BUILD SOFTWARE TO TEST SOFTWARE

Deliberate Practice of Software Testing

exactpro 🕞

TEST COVERAGE EXITUS ACTA PROBAT





Instead of running a set of test cases per each requirement, the Exactpro team structures the test scenarios around business flows within the Daily Lifecycle and production use cases. This secures the project with proper test coverage. The test library is designed based on the Theory of Everything (ToE) and considering the operating system from the End-2-End perspective, also taking into account the surrounding applications and integration with them. The captured test results are analysed with compliance checks on top of the test library execution providing both the observed deviations from the expected behaviour and paths for future coverage improvements.

CRADLE TEST DATABASE



exactpro.com info@exactpro.com Suite 3.02, St Clements House, 27 Clements Lane, London +44 (0) 203 319 1644



- Incorporated in 2009 with 10 people, our company has experienced significant growth as satisfied clients require more services; now employing over 560 specialists.
- Part of London Stock Exchange Group (LSEG) from May 2015. Exactpro management buyout from LSEG in January 2018.
- Headquartered in the UK, with operations in the US and Eastern Europe.
- Exactpro opened new R&D center in Tbilisi, Georgia in September 2018.

EXTENT Software Testing and Trading Technology Trends Conference 17 September 2019, London

extent.exactpro.com

exactpro 🌘









exactpro 🚱

iro 🚱

ctpro

IAT

....

exactpro 🚱

BUILD SOFTWART

Build Son to Test S

exactpro.com

exactpro 🕄



Believe in nothing. Even if it means testing the whole darn thing.

Build Software to Test Software



EXACTPRO'S TYPICAL CLIENT-DEDICATED TEST HARNESS TEAM ON AGILE PROJECTS





For the client, the Exactpro Team Lead and Test Harness Owner are the two go-to persons on the Team 3

4

Automation Developers and the Core Developer collaborate to enhance the test harness used by the Team and execute the developed automated tests

2 The Subject Matter Experts work hands-on with the client systems and existing test artefacts executing testing according to the prioritized work items

The Environment DevOps person ensures that the tests are configured for execution in an unattended mode



Client Onboarding and Certification



Migrating Client's FIX Customer Base to a New Trading Platform



In contrast to exchange platforms, broker systems normally allow a much wider range of possibilities for tuning the protocol for interaction with client systems.

One client can concurrently use several brokers for accessing an exchange. Brokers often try to reduce the number of changes in communication protocols which a particular client must implement.

This approach simplifies client interaction and leads to the existence of a large number of heterogeneous configurations on the broker platform end. If the internal changes of the broker platform take place, a need in a regression test occurs, to verify compatibility with the client systems.

The developed tool allows processing of the existing data from test and production environments and creating the needed library of active testing scenarios.

Production logs are loaded into the certification testing tool's database. The script generator is using the database for creating a sequence of test scenarios performed by an active test tool for functional testing. The logs, generated when test scenarios are executed, are loaded into the database to compare them with production logs.

Software Testing Humor C EXACTPRO TEST WORLD

Welcome to TEST WORLD

A World of Adventure A World of Danger A World of Fantasy Go-Live without Consequences Retrospectives without Judgements Testing without Limits



... inspired by HBO's West World



https://youtu.be/zgF4hCAnXvl



EXACTPRO TEST AUTOMATION APPROACHES: A CASE STUDY IN TRADING (FT & NFT)

These case studies discusses testing and automation challenges and solutions for the trading systems with different asset classes, features, protocols and process distribution complexity.

Exactpro tools are used for testing and monitoring a variety of financial services industry systems:



Operability, deployment, configuration, monitoring, fault tolerance, disaster recovery

Challenges:

- ➡ test coverage according to ToE (Theory of Everything) design
- control over test environments
- daily/weekly life cycle
- time consuming GUI tests
- dependency on a variety of preconditions
- ability to execute test scenarios in parallel
- ➡ production-like conditions, etc.

While designing the test library as the theory of everything concept, the analysis of the system complexity and business flows takes place. Parameterizations in each and every step of the business flows lead to a significant number of test scenarios.

Software Testing Humor 🙂

THE BEST A TESTER CAN GET



Five years ago, we launched our Build Software to Test Software tagline.

Since then, it has been an aspirational statement, reflecting standards that our teams strive to achieve.

But read software development blogs today and it's easy to believe that testers are not at their best. Many find themselves at a crossroads, caught between the past and a new era of agility. While it is clear that changes are needed, where and how we can start to effect that change is less obvious for many. And when the changes needed seem so monumental, it can feel daunting to begin. So, let's do it together.

