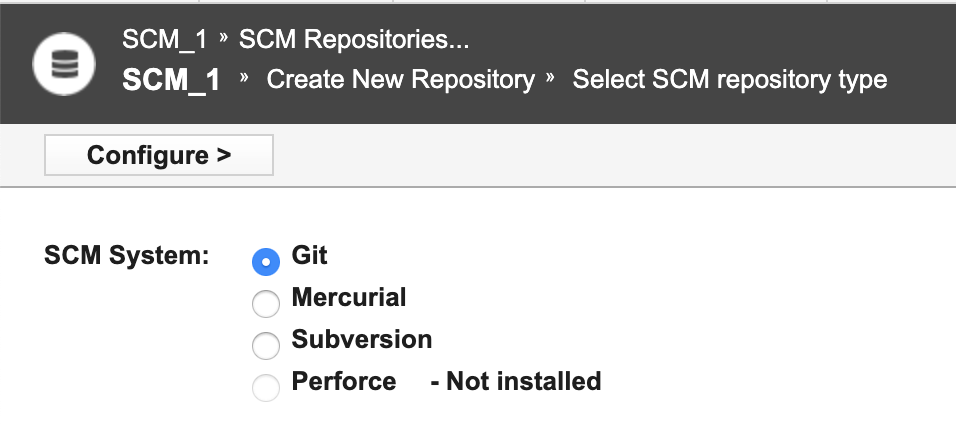
Design UI to add cleaning session

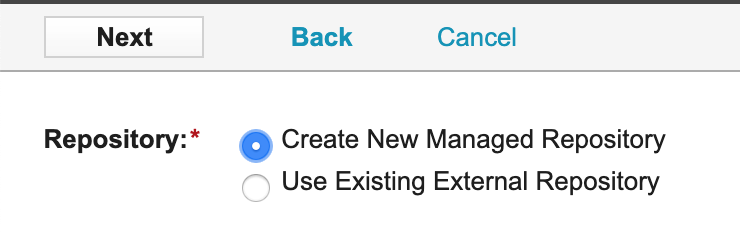
1. Navigate to Release Planner and view the Backlog of tasks
2. Prioritize the Backlog
3. Assign the top 3 tasks to sprint 1 in the Release Planner
4. Start Sprint
5. Assign the tasks to developer (Student123) in the Release Planner.

## lsWorkshop: Create a Managed Repository

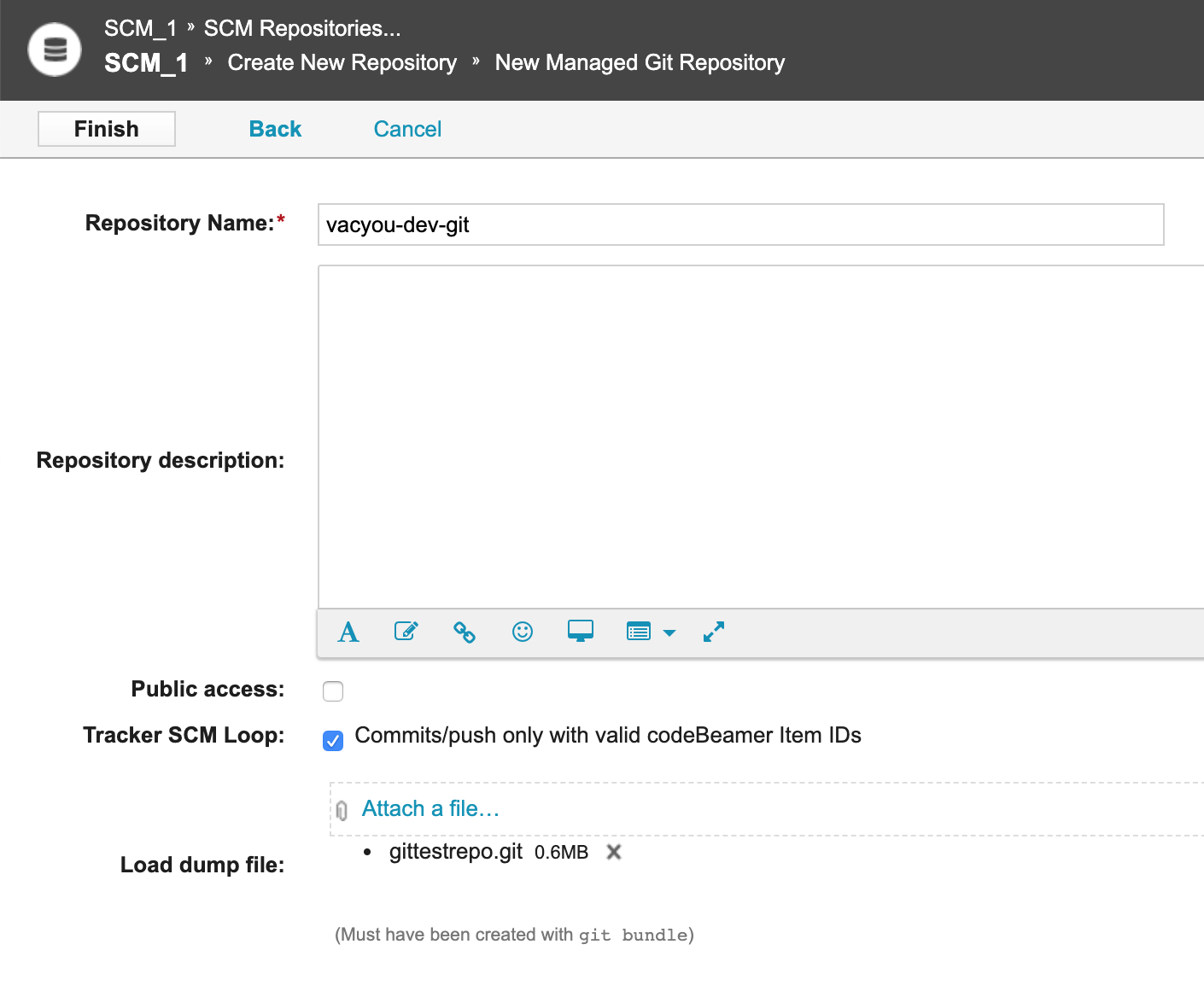
1. In the tab **SCM Repositories**, create a **GIT repository** from the attached dump



