exactpro
EXITUS ACTA PROBAT

20 Simple Questions
from Exactpro for Your Enjoyment
This Holiday Season

BUILD SOFTWARE TO TEST SOFTWARE

EXACTPRO'S BESPOKE TEST TOOL SUITE

Over the years, we have built unique solutions to test our clients' software.

Do you have systems that you would like to put into live service in 2020?

Did you know that in 2020 we will migrate all our test tools to a new microservices-based platform?

Our teams from London, New York, Tbilisi, Moscow, Obninsk, Saratov and Kostroma, where we relentlessly work on improving the resiliency of our clients' platforms, join in sending you our warmest greetings this time of year. We hope your holidays are filled with all the wonder and joy only this merry season can bring.
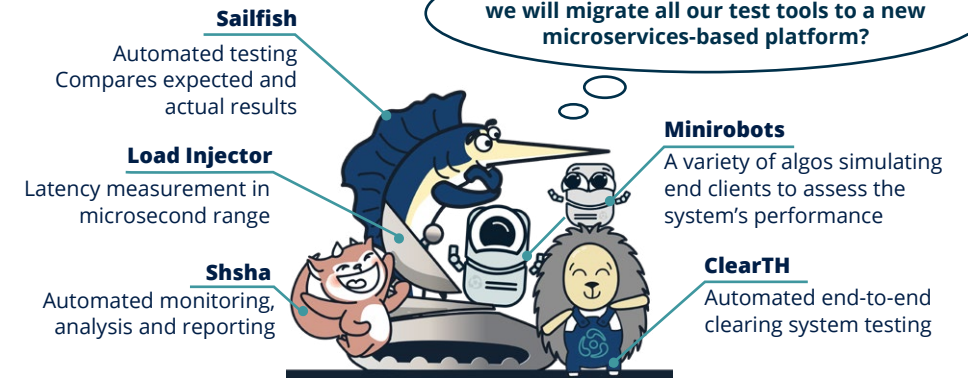
Sailfish
Automated testing
Compares expected and actual results

Load Injector
Latency measurement in microsecond range

Shsha
Automated monitoring, analysis and reporting

Minirobots
A variety of algos simulating end clients to assess the system's performance

ClearTH
Automated end-to-end clearing system testing

THE EXACTPRO CLIENT NETWORK

Toronto, London, Oslo, Stockholm, Amsterdam, Zurich, Vienna, Abu Dhabi, Nur-Sultan, Tokyo, Hong Kong, Sydney, Athens, Colombo, Johannesburg, Singapore, Jersey City, New York, São Paulo, Madrid, Paris, Milan

TESTING FOR TRADING INFRASTRUCTURES: FUNCTIONAL AND NON-FUNCTIONAL TESTING

Functional Testing: Active Real-time Testing

Sailfish is an active real-time keyword-driven test tool. The test libraries we developed for it include tests for various business contexts that range from Regulated Markets to MTFs, from Dark Pools to Clearing Houses and Brokerage Systems. They have been successfully implemented in a wide range of technical and middleware infrastructures. Sailfish can also be used as an exchange simulator for testing post-trade systems or smart derivatives. Sailfish is a web-based application which makes it possible to achieve fully autonomous scheduled test execution that does not require ongoing monitoring.

Functional Testing: Passive Testing/Monitoring and Client Onboarding

Shsha is a post-transactional passive-testing tool that allows to test the back-end of trading platforms, market data, post-trade, market surveillance systems without interacting with them. Shsha lets you comply with audit and regulatory requirements. Shsha can be used when a Client of an exchange performs onboarding certification tests.

Non-functional Testing: Stress Testing

Load Injector is a powerful load generator built to stress test scalable high-load trading infrastructures. It is an open-cycle load generator capable of supporting both model and measurement approaches of performance testing.

POST-TRADE SYSTEMS TESTING

ClearTH is a web-based application for testing post-trade systems. The tool simultaneously executes multiple end-to-end test scenarios in batches. This allows raising the level of test automation. ClearTH detects abnormal behavior of 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.

THE EXACTPRO SOLUTIONS ECOSYSTEM.
THE INTELLIGENT WAY TO TEST

Can machines learn from mistakes?

A COMBINATION OF FUNCTIONAL AND NON-FUNCTIONAL TESTING
Experience has shown that the best results are achieved at the confluence of functional and non-functional testing.

Realistic Simulation of Market Agents for Testing Purposes
Minirobots is an active multi-participant testing tool which simulates the behavior of real traders. The tool has complete autonomy in making routine decisions under specific market conditions. Depending on what the testing needs are, each of the robots can act independently or in tandem with the other ones.
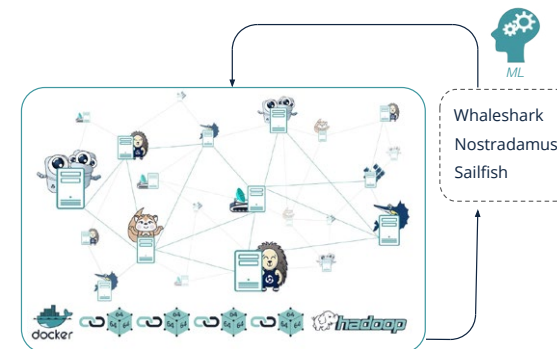
Machine learning (ML) and data science can transform the traditional approaches into new sophisticated testing techniques of the future. Among other solutions, Exactpro is building Cradle. It is a framework allowing us to store the output from hundreds and thousands of test nodes. Collecting the output does not merely come down to storing the pass/fail outcomes for every test execution. It also includes capturing the network traffic, the log files, system configurations, defects and their descriptions, etc.

Here is how Cradle works:
- upon test execution, every test node collates the data and contributes it as a block into data storage, which is part of the framework;
- the raw data is converted into sets of flat files that are structured and easily accessible via distributed queries;
- compliance checks are then performed as analytical queries against the whole test database that becomes available as a result of end-to-end testing, as opposed to running a single test case per requirement.

This approach allows our QA Engineers to maximise test coverage and create and maintain powerful test libraries, with a lot of analytical effort being put into the process.

CRADLE TEST DATABASE

Have you visited Exactpro's GitHub repository?

EXACTPRO SOLUTIONS IN OPEN SOURCE

Open Source (OS) software is gradually becoming the force driving technology transformation forward. It is being actively adopted by the financial industry as well, even though there are many obstacles to its industry-wide implementation. However, by its nature, OS is transparent, and being part of the OS community allows for many benefits. We welcome open collaboration. Our strategy is to host and continuously maintain our test tools on GitHub. As of today, we are proud to have released three of our solutions to Open Source.

Cradle will be a powerful tool to preprocess data for ML. The ML-enhanced solutions such as Nostradamus and Whaleshark are another part of the Exactpro solutions ecosystem. Here is how the ecosystem functions:
- the testing tools perform software testing based on the accumulated regression libraries and produce test execution reports;
- the reports contain both positive and negative execution outcomes. A negative outcome may be a type of bug revealed in the system. Bug reports contain a detailed description of the bug that's necessary for fixing it;
- bug reports and execution reports are valuable statistical data stored in Cradle;
- the ML-based analytical platform processes these reports to discover trends, detect and localize the reasons for failed reports, improve the testing strategy and quality, as well as significantly reduce manual interference.
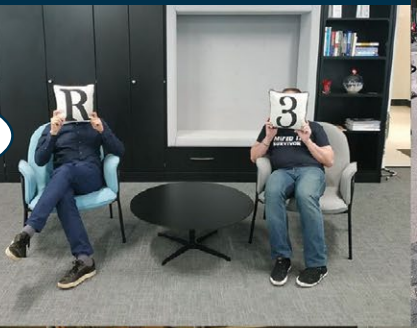
ML
Whaleshark
Nostradamus
Sailfish

The Exactpro solutions ecosystem leverages the power of knowledge about the system under test that software defects provide to project stakeholders. This helps to continuously enhance the software development and testing efficiency and ensure that our clients get the best technology that data science has to offer.
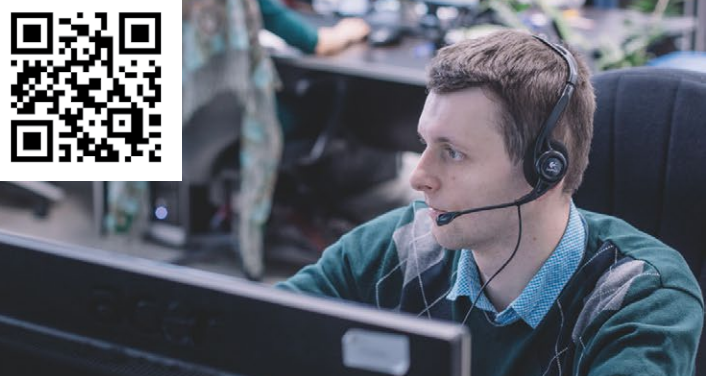
Sailfish — Functional testing of exchange, MTF and broker systems

ClearTH — Automated testing of post-trade infrastructures

Nostradamus — ML-based analysis of software defect reports extracted from defect repositories

exactpro.github.io

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

Are you tired of running in circles trying to implement Agile?