It's time we acknowledge that brands, like ours, play a role in influencing culture. And as a company that encourages testers to be their best, we have a responsibility to make sure we are promoting positive, attainable, inclusive and healthy versions of what it means to be a tester. With that in mind, we have spent the last few days taking a hard look at our past and coming communication and reflecting on the types of testers and behaviors we want to celebrate. We're inviting all testers along this journey with us – to strive to be better, to make us better, and to help each other be better.

From today on, we pledge to actively challenge the stereotypes and expectations of what it means to be a tester everywhere you see Exactpro. In the ads we run, the images we publish to social media, the words we choose, and so much more.

As part of The Build Software to Test Software campaign, Exactpro is committing to donate \$1,000 per year for the next three years to educational organizations executing programs in Georgia designed to inspire, educate and help testers of all ages achieve their personal "best" and become role models for the next generation.

Our tagline needs to continue to inspire us all to be better every day, and to help create a new standard for junior testers to admire and for experienced testers to achieve...

Because the systems we test today. Will be the systems in live service tomorrow.

We've all got work to do. And it starts today. Exactpro. Build Software to Test Software.

https://youtu.be/m3qhZkisejs



Exactpro Test Automation Tools for Trading Infrastructures



is an active real-time tool used for exchange, MTF and

broker systems testing. It can also be used as an exchange simulator for testing post-trade systems or smart derivatives. Sailfish is a web-based application which supports simultaneous connections of multiple users to both incoming and outgoing streams. It is written in Java.





is a post-transactional passive testing tool which allows testing the back end of trading platforms, market data, post-trade, and market surveillance systems without interacting with them. Shsha lets you easily comply with audit and regulatory requirements.



is a powerful load generator built to stress test scalable high-load trading infrastructures. Load Injector supports FIX (all versions), ITCH, LSE, Native, SOLA SAIL & HSVF, HTTP, SOAP, and various binary trading and market data systems protocols. The tool's architecture is flexible and allows adding other protocols.

Load Injector is an open-cycle load generator capable of supporting both model and measurement approaches of performance testing.

Case Study: Functional Validation of Auction Algorithms Using Model-Based Approach



Test Execution Control, Monitoring and Post-Run Analysis (NFT): in order to run automated non-functional tests, it is required to monitor various metrics of the system under test

and hardware usage. There is also a need to have an option to start/stop/adjust load injection or other actions based the current system state.

Case Study: Message Processing in Reconciliation Test Approach



Challenges in Test Data Preparation for the Non-Functional Tests:



Case Study: Non-Functional Testing (NFT)



Capacity & Stress Testing

Load tests simulating production load in terms of user connections, Order flows, Trades and Market Data feeds in parallel with extensive metrics monitoring.

Failover

Resilience tests to check the system's protection against various outages of different software and hardware components and making sure of the service continuity.

Latency

How fast is our system under various conditions? Latency (response times) measurement per critical system paths (Order Entry, Market Data, Trades, etc). Building latency figures for different load shapes, injection rates and input data composition.

Daily Life Cycle (DLC) Test

The test executed in conjunction with the Functional test team:

• Take the system under test through a Production-like schedule (all trading cycles and scheduled sessions);

- Apply appropriate load during various phases;
- · Perform some functional tests under load;

• Make data consistency checks (reconcile output from various sources and check data for integrity.



Test Execution Control, Monitoring and Post-run Analysis (NFT):

in order to run automated non-functional tests, it is required to monitor various metrics of the system under test and hardware usage. There is also a need to have an option to start/stop/adjust load injection or other actions based on the current system state.



Daemons collect system info, parse logs, execute commands, control test execution. Router is used to communicate with daemons.

Data Processor



Grafana



TestManager: Automated execution of test scenarios, collecting and processing test information. Transform, collect and store data for future use Data visualisation Data and reporting mana

Data analysis and management

ΤM

EXACTPRO TEST AUTOMATION APPROACHES: A CASE STUDY IN POST TRADE (FT & NFT)

Delivering large Post-Trade initiatives: key challenges and innovative QA

Challenges:

A clearing system has highly complex features requiring a high degree of accuracy and skilled resources:

- Reference data;
- Risk management;
- Schedule.

There are also external parties that contribute to the complexity:

- Markets;
- Participants with different types of connections;
- Settlement and payment systems;
- Market data providers, and interventions from the operations team.



