QA Engineer Final Projects

JIRA

- Test Plan
    - Your test plan should be thorough and cover all planned testing activities, but maintained to reflect changes to your plan/requirements.
        - Define the scope of what you will be testing, especially if you are not going to be able to fully test the product.
    - When the test plan is ready for review, move it along through the workflow to “In Review” for a peer and instructor’s approval.
    - Approvals are required, at least one peer approval and an approval from your mentor or instructor.
    - At the completion of testing, prior to submission, a Test Summary Report should be generated.
        - Refer back to the test summary report assignments for required content.
        - Add the report as a comment on the Test Plan.
        - Approvals should be the same as those who approved your test plan.
- Test Cases
    - All of your features to be tested in your test plan (and their pass/fail criteria) should have corresponding test cases.
    - When a test case is ready for review, move it through the workflow to be “In Review.”
    - All test cases require either a peer or instructor review.
- Bug Reports
    - When a bug report is created, it needs to be verified by a peer, who can move the bug into “To Do” from the “Open” status.
    - If the bug isn’t *actually* a bug, don’t delete it, move it to “Done.”
    - Go ahead and report suggestions or “this would work better if” ideas in ONE bug for the project, and you can move it into “In Review.”
- Other Info
    - Your instructor can create additional projects with separate boards at your request.
    - Link Bugs, Test Cases, Test Plans, etc.
    - USE THE WORKFLOW.

Automation

- NightwatchJS is the automation framework that will be supported, but you can use a different framework if you feel up to the challenge.
- Automated tests should reference the JIRA test case containing the steps it automates.
- Abstraction should be utilized:
    - Separate selector file(s) or Page Objects
    - Data file(s)
    - Functions are encouraged, particularly for data-driven testing, but they are not required.
- Your tests should be pushed up to a GitHub repository that will be accessible by your instructors.
    - .gitignore should include any system generated .log files or test-outputs, and your nightwatch.props.js file.

Group Work

- A test lead will coordinate with the instruction team, but the teams need to share the workload evenly. Don't take a team member's work, and don't let your team do your work.
- Test plans need to be approved by the entire team.
- Automation needs to be checked in to a shared GitHub repository
    - Individual pieces needs to be pushed up to GitHub in a separate branch of the same repository, where a pull request can merge the feature branch into the master branch.

Tools and Techniques Usage

- Make full usage of tools and techniques we have learned.
    - As mentioned, JIRA and NightwatchJS are the main tools.
- If the project uses an API, Postman tests are required.
    - SoapUI is optional.
- Boundary Value Analysis should be used - valid and invalid data tested.
- Use of Decision Tables and State Transition Diagrams/Tables is highly encouraged.
- Attach tables/supporting docs to the test plan or test case(s) they apply to.

Submission

- When everything is complete, your instructor should be involved - have the following ready:
    - All JIRA issues
    - Test Summary Report (draft)

    - GitHub repository
- Have the submission in by Wednesday before class is concluded, to address any polish or changes required.
    - Keep your instructor involved earlier if you are concerned that changes major changes might be required.
- Submissions will be accepted after the scheduled end of the course, including for badging. Just remember that badging is required prior to the school scheduling your certification exam.