exactpro
EXITUS ACTA PROBAT

Test Automation for Risk Management Systems

Presentation by Alyona Lamash
Head of Risk Management Practice

Contents

Structure of RM Systems

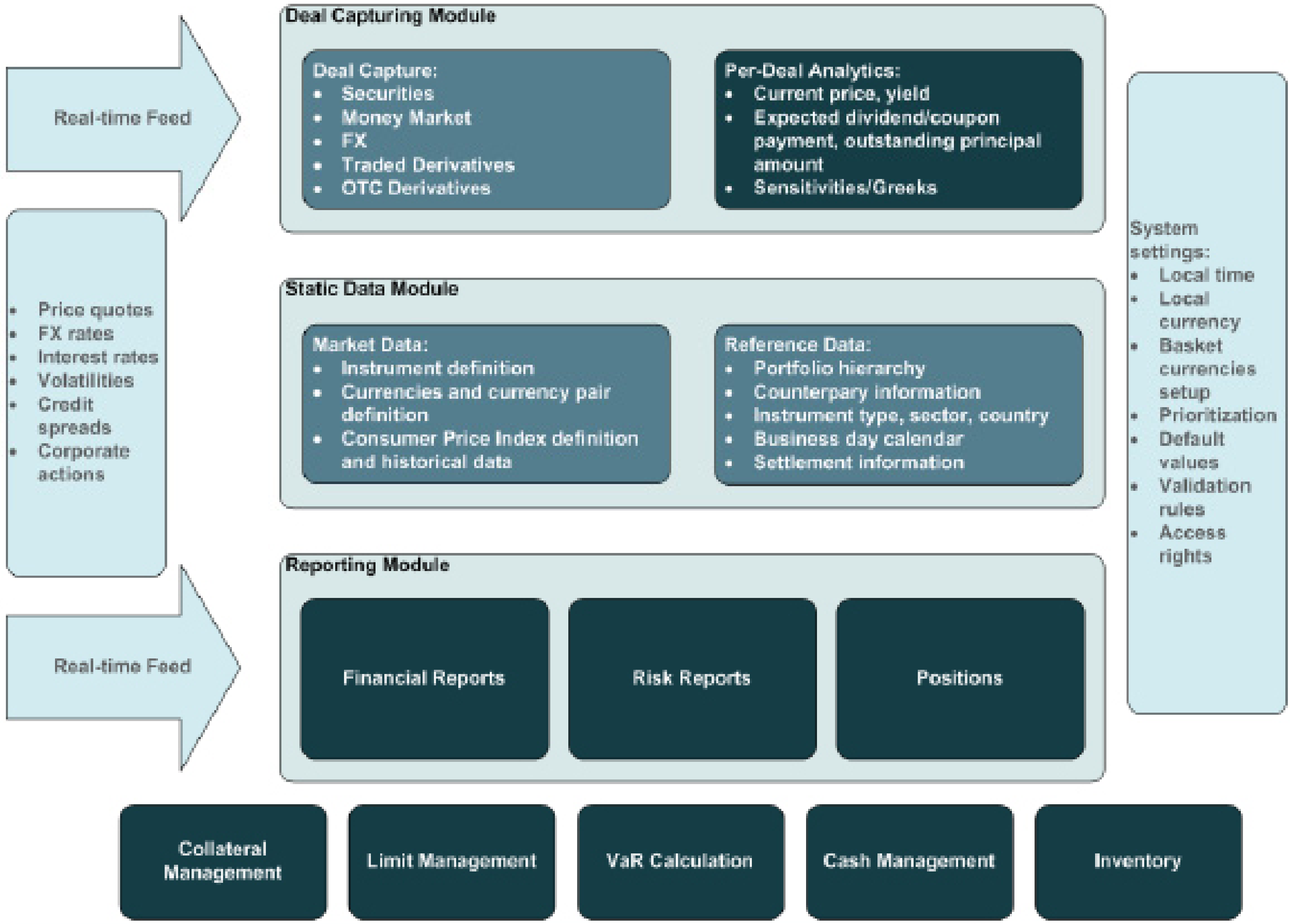
Four main inputs contribute to the efficient running of Risk Management systems:

1. Trading data: Timely delivery of trading data is important to avoid missing significant changes in a position or changes in important parameters of existing deals
2. Real time market data: Current market data are essential for calculations; independent feeds should be used to establish objective and authoritative valuations
3. Static data: The way financial instruments are defined plays a leading role in instrument analytics, because instruments with similar names may have different trading and valuation characteristics. Reference data are used to classify trades, to maintain trading hierarchies, to review positions in different contexts, to separate proprietary from trading positions and to aggregate instruments according to user-defined instrument types
4. System settings: Are applied when the system is installed and may only be changed by system administrators. Settings are most complicated in globally-customizable applications, where a change in one setting can impact many or all processes in the system overall. Correct settings and thorough control are critical for success

These inputs are utilized by different analysis and control systems (such as Risk Monitoring & Control, Collateral Management, Limit Management, Cash Management, Inventory).