A clearing system runs some routine processes repeated every day, in a predefined order and at predefined times: - Trade upload;

- Smart computations;

Trade Capture

Market

- Settlement sessions;
- Collateral uploads;
- Reporting, etc.

The processes are dependent on each other. There are also several data inputs into the system that are kept in the system and present an additional challenge.

- There is a multitude of components in modern complex post-trade infrastructures;

- Upstream and downstream system dependency;
- The participant structure is very complex;
- Trade/Xfer/Position/Account life cycle;
- The number of Asset Classes may vary;
- The complexity of the Risk calculation process;
- Access via a set of API endpoints.







Operations Ref. Data Management Client Interfaces Client Client Client Client Client Client Client Client Client Consider Client Consider Client Consider Client Consider Consider Client Consider Consid

Regulatory Reporting Ris



Case study #1: Collateral Deficit

- trades need to invoke the initial margin call;
- margin computations are checked;
- the inputs from the participants that deliver collateral to cover the risk exposure by CCP are simulated;
- additional computations with the regulated collateral are done.

Case study #2: Delivery Default

- Multiple-day scenarios starting with a trade upload and continuing with settlement during the next days;
- Within this time, various inputs from settlement systems are simulated, where one participant fails on these positions and another one settles, etc.;
- Processes that happened after the settlement period took place are checked.

Addressing Complexity: Testing of Complex Multistep Scenarios

There is a need for testing multi-step and multi-day scenarios, there could also be a need to test several scenarios at the same time. Our solution includes using an automated test tool which allows running several scenarios, organizing the test steps in batches and aligning the test schedule with the system schedule.

Our test tool allows us to execute all the test steps in a predefined order, which facilitates aligning the testing order with the events in the clearing system. Simulators are developed and integrated with the test harness which controls both the simulators and the execution of test cases.

Case study #3: Corporate Action

- Executed over several days;
- Starts with a trade upload;
- Need to verify that corporate actions are only applied to the positions that are eligible;
- Need to check that further computations take the new and original positions into account properly.

Collateral Deficit	Settlement Failure	Corporate Action
1. Set Collateral	1. Send Trades	1. Send Trades
2.Send Trades		
G1. TRADE UPLOAD1		
3. Verity Positions	2. Verity Positions	2. Verity Positions
	3. Configure Sim to reject SI	
G2. MARGIN RUN1		
4. Receive Margin Call		3. Invoke Divident CA
5. Transfer Collateral		
G3. SETTLEMENT RUN1		
	4. Verify Positions	4. Verify Positions
G4. MARGIN RUN2		
6. Verify: No Margin Call		
G5. MARGIN RUN1 (T+1)		
7. Verify: No Margin Call	5. Verify: No Margin Call	
	6. Configure Sim to reject SI	
G6. SETTLEMENT RUN1 (T+2)		
	7. Verify Positons	5. Verify Positons
G7. MARGIN RUN2 (T+2)		
8. Verity: No Margin Call	8. Verify Positons for BuyIN	
G8. SETTLEMENT RUN1 (T+4)		
	9. Verify BuyIN SI	6. Verify Positions
	10. Verify Positons after BuylN	

EXACTPRO TEST AUTOMATION APPROACHES: A CASE STUDY IN POST TRADE (FT & NFT)

Addressing Complexity: Risk Calculation Algorithms

1. Testing of this area is a challenge due to the fact that the underlying mathematical concepts are quite complex and large volumes of data are used.

2. One solution is to test a model which is based on simplified assumptions, but replicates the algorithms and the calculations done by the system, and compare the actual results of the system under test with the expected test results computed by the model.

3. Another solution is to define a reliable set of test results from testing the previous version, define a baseline from regression and compare it with the results that come from testing the system under test.



1. We need to run a substantial number of test scenarios AND also run them in an appropriate time frame;

2. Our approach is to create different test cases and test tools which are capable of running them;





Addressing complexity:

- 1. **Simulators** are developed as a part of the test harness. Understanding the upstream and downstream systems and having the format and layouts of the interim protocols to those apps allows to emulate these streams and to continue the test design. This approach provides flexibility and independence from the availability of the surrounding systems.
- 2. The test harnesses are configured to support **APIs to mimic participants' activities**, **market updates**, other processes.
- Market data streams are defined by avoiding chaotic updates and replacing them with emulation updates. The emulation updates contain predefined datasets, so QA can rely on this "controlled" data; and design the expected behavior based on it.
- 4. **Test harness controllers** are created, especially when the system lives very complex daily, weekly and monthly cycles.
- 5. **Test schedulers** are developed to incorporate all end-to-end test scenarios into one test library run we internally call "The Big button".

