

Lead Software Engineer (C, C++)

Job Description Summary

Own complex problems having dependency across services and facilitate cross-functional team interactions to drive resolution

Write code to build and enhance applications/services and promote code-reviews, code scanning, and other standard development practices to deliver high-quality artifacts to production.

Define, design, and develop procedures and solutions at a service level to meet the business requirements/enhancements

Drive prioritization decisions and trade-offs in working with product partners

Identify opportunities and build roadmaps to enhance primary service/function

Automate and simplify all aspects of software delivery and development actively evangelizing the need to automate and simplify where needed.

Drive seamless integration across all connected services to meet end-user expectations

RESPONSIBILITIES

Drive blameless postmortems culture to identify root causes of incidents and implement learnings

Introduce new technologies and architecture by following enterprise guidelines

Advocate for engineering principles outside of current organization/platform

Provide development and architecture guidance to team members

Build relationships and effective partnerships across organizations

Write and evaluate recommendations for job promotions based on an unbiased view of one's accomplishments

Conduct technical interviews for hiring engineering staff and raising the performance bar

Lead by example with hands-on approaches to demonstrate engineering excellence

Encourage staff to share and seek knowledge within their Guild/Program to drive reuse of patterns/libraries/practices and enhance productivity

JOB SPECIFIC EXPERIENCES

Has the ability to write secure code in three or more languages (e.g., C, C+, C#, Java, JavaScript) and familiar with secure coding standards (e.g., OWASP, CWE, SEI CERT) and vulnerabilities

Has skills in building applications using open frameworks to achieve reuse and reduce development times (e.g., Spring Boot, Steeltoe, Angular, DXP, others)

Understands internals of operating systems (Windows, Linux) to write interoperable and performant code

Able to perform debugging and troubleshooting to analyze core, heap, thread dumps and remove coding errors

Has skills to document and coach team on the development practices and coding guidelines (e.g., branching, peer reviews, library use, logging, scanning rules, test-driven development, error handling)

Understands use cases for advanced design patterns (e.g., service-to-worker, MVC, API gateway, intercepting filter, dependency injection, lazy loading, all from the gang of four) to implement efficient code

Has skills to undertake a technical review of code across applications and their dependencies to look for anti-patterns and promote continuous refactoring

Understands and elaborates technical debt and operational issues to drive prioritization discussions with stakeholders to improve the run experience

Understands system architecture to plan for platform and infrastructure capacity (e.g., database, compute, network, storage) and drives the dependency prioritization to reduce the delivery lead time

Has skills to understand customer journeys and ensure a Mastercard good experience by continuously reducing Mean time to mitigate (MTTM) for incidents and ensuring high availability (99.95% as a starting point)

Has skills to simplify deployment and eliminate software and infrastructure snowflakes using standardized platforms, ephemeral instances, and automation

Has skills to orchestrate release workflows and pipelines and apply standardized pipelines via APIs to achieve CI and CD using industry-standard tools (e.g., Jenkins, Bamboo, AWS/Azure pipelines, XL Release, others)

Able to configure rules and build automation for code with vulnerability scanning and software composition analysis using standard tools (e.g., Sonar, Checkmarx, Nexus, JFrog XRay, Veracode, others)

Has skills to define, organize, and report on test runs for major, minor, and hotfix releases (including unit, component level, system level, customer journeys, past customer issues, and regulatory controls)

Has skills to conduct various performance tests (e.g., load, spike, breakpoint, endurance) to

Understands application/service limits and behaviors