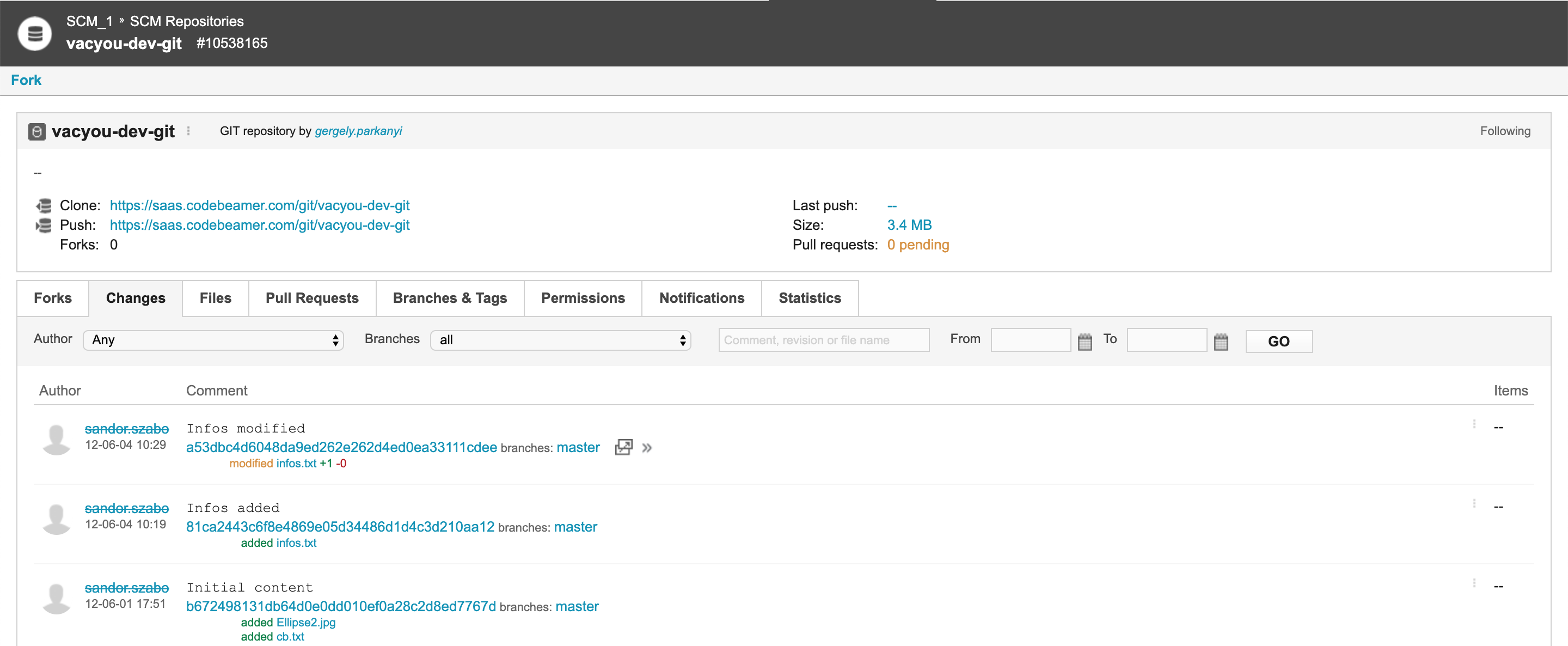




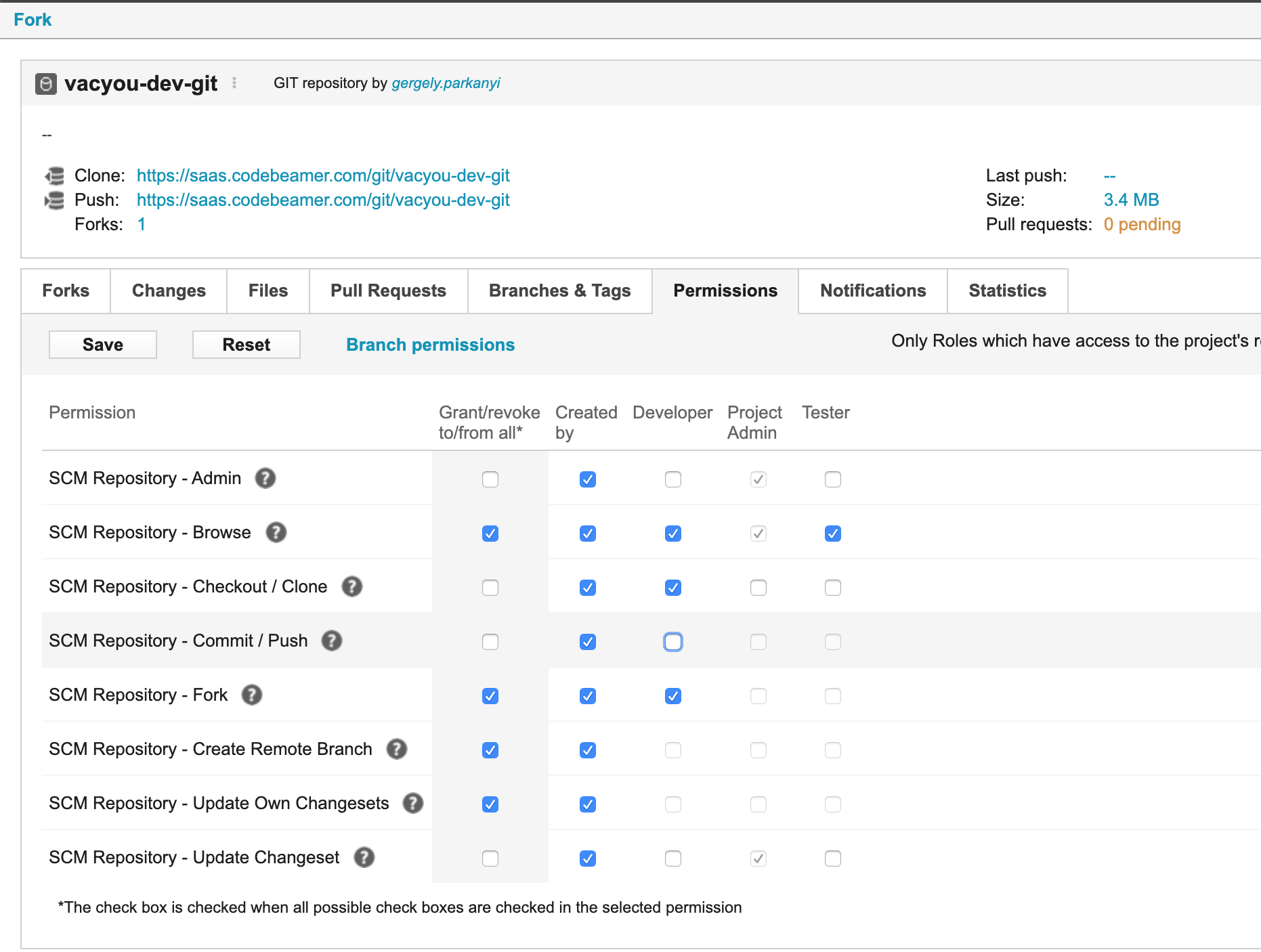
Enter Repository Name, description. Attach the dump file distributed before the training. Select the checkbox for Commit / Push only with valid codeBeamer Item ID:



1. The repository has been added successfully



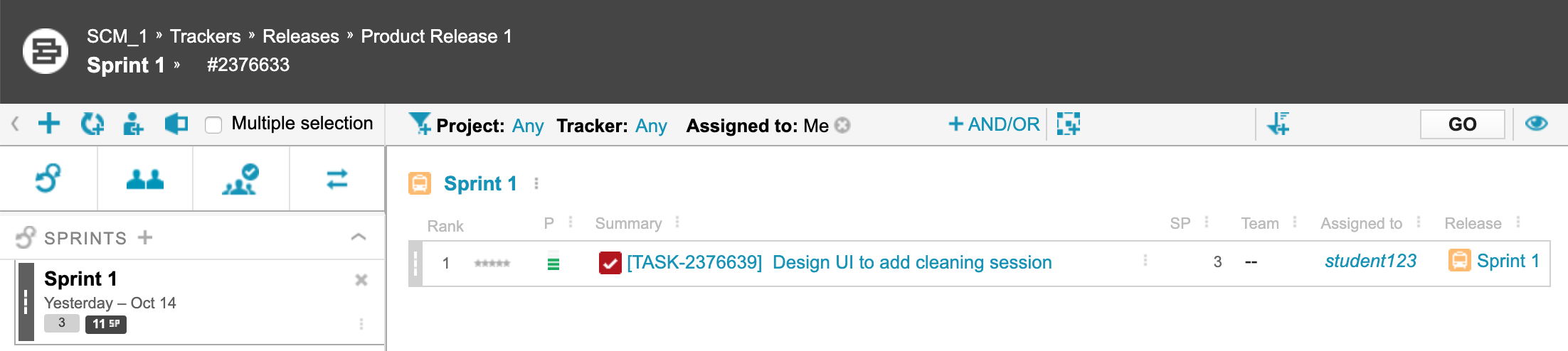
1. Revoke the Commit / Push permission from the Developer Role



