CASE STUDY

Revamping Legacy Systems: A Custom Dashboard for Real-time Visibility

SQA Group partners with insurance company to enhance visibility and streamline workflows

CHALLENGE

American Family Insurance, in the first of year of a multiyear project to modernize the tech stack for their core business applications, was experiencing coordination and integration challenges. Most of their legacy systems had been written as stand-alone applications, with basic and oftentimes manual hand-offs between components.

With 47 Scrum teams organized in 8 major value streams, understanding the dependencies in a complex, multi-stage process was a challenge. Integration testing was fragmented and error-prone, with releases frequently facing compatibility problems. Nearly 50% of defects found were regression errors – i.e., a fix in one place had unintended consequences in another section of the code base. The team needed real-time understanding of the qualitative state of the code and readiness of components to be promoted to the next step in the path to production.

IMPACT

The team at American Family Insurance, regardless of organization level of specialty, gained instant visibility into:

- The state of their area of responsibility
- Upstream dependencies affecting their area and downstream areas they were impacting
- An intuitive visualization of the relative severity of all bottlenecks throughout the entire project
- A real-time reporting mechanism for communicating status and tasking project-wide

The time it took to alleviate bottlenecks dropped from weeks to hours, and the project regained velocity in terms of how fast code could move to Production.

SOLUTION

American Family Insurance partnered with SQA Group to develop and implement a low cost of ownership consolidated and fully automated dashboard integrating outputs from multiple development and testing streams, build and release pipelines, and unrelated management/reporting systems that created a “single source of truth,” as well as provided a continually updated real-time snapshot of the state of the entire project.

The heart of the dashboard was a cloud-native custom graphical display which retrieved data from the dozens of applications and systems in use via a dashboard web service in Python running Lambda functions. Major data sources included a JIRA service for feature development status and Zephyr service for test development and execution and a Metrix database housing data. Data structures were rendered dynamically using Dynamo DB. Because of the extensive use of just-in-time Lambda functions and non-persistent data sources, the footprint of this complex, end-to-end system was very low.

This system included an algorithmically-driven and graduated color-coded readiness display, giving in one view the “hot spots” for all scrum teams, all value streams, and all build and release pipelines contributing to a program increment. The component objects of the Dashboard were clickable to subordinate drilldown views providing real-time details regarding all steps in the path to production, allowing active edits of the feeding systems by viewers with appropriate permissions. It also included a reporting structure with dozens of standardized reports and a toolbox for building custom reports.