VIEW POINT

DESIGNING QUALITY BLUEPRINTS FOR PRODUCTION STABILITY IN FINANCIAL SERVICES

Infosys®
Navigate your next

It is nearly midnight, alert messages arrive in rapid succession followed by a phone call. A low voice says, "A critical service related to fund transfers is not responding. Investigations have begun, but we may need help from other teams including someone from testing."

Financial firms looking to enable digital transformation are adopting microservices-based architecture and integrating with many third-party service providers in order to drive seamless customer journeys. But the reliability of IT systems depends on internal systems as well as external services. While robust software testing approaches ensure stability/reliability of internal systems, these may not cover the entire breadth of external services.

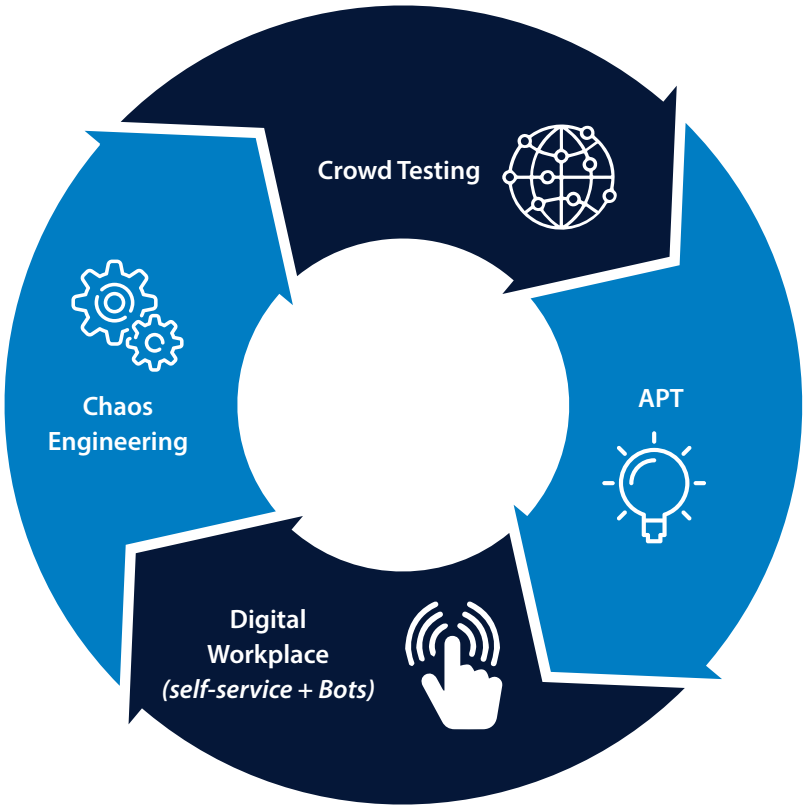
Ultimately, financial services institutions will have to deal with unplanned outages like the one described earlier. Application managers dread situations related to failures in payment services, non-responsive online banking, loan processing delays, etc., and strive to avoid them. Thus, banks and financial services firms are re-imagining their validation strategies to support frequent testing in production environments and preempt outages by identifying the causes.

Achieving Internal and External Reliability – A blueprint for Testing in Production

Traditional software testing approaches can validate whether the software or system can survive the artificial realm of test environments – but not the real world of production. With the emergence of new trends like No-Code/Low-Code solutions, SaaS and Cloud options, entire systems are being developed closer to production, thereby pushing the boundaries of testing.

