**Software Quality Assurance**

**Question**: IT graduates need to have technical and problem solving skills to visualise, specify and build innovative and quality solutions to complex problems. Software Development is a profession and it’s essential to realise that there is a professional context to work and SQA is of paramount importance in this respect. There is change and evolution in terms of what constitutes a modern SQA environment and different organisations can be at different stages in terms of their environment. The changes to what constitutes SQA results from factors such as: ideas and research on SQA, technical innovations in relation to SQA such as DevOps and changes to the nature of the software systems we are building e.g. IOT, big data etc. Currently there is broad consensus in terms of what constitutes modern SQA environments.

At the onset of your professional career it’s important to have a good understanding of what constitutes modern SQA environments. Demonstrate your understanding of this by writing a white paper (2 to 3 page synopsis) outlining your views and opinions and appropriate supporting information based on the material and resources in the SQE module and your own research.

You should cite approx. three references in support of your argument and list the references at the end and the audience for the deliverable is IT management, IT project managers and senior developers.

**Testing**

**Black Box Testing**

Tests UI and Layout. User Experience. Tests in real world scenarios.

**White Box Testing**

Tests at code level

Has technical abilities of a developer

**Grey Box Testing**

**Milestones**

Alpha – High priority items addressed

Beta – Medium priority items addressed

Release – Low priority items addressed

Bugs addressed not necessarily fixed.

**Ranking Issues by Severity**

1. Crash

2. Non-Functioning

3. Incorrectly Functioning

4. Incorrectly Functioning with Workaround

5. Performance

6. Cosmetic

**Using a Matrix to Define Priorities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |

**Tracking Bugs**

Atlassian JIRA, Bugzilla, CodePlex etc

* 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 (Allows conversation to take place to preserve information for future)
* Conversation
* Attachments (eg. Screenshots or System Logs)
* Workflow Management
* Linked terms and bugs (Keeps all relevant information for a set of related bugs together)

**Test Case Management Systems**

* Managing individual test cases
* Tracking progress and results
* Multiple devs can work on bugs and provide a single report once complete

Choosing a TCM

* Ease of use
* Easy adoption
* Transparency
* Easily assign cases to suites
* Easy to determine what to run
* Easily assign test passes
* Real time reports
* How well it integrates with bug base
  + Test case should easily translate into bug report
  + Bug report should easily translate back to test case

**Bug Recording and Bug Reports**

Summary, Steps to reproduce, Results

**Bug Model**

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Number of Bugs Logged | Number of Bugs Fixed | Bug Count |
| 1 | 20 | 10 | 10 |
| 2 | 20 | 10 | 20 |
| 3 | 20 | 10 | 30 |
| 4 | 20 | 10 | 40 |
| 5 | 20 | 10 | 50 |
| 6 | 20 | 10 | 60 |
| 7 | 20 | 10 | 70 |
| 8 | 20 | 10 | 80 |
| 9 | 20 | 10 | 90 |
| 10 | 20 | 10 | 100 |
| 11 | 20 | 10 | 110 |
| 12 | 20 | 10 | 120 |