Project teams often try to superficially adopt the Agile principles. This leads to a pseudo-agile project implementation. Moreover, an Agile cargo cult develops across the organisation. It manifests itself in "fake testing" practices, such as blending testers into Scrum teams or doing upfront test automation (TDD). These ideas have a damaging effect.

A true Agile implementation lies with the adoption of the core values of the Agile methodology, which fundamentally changes the internal culture and the way the organisation or a specific project works.

Here is what the Agile workflow at Exactpro looks like:
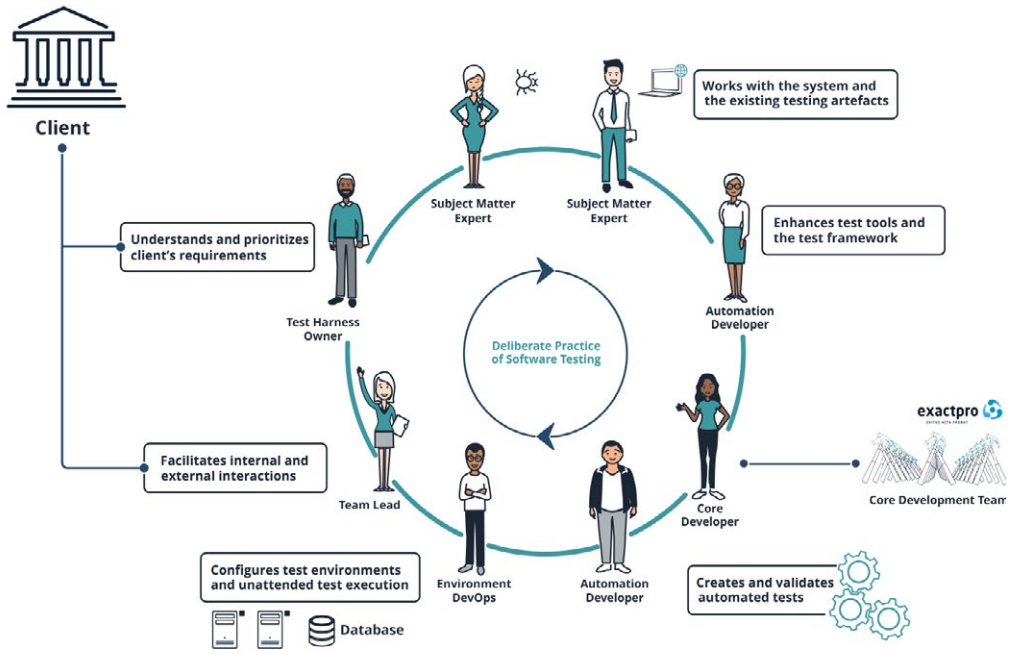
Exactpro's **Team Lead** and **Test Harness Owner** are the client's two go-to persons on the Team. The **Subject Matter Experts** work hands-on with the client systems and the existing test artefacts, executing testing according to the prioritized work items. **Automation Developers** and the **Core Developer** collaborate to enhance the test harness used by the Team and execute the automated tests. Agility is achieved via clear role distribution, close collaboration and excellent communication between the parties.

exactpro.com

ONBOARDING BENEFITS WITH SHSHA

Would you like to improve your client conformance process?

Client onboarding is a type of testing conducted to ensure the compatibility of the trading and post-trade infrastructures with transaction participants via financial industry protocols. Client onboarding is a mandatory compliance procedure for any exchange operator/broker platform and key to carrying out financial transactions. Client onboarding involves representatives of disparate organizations. Therefore, the process requires a substantial amount of coordination and teamwork. Shsha is Exactpro's proprietary passive monitoring tool that will help you streamline your onboarding processes.

Seamless automated onboarding via leveraging passive testing methods

Shsha does not affect the system under test or create additional message streams. It observes the system and analyses the results without interaction.

Increased transparency via aggregated analytics

Shsha monitors member activity and message flows, e.g. traffic capture of network connections or data from log files, and presents the results in a user-friendly way.

Support of most protocols, easy customization

Supports various industry-standard (FIX and dialects, FAST, SWIFT, ITCH, HTTP, SOAP, etc.) and proprietary (MIT, SAIL, HSVF, RTF, RV, Reuters, Fidessa OA, Quant House, etc.) protocols.

Based on typical real-life scenarios to produce meaningful results

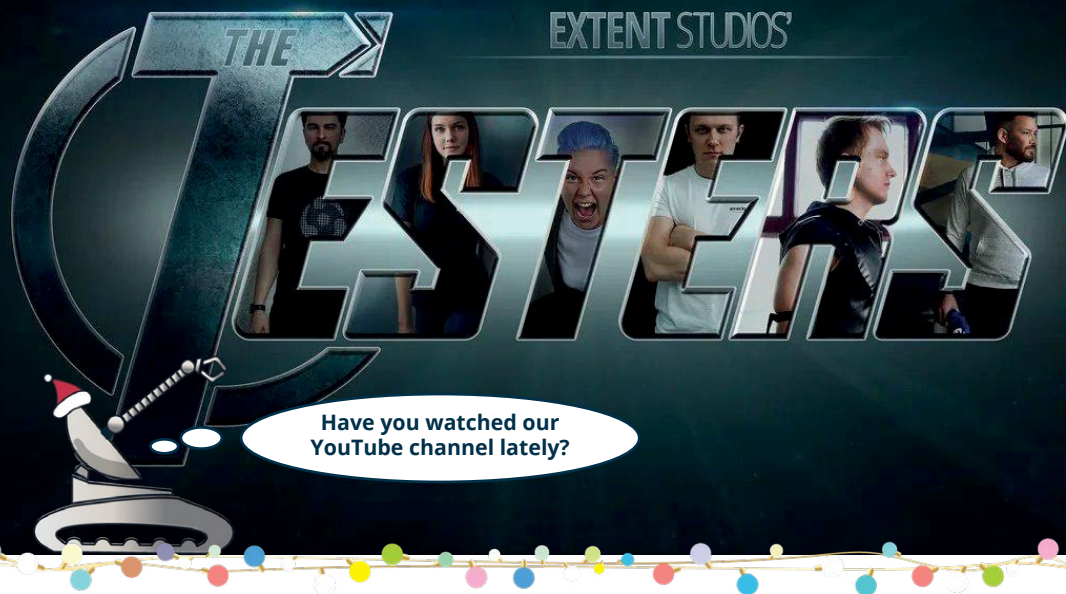Shsha's test scripts are constantly updated to guarantee regulatory compliance.

Results comparison and parallel run capabilities

Shsha allows processing of the existing data from test and production environments and creating a relevant library of active testing scenarios.

Highest-caliber QA maintenance teams

Exactpro is a leading provider of quality assurance (QA) services for market infrastructures.

Shsha has no impact on the system under test, leaves no footprint on the infrastructure, and is highly versatile in terms of protocol support. It can be used for self- and external certification.

Client Onboarding and Certification

Client Testing Environment

SOR Client C

FIX 4.2/4.2

FIX 5.0

LAS

FIX 4.2 CCP

CCP

All conformance and reconciliation reports are stored as SQL queries in a folder used by the Frontend. No changes in the application code are required to add a new reconciliation report.

When an SQL query is executed, it returns a list of captured messages and their status. The Frontend highlights the messages returned by the query.

SQL queries can also be used to return message stats, the number of logons/unparsed messages per client, etc.