A Holistic Integrated Automated Test Solution:



EXACTPRO TEST AUTOMATION APPROACHES: CASE STUDIES IN COLLATERAL MANAGEMENT

These case studies discuss testing of a collateral and liquidity management system for a leading global rates and multi-asset clearing house and a multi-national central counterparty.

Case Study 1: Testing Collateral and Liquidity Management for a Global Clearing House



Challenges:

- · To identify the impact across all available reports & outputs from Calypso;
- To identify the expected result several reports based on multiple days' market data;
- To secure proper coverage with a set of input data (for instance, account structure, its roles, collateral types, cover preferences, markets, products, rates, place of custody) which allows emulating all required combinations and cases;
- To simulate a cover distribution model keeping in mind the coverage matrix above in order to derive the expected behavior and outputs for actual results reconciliation.

Case Study 2: An End-to-End Scenario for Collateral Fee Charge to Verify that Clearing House Charges Set for Custodians are Accurate



Case Study 3: An End-to-End Scenario for Cash Withdrawal Flow



End-to-End business flow steps covering emulation and validation of the flow:

- 1. Cash Withdrawal instruction via Web UI;
- 2. Cash Withdrawal authorization via Web UI;
- 3. Calypso side: validation of related Margin Call, Transfers and Messages and relevant limit breaks on the task station;
- 4. Verifying generated MT202;
- 5. Emulating ACK message;
- 6. Checking that transfers are settled and Instruction completion message appears in Web UI;
- 7. Emulating MT950.

EXACTPRO TEST AUTOMATION APPROACHES: A CASE STUDY IN RISK MANAGEMENT

This is a case study about challenges of testing Risk Management systems and our test automation and testing approach developed and implemented for our client, a Central Counterparty responsible for clearing and risk management of CCP-eligible transactions on a leading European exchange.

Challenges:

1. Functional Area:

the underlying mathematical concepts are quite complex, and large volumes of data are used;

- 2. Calculation of Margins: Margins are calculated using:
- the MVP (Method for Portfolio Valuation) methodology for Governments Bonds;
- the MARS (Margining System) methodology for Equity Derivatives and Equity Cash products;

3. The Aim of Initial Margin (IM):

Initial Margin is called on a daily basis to cover theoretical costs of liquidation in order to close the open positions in the worst possible market scenario, within a maximum price variation range called "Margin Interval". The "Margin interval" is specific for each financial instrument;





Clearing and Settlement System

4. Intraday Margin:

intraday margins are called by the margining system if sudden sharp price variations occur or in the case of a Member's excessive overall risk exposure;

5. Default Fund:

a Default Fund is managed by the system as additional protection aimed at covering risks associated with sharp price/interest rate movements. The Default Fund amounts are calculated as a result of periodic stress tests.

Test Automation Approach:

Our Solution is to test by using a model based on simplified assumptions, but replicating the algorithms and calculations done by the system under test, and compare the actual results taken from the system under test with expected test results computed by our model.



A **Risk Simulator** was developed and integrated with our test harness created to replicate Margin Run activities, including message flows and margin values.

Smart Actions were developed to calculate margin values and perform default fund verification and reporting.

Risk Engine Log Parser and Risk Parameters Analyzer were created to analyze and verify margin and risk parameters based on input data

from both Margining and Clearing systems. It provides the user with a possibility to check the actual margin values received in the systems against the calculated expected values.

Case Study: An End-to-End Scenario to Verify the Process and Parameters of Margin Calculation for All Cleared Instruments by Using a 'Risk-Based Margining' Algorithm



DELIBERATE PRACTICE OF SOFTWARE TESTING

EXACTPRO TEST AUTOMATION APPROACHES:

Being an R3 technology partner and a member of Hyperledger, we understand the software testing challenges of a DLT-based post-trade infrastructure. We have developed a DLT post-trade case study based on open source Corda and Hyperledger technology and designed a comprehensive test automation framework adopting End-to-End testing approaches from multiple Exactpro post-trade projects conducted with the help of our core testing tool ClearTH.



A Test Automation Framework for DLT Projects

ClearTH provides an innovative way to test Clearing, Settlement and Back-Office systems. ClearTH is a unique Exactpro tool able to simultaneously execute multiple end-to-end test scenarios in batches. The tool easily detects abnormal behavior in the system under test and effectively predicts potential issues. It offers many built-in actions to cover the majority of activities in post-trade systems.