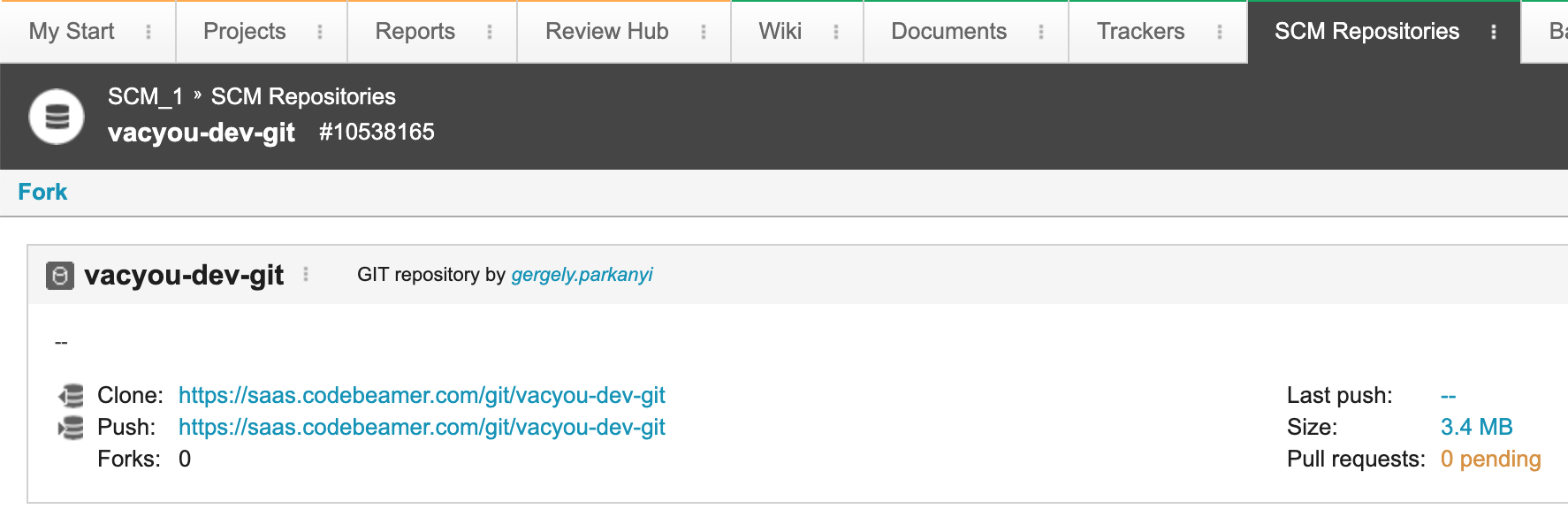
## A Day in the Life of a Developer

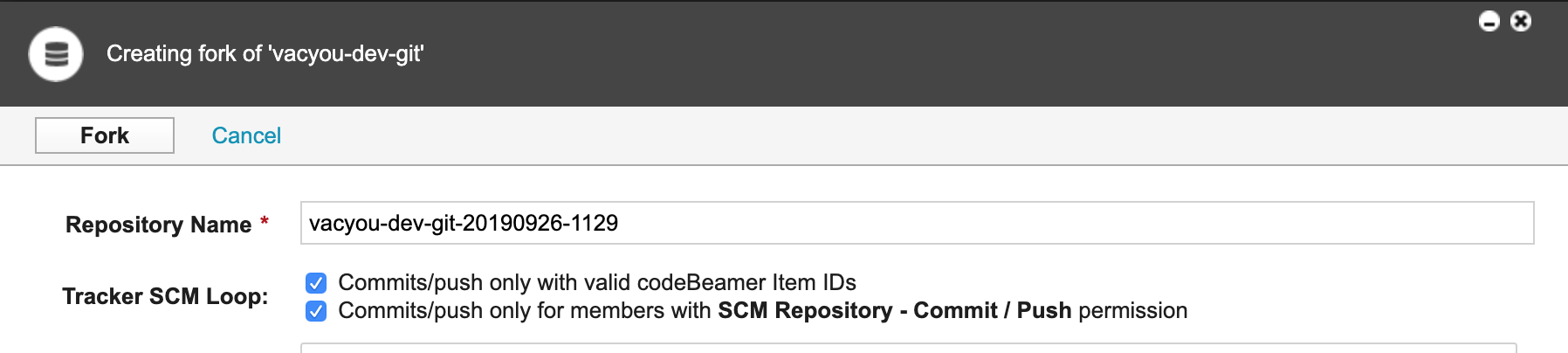
### Workshop: Create your fork

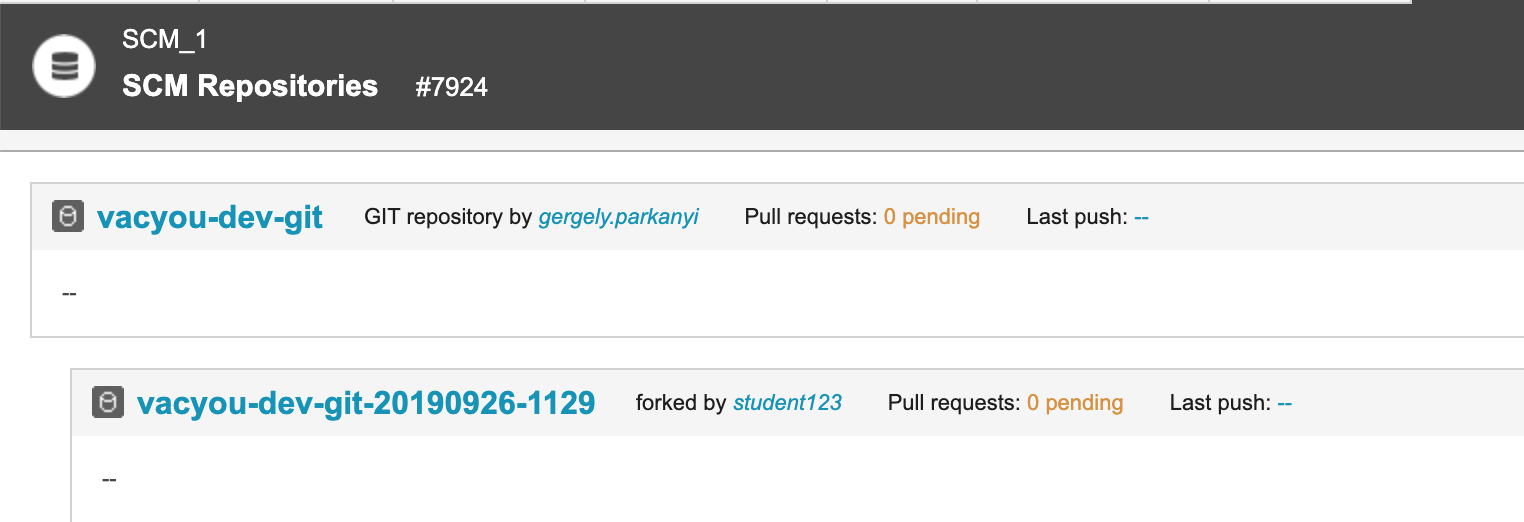
1. Log in as a developer.
2. Review Release planner. Filtered to your tasks



1. Create your own fork of the master GIT repository







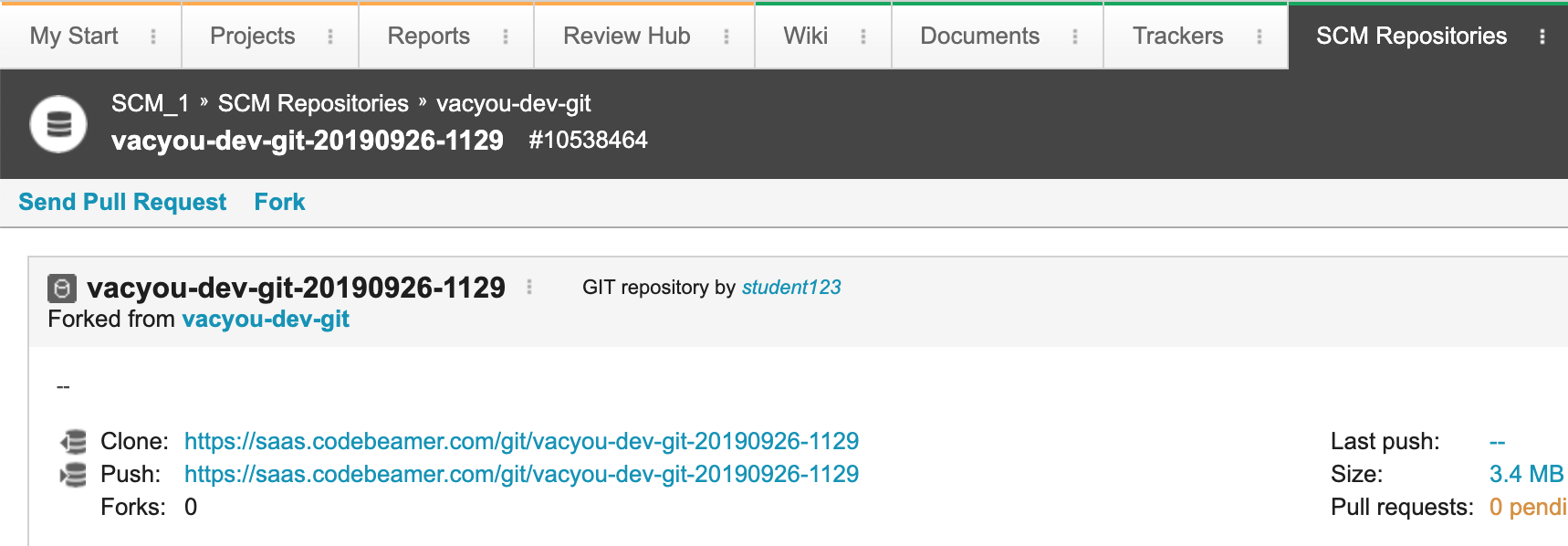
1. Create a development folder for your local repository.
2. Install git on your local machine
   * <https://git-scm.com/download/win>
   * <https://git-scm.com/download/mac>