Network Capture

Shsha

XML Dictionaries, used to configure decoders for particular drop/message specification

Conformance Testing Solution Frontend

MySQL Database

SQL Tables

FIX, Native and ITCH Decoders

Store PCAP files into a folder, accessible from Conformance Solution

FIX 5.0 Trading (TCP)
FIX 5.0 Drop Copy (TCP)
FIX 5.0 Post Trade (TCP)
Native Real-Time (TCP)
Native Recovery (TCP)
ITCH/FAST Replay/Recovery (TCP)
ITCH/FAST Real Time Channels (UDP)
SWIFT ISO 20022/15022

Client Systems Passing Conformance

Client A

Client B

Create a folder with PCAP files for every day, so client onboarding team historical data

All captured messages with timestamps and tags

Unparsed messages - failed protocol level validation

Individual tables for each message type

PCAP files upload history and processing stats

EXTENT STUDIOS'

THE TESTERS

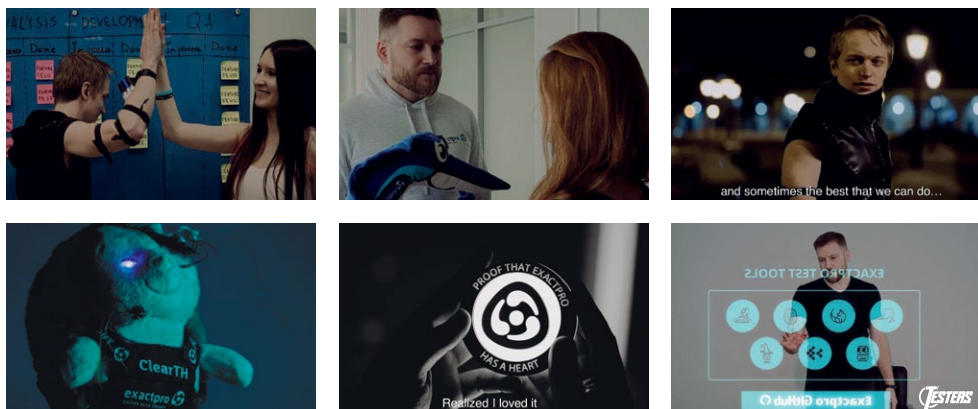**Have you watched our YouTube channel lately?**

Exactpro Testers: Endgame

The grave course of events set in motion by the latest change request that wiped out half the functionality and fractured the automated test ranks compels the remaining Testers to take one final stand in grand conclusion to twenty-two sprints.


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

Do you have an end-to-end integration testing environment?

This case study discusses testing and test automation challenges and solutions for 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
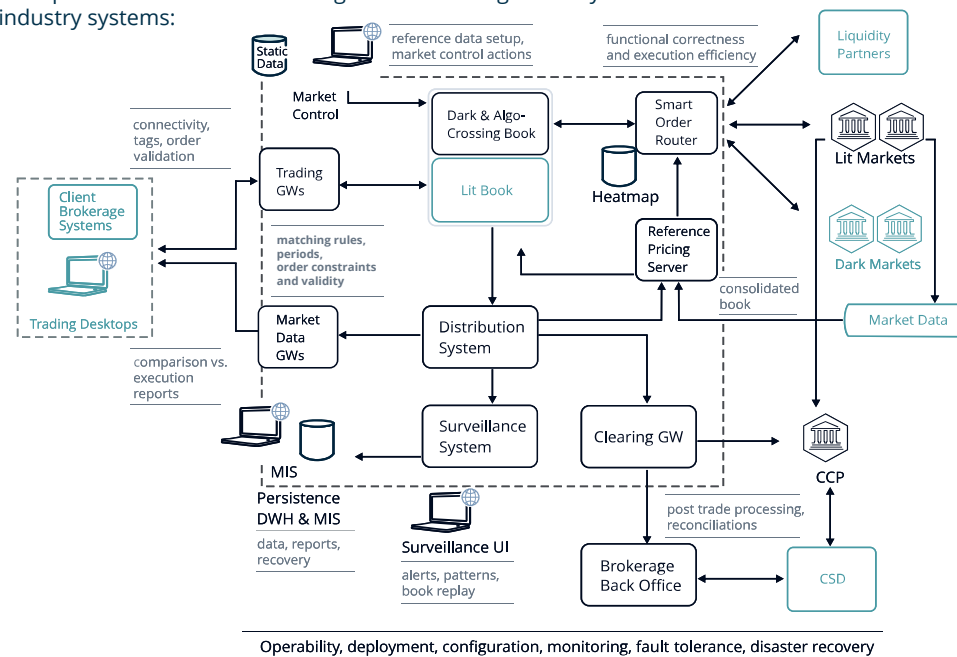
Challenges:

➡ 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.

The method of designing the test library in line with the theory of everything allows performing the analysis of the system complexity and business flows. Parameterizations in each and every step of the business flows lead to a significant number of test scenarios.

DELIBERATE PRACTICE OF SOFTWARE TESTING

Nostradamus
DEFECT REPORT ANALYSIS

Software defects:
noise or hidden value?

exactpro
EXITUS ACTA PROBAT

Nostradamus is an open-source tool for analyzing software defect reports extracted from defect repositories. For export of bug descriptions, the tool can work in XML format. The tool uses Machine Learning (ML) techniques to determine important links between various defect attributes and generate certain bug metrics, such as the probability of a bug being fixed/rejected, including time to resolve. Nostradamus calculates various statistical data including distributions and values of aggregate function. It can also build the top of the most frequent terms and the top of the most significant words, etc. This knowledge allows to achieve various IT-related goals, for example:

• more accurate planning and goal setting for project managers;
• improving the defect report quality for QA analysts and junior analysts;
• discovering hidden development dependencies for system architects and developers.

Nostradamus is written in Python and uses its libraries such as pandas, scikit-learn, numpy and scipy.

The tool's architecture is as follows:

Exactpro's Nostradamus tool is available on GitHub at
https://github.com/Exactpro/nostradamus
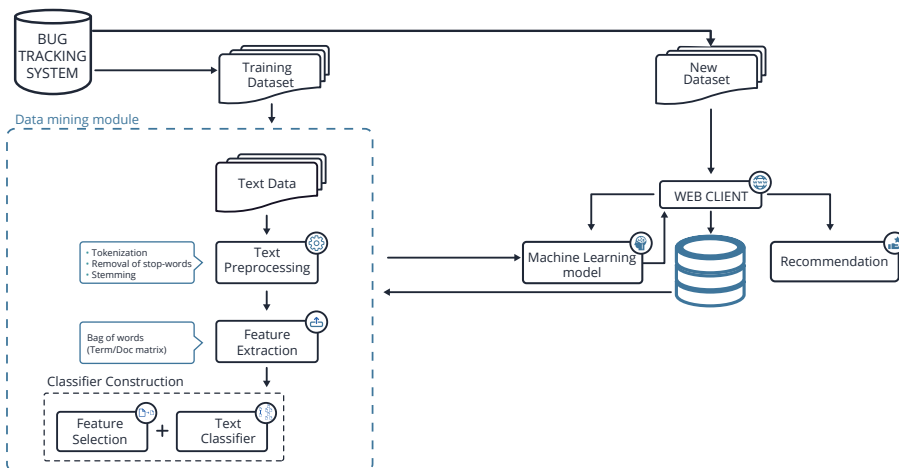
PREDICTIONS

Nostradamus calculates various probabilities based on a text that has been entered. These probabilities are calculated according to built-in models that consider the specifics of the uploaded bug reports. Every text change leads to a change in probabilities of all indicators. Thus, the submitter can evaluate the text of the description. He/she can also improve it by increasing or lowering the probability of a certain indicator. In the Nostradamus GUI, the calculated probabilities are presented as pie and bar charts.

Nostradamus is capable of calculating the metrics based on the analysis of a single bug report or a set of bug reports.
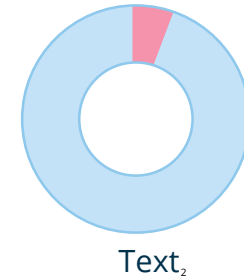
$Text_1$ = It is impossible to deploy application because of missing libraries. Please investigate possible reasons of this unexpected situation. This issue appears reguarly in the new version.

$Text_2$ = It is impossible to deploy application because of missing libraries which are not picked up from the drools build. Please see the following exception and steps to reproduce.

