**Assignment 1 – Software Quality Assurance**

Modern Software Quality Assurance is made up of many components.

**Testing:**

* Black Box Testing or Behavioural Testing. It is a method of software testing that examines the functionality of an application without looking at the inner workings of the software. Testers test the UI and layout. They come at testing from the perspective of a user. They attempt to test the application in real world scenarios.
* White Box Testing. It is a method of testing the inner workings and code of an application. It does not involve testing the functionality of the application. Testers have the technical abilities of a developer.
* Grey Box Testing. It is a combination of Black Box and White Box Testing. Testers partially know the internal structure of the application. They require high-level and detailed documents describing the application, which they collect in order to define test cases.

**Tracking Bugs:**

Software such as Atlassian JIRA, Bugzilla or Pivotal Tracker can be used to track bugs and usually allows the following functionality:

* Project name
* Issues
* Feature/Ownership areas
* Defect Frequency (How often a user will experience the defect)
* Priority (Assignment developers should know what needs to be worked on)
* Issue Template
* Target Fix Date (When a bug should be fixed about)
* Build Information
* Conversation (Allows conversation to take place between developers. Useful for future reference)
* Attachments (eg. Screenshots or System Logs)
* Workflow Management
* Linked terms and bugs (Keeps all relevant information for a set of related bugs together)

**Using a Matrix to Define Priorities**

This is just one example, methods can vary. The top row represents percentage of users affected by the issue. The left column represents the severity of the bug. Orange is high priority, yellow is medium priority and green is low priority. Developers must find out the severity and percentage affected of each issue. This allows them to clearly see what needs to be prioritised.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **100% - 75%** | **75% - 50%** | **50% - 25%** | **<25%** |
| **Crash** | Priority 1 | Priority 1 | Priority 1 | Priority 1 |
| **Non- Functioning** | Priority 1 | Priority 1 | Priority 1 | Priority 2 |
| **Incorrectly Functioning** | Priority 1 | Priority 1 | Priority 2 | Priority 2 |
| **Incorrectly Functioning with Workaround** | Priority 1 | Priority 2 | Priority 2 | Priority 3 & 4 |
| **Performance** | Priority 2 | Priority 2 | Priority 3 & 4 | Priority 3 & 4 |
| **Cosmetic** | Priority 2 | Priority 3 & 4 | Priority 3 & 4 | Priority 3 & 4 |

**Milestones:**

Projects should be broken up into milestones such as:

* Alpha: All high priority items addressed.
* Beta: Medium priority items addressed.
* Release: Low priority items addressed.

This helps keep developers on track and release better software. By the time the software is released it shouldn’t have any severe issues as they would have been addressed in Alpha. Addressing an issue does not necessarily mean fixing it, but that there has been a discussion about the issue and a plan is in place to deal with it if necessary.

**Bug Reporting**

**Bug Model**

**References:**

<https://www.lynda.com/Developer-Programming-Foundations-tutorials/Welcome/126119/146201-4.html>

http://softwaretestingfundamentals.com/black-box-testing/