**Career Summary**

I am an accomplished and result-driven Sr. Release Management Engineer with vast experience in different software systems. A hands-on software engineer who provides performance-oriented contributions and has a bias for action.

**Summary of Qualifications**

* Widespread experience in the field of Software Engineering.
* Exposure to a wide array of software products and interfaces such as Web, Desktop, Mobile.
* Utilized the pyramid model of testing. This encompasses: -
* & At the bottom of the pyramid- a maximum number of unit tests
* & At the middle of the pyramid- a layer of API tests.
* & At the apex of the pyramid - a small layer of GUI (Graphical User Interface) tests.
* Used multiple programming languages and various testing frameworks to automate the test cases.

**Technical Skills**

|  |  |
| --- | --- |
| **Core competencies** | Strong collaborative skills, analytical skills, defect management, RCA, Process Improvement, QE Audit |
| **Programming Language** | Java (Sun Certified Java Programmer), MATLAB, .NET, C, C++, C#, Android SDK Testing, IOS ,Python |
| **Framework** | JUnit, Selenium, WebDriver, SoapUI, JMeter, TestNG, FuncUnit, Mocha, Cucumber |
| **Bug Tracker and Testing Tool** | Bugzilla, JIRA, HP Quality Center, VersionOne, Zephyr |
| **Continuous Integration (CI)** | Jenkins, CircleCI |
| **Operating System** | Windows, Linux, Mac |
| **Web Technology** | JavaScript, Perl, HTML, DHTML, CSS, DOM, XML, Dreamweaver, Python |
| **Database** | MySQL, Oracle |
| **Web/Application Server** | Apache |
| **Networking Concepts**  | TCP/IP Protocol, UDP, FTP, OSI Model, Client-Server Architecture |
| **Integrated Development Environment (IDE)** | Eclipse, IntelliJ IDEA, Visual Studio, Xcode |
| **Version Control** | Perforce, Tortoise SVN  |
| **Build Tool** | Maven, Testflight, Firebase |
| **Cloud computing platform** | AWS (Amazon Web Services), Azure |

**Professional Experience**

**Principal Software Engineer and Release Management**

**June 2022 to Present**

* In this role, I work on their web applications and mobile applications. I am responsible for the end-to-end development including functional spec, development, QE, release management and maintenance.
* In terms of release management, we have a dev complete date (no new feature code to be submitted after this date). 1 week after that we have code complete date (no new code to be submitted after this date). All P1 and higher bugs need to be fixed by dev complete date After code complete date, only critical (release blocker) bugs can be fixed. A release candidate branch is cut and regression testing is performed on it.
* validate OKTA adaptive multi-factor authentication and authorization used by applications accessed externally and from iOS/Android devices.
* Mobility testing on CYIPS, Waypoint, eLogbook mobile apps.
* Involved in testing application on different mobile platforms such as iOS.
* After regression is complete, a Go/No go meeting is held between QE, Dev and Release Management to determine if the release candidate is ready for shipping, delayed or rejected. If the team members determine that a release candidate is ready for shipping, we got a phased release. In this we initially roll out build to internal customers. Then 10% of our external customers. Then progressively to all our customers. During this phased roll out we monitor for critical defects and crashes (especially for the mobile builds). Depending upon the SLA requirements of this monitoring, if some aspects are Not met (like too many crashes, a P0 bug etc.), we might roll back the release.
* We build our services in the Back End using microservices architecture. There is an API gateway which handles the client requests and directs them to the corresponding service.
* We are a SAFe Agile shop.
* I introduced a process for requirements gathering, requirements review and mapping the requirements to customer needs.
* I introduced a process for Sprint Retrospective. In this we discussed what went well and what things we could improve upon. Based on improvement points, we spun action items and improved our velocity by 20%.
* I improved the QE operational process ranging from weekly updates, better RCA (Root Cause Analysis) on customer bugs, efficient defect logging, and advocating for QE as a shared responsibility. I created a new process around test case management using Zephyr. The guiding principle is that only new test cases need to be logged in Zephyr to prevent test case duplication. I created a new process around mapping test cases to user requirements and updating the traceability matrix.
* I along with my SDET engineers and QE Managers created automation from scratch for the IOS product. We used Appium as our core framework. We built the model classes and CI/CD pipelines. We prioritized P0 test cases and components that were used most by our customers. Through automation projects, we were able to reduce our regression time by 60%.
* Weekly iOS and Android release management, working with all teams across mobile platform, communicating the update of the process.
* ​Technology stack.
* **Web Front End:** React.js. Back End: Java/Spring Boot.