Rejected Pie Chart   Rejected / not Rejected

Rejected Pie Chart   Rejected / not Rejected

$Text_1$

$Text_2$
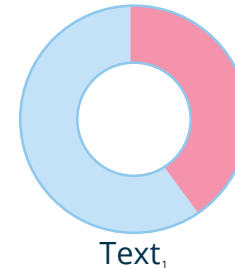
RECOMMENDATIONS: Significant Terms are Highlighted
To help the defect submitter with improving the description, the significant terms are highlighted.

It is impossible to deploy application because of missing libraries. Please investigate possible reasons of this unexpected situation. This issue appears regularly in the new version.

Resolution — Rejected
Priority
Areas of testing

EXTENT CONFERENCE *talks*

TMPA EXTENT CONFERENCE

GeoSTQB

Exactpro opened its first office in Georgia in September 2018. Since then, creating and nurturing the Quality Assurance community of professionals in the region has been one of Exactpro's top priorities.

In 2019, two **EXTENT Talks** conferences have taken place. EXTENT Talks bring together IT specialists working in various industries and seeking professional growth, practitioners from IT firms, as well as Quality Assurance enthusiasts of all backgrounds interested in actively participating in IT events in the region.

In November 2019, the International Conference on Software Testing, Machine Learning and Complex Process Analysis (**TMPA**) was held in partnership with Exactpro, Springer and ACM Sigsoft at Ivane Javakhishvili Tbilisi State University. TMPA is an annual scientific conference committed to exploring innovations in software testing, verification and analysis, including the applicability of the latest data science methods to the analysis of software quality.

Exactpro is proud to have initiated the creation of a new local ISTQB member board – **GeoSTQB**, the Georgian Software Testing Quality Board. The Board will promote the importance of professional excellence among software testers and will soon administer certification of IT professionals in Georgia according to the highest international standards.

EXTENT CONFERENCE

SOFTWARE TESTING AND TRADING TECHNOLOGY TRENDS CONFERENCE

An annual forum for sharing innovative trading and post-trade technology ideas and expertise among specialists working in the global financial markets industry.

The EXTENT-2019 agenda featured the following sections:

- Post Trade and Distributed Ledger Technology
- Non-functional Testing in the Cloud
- Regulatory Impact on Software Testing
- Resilience of Financial Market Infrastructures
- Agile Software Testing and DevOps
- Artificial Intelligence and Software Testing

The EXTENT series events highlight the latest trends in developing program and hardware platforms used at exchanges, brokerages, investment banks and other trading participants, focusing on quality assurance and efficiency of such platforms.
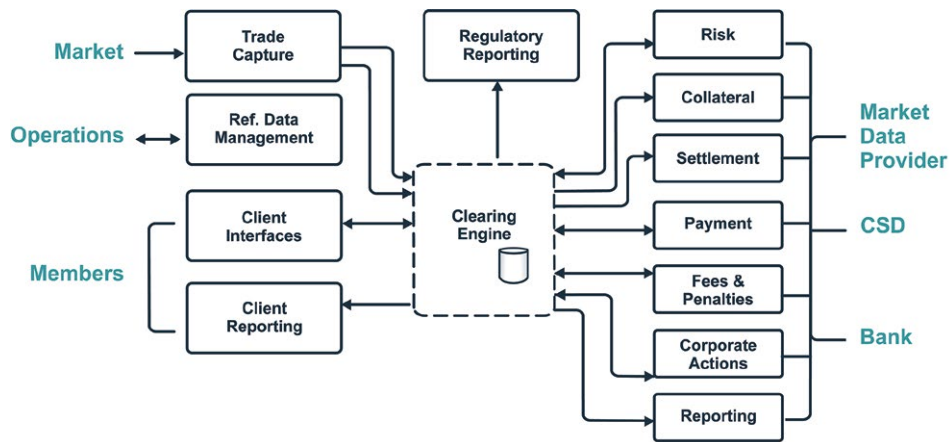
See you in London in 2020!

TMPACONF.ORG

EXTENT.EXACTPRO.COM

Are you sure that you are not underestimating the complexity of your post-trade platform?

TRADE/POSITION/ACCOUNT LIFECYCLE

Another point of paramount importance is that the full position lifecycle needs to be taken into account. One lifecycle could span multiple days and include dozens of steps. It would be wrong to focus on a specific point of the clearing process, such as trade input or a settlement session, avoiding the other processes. It is vital to consider all the stages of a position lifecycle to complete the scenario. For instance, to test a Collateral Deficit scenario, we need to input trades, initiate risk computation, receive a margin call, simulate collateral delivery from the member, and then invoke the next round of risk management computation, to be able to see how well the scenario worked. A Delivery Default scenario, which is also a multi-day one, is even more complex: after adding a trade, it is necessary to keep tracking the position till the Settlement Date and simulate a delivery failure, then keep tracking it forward to initiate buy-in and cash settlement processes. The implementation of Corporate Action also requires several days to make sure that all the necessary steps are covered before and after the execution of corporate action.

RISK CALCULATION

Risk computation, which is the heart of every clearing system, requires real-time and historical data, and is normally based on complex mathematical models. It uses powerful computation algorithms, such as, among others, VaR and Expected Shortfall, yield curve shifts, SPAN, etc. Risk calculation is time-consuming and requires a high degree of knowledge from the tester.
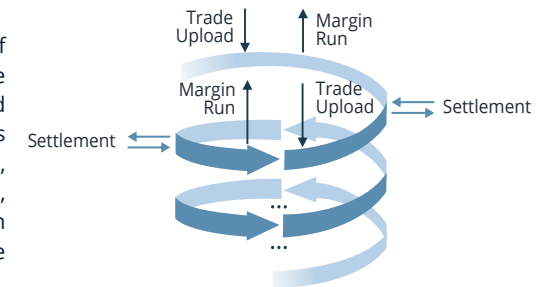
ASSET CLASSES

Clearing different asset classes involves different processes in the CCP and different algorithms, with cross-asset-class margining adding even more complexity to the challenge.

UPSTREAM/DOWNSTREAM SYSTEMS

There is also the challenge of upstream/downstream systems dependency on the clearing system's environment. Such systems have limited availability during testing, so it is extremely important to be able to properly simulate them.

API

During testing, access to a variety of API endpoints such as FIX, MQ, SWIFT, FPML, as well as custom interfaces, should be supported by test tools.

CLEARING AND SETTLEMENT INFRASTRUCTURE

A typical clearing system infrastructure consists of a number of internal components, where each one is responsible for a variety of tasks, or possibly incorporates several tasks at a time. There is a significant amount of connections to third-party systems, or in-house external systems such as markets, market data providers, CSD, payment systems, and, of course, there is connectivity with clearing members. Having a lot of experience in testing such systems, the Exactpro team is used to encountering challenges on a daily basis. This case study focuses on the ones that are most prominent, and on the way Exactpro addresses them.
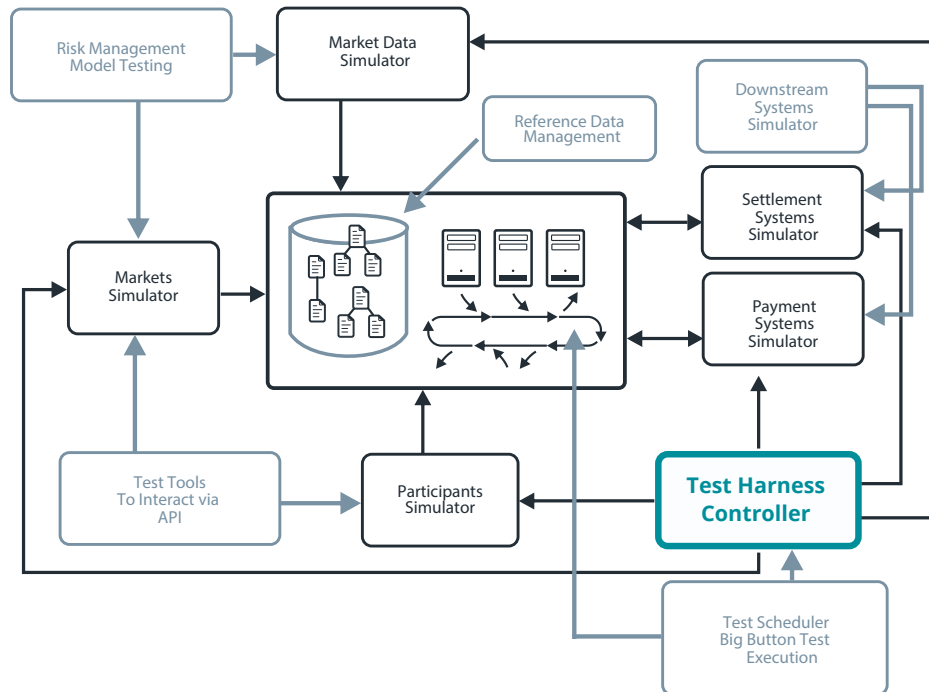
PARTICIPANT STRUCTURE COMPLEXITY

The participant structure complexity is a challenge that cannot be avoided. A company may be represented by various entities: these entities might trade at different markets, trade different asset classes and even their margining can be done in a consolidated way or in segregated currencies. The links between Non-Clearing Members (NCM) and Clearing Members (CM), their changing roles in different markets, the array of accounts – all of that adds challenges to the software testing process.