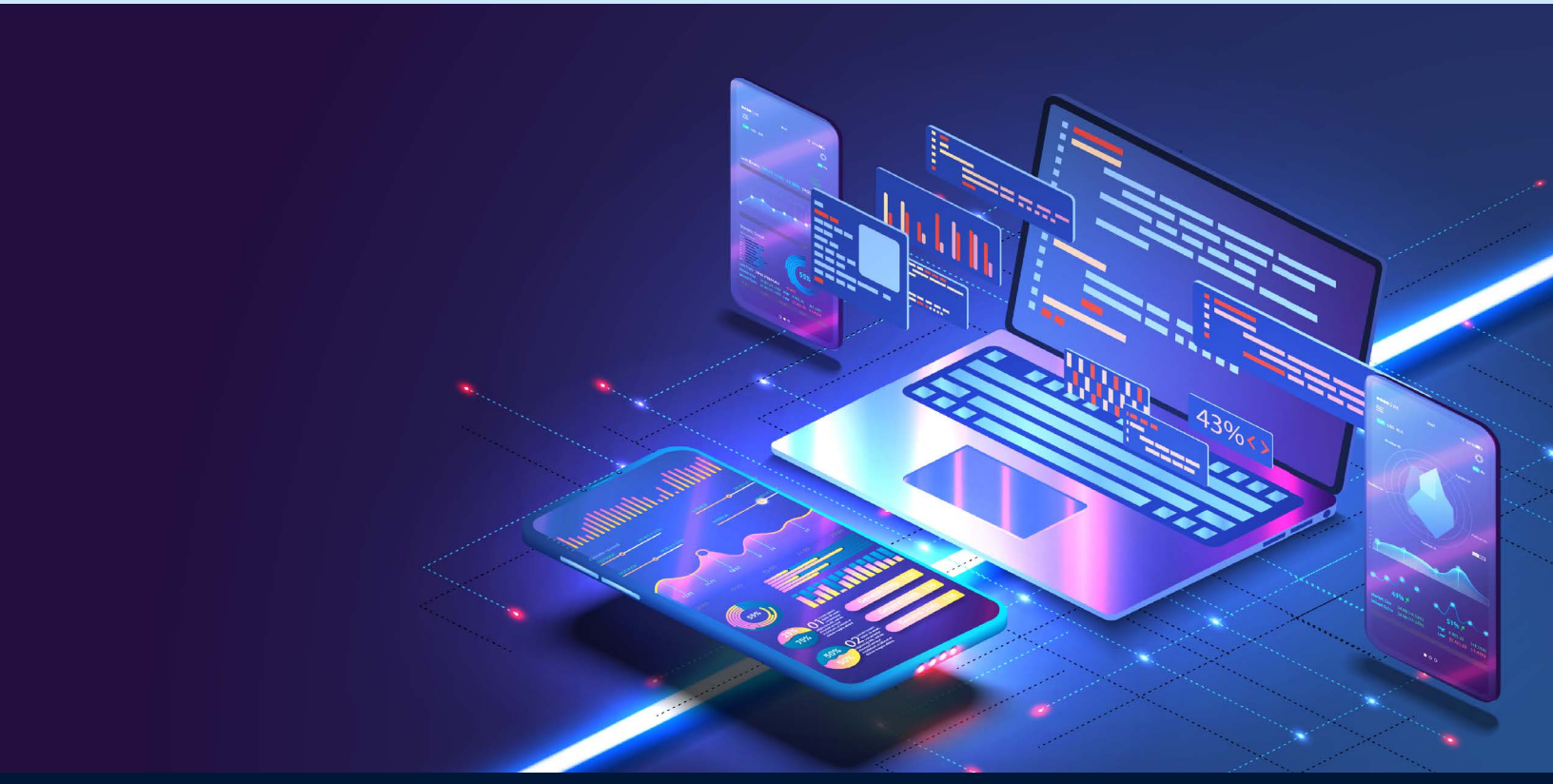
First, let us see how testing and production stability are related. There is a relationship between Testing and Mean Time to Repair (MTTR): While passing a test (or a series of tests) does not prove reliability, failing these tests indicates low reliability (1). Systems spend more time in operations than in development and quality assurance. Thus, systems built for quality assurance and not for production lead to higher costs (2).

A successful approach is to identify the types of tests that will detect problems earlier, thereby preventing production incidents and achieving zero MTTR.



Here are some test strategies that establish stability and reliability of a system in production:

- **Automated production testing (APT)** – Automated production testing is a set of automated tests that is carefully curated to ensure that core features of any given application are up and running. These automation suites are modeled to be executed in a production live environment and enriched regularly based on production issues.
- **Self-service workplace** – An APT suite alone is not enough since the operations team cannot execute this on-demand. The next step is to make these tests available on a self-service workplace for on-demand execution. All processes related to running the suite like script failure analysis, execution monitoring and result reporting should also be automated using robotic process automation (RPA).
- **Chaos engineering** – While first two approaches focus on the functionality of a system in production, chaos engineering is about testing the resilience of distributed architecture. These tests measure the ability of a system to absorb the impact of a problem in one or more parts while continuing to provide an acceptable service level. Typically, these tests cover resources (CPU, I/O, memory), state (processes and hosts) and network (latency and DNS). They require specific tools like Gremlin to simulate the attack (3).
- **Crowd testing** – While above three strategies are system-centric, crowd testing is a strategy aimed at gathering real-world feedback in real-time to deliver a quality digital experience to end customers. Some common use cases are using resources from the gig economy to test digital and connected experience or conducting beta testing for client facing web/mobile apps.



An Infosys customer, a large asset servicing and wealth management company, has been using this blueprint for the past couple of years to drive huge cost savings by preventing over 300 production incidents. Currently, the first two levers of the blueprint (APT and self-service) have been implemented across a portfolio of nearly 150 critical apps. The other two levers – chaos engineering and crowd testing – are being deployed.

About the Authors

Sreekanth D,

Practice Engagement Manager at Infosys, focused on implementing AI & Automation solutions in Software Testing

Amy Dixon,

AVP, Client Services at Infosys, focused on Quality Engineering and Digital transformations for large enterprises

References

1. Google SRE - https://sre.google/sre-book/table-of-contents/

2. Michael T. Nygard - https://www.oreilly.com/library/view/release-it-2nd/9781680504552/

3. https://www.gremlin.com/

4. https://www.applause.com/

5. https://www.testbirds.com/services/quality-assurance/

For more information, contact askus@infosys.com

Infosys®
Navigate your next