# Exactpro overview

Exactpro provides software testing services for mission critical technology that underpins global financial markets. The firm is experienced with trading and clearing and settlement platforms, market data systems, collateral management and risk management systems, central data warehouses, regulatory reporting, etc.

A specialist firm focused on functional and non-functional testing for complex, distributed, non-deterministic and artificial intelligence platforms

Headquartered in the UK, Exactpro has delivery centres in Georgia, Sri Lanka, Armenia, Lithuania, and the UK, and representative offices in the US, Canada, Italy, and Australia.
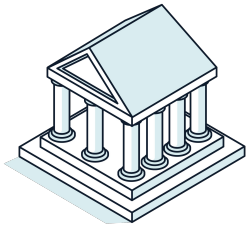
Part of the London Stock Exchange Group (LSEG) from May 2015 until January 2018, when the Exactpro management proceeded through the buyout of the company from LSEG.
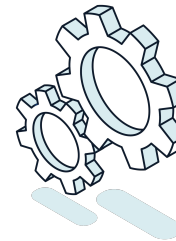
Incorporated in 2009 with 10 core members, Exactpro now employs circa 500 specialists.
Most of our clients are regulated market infrastructures.

# Software testing for financial market infrastructures

- End-to-end testing for complex transaction processing platforms: extensive test libraries covering complex business life cycles, system verification at the confluence of functional and non-functional requirements

- High-volume automated testing: autonomous test execution of more functional tests under load to improve test coverage, system quality, and resilience

- Technology stack diversity: testing financial platforms built with traditional/emerging technologies, support of various protocols as well as API, UI, DLT, and cloud endpoints

- Data governance: test automation framework enabling better access to test evidence and smart analytics for governance and regulatory compliance

- AI-driven testing: system modelling and simulation, test script generation and optimization, smart test execution, support in test results interpretation

# Pace of innovation and software quality



Quality

Cost

VALUE

Speed
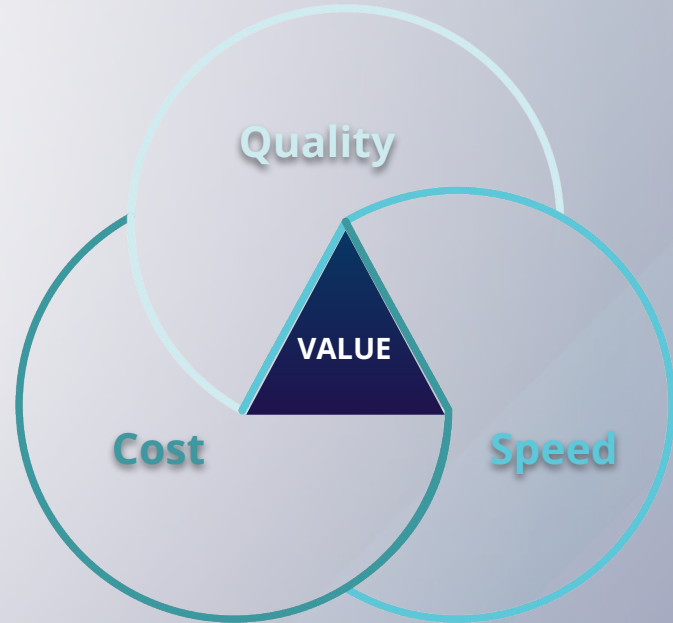
# Problem of automation in software testing

exactpro
EXITUS ACTA PROBAT

**Effort:**

| Cognitive (Medium) | Mechanical (Easy) | Cognitive (Hard) |



**Task:**

**Select** actions to perform

- What actions are available?
- Which of them to choose?
- In what order?