INNOVATION IN QA. CLEARTH

If you combine all the points mentioned above, you will inevitably get a massive test library. That is why Exactpro has come up with solutions to deal with such challenges. We offer the Holistic Integrated Automated Test Framework as one of them. ClearTH is an original Exactpro tool, able to simultaneously execute multiple end-to-end test scenarios. Most of them span days and involve a large number of steps. Moreover, these steps should be aligned with the CCP processes. Using ClearTH, the aforementioned examples of test scenarios (Collateral Deficit, Delivery Default, and Corporate Action), as well as many others, can be performed in batches. Test execution is split into Global Steps, and each scenario is aligned with these Global Steps. This way, we can execute as many scenarios as possible at a time and make sure that we incorporate them into the CCP schedule.

RISK COMPUTATIONS

Another case study is the verification of RM computation, which usually involves quite complex algorithms. In some cases, we can create a model and perform calculations using our software and then compare the results with the ones achieved by the system under test. In more difficult cases, a number of tests can be run and used as a baseline for future versions of the system. The Exactpro tool to implement this is called Risk Analyzer. It can be used as a stand-alone tool, but in this context it is referred to as an essential part of the ClearTH solution.
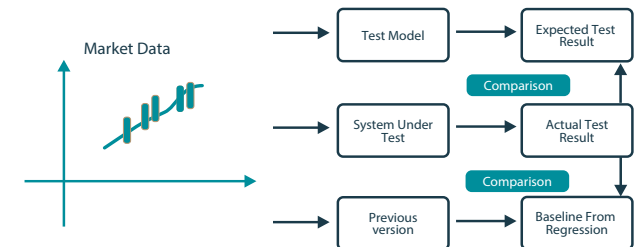
HiVAT — HIGH VOLUME AUTOMATED TRADING

Sometimes, it is not enough to have a set of predefined test cases, as certain issues may only arise when the system is under a high load. So, in the first testing round, we input a random load, make snapshots of all the valuable points, aligned with the CCP lifecycle, check logs and verify the consistency, monitor the GUI and the API. During the next round, we execute the test with the new software version or under different conditions (such as failover) and compare those snapshots. Any discrepancy may indicate that the system has a bug.
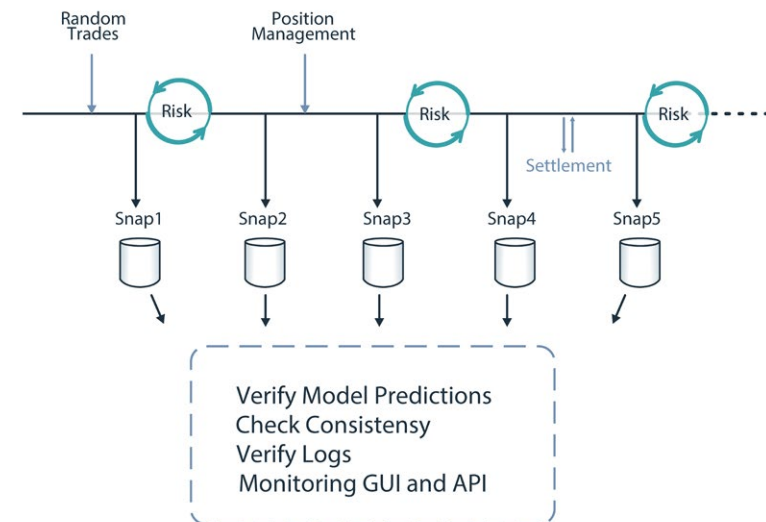
Would you like to team up and build next-gen platforms on R3 Corda?

Being an R3 technical delivery 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. This unique Exactpro tool able to simultaneously execute multiple end-to-end test scenarios in batches. It detects abnormal behavior in the system under test and effectively predicts potential issues. It offers many built-in actions to cover the activities typical of the business flows in live post-trade systems.
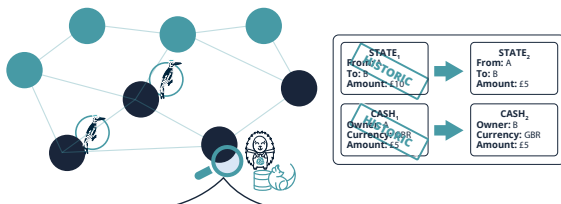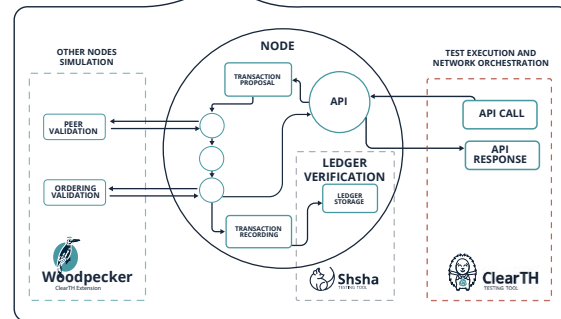
ClearTH:
- verifies each stage of the DLC;
- has an integrated schedule;
- automatically runs test scripts;
- creates multiple-day test scenarios;
- performs multiple concurrent tests;
- has integrated simulators;
- supports SWIFT ISO protocol.

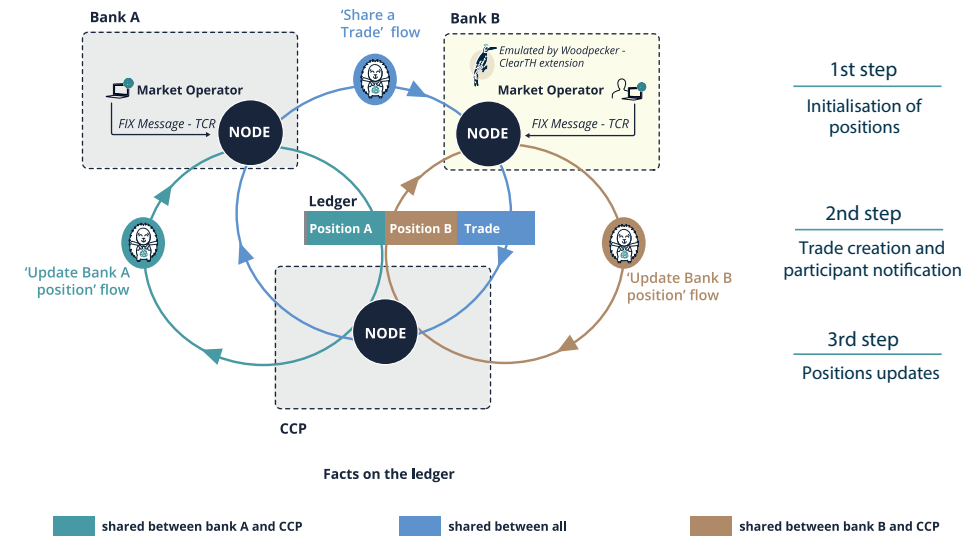Woodpecker is a ClearTH extension. Its main features are:

- It is highly customizable (data source, data types, load, time, interface).

- It is easily transferable to different environments due to its ability to extract data right from the system under test.

- Testing scenarios are created without humans cover a vast majority of diverse conditions/data combinations.
- It's a productive intersection of functional and non-functional testing approaches.

