ELECSOFT

CASE STUDY

INTERNAL ENTERPRISE API PLATFORM – CONTINUOUS TESTING FRAMEWORK

THE CLIENT

The client was a leading Canadian financial services company, known primarily for its life insurance services. They had initiated a new program to modernize their software architecture, providing internal applications with a standardized framework to interface with each other.

THE CASE

Without a standardized framework the client's software architecture was causing time delays and security concerns in how data was shared amongst internal applications. The client aimed to solve this problem with an internal Enterprise API Platform that would allow organization wide applications to share data and back-end services with greater ease. Internal applications, such as the client's Insurance Claims Application, were onboarded to the platform and used as the baseline for performance and security metrics.

Initially, the client's internal teams were manually validating the API integrations from each application that was to be onboarded to the platform. This was leading to lengthy test cycles and test management challenges due to conflicting application needs and strategies. The manual testing was also not sufficient in detecting cross-cutting security concerns among the API Platform and consuming applications.

The client needed independent and unbiased QA services to support the transformation process for their platform APIs and API adoption. They needed their QA provider to develop a standardized automation framework that would significantly reduce test cycles and accelerate platform adoption across the organization.

THE SOLUTION

Elecsoft was engaged to develop a thorough automated QA framework for the client. **Elecsoft**'s team of QA engineers and members of the client's Testing Centre of Excellence (TCOE) reviewed, evaluated, identified test cases, and developed a comprehensive automated QA framework for non-functional requirements.

Elecsoft strengthened the client's automation strategy by developing baseline requirements from existing application load, performance, and security metrics, and was responsible for developing and executing scripts to test 2-3 API deployments per month on a rolling basis.

The following **Automation Solutions** were delivered:

Develop Standardized KPI metrics baselines for applications consuming API platform services.
Develop CI/CD pipeline components to trigger post deployment test execution and logging.
Develop automated security validation scripts by using light 4j Proxy Framework
Develop on-demand chaos monkey test automation suite with validations of service recovery and expected behaviour.

TOOLS USED

SwaggerHub	**ReadyAPI**	**Light 4j Proxy Framework**
Jenkins	**JAVA Scripts**	**RESTAssured**

QA DELIVERABLES

Document Overall Test Strategy and Test Plan
Document the test scenarios/cases for all in-scope systems
Co-ordinate and execute Non-functional testing (integration, Load, Performance, Security) for all in-scope systems
Lead daily triage calls for each phase of testing
Provide daily test execution status on progress of each phase of testing
Test Closure Report

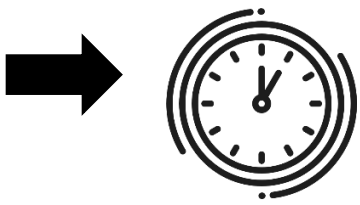
THE OUTCOME

The suite of test handlers and automatically generated tests created over 100% efficiency in testing cycles as entire test pipelines could now be run in under **5 minutes** as opposed to **5 days** with a manual approach. The testing pipeline also became a key accelerator in the client's continuous delivery model and was used for each deployment from development to production, enabling quick platform adoption on a monthly basis. The reduced test cycles and accelerated platform adoption resulted in time saving, cost cutting and an overall more secure interface for applications throughout the organization.

Manual Approach

5 DAYS

Automated Approach



5 MINS