**Execute** actions against the System Under Test (SUT)

Action types:
- Set-up actions
- Experiment actions
- Observation actions

**Interpret** the outcome of the actions

Non-exclusive possibilities:
- Test results match our model of the SUT
- Something is wrong with our actions
- Something is wrong with the SUT
- Something is wrong with our model

Provide the information about findings to the stakeholders for them to make decisions

**Evaluate** obtained information and the list of available actions – repeat the steps

# Modelling and simulation



**Test basis**

FIX orchestra

✓ all instruments and their trading control permutations

✓ all users and their trading control permutations

✓ all relevant reference data actions

✓ protocol specifications

✓ various input actions check lists

source script modification

system modelling ('digital twin')

input dataset

expected outcomes dataset

# Test execution and test library optimization

**exactpro**
EXITUS ACTA PROBAT
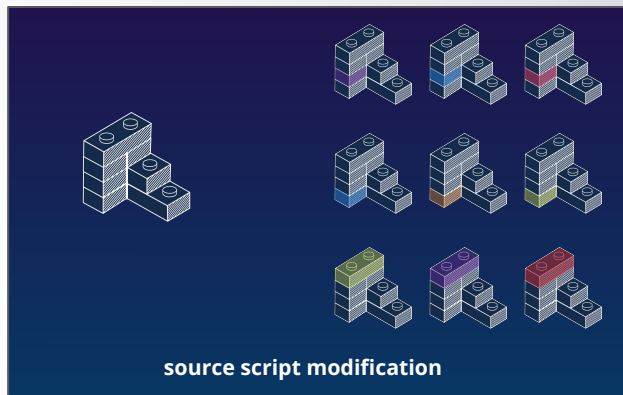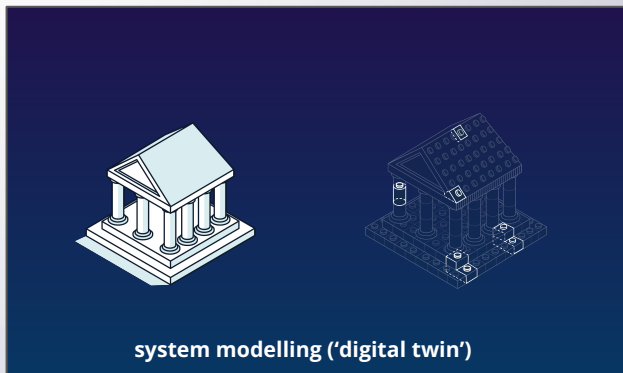
**test basis**
- ✓ all instruments and their trading control permutations
- ✓ all users and their trading control permutations
- ✓ all relevant reference data actions
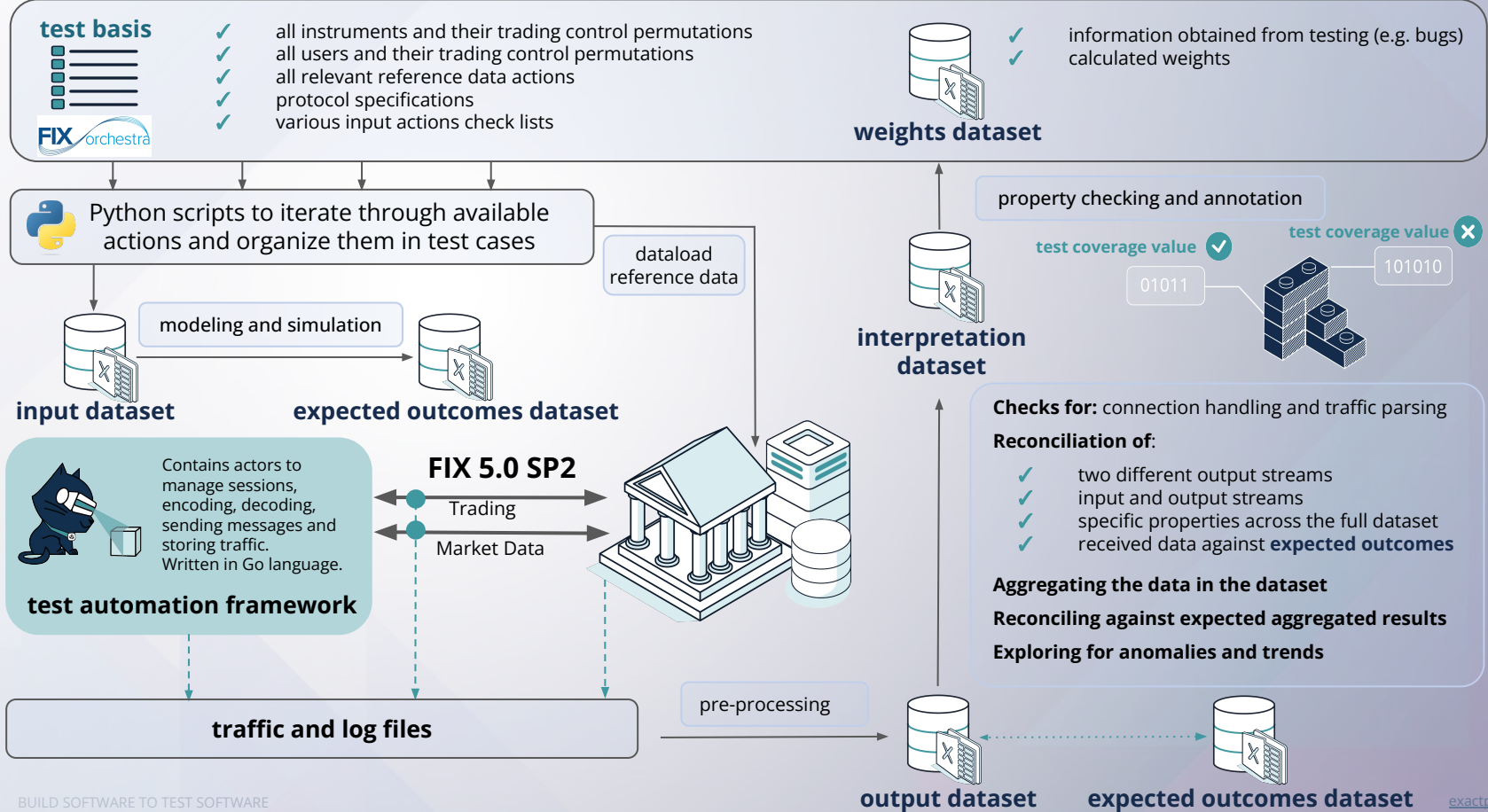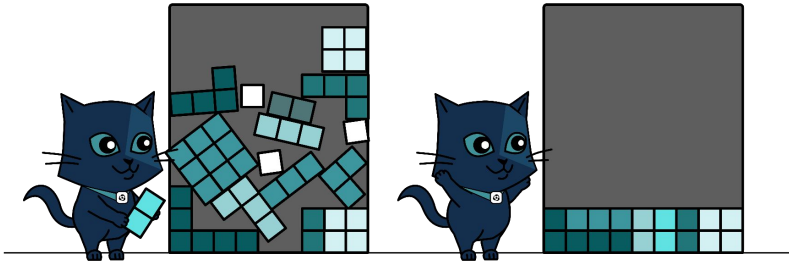- ✓ protocol specifications