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

Facts on the ledger

shared between bank A and CCP	shared between all	shared between bank B and CCP

1st step — Initialisation of positions

2nd step — Trade creation and participant notification

3rd step — Positions updates

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
EXITUS ACTA PROBAT

exactpro
EXITUS ACTA PROBAT
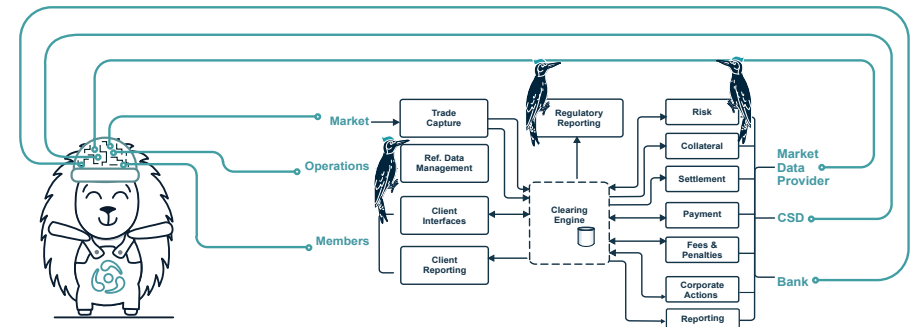
ICST 2019 IEEE — ClearTH FOR DLT-BASED POST-TRADE SYSTEMS RESEARCH BASED ON R3 CORDA TECHNOLOGY HAS WON THE BEST TOOL DEMO AWARD

The paper was presented at the 12th IEEE Conference on Software Testing, Validation and Verification (ICST) in April 2019 in Xi'an, China.

Abstract — The paper presents an overview of a test automation framework aimed at end-to-end functional and non-functional testing of DLT-based hybrid financial software for post trade. The proposed solution comprises the components designed for testing user-facing parts of the SUT as well as business logic specific for different DLT-based architectures. This combined approach is seen as a viable solution of the problem of the SUT complexity as well the variety of possible DLT architectural decisions.

r3. DAML HYPERLEDGER GENERAL MEMBER

IEEE — USER-ASSISTED LOG ANALYSIS FOR QUALITY CONTROL OF DISTRIBUTED FINTECH APPLICATIONS

The paper was presented at the IEEE International Conference On Artificial Intelligence Testing (AITest) in April 2019 in San Francisco Bay, CA, USA.

Abstract — Testing of distributed systems is a complex task, which is hampered by the impossibility of guaranteed reproduction of errors associated with race conditions. Even minor instrumentation of the system significantly changes its characteristics, which becomes critical, especially for load testing. All of that increases the importance of quality control methods based on the system log analysis. In this paper, we present our experience of semi-automated analysis of the behavior of clearing and settlement system by utilizing its logs for the purpose of identifying and classifying errors.

AiTest

IEEE — DEVELOPMENT OF INTELLIGENT VIRTUAL ASSISTANT FOR SOFTWARE TESTING TEAM

The paper was presented at the 19th IEEE International Conference On Software Quality, Reliability and Security (QRS) in July 2019 in Sofia, Bulgaria.

Abstract — This paper is our view on incorporating virtual agents into daily operations of a software testing team. An important property of intelligent virtual agents is their capability to acquire information from their environment as well as from available data bases and information services. Research challenges and issues associated with the development of intelligent virtual assistant for software testing team are discussed.

MACSPro' 2019 Modeling and Analysis of Complex Systems and Processes — TOWARDS A FORMAL MODELLING OF ORDER-DRIVEN TRADING SYSTEMS USING PETRI NETS: A MULTI-AGENT APPROACH

The paper was presented at the Modeling and Analysis of Complex Systems and Processes (MACSPro'2019) conference in March 2019 in Vienna, Austria.

Abstract — Electronic trading systems provide the computational support for stock exchanges. Liquid markets use order-driven systems, i.e., where client requests, for trading financial instruments, are served through individual orders. This paper presents Petri net models assembling some crucial processes executed within order-driven systems such as orders submission, application of precedence rules, and the order matching mechanism. Such processes were modelled as types of agents running in a multi-agent system (MAS) using nested Petri nets (NP-nets) — a convenient formalism for modelling MAS. With NP-nets, we focus on the control-flow perspective (causal dependence between activities executed by agents) and in the synchronization between agents. Conversely, we have used coloured Petri nets to extend the model including orders as objects with attributes. Thus, this work with Petri nets represents an experimental & initial research phase to validate trading systems using related methods such as process mining, simulations and model checking.

To learn more about Exactpro's research, visit
https://exactpro.com/ideas/research-papers

WOULD YOU LIKE US TO CREATE A VIDEO BASED ON YOUR PROJECT?

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

Have you ever questioned the nature of your reality?
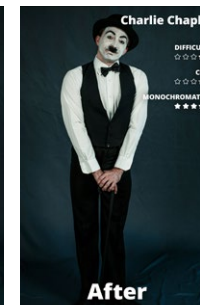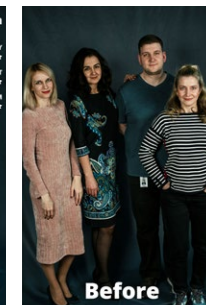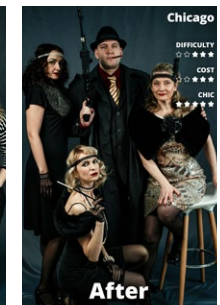
... inspired by HBO's West World
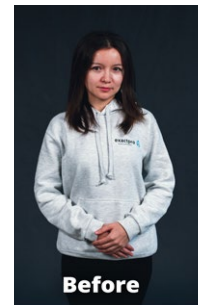
DOLORES   MAN IN BLACK   TEDDY   DR.ROBERT FORD

What is the right balance between shift-left testing and software independence?
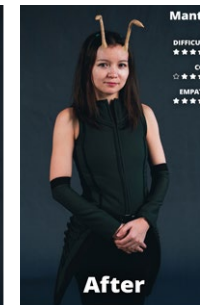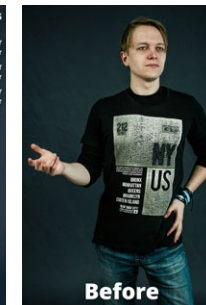
ARE COMPANIES RIGHT TO OUTSOURCE SOFTWARE TESTING?

By Iosif Itkin, CEO and co-founder, Exactpro, and Daria Degtyarenko, Researcher, Exactpro. With regulatory requirements and business pressures putting an ever-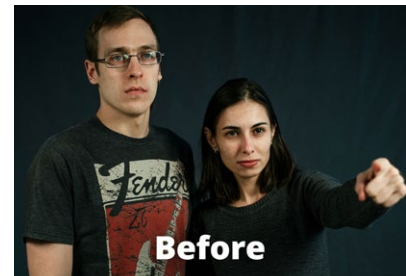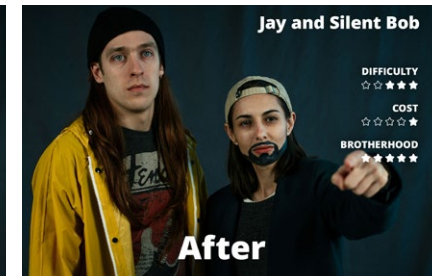tightening squeeze on financial services, firms tend to look for the most efficient ways of ensuring the quality and stability of their platforms, with some turning to quality assurance outsourcing. However, outsourcing might not always be the best plan, for four key reasons: responsibility, capability, agility and quality.

Responsibility. In its SYSC 8.1 General outsourcing requirements, the Financial Conduct Authority (FCA) states that a regulated entity must not 'delegate responsibility' while outsourcing 'critical or important operational functions or any relevant services and activities'. The quality of software systems and platforms underpinning the financial market infrastructure's (FMI's) operations and services is certainly important and, arguably, critical. Therefore, it is a regulatory directive that quality assurance must not be delegated.

Capability. To stay competitive, any company must be technology-focused. No company would want its core competency to be building low-quality products, so quality assurance is not a peripheral function. Software testing cannot be viewed as an ancillary activity for a capable, competitive firm.

Agility. Relying on external vendors could hinder the flow of information necessary for companies focused on reducing time-to-market and undergoing agile transformations. Flexible software development and acceptance-testing methods require synchronisation among all team members, which can become difficult when outsourcing, because the teams are divided by firewalls, time zones, distances, and cultures.