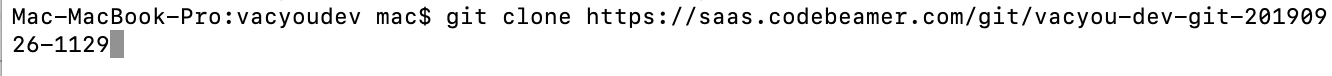
### git Workshop: Clone the repository to your local machine

Open up terminal (mac) command prompt (pc)

* Navigate to the folder created above
* Git clone {clone URL}

You will find it in the repository details page:



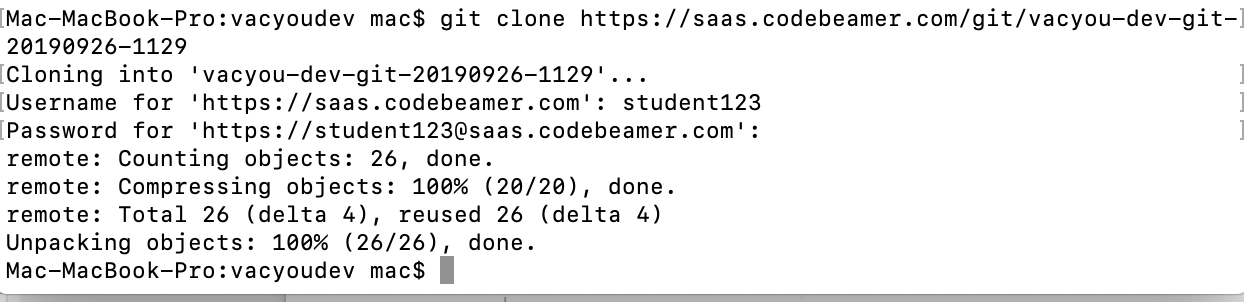


Note:

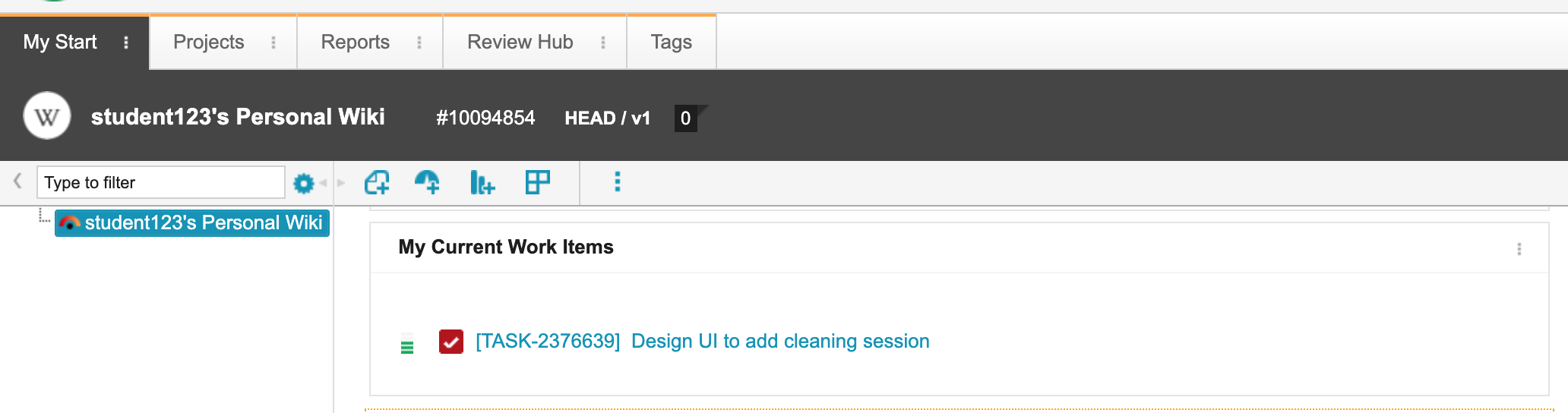
If you are using a new user to access the repo



Cloning:



1. Find the task in the MyStart page



1. Change status of work item to In Progress by selecting the Start workflow action.

### UIWorkshop: complete your coding task

1. Create a new text file and copy this code into it:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>title</title>

</head>

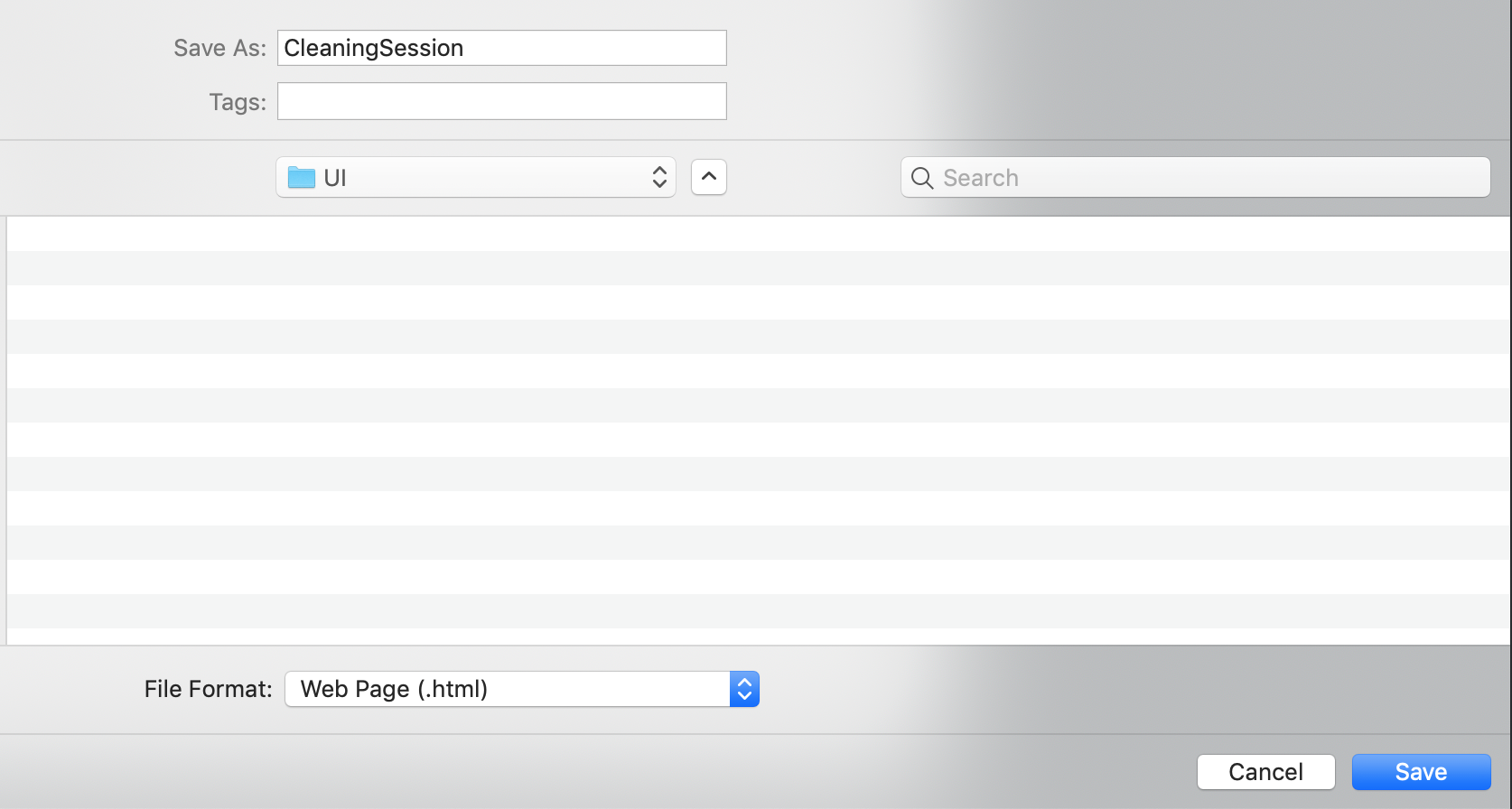
<body>

<h2>Cleaning session</h2>

</body>

</html>

1. Create a new folder called UI in your local cloned repository folder and save the text file in .html format.



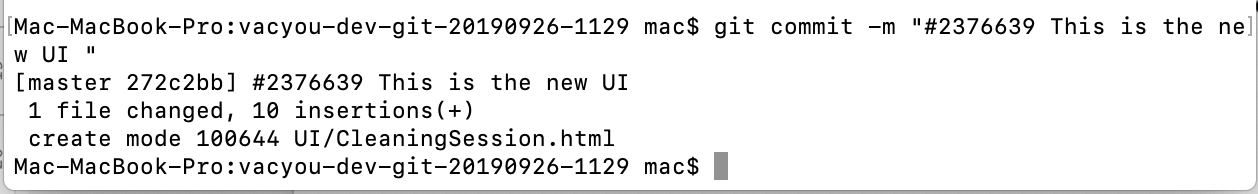
1. Test the code by opening the file up in a browser.

### Workshop: Commit and Push to Fork. Create a pull request.

1. Navigate into the sub-folder created by the clone command.



1. Commit the change to your local repository. It is very important that you use a reference to your own task number, not the one shown in the scrennshot below.

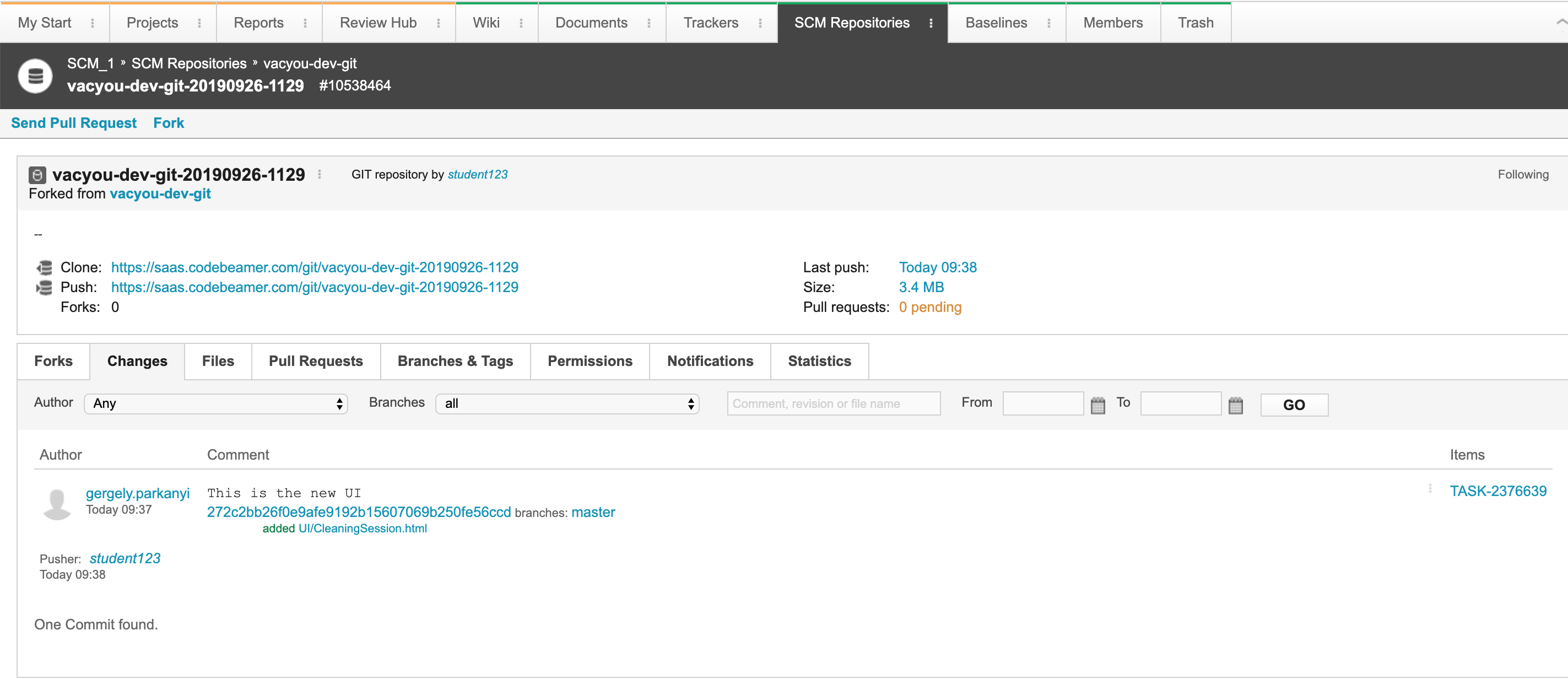


*Hint: In terminal cd v + tab will auto-populate the folder name if your folder start with v.*

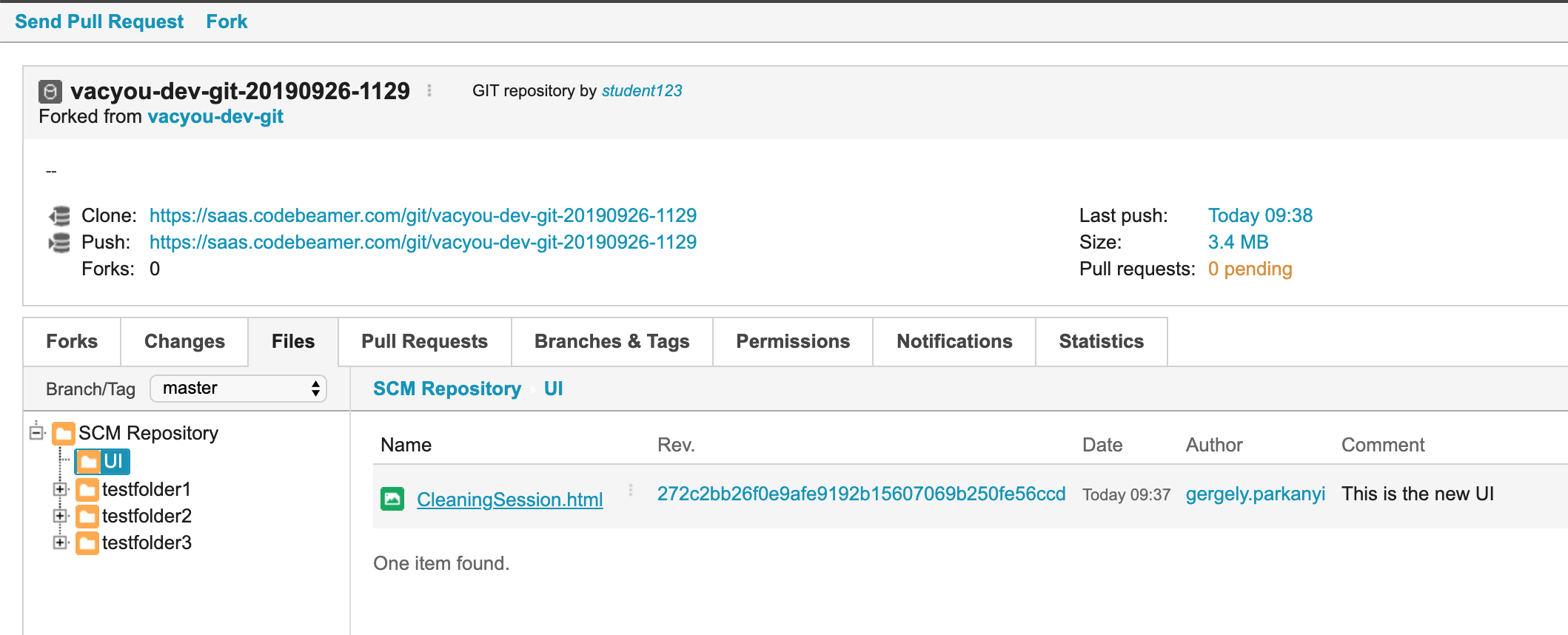
1. Push the change to your **Fork repository**



