SOFTWARE QUALITY ASSURANCE ENGINEERS AND TESTERS

A DEEP DIVE FOR SKILLS-BASED HIRING

REV: 04/04/16
Occupation Overview: Software QA Engineers & Testers

Foundational Competencies

- **Critical Thinking:** Using logic and reasoning to identify the strengths and weaknesses of alternative solutions, conclusions, or approaches to problems.
- **Reading Comprehension:** Understanding written sentences and paragraphs in work-related documents.
- **Active Listening:** Giving full attention to what other people are saying, taking time to understand the points being made, asking questions as appropriate, and not interrupting at inappropriate times.
- **Speaking:** Talking to others to convey information effectively.
- **Writing:** Communicating effectively in writing as appropriate for the needs of the audience.
- **Programming:** Writing computer programs for various purposes.
- **Monitoring:** Monitoring/assessing performance of yourself, other individuals, or organizations to make improvements or take corrective action.
- **Complex Problem Solving:** Identifying complex problems and reviewing related information to develop and evaluate options and implement solutions.
- **Quality Control Analysis:** Conducting tests and inspections of products, services, or processes to evaluate quality or performance.
- **Systems Evaluation:** Identifying measures or indicators of system performance and the actions needed to improve or correct performance relative to the goals of the system.

Occupation-Specific Competencies

- **Basic General Database:** Demonstrated proficiency with SQL basics (e.g., selecting, inserting, updating, deleting records), at least one database management software application, and database fundamentals such as normalization, schemas, and relationships.
- **Basic Software Quality Assurance:** Demonstrated ability to design individual tests, create test scripts, and conduct tests and inspections of software to evaluate performance against requirements, quality, and performance. Includes the use of test tools such as LoadRunner and JMeter.
- **Basic Software Development:** Familiarity with the use of object-oriented techniques, user experience and responsive design, web mobility, back-end processes, communication tools (such as AJAX), web services (including REST), a web framework, version control, and a development lifecycle methodology (such as Agile).
- **Basic Core Coding Languages:** Familiarity with the concepts and ability to develop very basic front-end, back-end and/or mobile applications utilizing common coding languages (e.g., Java, C#, Objective C, JavaScript) on a development platform and integrating data storage (including SQL), libraries, methods, interfaces, and/or objects.
- **Basic Systems Design and Implementation:** Demonstrated ability to assist customers in the gathering of requirements and design, implement, and support simple technology solutions to existing business problems.
- **Basic Web Development Languages:** Basic familiarity with such commonly-used web development languages as AJAX, XML, HTML 5, and JavaScript.
- **Intermediate Scripting:** Demonstrated proficiency developing and using simple scripts that utilize common scripting languages such as Perl, Python, or shell scripts.
- **Basic Tech Support:** Familiarity with the use of some components of commonly-used computer hardware, software, applications, etc. and a basic ability to diagnose customer problems and provide troubleshooting and issue resolution support.
- **Intermediate Testing:** Demonstrated ability to design tests, create test scripts, ensure that test cases mimic user usage, and execute and validate unit, system, and performance test routines for a team; Demonstrated ability to use appropriate test tools.
- **Intermediate General Information Security:** Demonstrated ability to install, configure, troubleshoot, test, and maintain in a secure manner the portion of the IT environment under their responsibility (networks, communication, hardware, software, and other devices) to ensure their confidentiality, integrity, and availability.

Job Description (Example)

Develop and execute software test plans in order to identify software problems and their causes.

- Work in a cutting edge agile software development environment as a Software Test Engineer ensuring the quality of the architecture of our Enterprise software applications.
- Write automated tests at the services/integration and UI layers, using software automation tools.
- Leverage agile testing framework, design and code functional/integration test cases and scenarios pertaining to various components of the application that ensures the software meets the intended requirements that meets all established quality standards specified in test plan.
- Execute test cases and scripts; perform data validation; implement and maintain regression test cases.
- Assist developers with debugging defects/Prod tickets by reproducing them in the test environment; log and track problems (bugs).
- Work closely with product owners to understand product specifics and to create corresponding test plan, quality standards, and strategies for the project; participate in the constant improvement of our test automation and continuous integration practices.

Activities (Example List)