Quality. 'Quality cannot be tested into a product' — so goes the aphorism, meaning that product quality is not solely determined via testing. Firms focused on quality work on holistic approaches to software development continuously analyse testing outcomes, measure attributes related to product quality and invest in improving risk management tools with the aim to prevent rather than cure. 'Quality is everyone's responsibility', as another old saying goes. Why would anyone want to rip out an integral piece of the intertwined quality puzzle by outsourcing software testing?

Exemplary organisations take their regulatory responsibilities and technology capabilities very seriously. They relentlessly strive to achieve quality and agility in their delivery. So surely the case for DIY software testing is watertight. However, true software testing is not there to confirm perfection. Software testing exists to uncover something that somebody doesn't want to exist. It is about providing objective information about the system under test in a form beneficial for the stakeholders.

To paraphrase James Madison, one of the founding fathers of the US, if companies were angels, independent software testing would be unnecessary. If angels were to develop software, neither external nor internal controls would be necessary. Software testing, if independent, does not set managers free from responsibility. Rather, it pressures them toward higher levels of regulatory compliance.

When Tolstoy wrote in "Anna Karenina": "Happy families are all alike; every unhappy family is unhappy in its own way", he meant that in order for a family to be happy, it must succeed with respect to a specific range of criteria, and failure on any single count leads to unhappiness. Similarly, to be successful, all production systems must work well within a narrow range lof criteria that assures their users' and stakeholders' satisfaction. In order to consistently reach that benchmark, it is of paramount importance for a technology firm to have a strong in-house capability to build, check and monitor systems to ensure that the family stays happy.

Software testing is focused on exploring the darker side of systems. Research confirms that developers are less likely to deploy advanced software testing techniques, such as passive testing, in part due to the congruence bias, i.e. overreliance on proving a specific hypothesis that directly results in neglect of indirect testing. Thus, leaving software testing to developers or cross functional teams could mean prioritising the component level and 'happy path' checking over proper risk assessment of an interconnected system as a whole.

Fundamentally, biases are not the enemy, because they exist for a reason: to increase efficiency. Superiority requires focus, and a good strategy always requires making a choice. In this particular case, it is whether to prioritise improving the core product versus building a sophisticated test harness. The creation of test harnesses is the core capability of an independent software testing business, so it doesn't make sense to hire smart testers and then tell them what to do. By the same token, outsourcing does not guarantee software testers' unbiased independence.

Many articles have been written about cross-functional, self-organising teams and confining software testing to sprints. Genuine agility is antifragile. Iterative processes considerably benefit from prompt independent feedback. In implementations that are agile in name only, there will always be a way to blame inefficiency on faulty communication. Reliable systems are not built on trust. Rather, they are built on the absence of trust. That is why independent software testing is fundamental to well-functioning financial firms, similarly to the importance of free press to a well-functioning democratic system.

Independent testing alone is not enough to achieve quality, especially when introduced at a late stage of software delivery. The best way to limit the total cost of the project is to incorporate independent testers into self-organising teams as observers and contributors, so they can detect problems as soon as possible. Whether the issue is a software defect or a process deficiency, it is never too early to embrace reality and deal with it.

If outsourcing software testing is a necessary evil, the shame of delegation reflects a gap between the status quo and the ideal world. Relying on outsourcing is not perfect. However, if applied properly, outsourced testing can at least create awareness that a gap exists and, possibly, can help to narrow it.

https://a-teaminsight.com/are-companies-right-to-outsource-software-testing

WFE Technology Conference: What is the main takeaway from doing QA for exchanges?

"It is both a pleasure and an honor to participate in The WFE's Technology Conference and to address such a distinguished audience.

Exactpro is a specialist firm focused on software testing for exchanges, clearing houses and CSDs. We started as a small startup in 2009. We were part of a global exchange group for two and a half years. Now we work as an independent business employing over 500 specialists.

We work across a wide variety of platforms. Platforms from the Past, the Present, and the Future. We support our clients who run systems that were built decades ago. We support our clients running modern low latency platforms, including those based on FPGA and Infiniband. We also support our clients designing the systems in the cloud, applying distributed ledger and machine learning technology.

Exactpro has the privilege of delivering services to our partners worldwide, from Sydney to Athens, from Johannesburg to Umeå. Large-scale platforms. Small platforms. Everything in between. I would like to share the main takeaway from our work.

The quote by Ray Dalio, founder of Bridgewater Associates, is the best way to describe it. "Truth — or, more precisely, an accurate understanding of reality — is the essential foundation for any good outcome". We all want better software, more resilient systems, shorter time to market and happy market participants. What is important, however, is that in technology, the truth is not given to us in the form of a revelation. Finding the truth is relentless learning. Software testing is relentless learning.

It requires courage to look at something complex and accept that it might not be easy to understand it. The level of safety achieved in the aerospace industry is a result of constant learning. Learning does not make people happy. In fact, we learn faster from unpleasant things.

Humans desire to be happy, which sometimes leads them to focusing on what is easy, instead of what is right. A committee working on a nuclear plant might spend more time discussing a bike-shed for the plant's employees and less time on the plant's safety features. It is easier to understand how bicycles work, and the parking lot will be used every day. On the other hand, one hopes that the day when you will need to resort to all these sophisticated and expensive means of protection will never come.

People look at sophisticated monitoring, onboarding and testing systems and claim that it is possible to achieve the same outcome by using a much simpler approach. Create simple scenarios, confirm that the system works as expected and avoid being grilled on hot stones by managers and regulators.

There are those who tell their technology partners, "It's time to try to defy gravity. But planes look difficult. Take our money, go into the woods, bring us sticks, bring us feathers, bind them together..." Others do. But not us. We build software to test software.

We do whatever it takes to help our clients embrace the nature of their reality. We are very grateful that we have an opportunity to do it for some of the most innovative exchanges and clearing houses globally."

WORLD FEDERATION OF EXCHANGES

Would you like to participate in developing cross-industry quality and reliability standards?

Earlier this year, Exactpro initiated the formation of MOST (Monitoring, Onboarding and Software Testing) working group within the FIX Trading Community. The group is co-chaired by Krishna Tharnoju, Consultant at Pictet Asset Management, and Iosif Itkin, CEO and co-founder of Exactpro. The goal is to address the need for further standardization and innovation.

The group's mission is to raise the quality and reliability of platforms operated by FIX Trading Community Members. This is being implemented via developing a set of recommended practices for software testing of FIX-related financial applications and creating technical guidelines for their quality and reliability assessment. The group meets on a monthly basis and also collaborates with the FIX Orchestra Working Group and the Cybersecurity Working Group to ensure alignment across a number of related issues.

Exactpro Partners
Exactpro partners with numerous global organisations on multiple projects worldwide.

Exactpro specialists relentlessly work on extending the regression libraries and performing gap analysis. We are building faster software to execute more tests and run them concurrently, under load. The same sets of tests are repeated multiple times so that they can be improved and made more diverse based on the collected data. We also intensify the usage of chaos engineering in our non-functional, failover and recovery testing. In order to collect, store and process all these data sets, we need to enhance the hardware as well. We are currently building the Exactpro cloud to implement efficient testing infrastructures and raise the effectiveness of our company's suite of test tools.

The Exactpro test cloud will serve as storage for all the test data and a cradle for the next generation of testing tools. It will contain built-in test coverage and compliance reporting. Its pricing will be tailored to massive defect mining processes. The existing cloud computing offerings typically work well when one needs to obtain capacity for a short period of time. However, defect mining implies that one has a complex system and has to constantly run tests against it under load in order to get as many defects hidden in the softwareas possible.

This year, our focus has also been on increasing the number of projects using cloud. In parallel with creating the Exactpro cloud, we have worked on the following three initiatives:
1) Aligning non-functional testing (NFT) with Kubernetes and working out an NFT testing approach for the cloud. There were multiple adjustments that had to be made, compared to ordinary enterprise NFT.
2) Adding support of Docker in Exactpro tools. The tools are now delivered as Docker images.
3) Working with AWS, Azure and Google Cloud as part of our ongoing projects.

MIND THE GAP BETWEEN TESTING AND PRODUCTION: APPLYING PROCESS MINING TO TEST THE RESILIENCE OF EXCHANGE PLATFORMS

focus Monthly insight from the WFE and our member exchanges

Do you want to know the gaps in your test coverage?