- ✓ various input actions check lists

**FIX** orchestra

- ✓ information obtained from testing (e.g. bugs)
- ✓ calculated weights

**weights dataset**

Python scripts to iterate through available actions and organize them in test cases

property checking and annotation

dataload reference data

**test coverage value** ✓
01011

**test coverage value** ✗
101010

modeling and simulation

**interpretation dataset**

**input dataset**          **expected outcomes dataset**

Contains actors to manage sessions, encoding, decoding, sending messages and storing traffic. Written in Go language.

**test automation framework**

**FIX 5.0 SP2**

Trading

Market Data

**Checks for:** connection handling and traffic parsing

**Reconciliation of**:
- ✓ two different output streams
- ✓ input and output streams
- ✓ specific properties across the full dataset
- ✓ received data against **expected outcomes**

**Aggregating the data in the dataset**

**Reconciling against expected aggregated results**

**Exploring for anomalies and trends**

pre-processing

**traffic and log files**

**output dataset**          **expected outcomes dataset**

exactpro.com

# Test library optimization

AI modeling for test script generation →
aiming for coverage



Symbolic AI for coverage point analysis →
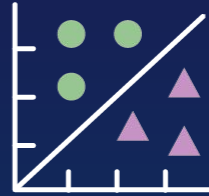test library optimization