**Example Quality Strategy**

**Overview**

We believe that quality is everyone's responsibility.  Quality is not a box to be checked, but a process that moves from writing the story, through crafting acceptance criteria (AC), through coding the story and writing unit tests, through the story handoff, through manual, security, and performance testing, and through the creation of test automation.

**Creating and Grooming Stories**

Stories are generally created by the Product Owner (PO), although other team members can create them, and they are discussed and pointed in a grooming session attended by the entire team.  AC are written by the PO at the grooming session, and a high-level test plan is discussed in order to estimate the testing effort.

**Development Process**

Unit testing will be created by the person developing the story, and will be considered part of the Definition of Done.  Pull Requests (PRs) will be approved by the Software Engineers (SEs), and additionally, unit tests will be viewed by the Software Test Engineers (STEs).

**Feature Handoff**

When a story is ready for testing, the person who did the development work will deploy the code to the QA environment and conduct manual exploratory testing to ensure the feature is working as expected.  Then a handoff meeting will take place between that developer and the STEs.

**Test Plan Creation**

The STEs will be responsible for creating and executing a manual test plan for the story.  This plan will be stored in the Test Plan section of the team's wiki, and can be used for regression testing by the entire team in the future.

The SEs and STEs will determine together what automated tests should be created for the story, taking into account integration, API, UI, security, and performance testing.  See the Automation section for details on how the automation will be implemented.  Automation stories will be created by the STEs and can be worked on by both SEs and STEs.

**Test Harnesses**

Any SE or STE on the team can create test harnesses for the whole team to use in testing.  Examples might be a tool to mock a messaging service, or a shared Postman collection that can be used by the entire team.

**Manual Testing**

The STEs will be doing the bulk of the manual testing of the product, on both web and mobile.  However, there will be times that other team members will be conducting manual testing, as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Test | When | Environment | Testers |
| Exploratory testing | Before story handoff | QA | Developer of the story |
| Exploratory testing | Before test automation | QA | SEs and STEs |
| Exploratory testing | Before release | Stage | Entire team |
| Acceptance testing | As the feature is released | Production | Entire team, with final signoff from PO |
| Regression testing | When there is a significant event, such as an environment upgrade | Affected environment | Entire team |

**Test Automation**

We will support the following types of automation:

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Test | When | Environments Used | Tools Used |
| Unit | With every build | Development | NUnit |
| Integration | With every build and deployment | All | C# and Moq |
| API | With every build and deployment | All | Cypress |
| UI | With every deployment | All | Cypress |
| Performance | Before deployment to Production | Stage | JMeter |
| Security | Before deployment to Production | QA, Stage | Burp Suite |

It is the responsibility of all the SEs and STEs to create, monitor, and maintain the test automation.  If an SE breaks an automated test through new feature work, that test should be fixed by the SE.

**Bugs and Tech Debt**

When a bug is found during testing of a story, the team will decide whether that bug should be fixed immediately or logged as a separate issue, with preference given to fixing the bug immediately.

The team is committed to giving 20% of their sprint time to address tech debt.  Which debt stories will be worked on will be determined by the team during sprint planning.

**Releases**

The Engineering Manager will be responsible for scheduling releases and updating release notes.  The lead STE will create a release test plan for the team.  Any member of the team can trigger and monitor the release process, and any member of the team can conduct the acceptance testing.  We will share deployment and testing duties on a rotating basis.  The PO will update the team's documentation.