

Confident Innovation: Transforming Software Testing to Drive Change

Rapid innovation vs growing complexity

Perhaps the most striking characteristic of today's financial services landscape is its rapid expansion towards new technologies, new use cases and their implementations. The speed of innovation goes hand in hand with growing complexity: year after year, financial technology platforms become more sophisticated – multi-threaded, distributed, handling increasing volumes of data and aiming at faster processing speeds. A wide variety of emerging technologies creates a new reality, in which new approaches and technology solutions reshape existing subject domains and cause a significant requirements shift. To claim their place in the industry, fintech players – technology providers and end-user institutions alike – have to respond to this trend. On the one hand, they need to rapidly innovate in order to keep their competitive edge. On the other hand, organisations must fulfil their obligations to their clients by ensuring the reliability and robustness of the solutions they build.

Traditional use cases vs new implementations

Technology platforms behind the financial services industry reflect the business flows and operational life cycles typical of particular financial subdomains. Traditional implementations of financial use cases are inherently complex, as is the underlying business logic. New implementations, while retaining the innate functional complexity, most commonly represent a shift towards a more hybrid and distributed architecture that provides greater flexibility and customisation.

Some examples of technological advances used to build present-day fintech applications include cloud implementations, microservices-based architectures, and flexible API interfaces. The recent launch^[1] of a digital wealth management sandbox^[2] illustrates exactly this trend: while addressing the variety of traditional wealth management use cases, the solution takes advantage of new technologies. The platform has a modular architecture: its order management, payments, KYC/AML, portfolio management and rebalancing, and tax wrappers functionalities are implemented as microservices. The connectivity leverages REST and WebSocket APIs enabling simple, fast, and secure integration with third-party applications. Another important feature of the solution is that it is provided as a cloud-native platform freeing the user from the need to manage and maintain the infrastructure.

On the business side, the platform allows the implementation of various wealth management scenarios. However, the complexity of the underlying workflows imposes more stringent requirements on the verification and validation process: with the new tech at hand, one needs a testing approach capable of matching the sophistication level of the platform under test.

Responding to change – Transforming testing

To align the benefits introduced by the new technology with the present-day requirements for reliability and robustness, the approach to testing such platforms also needs to be revised.

To show how testing should and can be adapted to respond to change, we will continue with the example of the innovative wealth management framework mentioned above and outline a test approach tailored to its specifics.

From the functionality perspective, the sandbox allows for the workflows related to automatic portfolio evaluation, portfolio rebalancing and readjustment, and other scenarios, all of which require a certain level of sophistication on the technical side.

To assess the quality of the platform, the testing tool, as well as the test library, needs the capabilities to account for the specifics of the implemented workflows.

For example, since the portfolio rebalancing functionality is governed by a set of conditions, the testing tool must be able to perform complex calculations and rule-based checks. The test approach should involve a flexible way to design a test script enhanced with capabilities of in-built actions to govern the atomic interactions with the system under test, along with an extensive modelling for data generation and implementation of multivariable conditional checks.

Another feature that may come in handy is scheduling: automated regular checks run as part of the test harness can prove useful for automatic holdings evaluation. Together with extensive modelling and rule-based checks, one can also take a step further and verify the instructions to correct the deviating portfolios.

Yet another must-have for a test harness for a wealth management use case is the simulator capability to emulate upstream or downstream systems involved in business flows. In the example use case, one of such elements is a market data feed providing financial information that serves as an input for the subsequent business flows within the wealth management system under test. When emulating a testing dataset, it is important not only to collate a representative array of real-world market data, but also to generate data inputs creating a much broader set of conditions to achieve higher coverage of test scenarios.

To align the benefits introduced by the new technology with the present-day requirements for reliability and robustness, the approach to testing such platforms also needs to be revised.

Testing to speed up development?

A test harness^[3] implemented in line with the above requirements – modelling, test data generation, simulation, scheduling – can be a powerful tool capable of providing valuable information about the test object. However, one of the most crucial characteristics of information quality is its relevance: the information is only useful if it is provided in a timely manner. Thus, it is also very important that the test harness be able to be part of the software delivery cycles and support daily releases as well as larger deployment milestones. For that, the test harness should easily integrate with the test system's CI/CD pipeline within multiple test environments that support test execution for fast narrow-scope checks as well as extensive end-to-end test libraries covering multi-day business life cycles. While test execution can be done autonomously, the test framework should allow for comprehensive analysis of test execution results to be performed by test analysts. This way, software testing does not delay time-to-market, but rather creates a continuous feedback loop increasing the delivery speed.

Disrupting the industry by staying resilient

The financial services industry is rapidly evolving: financial institutions, together with technology partners, are finding new technologically advanced ways to implement traditional and new business use cases. At the same time, the new technology landscape is shifting towards tougher requirements for dealing with software complexity, data volumes and processing speeds. Both trends generate the need for a new approach – building test tools that match the sophistication levels of modern technology solutions. With such test harnesses, market participants and technology providers will be able to innovate responsibly and confidently through a continuous feedback loop in which the development of technologically advanced software is perfectly aligned with technologically advanced software testing.

With such test harnesses, market participants and technology providers will be able to innovate responsibly and confidently through a continuous feedback loop in which the development of technologically advanced software is perfectly aligned with technologically advanced software testing.

Notes:

[1] <https://exactpro.com/news/exactpro/wealthos-sandbox-publicly-available-after-testing-exactpro>

[2] <https://wealthos.cloud/signup/>

[3] <https://exactpro.com/test-tools/th2>