Being highly sophisticated by design, exchange platforms require continuous testing to ensure their resilience. Defects can slip through gaps in test coverage. Thus, identifying and closing such gaps are of the utmost priority. Various techniques are available to support this process.

One of them is called process mining. It is a family of business analysis techniques used to extract information about distributed systems from their execution logs and network captures, which are in no short supply when it comes to complex technology platforms processing huge amounts of inward, outward and internal messages daily.

To illustrate how the process mining approach can be used to enhance functional validation, let's consider the simplest example of a trade order lifecycle.

In general, an order lifecycle comprises several stages. After a participant establishes the connection and sends an order, the system validates the request, and, based on this, places or does
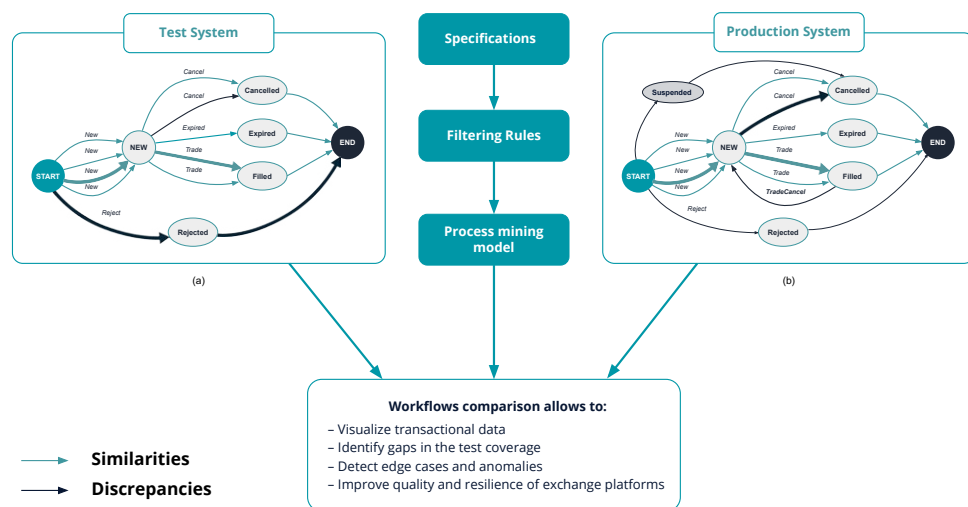
not place the order into the order book for further activities. The system assigns an order status tag to this request at every step of its lifecycle, with each order status change communicated to the participant via execution report messages. Even for this simple example, there are numerous potential execution scenarios involving different sequences of order status changes.

Represented as a graph, these changes help to visualize and understand the test coverage (see the diagram). Having visualised test execution data, one can discover that, for example, the transactions extracted from the test environment lack **Suspended** order statuses and **TradeCancel** state transitions (a). In contrast, similar visualisation based on the production transactions dataset will show that there are **Suspended** order statuses and **TradeCancel** transitions present and there are much fewer **Reject** order statuses and more **Cancel** transitions (b).

As demonstrated in the diagram, the process mining approach involves several steps. First, the transactional data is captured from the systems under test or production environment. This data includes test execution results, application process logs and network captures. Then, the data is combined into a single database and transformed into a set of flat files which are subsequently used to build a process mining model. Additionally, system specifications and rules filtering can be used to simplify the model.

With process mining techniques in place, the tools can compare both the test and production graph representations and highlight the discrepancies, which allows QA engineers, subject matter experts and compliance analysts to identify the functional areas requiring more extensive test coverage, thus helping to enhance test libraries. Another benefit of using this approach is that functional and non-functional testing libraries can be compared against each other in order to understand whether they cover the same transitions.

Test coverage analysis is a crucial part of proving the resilience of mission-critical platforms. The described approach allows the visualisation of transactional data in order to obtain a direct representation of coverage and, ultimately, close the testing gaps to assure the orderly functioning of the global financial markets.

Test System (a) / Specifications / Filtering Rules / Process mining model / Production System (b)

Workflows comparison allows to:
– Visualize transactional data
– Identify gaps in the test coverage
– Detect edge cases and anomalies
– Improve quality and resilience of exchange platforms

→ **Similarities**
→ **Discrepancies**

TESTING
THE INTELLIGENCE OF AI

Are you ready to test your AI-based system's ability to generalize?

In what way is testing AI systems different from testing traditional systems in the financial services industry? Do they need a specific test approach?

A recently issued A4Q AI and Software Testing Foundation Syllabus provides the following answer: in addition to traditional quality characteristics of IT systems, AI-based systems are required to be able to learn, to demonstrate an ability to generalize and to be trustworthy.

As for the **ability to learn**, the target of the testing activities is a specific algorithm which is assessed based on a set of metrics proposed by scholars and industry data scientists. In terms of the test approach, this characteristic requires knowledge area-specific checks and presents a challenge arising from the necessity of having massive amounts of thoroughly selected data.

While the ability to learn is most actively explored by data scientists, **trustworthiness** of AI-based systems seems to be a more widely discussed characteristic. The expert groups established as part of governmental and international cross-industry organisations set out components of trustworthiness, such as ethics, compliance, robustness, transparency, safety, etc.

The main focus of the end-to-end testing of AI-based system is the **ability to generalize**. Will the system be capable of adapting to different and previously unseen scenarios? This is the very question that we ask ourselves when testing software for complex distributed multi-threaded non-deterministic platforms in financial market infrastructures. The tools and methods applicable to validate such platforms are also relevant to testing AI-based models within their surroundings.

The validation and verification of all sophisticated systems are affected by the cognitive patterns common for human judgements. Meta-research studies confirm that the key cognitive pattern affecting any type of software testing is *confirmation bias*.

It is also well established that this bias severely affects our ability to consume new media and information. It would be fair to say that machine-readable news analyzers, responsible for extracting sentiments and data from the underlying texts, present a perfect playground to evaluate a doubled-down effect of the confirmation bias.

When testing AI-based systems, the *anthropocentric bias* can obscure test design and execution. For example, in conversational assistants, which are widely used in banking, insurance, and portfolio management services, the problems can be caused by anaphoric relations, punctuation, misspellings in the input. To avoid it, one might want to rethink the equivalence classes and significantly broaden the user input datasets used during end-to-end and negative testing.

In a high-volume low-latency environment which is common for AI platforms (e.g. in algo-trading), it is often impossible to predict the exact outcome of a test scenario. Such cases require advanced software testing techniques, e.g. passive testing and post-transactional verification, which can be neglected due to the *congruence bias*.

QA activities around sophisticated systems powered by big data and non-deterministic methods can often be susceptible to a cognitive bias called *Parkinson's Law of triviality*, a tendency to give disproportionate weight to trivial issues. One can suspect that the validation priorities were misplaced while applying software testing to monte-carlo and statistical-based pricing calculators in mortgage-based structured products.

Another bias affecting testing AI is the *automation bias*. One of the approaches to overcome it is software testing tools diversity – usage of the instruments based on different technologies and approaches. In case of AI-based fraud detection and market surveillance systems, it is the usage of script-based tools, participant simulators, built-in custom alerts, etc.

Zero-risk bias, a tendency to prefer the complete elimination of a risk even when alternative options produce a greater reduction in risk overall, is frequently observed in software testing in general. It happens when all the effort is concentrated on the elimination of the risk associated with a single module or a new functionality without proper end-to-end testing of the whole system. An example of AI-based system susceptible to this risk is an insurance claims management application, which analyzes historical data and compares it to the predicted total loss cost in order to make a decision regarding the management of the claim.

Everyone is concerned with the unpredictability of AI-based systems. Yet, humans tend to underestimate the complexity of the existing distributed non-deterministic platforms. Thinking about complex distributed platforms built with traditional technologies as of 'less challenging', compared to AI systems, is an example of work of the *illusion of control* bias which makes people happier and less stressed, though not better performing.

Software testing is not about happiness. Software testing is about pain and unhappiness. We truly learn something only when we suffer and software testing is relentless learning.

The full version of the white paper is available on our website.

DELIBERATE PRACTICE OF SOFTWARE TESTING

Would you like to provide
your feedback on our work?

exactpro.com/feedback

exactpro.com