



Software Crafter – QA

Work in an exciting environment on software projects from NGOs to Banks, from IoT to Cloud, with a culture that focuses on quality, teamwork, and individual growth.



Do not print this document.
Don't waste paper.
Save the environment.

Who we are

Nelkinda Software Craft, founded in 2015, is a culture leader in quality software development. Through our project and coaching work, we are advancing the way by which organizations develop software. Our clients range from startups to multinational banks. Having organized, hosted, and supported more than 100 events in the software development community over the past 5 years, we are one of the most active contributors to the international software craft scene.

Job Brief

Industry	Computer Software, Training, and Consulting
Occupational Category	Software Crafter QA
Employment Type	Full-time employment, 40 hours per week
Job Location	India/Pune/remote
Date Posted	2021-03-11
Probation Period	6 months
Join Date	Immediate/ASAP

Job Brief Description



Angel says: We also accept applications from applicants who do not satisfy all the aptitude requirements but have the right attitude to quickly upskill themselves.

Who you are

- You have 3-5 years of work experience and know at least one of the following languages: C, Clojure, Go, Java, JavaScript, Kotlin, Lisp, Scala, or TypeScript extremely well.
- You have a passion for solving problems and providing not only workable but also maintainable solutions.
- You understand that learning is a never-ending journey. Because of that, you are always eager to learn new tools and technologies. You keep in touch with the latest improvements in software engineering by reading books and blogs, following industry leaders on social media, and watching videos. You support team members and clients on their learning journeys.
- You have strong analytical and reasoning skills based on the scientific method with an ability to structure and visualize processes and outcomes.
- You have outstanding communication skills in English and the ability to work collaboratively.
- You have a degree in software engineering, computer science, mathematics, electronics engineering, or a related field, or equivalent work experience.
- You have mastered behaviour-driven development/specification by example, acceptance tests, automation, SOLID and package principles, 4 rules of simple design, test automation pyramid.
- You practice good test management and reporting.
- You embrace change and harness it for competitive advantage.
- You do not judge tools and technologies by their age but by their usefulness and fitness for purpose. You can use old and new tools and technologies alongside each other to combine their strengths.



What you do

- You collaborate for requirement analysis with the stakeholders/BA and help the stakeholders with requirements to solve their business problems.
- You ensure a shared understanding of and compliance with requirements using specification by example and behaviour-driven development (BDD).
- You automate and maintain the test suite to ensure fast and flawless execution.
- You optimize for speedy feedback that allows running test cases and entire test suites any times a day on developer machines as well as CI/CD pipelines.

Responsibilities

- You actively involve in kick-offs/3 Amigos and desk checks so that you can provide feedback earlier in the lifecycle of product development.
- You ensure that there is minimal overlap of tests in the test pyramid and confirm that the unit and integration tests cover almost all feasible tests.
- You safeguard the quality of work and process of a team of aspiring extreme programmers by sharing your test knowledge and helping the team develop automated tests throughout all layers of the test automation pyramid.
- You help our clients to improve their software development process to achieve the same level of quality and agility as we live and breathe at Nelkinda Software Craft.

What we offer

- The opportunity to work with a large variety of projects, clients, and technologies.
 - A health insurance plan with family coverage.
 - A detailed individual growth plan and support to help you become a master software crafter.
 - An allowance for books, conferences, internet, and training.
 - An environment that allows you to grow and shine as an individual in the international software craft community through our various event platforms.
-



Nelkinda Software Craft

- A value system that ranks principles and practices higher than tools and technologies.
- Opportunities to contribute to open source projects.

How to apply

- Send your application letter and résumé in PDF or XHTML5 format to career@nelkinda.com.

How we interview

- 60 minutes video call on Discord, Google Meet or Zoom.
- Offline coding assignment plus 60 minutes video call in which you have to explain your solution to the coding assignment.
- 120 minutes remote pairing session on Discord, Google Meet, or Zoom.

How to prepare for the interview

Make sure you have your laptop ready with a good webcam and microphone or headset and that video calls work, the audio is good, and you know how to share your screen. Test with a friend.

An interview is a mutual activity, prepare a list of questions you would like to ask.

- Behavior Driven Development <https://www.agilealliance.org/glossary/bdd/>
- Cucumber <https://cucumber.io/>
- Extreme Programming <https://www.agilealliance.org/glossary/xp/>
- Flaccid Scrum <https://martinfowler.com/bliki/FlaccidScrum.html>
- Git <https://git-scm.com/>
- GitHub <https://github.com/>
- JUnit 5 <https://junit.org/junit5/>
- Manifesto for Agile Software Development <https://agilemanifesto.org/>
- Manifesto for Software Craftsmanship <https://manifesto.softwarecraftsmanship.org/>
- Principles of Object-Oriented Class Design <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>
- Refactoring <https://refactoring.com/>
- Scrum <https://www.agilealliance.org/glossary/scrum/>



- Specification by example https://en.wikipedia.org/wiki/Specification_by_example
- Test-Driven Development <https://www.agilealliance.org/glossary/tdd/>
- The Four Rules of Simple Design <https://martinfowler.com/bliki/BeckDesignRules.html>
- The Three Laws of TDD <http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd>
- The Transformation Priority Premise <https://blog.cleancoder.com/uncle-bob/2013/05/27/TheTransformationPriorityPremise.html>
- Trunk Based Development <https://trunkbaseddevelopment.com/>