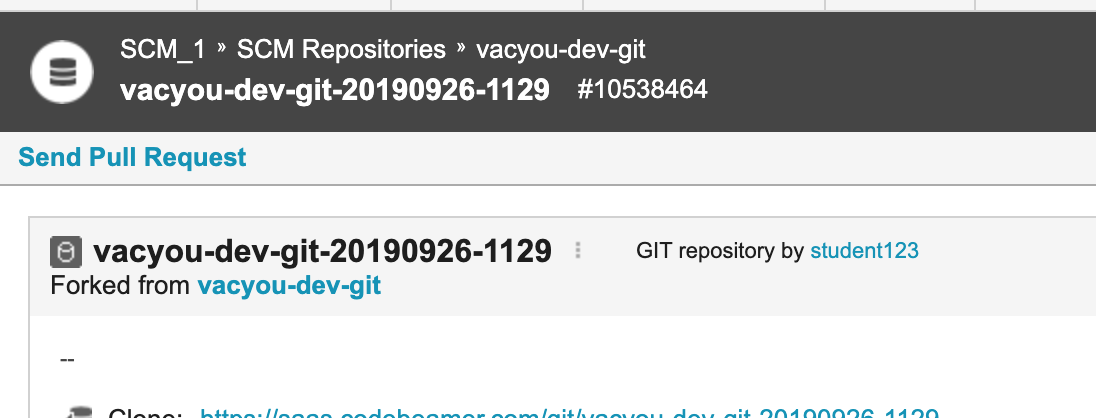
1. In codeBeamer check the **changes** and **Files** tab of your fork repository
2. #5972
3. #15880176 - Copy temperature value to SRAM



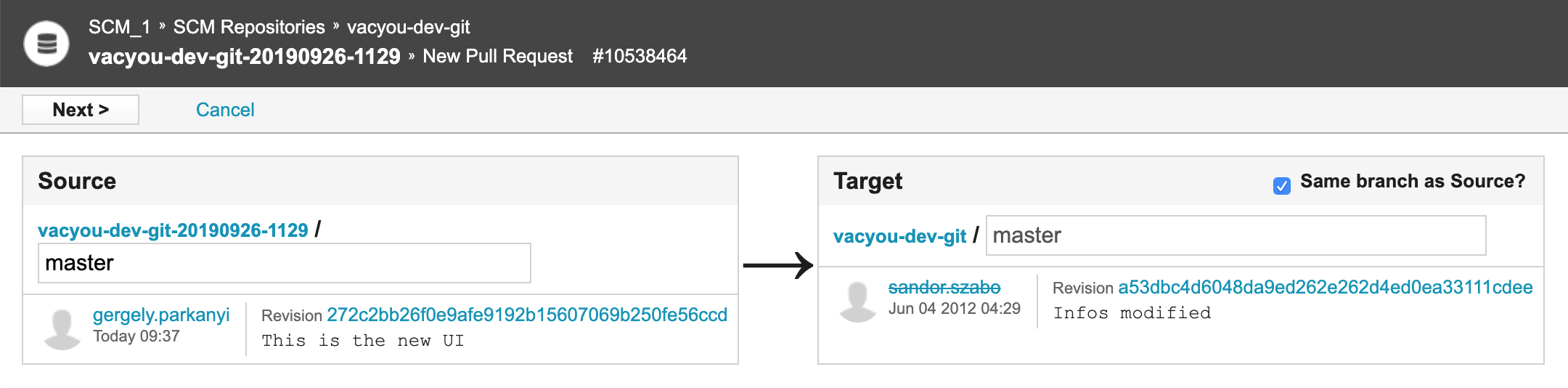
1. Navigate to the files tab!

