

Continuous Integration and Coverage: Team Assignment 8

Published

 [Edit](#)

This is a team assignment.

First setup your github repository with a continuous integration tool, and make a pro forma commit to check that your full test suite indeed executes.

Then introduce a statement or branch coverage tool that plays nicely with your unit testing and CI tools, and measure the percentage covered. If this is not 100%, figure out what code isn't being exercised and why, and add additional unit tests to try to increase coverage. Note you may not be able to exercise all code automatically generated by the framework. But keep in mind that you can use unit tests, and mocks/stubs employed in unit tests, to invoke subroutines with parameters that would be hard to force during full system execution and to return values that might be difficult to achieve during full system execution, respectively, e.g., security violations, network errors, file system full. Add the coverage report to your github repository as a version controlled file; make sure to commit the revised report after every further change to either the code or the test cases. Also write a brief document that explains any remaining problems in reaching 100%, and include this in your github repository.

Finally, devise a set of full system tests for your application, corresponding to your user stories or use cases. Some of these might be manual scripts rather than automated, that's ok as long as you indeed manually conduct the tests, observe the results, and record any bugs found as issues. Include any scripts, data sets, etc. in your github repository.

Make sure your issue tracker accurately records the status of all bugs found so far throughout the project, whether fixed or not. Also make sure your task board for the second iteration is up to date.

Plan a FIVE-minute "showcase" demo that highlights your application and demonstrates MVP (minimal viable product). Write a human-executable script explaining what you plan to show and what you plan to say, practice the demo to make sure everything works and fits in the time allotted, and include this script in your github repository. Make sure to tag the exact file versions to be used for the demo. Note this "showcase" demo will be presented on demo day. Thus you need to submit this document before the demo, not after.

Each team should submit a single file, i.e., one member should submit. The name of your file should include your team name, and the contents of the file should also include your team name. Otherwise all the file needs to contain is where exactly to find each of these materials.

Points 10
Submitting a file upload
File Types doc, docx, pdf, txt, xls, and xlsx

Due	For	Available from	Until
Dec 10, 2016	Everyone	Aug 26, 2016 at 12am	Jan 31 at 11:59pm

[+ Rubric](#)