ClearTH:

- · Verifies each stage of the DLC;
- · Has an integrated schedule;
- · Automatically runs test scripts;
- · Creates multiple-day test scenarios;
- · Performs multiple concurrent tests;
- · Has integrated simulators;
- Supports SWIFT ISO protocol.

Woodpecker is a ClearTH extension. It's main features:

• It is highly customizable (data source, data types, load, time, interface);

 It is easy to transfer to different environments due to its ability to extract data right from the system under test;

Testing scenarios created without humans cover a vast majority of diverse conditions/data combinations;

• It's a productive intersection of functional and non-functional testing approaches.



Case Study: Distributed Ledger Technology in Position Update Business Flow 3 Steps on the Ledger Side:



At each step, the test framework provides different methods to validate the endpoints, inputs and outputs. The actual business flow steps and validation check items are transformed into a **ClearTH matrix**. Any deviations from the expected behaviour can be analysed once the matrix has been executed.



EXACTPRO TEST AUTOMATION APPROACHES: A CASE STUDY IN MARKET SURVEILLANCE

This case study discusses testing and automation challenges and solutions for the market surveillance systems connected to various sets of trading systems and market data providers, involving various data mining processes and alerting mechanisms, and having different degrees of process distribution complexity.

Market Surveillance System Testing: Key Challenges

A market surveillance system is a complex system that includes a set of subsystems:

- A gateway subsystem that obtains data from different data sources such as:
 - trading systems
 - market data providers
 - trade reporting systems
 - reference data systems
 - unstructured data
 - 3rd-party data vendors: FX rates/ regulatory news announcements

- 2 A data enrichment subsystem:
 - internally calculated statistics
 - analytics module
- **3** Real-time and offline alert engines to detect abusive market behaviour:
 - rule-based engines
 - advanced engines that include machine learning
 - data mining engines



A CASE STUDY IN MARKET SURVEILLANCE





A data repository to store structured and nonstructured data:

- Oracle
- in-house data warehouses
- big data technology solutions such as Apache Hive/Apache Parquet
- data lake

A GUI module that provides drill down capabilities to investigate the detected alerts:

- · real-time monitoring dashboards
- offline monitoring: Market Replay system
- third-party dashboards

EXACTPRO TEST AUTOMATION APPROACHES: A CASE STUDY IN MARKET SURVEILLANCE



Automated testing using the Sailfish test tool – real-time data

Sailfish test tool is used for functional testing. Sailfish allows us to conduct end-to-end testing for the whole message flow starting from injection of data in the upstream system via the trading interface (e.g. FIX Gateway), receiving messages via the market surveillance stream gateway, and compares the received messages with the expected ones.





Testing by comparison – offline data

If there is no way to get a response from the market surveillance system in real time via a channel, testing by comparison is used. This type of tests comprises comparing the contents of output data from the system against the actual results.

Pre-conditions/assumptions:

- we have a "master copy" of the output data
- the master copy had been carefully verified in the previous release (i.e. assumption: master copy is correct)
- · changes on the upstream system side are minimal
- · no changes in the input data



A CASE STUDY IN MARKET SURVEILLANCE



The Data Consistency part

The data consistency verification test is performed after each operational cycle of the system.

The consistency test is based on the comparison of messages from different end-points of the market surveillance system under test and the external (integrated) systems.

The following end-points are used in the approach:

- data of the downstream gateways of exchange system and trade reporting system
- · data of the input gateway of the surveillance system
- · data from the Data Warehouse of the market surveillance system

We use the Shsha tool to parse the tcpdump files captured from the external systems (if the system itself didn't store the sent messages) in order to compare the output data of the external systems and the input data of the surveillance system. It allows us to perform the consistency check for the whole message flow.

The Next Generation of Market Surveillance Systems

Realizing the advantages of cloud computing services providers (AWS/Azure, etc.), the financial industry firms are now actively migrating their solutions – including market surveillance systems – into the cloud. Exactpro test harness tools are cloud-ready and can help clients to make the migration as smooth as possible.

The financial industry is also actively leveraging machine learning and artificial intelligence techniques. An increasing number of companies tends to apply AI to market surveillance systems for:

- smart market monitoring
- prediction
- fraud detection
- pattern recognition from data based on machine learning models, instead of rules-based algorithms

To address this development, Exactpro is doing research in the field and enhancing the existing test harnesses.

exactpro.com

Suite 3.02, St Clements House, 27 Clements Lane, London, UK, EC4N 7AE