



Case Study: Front-End Trading Applications Testing for Major Brokerage

1. BACKGROUND

Front-end applications are a major part of any trading platform. However, nothing exists in isolation, and our preferred approach is to test the system as a whole, which calls for tight integration between testing of front-end and back-end components of a trading platform.

This case study summarizes the approach Allied has taken when tasked with testing front-end applications for a major US brokerage, while also examining test coverage / test suite structure and the optimal test team composition.

2. CHALLENGE

Front-end offerings for the Trading Platform included the following electronic order channels differentiated by levels of sophistication and serving different Retail¹ customer segments:

- ▶ Basic Web client
- ▶ Automated phone channel
- ▶ Wireless channel
- ▶ Advanced Web client
- ▶ Basic Desktop client

¹ In this document we are focusing on Retail customer-facing electronic order channels only. Allied has, however, also tested Institutional clients and internal electronic order channels.

- ▶ Advanced Desktop client

Testing responsibility for the channels highlighted in bold (Active Investor segment) resided with Allied Testing. The other channels were owned by different testing organizations. Therefore, one of the challenges Allied faced was to establish a high level of coordination between the various channel testing teams involved in major releases, because all of the channels utilized the same mainframe back-end, while Active Investor segment channels also utilized distributed back-end components.

Types of releases:

- ▶ Mainframe back-end releases affecting both Non Active Investor and Active Investor channels
- ▶ Distributed back-end releases affecting Active Investor channels only
- ▶ Active Investor client releases that could also include changes in both mainframe and distributed back-end components to support new or modified functionalities (which, in addition to testing of the updated clients, also required testing with the production-current clients to verify backward compatibility)
- ▶ Proprietary Tickerplant releases that affected market data firm-wide

Given the frequent release schedule, Allied had to find a balance in testing of the various clients that would provide the necessary level of testing support while avoiding duplicate work efforts. Part of the answer to that challenge was building a robust automation framework, while another aspect of the solution was to optimize the team structure and the test scope.

3. ALLIED APPROACH

3.1. FRONT-END FEATURE SETS

From the front-end application perspective, the major functionality groups Allied specialists dealt with were as follows:

- ▶ Order Entry and Order Status for Real Orders -- Equity, Single-leg Options, Multi-leg Options:
 - Non-directed and Directed Orders: Routing options including basic compliance Smart Order Routing, Advanced Smart Order Routing, and major DMA venues
 - New orders, cancel/replace orders, and cancel orders
 - Basket orders
- ▶ Order Entry and Order Status for Virtual Orders:
 - Risk Management features such as Alerts and Conditional Orders, Bracket Orders, Trailing Stop Orders for Equities and Single-leg Options
- ▶ Balances, Positions (including Cost Basis and P&L, Portfolio Management tools), Executions / Transactions History
- ▶ Account access (Authorization and session management, including Web-Links automatic authentication for access to Basic Web client)
- ▶ Communication Tools (Urgent Messages and Online Chat)
- ▶ GUI (Navigation paths and screens configurations, customization options)
- ▶ Market Data Tools (Level 1, Level 2, Option chains, Charting, Watch Lists, Decision Support Tools, Back-testing tools, Market Scanners, etc)
- ▶ Deployment: Install and Update (specific to Desktop clients only)



- ▶ Cross-platform compatibility: hardware and software requirements, OS/SP, Browser, JVM, Silverlight, .NET, etc., (some elements were specific to Advanced Web client only)
- ▶ Cross-channel compatibility (e.g., verify that orders entered in Web client can be viewed and acted upon in Desktop client, compare Balances and Positions data between different clients, etc.)

From the back-end testing perspective, the Allied team also dealt with:

- ▶ Individual back-end components and system integration testing
- ▶ BackOffice testing
 - Risk monitoring tools
 - Regulatory Reporting: FINRA OATS, NYSE OTS, SEC 606 Order Disclosure, etc.
 - Various EOD and SOD processes: Balances and Positions reconciliation, Post-trade reports, Settlement export files, etc.
- ▶ Failover testing
- ▶ Performance testing

3.2. TEST TEAM COMPOSITION AND ALIGNMENT TO BUSINESS DOMAINS

It was obvious from the start that expert-level testing was required due to the high level of sophistication of the features in our front-end applications. Going in with “one size fits all” pool of testers was unpractical. Therefore we split our testing team into several sub-teams as follows:

- ▶ Front-end Test Team
- ▶ Trading Test Team (including BackOffice and Virtual Orders sub-teams)
- ▶ Market Data Test Team

With this team structure each team was aligned to a corresponding Business Domain, which allowed for sufficient degree of specialization.

However, the division of the Allied team into Front-end, Trading, and Market Data Domains did not mean that only the Front-end Test Team participated in the testing of new Active Investor client releases. Our approach was to have team members trained to the SME level in their respective domains, while at the same time keeping a required level of proficiency in other domains for back-up. This allowed Allied Testing to go “all hands on deck” during fire-drill situations, while being able to perform expert testing in the business-as-usual mode.

The other sub-teams included Automation Team and Environment Support Team.

Allied’s programmers developed a custom automation framework that removed the need to have scripting skills in order to be able to create new and update existing test cases. Our SMEs were able to run automated tests, implement and modify their test cases by changing data matrices used by the automation framework, while Automation Team was only responsible for framework maintenance.

4. RESULTS

- ▶ The feature-sets identified above served as the basis for mapping the test suite structure.
- ▶ The expert sub-teams were involved in developing test cases for corresponding Business Domains / Feature-sets, which allowed for building an extensive and coherent test suite for

each front-end application. At the same time, cross-domain training ensured that everyone was up to the task when Allied faced short-term workload spikes.

- ▶ Allied identified areas suitable for automation, and developed the automated test scripts from scratch. Our robust automation framework allowed us to avoid the costly process of developing manual test cases, converting them to automated test cases, and then having to involve developers to implement changes to the test cases.
- ▶ The Allied team was able to achieve an overall test case automation rate of about 70%, getting as high as 95% in certain areas such as Order Entry / Order Status. This freed our expert testers from the major part of routine regression testing and allowed them to efficiently focus on the more complicated functional testing.
- ▶ Allied's SMEs were heavily involved in release planning schedules, which allowed us to tailor the test scope for specific changes Mainframe and Distributed back-end releases. By analyzing individual release items we were able to prepare custom test suites for each release and avoid the necessity of running full regression tests for each client as we combined extensive testing of one client with touch tests of the other clients. In addition to that, Allied took the lead in coordinating the efforts of the other channel testing partners by providing them with clearly defined functional test cases.