LTIMindtree

Point of View

Embracing a Fail-Fast Culture through Resilience Engineering

The recent explosion of digital transformation initiatives has made the adoption of a distributed architecture for building cloud-native applications de facto. There is a sheer increase in the technical complexity for delivering an anti-fragile business-critical application. Embracing a fail-fast culture that conforms to stringent NFRs on scalability & high availability has become a necessity.

Resilience Engineering practices play a vital role in building and sustaining the resilience and reliability characteristics of the applications. IT Resilience cannot be made possible by just adopting operational strategies like moving to the cloud, auto-scaling of infrastructure resources, disaster recovery plans, etc. The strategies for assuring IT resilience are not popularly adopted by businesses until the impact is realized. A recent article from Mckinsey states, “IT resilience is still not considered as a business problem until it has a financial impact through customer attrition, or they are called out by regulators.”

Instead of bringing a strategy for fixing the problem reactively, Resilience Engineering brings a holistic strategy for prevention, protection, detection & recovery. It helps build a delivery culture that designs & develops applications “First Time Right” with resilience characteristics and adopts continuous validation strategies for operational excellence. It helps in injecting failures and promotes early and effective recovery-driven software development practices.

IT disruptions can be majorly avoided by adopting the below key Resilience Engineering interventions,

Resilient Architectural Design – Adopt distributed architectural constructs for ensuring redundancy, fault tolerance, auto-scaling, load-balancing, high availability, failover & sustainability. Avoid cascading failures through the adoption of design patterns like timeouts, circuit breakers, fall-back, rejection, idempotent operations, etc.

Technical Debt Reduction - Create awareness & adopt best practices to control technical debt. Implement strategies to prioritize continuous measurement and reporting of technical debt. Reduce code debt to avoid 'code smells' by including refactoring activities as part of the 'definition of done'. Practices like infrastructure debt control through IaaC mechanisms & test debt control through test automation help.

Early & Continuous Chaos Engineering - Strengthen the NFR validation strategies and promote early chaos experiments as part of the pipeline to provide early feedback. Start with a smaller scope and increase the blast radius as you shift right in the SDLC to validate the resilience mechanisms. Build & validate the hypothesis by injecting failures at various levels including application, network & infrastructure to validate the IT system preparedness to handle unknown disruptions.

Full-stack Observability - Embrace the power of observability solution for end-to-end visibility of failures. Quicker detection and effective root cause analysis of failures will help reduce Mean Time to Detect (MTTD) & Mean Time to Repair (MTTR). A unified correlated view of logs, events, traces & metrics becomes vital in a hybrid cloud environment running thousands of microservices to quickly nail down the failure reasons and fix the issue to increase the system uptime.

Site Reliability Engineering – Effective balance of release velocity and system reliability through effective error budget policy and ensuring well-connected feedback loops are established to well utilize the wisdom of production. Promote canary deployments. Track SLOs (Service Level Objectives) continuously to enhance customer experience and accelerate TOIL automation opportunities to improve system efficiency and achieve high availability SLOs.

Instead of a mere focus on operational aspects that are prevalent in the market, Resilience Engineering aids in bringing a cultural change to foresee technology problems and build an IT system that can handle unpredictable failures, validate the resilience mechanism early and continuously and promote self-healing capabilities to achieve zero-touch operations.

About Author

Ramya Ramalinga Moorthy is associated with **LTIMindtree** as **Industrialization Head - Performance & Resilience Engineering practice** She carries 19+ years of experience in Non-Functional Requirements compliance space helping several fortune 500 clients with engineering strategies to validate their systems against Performance, Scalability, Availability, Capacity, Security, Resilience & Reliability. She is a certified SRE consultant and ethical hacker. She is a renowned author & conference speaker.