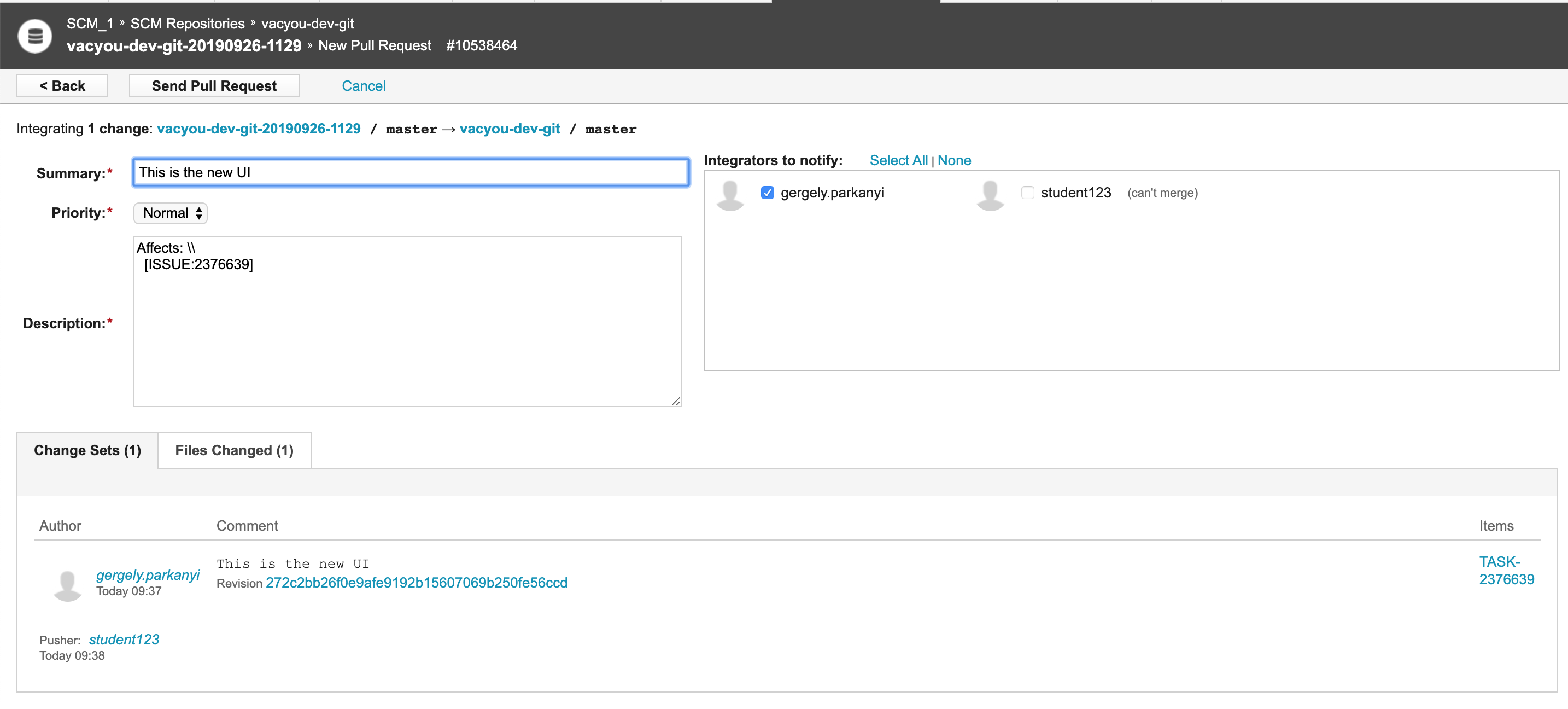


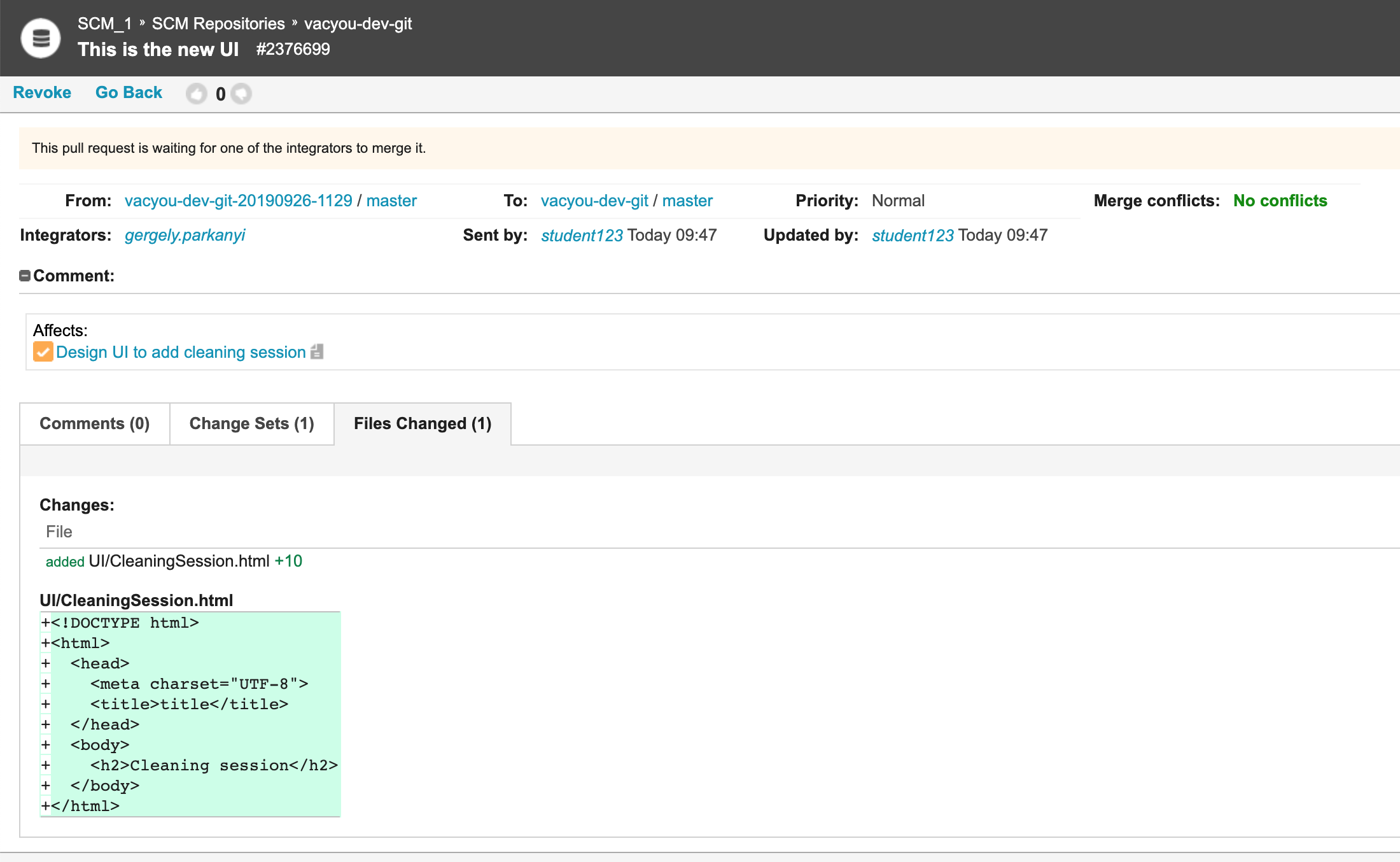
1. You should now see your newly added program.
2. Select the link above:



1. Create a pull request







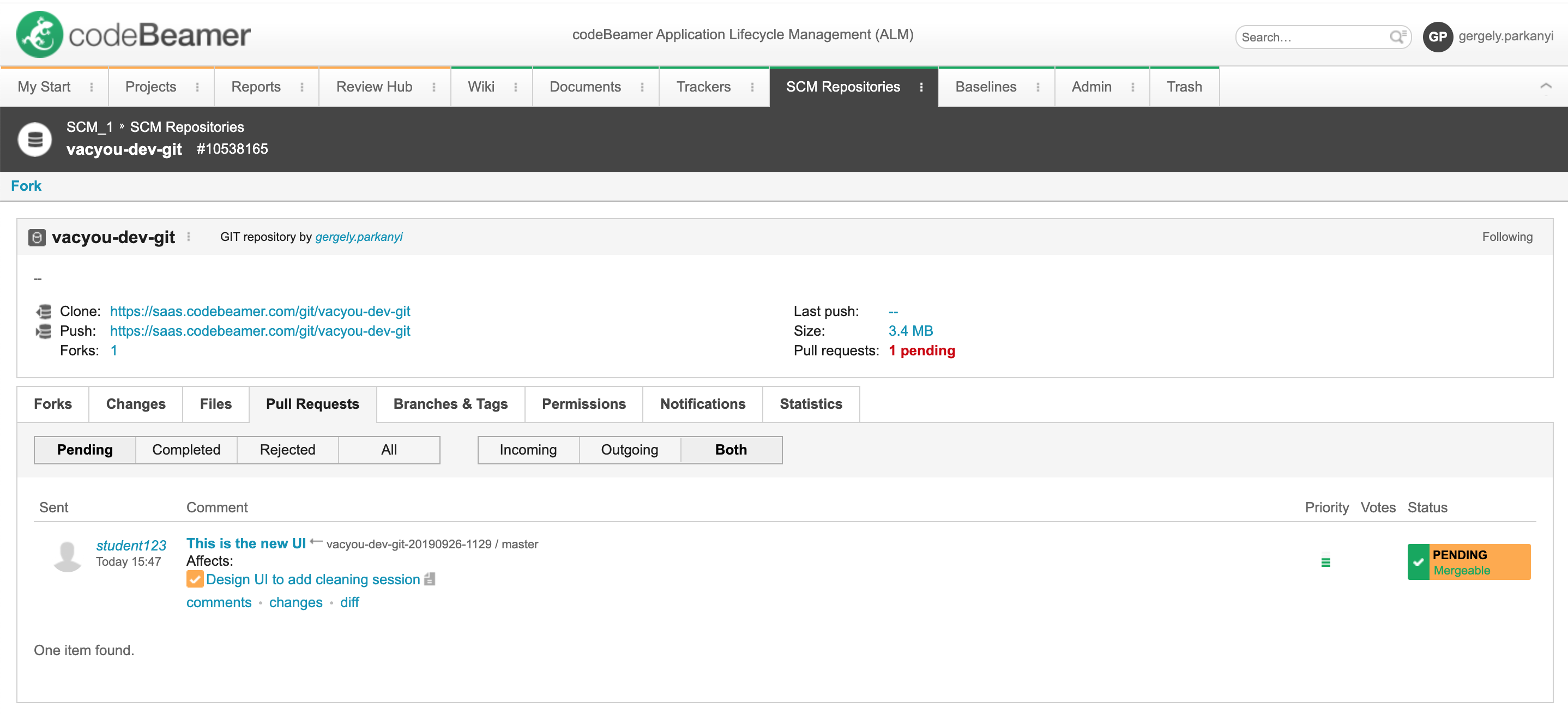
1. Change the status of the task by selecting „Resolve“. New status is „to Verify“



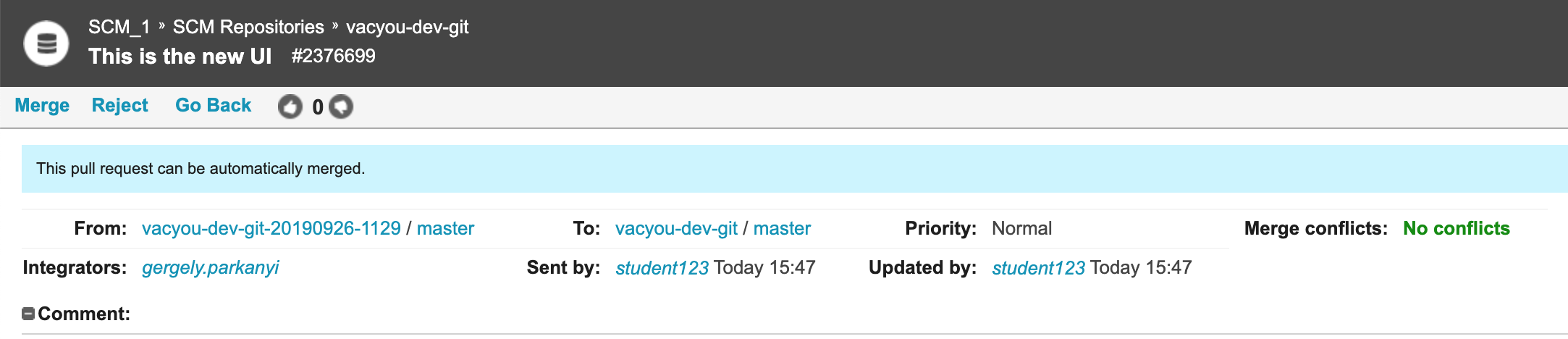
## Source Code Review. Development Management Tasks