Mobile (Android and IOS) Front End: React Native. Mobile Back End: Firebase

**Database:** PostgreSQL.

**Release Manager**

July 2020 to May2022

**Responsibilities**

* Qualtrics, a subsidiary of SAP, provided premier experience management solutions across the corporate spectrum. I managed the QE Department across 4 teams: TextIQ, Statwing, Odo, and Salesforce.
* In terms of release management, we have 2 kinds of releases. A Beta release and a General Release. During the Beta release, the build is only exposed to internal customers. The SLA’s are lighter (execution of all manual tests, only P0 bugs fixed etc). Based on the beta release, we receive feedback from internal customers and the QE’s log defects.
* Once the feedback is incorporated and bugs are fixed, we prepare for the General Release. The SLA’s are more strict (all required tests automated, all bugs P1 and higher fixed, no release blockers). Regression testing is conducted on the release candidate. Once a release candidate is approved, the final sanity checking is done on the release date. A phase release is done to roll out to different brands.

**My role covered these aspects:**

* & sourcing and interviewing candidates,
* & driving quality process for each product team,
* & mentoring and taking care of my direct reports.
* In terms of sourcing and hiring, I was sourcing for good candidates for open positions on my team since our recruiting team was spread too thin. In terms of interviews, every Engineering Manager had to take a certain quota of interviews to fulfill their interviewing OKR. I took 10-12 interviews every quarter and I interviewed Quality Engineers, Developers, Support Engineers across the junior, mid-level, and senior levels. As an org, our goal was to build a high-performing diverse team.
* In my org, I had new members give feedback on our onboarding process so that we could improve it iteratively.
* In terms of driving the quality process for each product, I had regular syncs with Dev Managers, Dev Sr Managers, peer Developers, and other stakeholders to know their pain points and resolve their asks. Apart from focusing on the granular testing activities (finding bugs, automating use cases, tracking Sprint work), I worked on strategic quality initiatives. These include conducting QE-specific RCA’s to find gaps in the Dev-QE practices. Writing quality vision documents and NorthStar documents to identify a vision for our quality procedures and incrementally improve them. Assess the test maturity models of our Dev components across functional, CI/CD, proper env usage, and escaped defects benchmarks. Have Dev-QE collaborative sessions and encourage a customer-focused mindset across the team. Our testing stack comprises an inhouse Python-based framework.
* ​​I conducted multiple RCA’s (Root Cause Analysis) and 2LA’s (Second Level Analysis). Based on the trends and patterns, I suggested action items for the Dev and QE teams. I spun time driven mini projects around these action items and this lead to a reduction of customer bugs by 65%.
* In terms of KPI tracking, I kept track of the following metrics: average test pass rate, test fickleness report, test pyramid numbers, QE effort for each feature, monthly customer defect count across various brands, sentiment analysis of engineering data.
* I take care of my direct reports and keep them motivated and happy. I do this by regular having 1:1’s with them (once a week to biweekly meetings), identifying crystal clear goals for them, resolving their ambiguities, setting expectations with the Dev team and other cross-functional stakeholders, working on issues that they escalate to me, and are unable to resolve themselves, empowering them to contribute to RCA’s, empowering them to enhance the existing QE process, understand their operational pain points and resolve those.
* Have the QEs involved earlier in the SDLC process so that design flaws and bugs could be identified early in the release process.
* To ensure my direct reports were meeting expectations, I regularly solicited feedback from their peer Devs and Dev Managers. I did a monthly OKR assessment to ensure that they are on track for their deliverables.
* As a part of people management, I promote and recognize great talent. It is important to retain great talent for long-term success.
* I had skip-level meetings with my Quality Engineers to ensure that they are taken care of and to solicit feedback about their Engineering Managers.
* ​Technology stack.
* Web Front End: React.js. Back End: Node.js. Kafka was used to process distributed data between services.
* The web resources are containerized using Docker and hosted on AWS (Amazon Web Services).
* We automated out DevOps test automation (automating the testing of builds in the DevOps process) using a Tricentis module.

PyTorch was used to build our sentiment analysis module.

QE Manager

**Feb 2020 to July 2020**

**Responsibilities**

* B2W One is a premier unified platform that allows proprietors of the heavy civil construction industry to accurately estimate bids, execute complex projects, and save historical data for future reference.
* Although this role was titled as a QE Manager, it was closer to a QE Director role. I was managing the entire QE Department (25 people including an outsourced team in Russia) and reporting to the CTO. I enhanced existing QE processes to create a lean QE life cycle that ably supports the SDLC. Example- I noticed that there were some gaps in the bug reproduction steps and in some cases there was an ambiguity in understanding between the Dev and QE members. I added a process wherein the reproduction steps would be aided with a video to remove any conceptual misunderstandings.
* I did hands-on Quality Engineering work, which includes testing the product, logging defects, and automating the use cases. Example- When I joined the company, there was no API automation and contract-level testing. My team and I did this project in 3 phases. 1) Identify all the API calls. 2) Do a manual iteration of all the endpoints in Postman. 3) Migrate these calls in an automated Java-based framework. This increased the code coverage by 35%.