- Design test plans, scenarios, scripts, or procedures.
- Test system modifications to prepare for implementation.
- Develop testing programs that address areas such as database impacts, software scenarios, regression testing, negative testing, error or bug retests, or usability.
- Document software defects, using a bug tracking system, and report defects to software developers.
- Identify, analyze, and document problems with program function, output, online screen, or content.
- Monitor bug resolution efforts and track successes.
- Create or maintain databases of known test defects.
- Plan test schedules or strategies in accordance with project scope or delivery dates.
- Participate in product design reviews to provide input on functional requirements, product designs, schedules, or potential problems.
- Review software documentation to ensure technical accuracy, compliance, or completeness, or to mitigate risks.
<table>
<thead>
<tr>
<th>Most Common Required Competencies</th>
<th>Most Common Break Point Competencies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Quality Control Analysis:</strong> Understanding how to identify and test system boundaries, then conducting tests and inspections to evaluate quality or performance; leveraging basic concepts of coding to find areas of weakness (unintended values, system architecture, process flow); ability to focus on both micro and macro components of a product's success.</td>
<td><strong>1. Quality Control Analysis:</strong> See previous.</td>
</tr>
<tr>
<td><strong>2. Critical Thinking:</strong> Using logic and reasoning to first identify the requirements and purpose of a program to determine potential weaknesses; capturing and prioritizing the impact of identified bugs; taking on viewpoint of multiple possible users.</td>
<td><strong>2. Active Listening:</strong> Taking time to understand the points being made, asking questions and clarifying details as appropriate; incorporating details from interactions into testing plans and reports; drawing appropriate conclusions using previous experiences and knowledge.</td>
</tr>
<tr>
<td><strong>3. Communication:</strong> Applying reading, writing, and speaking skills to gather detailed system requirements for testing, thoroughly and accurately recording results, and presenting back to appropriate parties.</td>
<td><strong>3. Critical Thinking:</strong> See previous.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Most Preferred Competencies</th>
<th>Most Hard-to-Find Competencies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Systems Evaluation:</strong> Identifying measures or indicators of system performance and the impact of a change within the system, relative to the goals of the system; differentiating between perceived and actual performance value; familiarity with individual components and their connection to overall system.</td>
<td><strong>1. Programming:</strong> Understanding the basics of computer programming (e.g., system structure, end-end data flow, application processes) to plan and act on testing procedures; ability to mimic the mindset of a developer during testing.</td>
</tr>
<tr>
<td><strong>2. Active Listening:</strong> See previous.</td>
<td><strong>2. Critical Thinking:</strong> See previous.</td>
</tr>
<tr>
<td><strong>3. Critical Thinking:</strong> See previous.</td>
<td><strong>3. Complex Problem Solving:</strong> Identifying unique issues and reviewing related information or patterns to categorize into larger system bugs; chasing issues back to origins and thoroughly recording possible causes; understanding interaction of smaller components in overall system.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Most Evolving Competencies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Programming:</strong> Evolution driven by rapid change in the landscape of technology and its application to all aspects of business; changes make it important to be aware of how new architecture, processes, and tools are integrated into applications to inform new ways of testing.</td>
</tr>
<tr>
<td><strong>2. Critical Thinking:</strong> Evolution driven by expanding number of possible end-users as more software packages are created with customization in mind; changes make it important to capture the multitude of ways software may be used in testing and bug identification.</td>
</tr>
<tr>
<td><strong>3. Complex Problem Solving:</strong> Evolution driven by software becoming progressively more multi-faceted, with greater interconnectivity both internally and externally; changes make it important to expand how you design and implement QA testing (applying more efficient testing tools, expanded user pathway testing, increased reliance on multiple data connections).</td>
</tr>
</tbody>
</table>
## Prioritized Occupation-Specific Competencies: Software QA Engineers & Testers

### Most Common Required Competencies

<table>
<thead>
<tr>
<th>Rank</th>
<th>Competency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Intermediate Testing</strong>:</td>
<td>Proficiency with how to design tests, create test scripts, ensure that test cases mimic user usage, execute and validate unit tests, and use appropriate test tools for their own changes. Proficiency with system and performance testing.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Intermediate Scripting</strong>:</td>
<td>Demonstrated ability with the use of common scripting languages such as PERL, Python, or shell scripts.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Basic Tech Support</strong>:</td>
<td>Familiarity with the use of some components of commonly-used computer hardware, software, applications, etc. and a basic ability to diagnose problems and provide troubleshooting and issue documentation; ability to independently operate technological tools.</td>
</tr>
</tbody>
</table>
### Occupation Deep Dive: Software QA Engineers & Testers

#### Job Titles Within This Occupation
- Quality Assurance Engineer
- Quality Assurance Analyst
- Quality Assurance Tester
- Software Test Engineer
- Software Tester
- Quality Assurance Specialist
- Software Quality Assurance Engineer
- Product Assurance Engineer
- Quality Assurance Director
- Software Quality Engineer
- Automation Engineer
- Web QA Engineer
- Mobile QA Engineer

#### Certification and Education Preferences (Example)
- Certified Software Quality Analyst (CSQA)
- Certified Software Test Engineer (CSTE)
- ISTQB Certified Tester, Foundation Level (CTFL)
- Quality Improvement Associate Certification (CQIA)
- Certified Software Test Professional (CSTP)

#### Tools Used (Example List)
- SQL
- JAVA
- LINUX/UNIX
- Oracle
- Scrum
- Python
- Agile Development
- Lode Runner
- HP Quality Center
- Selenium
- Browser Technology
- SOA
- Jira
- Team Foundation Server (TFS)
- GitHub

#### Other Relevant Foundation Competencies
1. Critical Thinking
2. Reading Comprehension
3. Active Listening
4. Speaking
5. Writing
6. Programming
7. Monitoring
8. Complex Problem Solving
9. Quality Control Analysis
10. Systems Evaluation
11. Active Learning
12. Systems Analysis
13. Science
14. Judgment and Decision Making
15. Operations Analysis
16. Coordination
17. Time Management
18. Troubleshooting
19. Social Perceptiveness
20. Technology Design
21. Learning Strategies
22. Operation Monitoring
23. Instructing
24. Persuasion
25. Installation

#### Other Relevant Occupation-Specific Competencies
1. General Database
2. Software Quality Assurance
3. Software Development
4. Core Coding Languages
5. LINUX/UNIX
6. Systems Design and Implementation
7. Basic Web Development Languages
8. Scripting
9. Microsoft Stack
10. Tech Support
11. Microsoft Office
12. Business Process and Analysis
13. Project Management
14. Validation
15. Core Operating Systems
16. General Data Techniques
17. IT/Hardware
18. UI/UX Design
19. General Networking Tools and Concepts
20. Business Solutions
22. Network Protocols
23. Database Administration
24. Graphic Design
25. Web Development Concepts