### Workshop: Track and approve source code changes

1. Log in as Development lead
2. Navigate to the repository and review pull requests:



Hint: *If there are conflicting changes, it cannot be merged automatically*



1. Complete a Source Code Review by clicking the „Merge“ link

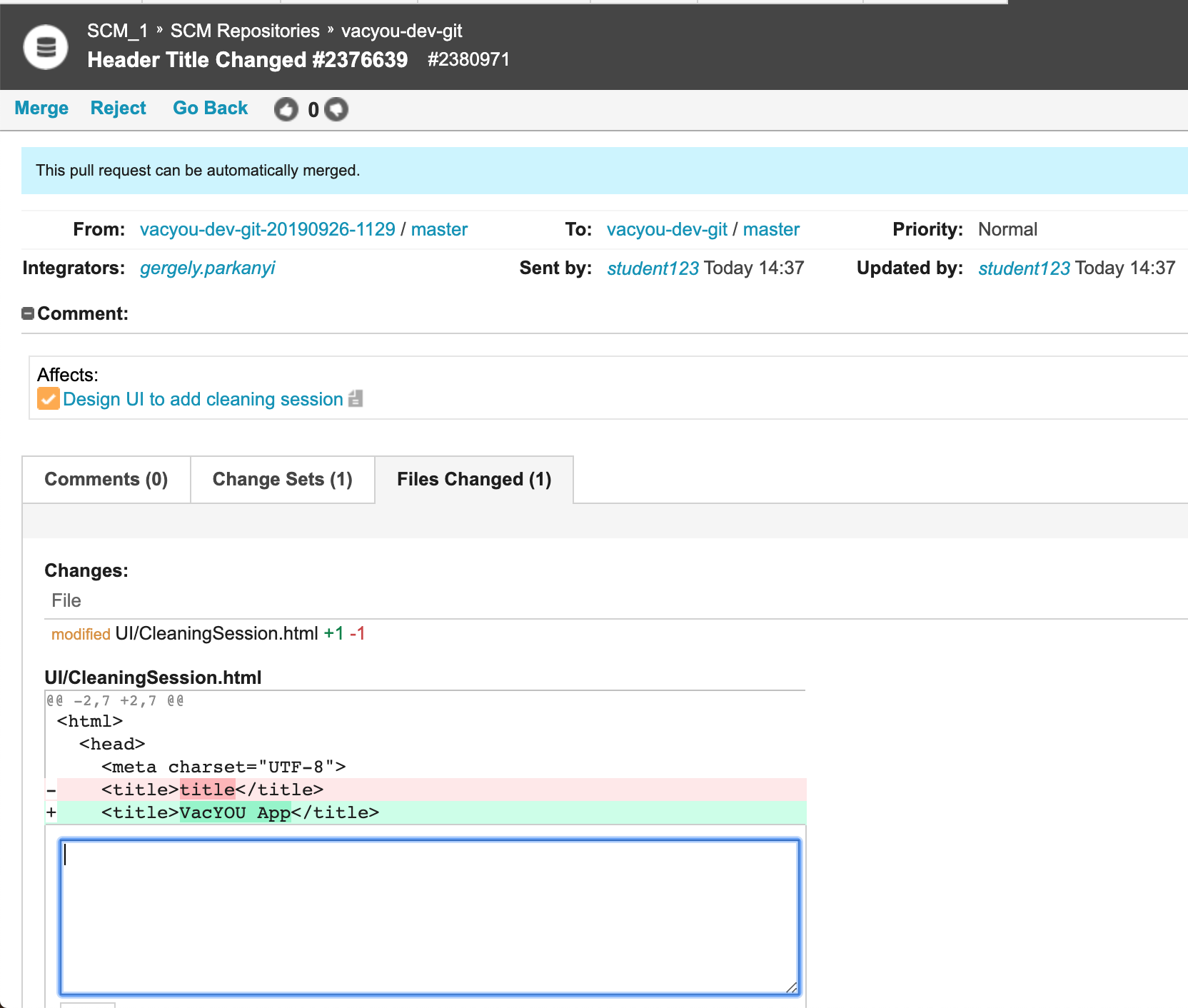
You will get the following confirmation:



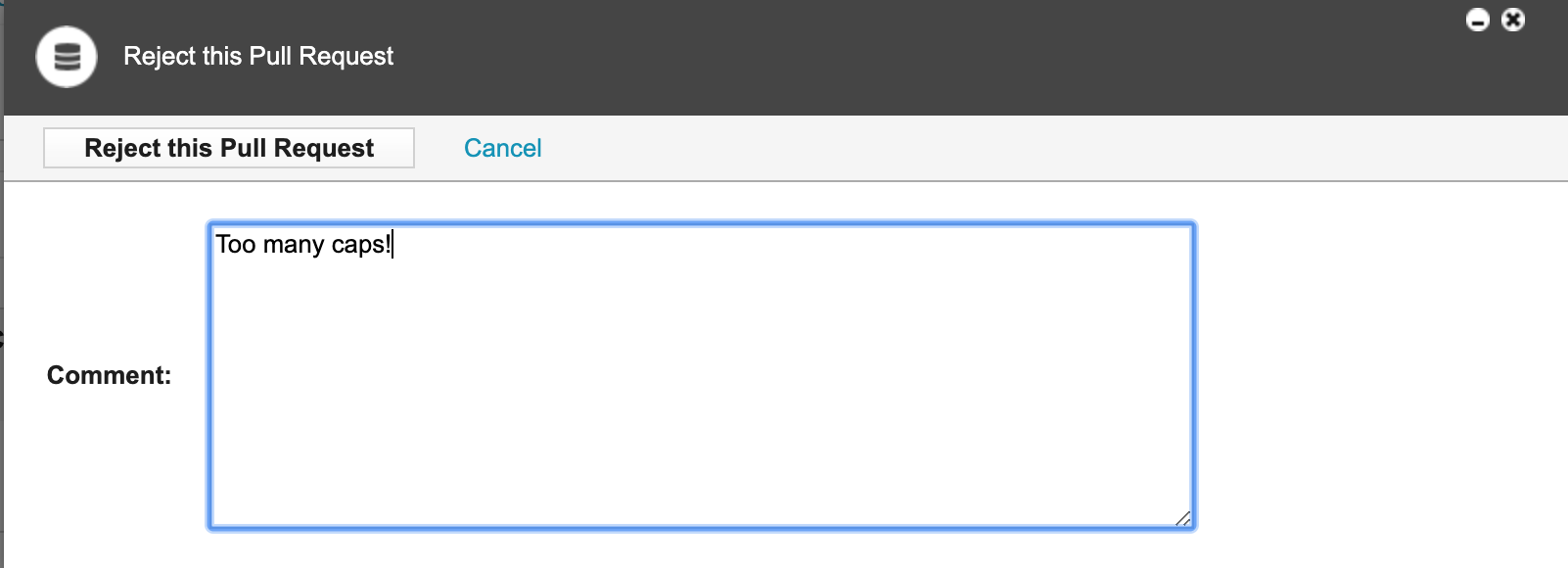
1. Change status of the task to inQA
2. Check the status of the Task on the Kanban Board. Start and finish on the Kanban Board.

### Workshop: Reject Source Code Changes

Repeat all the steps above carefully to have something to reject.



Double click the line and add a comment



Log back as Student123.

Review your Activity Stream. At the top of the list your will see an “Item Changed” message.

## Overview of useful Git and general Command Line(Pc) / Terminal(Mac) instructions

|  |  |
| --- | --- |
| Ls | List the contents of a folder |
| Tab | Cd x+tab (where x is the first letter of a valid folder) changes directory to that folder |
| Cd .. | Change directory |
| Git status |  |
| Git add |  |
| Git status |  |
| Git commit | -m “#cB Task Description of Change” |
|  | How to fix the “fatal: unable to access 'https://saas.codebeamer.com/git/TCS\_SCM\_3/': SSL certificate problem: unable to get local issuer certificate” error: |
| Git config | For Windows first try:  git config --global http.sslbackend schannel |
| Git config | In the local repo folder:  git config http.sslVerify false  at the end of the training:  git config http.sslVerify true |
| Git push |  |