

WHAT FINANCIAL BUSINESS LEADERS OUGHT TO DEMAND FROM DEVOPS

Mr. Ritesh Srivastava,
Vice President - Engineering, Quadratic Insights Pvt. Ltd.

In many ways, it is not inaccurate to describe today's most successful financial organizations as IT organizations with banking & insurance licenses, rather than as financial organizations with IT departments. The competitive pressure from smaller, specialized, online companies is forcing all players to continuously improve in terms of IT.

Innovation and speed-to-market of financial products have often been hampered by IT factors, such as long lead times for starting projects and for moving them into production; late stage code merges leading to software quality issues; manual handovers and approvals; infrequent version releases; and poor coordination between Development and Operations.

Financial projects have so much at stake that wise IT engineers will hand-pick DevOps priorities to match each project's business objectives and exposures. Unless business leaders in financial organizations are aware of what exactly to demand from the IT DevOps teams (internal or external), they do not have true control over today's complex projects.

Compared to other industries, financial industry business leaders need to place an unusually high priority on the following aspects, while rolling big data, AI, automation, advanced analytics or software development projects.

Maintainability:

Improved analyses, features, and data must roll out very frequently, with very low latency (delay). Financial project requirements, hence the corresponding software requirements, need to be agile and change over time. New analyses need to be performed; new financial instruments need to be made available to model. Whatever the need, your code may very likely need to be modified. To handle such changing needs successfully, your software needs to be designed

from the beginning with extensibility in mind. It needs to rely on a combination of strong type systems and thorough testing, to make rapid and major refactoring possible.

Data integration:

The ability to ingest and digest a never-ending stream of new information; the more feeds (atomic inputs) that can be handled, the better. One common theme in financial projects nowadays is the wide array of data pulled in, its variable quality and formats. Your offering may require interaction with anything from live feeds from a stock exchange, legacy files from a mainframe, Microsoft Access databases, or the Twitter firehose. To work with such varied data, it is important that all your software first cleans up, or sanitizes, the data that it needs. Your software should use strict parsing rules. Your software should come with a monitoring solution that notifies you when it sees inexplicable data in one of your data feeds.

Automated Deployment:

At Quadratic, our DevOps teams routinely huddle with developers & data scientists, to understand the key aspects of design. This understanding helps them use the right automation & configuration tools to build the complete pipelines for continuous integration & continuous delivery (CI/CD). With such an approach, we can deploy 50-node dynamic clusters in just a few hours.

Automated Testing:

An uncaught mistake or security loophole can quickly run into hundreds of thousands of dollars in cost, sometimes much more. In case your software build systems allow you to release code that has not yet been through your test suite, or even worse, they allow you to be unsure of whether the released code was tested, you are entering the land of nightmares. Your software teams and vendors must be aware that excellent Quality Assurance Automation is a core part of your build systems. At Quadratic, we follow Test Driven Development [TDD], which when combined with continuous integration and

continuous delivery (CI/CD), allows code updates to be released as frequently as daily. An assurance from your software team or vendor that they "write quality code in the first place" is great but is not a substitute for automated testing. Multiple tools exist today that help with automating penetration testing, security testing, and performance testing.

Compliance, regulation, and auditability:

To minimize chances of malfeasance, sensitive systems should always be modular in design, and unrelated components should run in separate processes -- ideally in separate virtual machines (VMs) separated by firewalls. A defect, code injection, privilege escalation, or social-engineering attack on one service or component should still not be able to tamper another service. While building deployment blueprints, Quadratic DevOps teams work closely with the architects to ensure that this practice is followed fully.

A good engineering process is not completed until all solutions are online and running with high quality, integrated with their data feeds, storage, networks, and administration systems. Such processes involve elaborate DevOps steps, such as dynamically creating and using virtual servers, tools, storage devices, accounts, software, and network configurations.

Let me conclude by saying that there are a lot of ways to build financial software, some good, many poor. Given the complexity of systems today, and the need to be quick, agile and responsive in the market, financial business leaders need to consciously set high-level expectations from DevOps (both from internal or external development teams).