**QE Lead**

**June 2018 to Feb 2020**

* I hold biweekly Sprint retrospective meetings with my team. We discussed what things went well and what things could we improve upon. Based on this discussion, we come up with course correction measures to incrementally improve our team output.
* 2 main metrics which I keep a track of are Velocity and Predictability. Through Velocity, I assess the quantified output of my team. Through Predictability, I assess how accurately my team members can estimate tasks
* Tested the Bot Manager product. Bot Manager is an advanced bot detection/mitigation mechanism. It can enable our customers to prevent bot-related damages at the edge server.
* Tested the Security Center product. It is a web-based dashboard that provides data on security events, traffic, and attack activity, all separated into 4 categories: DoS attack, application-layer attack, bot activity, and custom rule activity. API testing is done through in-house Java-based frameworks written in TestNG. GUI testing is done through a Selenium-based framework.

**Quality Engineering Lead**

**February 2013 to May 2018**

**Responsibilities**

* My role as a Technical Lead has 2 aspects. The 1st aspect is to provide an individual technical contribution to the Quality Engineering aspects of the Software Product. This includes testing the product for bugs and writing automated test scripts to enhance the test regression suite. The 2nd aspect is a management role, in this, I provide guidance to my direct reports and ensure that they are performing their duties in an efficient, methodical manner. I need to ensure that all my direct reports are happy and satisfied and they always remain excited so that they give high output without overworking.
* Worked with cross-teams such as Software Development and Usability to hash out the requirements and the design of the product. Represented the quality aspect in such meetings. I attempted to discover the quality flaws as early as possible in the Software Development Life Cycle. Worked on the Application Deployment Team. Worked in an agile environment. Wrote code in MATLAB to test native libraries. Tested the user interface as well as the backend API. Tested the web interface. Automated the test scripts to reduce dependency on manual verification. Wrote multiple test frameworks from scratch. The desktop-based test frameworks were built using the Java Robot library.
* The web-based test frameworks were built using Selenium. Refactored the existing test code to make it more efficient. Wrote test scripts in FuncUnit and Mocha to test JavaScript-based products.
* Suggested some modifications in the workflow process whereby the efficiency of the team can be improved. This included paper prototyping and integrating a QE-only sprint to keep track of the QE-specific work. Under this system, I encouraged the team members to granulate their bigger projects into smaller pieces and work on completing one milestone at a time. The QE Sprint was closely tracked, and we ensured that the team members were Not over-committing.
* Kept track of different testing metrics such as failed test case percentage, number of market bugs per developer, critical defects percentage. Interacted with offshore and onshore teams to track project deliverables.

Sr Quality Engineer June 2012 to Feb 2013

**Responsibilities**

* Sabre Holdings is one of the world’s largest travel technology companies. Worked on the SabreSonic web team which primarily focuses upon the JetBlue corporation. Worked in an Agile environment. As the primary POC, I had to manage offshore teams and ensure that they were up to date with the expected deliverables. Wrote functional test plans and executed end-to-end testing. I tracked defects found during the software release phase. I created mock data which was used as test data in the software development life cycle. Implemented performance testing. Extensively tested the application for security flaws. Used Selenium RC to execute automation, wrote selenium-based scripts in the TestNG framework to automate the test cases. Used tools like HP Quality Center and Verson1. Used SoapUI to test RESTful web services.

Sr Quality Engineer November 2011 to June 2012

**Responsibilities**

* Worked on the Yahoo product “Apt”. It is a web-based application that provides advertising solutions to third-party sponsors. Worked on an internal Yahoo product, Media Planning Application. This application provides customer-specific data to Yahoo users and gives an overview of the Yahoo advertising entities. Worked in the UI framework development team for re-factoring the automation code for the product, Apt. Used tools like AX247 (Automation Test Execution Engine), Bugzilla, Test Manager.
* Implemented UI automation for these products. Wrote selenium-based scripts in core Java. JUnit framework is applied to perform functional testing and unit testing. Used Selenium RC, Webdriver, and JMeter to execute the automation. On the local machine, Maven is used to build the test scripts. Verification was done by executing test scripts on VMs (Virtual Machines), located in remote servers. Used soapUI to test web services. Applied Cucumber tool to shift the paradigm towards Behavior Driven Development (BDD). Cucumber is a tool that uses a language, Gherkin and this tool allows the usage of plain text descriptions as automated tests. Worked in an Agile environment.

**Certifications:** Sun Certified Java Programmer