Some calculations applied at the risk reporting level can also be used by the Limit Management system or can be calculated separately. Different functions can be implemented in one system.

Structure of RM Systems (continued)



Business Logic Verification

There are three layers of risk calculation and therefore three verification phases:

1. Position collection layer
    - All necessary trade parameters are transferred
    - Corresponding real-time market data is loaded
    - Trades and/or trade legs are aggregated into positions appropriately; classification of positions is applied according to selected parameters
2. Position evaluation layer
3. Risk calculation layer
    - Sensitivities / Greeks
    - Scenario analysis
    - Position VaR and diversified portfolio VaR
    - Stress Testing

Test Automation can be applied at each level, provided that:

a) The environment is replicated for each series of tests
b) System settings are predefined and thoroughly controlled before test execution
c) All necessary static data and test scenarios are maintained

The reliability of tests also involves predefined values for the "real-time" market, so that newly calculated values coincide with benchmarks

End-to-End Testing & Automation

Risk Management systems operate in conjunction with other applications, so integration between components must be verified by end-to-end testing.

This involves additional test steps to cover business processes, simulation of data flow from external sources, careful selection of specific test data and others that are worth doing at the business process level rather than the functional testing level.

In practice, test automation should be mandatory for:

- Types of data that cannot be entered manually: exchange feeds, simulations, FIX connections to clients
- Data reconciliation between several systems
- Iterative actions with heavy data items that do not require specific re-arrangements at every turn

Examples include:

- Replication of the test environment by executing scripts aimed at specific data entering
- Setting up static data to test trading, deal entering, position recalculation and risk figures calculation functionality

Automation is most commonly used when testing multiple items or multiple replication of similar steps, for example:

- Loading a large number of trades into one system and reconciling it with another
- Testing updates of data formats to verify that data in the new format is recognized and accepted
- Repetitive actions with different instrument types, different fields or in different parts of an application
- Tests which vary each parameter, one-by-one, with a limited impact on the output result

A Typical Automated Test

Froglogic Squish was used to test the data entry, calculation and report functionalities of Reuters Kondor+ modules. Various methodologies were tried, to test business logic with an interface testing tool. The following strategy was developed:

- The test script consists of three types of file:
  - "Dataset" represents Excel tables in spreadsheets named "Input" and "Output". Header lines contain mnemonic names for the fields in the application's windows; the remaining lines represent test data that must be input in the fields of the window or checked with output results;
  - "Linker" files bind mnemonic names and real fields in the application's windows; they are applied to several versions of the application or several windows with the same functionality but different widgets;
  - "Workflow" describes the order for each field to be touched by Squish during execution
- All files are processed by this test automation system, so that fields are mapped by "mnemonics" and then transferred to the script code
- Script execution consists of running Squish through all fields in a specific window that are included in the prepared files, so that the data is entered and then processed by the system. Then the script is run through the window again, in order to compare the displayed or reloaded values with predefined benchmark values from the "Output" worksheet of the "Dataset" file
- The run is executed as many times as there are lines in the worksheet of the "Dataset" file and Squish fills or checks only the fields with symbols in the corresponding column
- Script execution results are then analyzed by a QA engineer

Sanity Checks

The verification of calculations does not necessarily involve direct computation.

A number of invalid assumptions or defects in a calculation algorithm can be detected, without time-consuming recalculation, by using "sanity checks". These are based on fundamental relationships and allow quick verification that the underlying assumptions hold true. If a sanity check fails, the root cause of the issue must be investigated

Examples of expected results from an automated test script that can be verified by sanity checks include:

- The delta of European call options is always positive and varies from 0 to 1
- Theta (time decay) for American options is always negative
- Bond prices and yields move in opposite directions
- Long-term bonds with predicted cash flows have positive convexity
- The sum of premiums of barrier 'up-and-in' and 'up-and-out' call options equals the premium of the corresponding vanilla call option

Collaboration with Business Users

It is vital for the success of testing that Business Users are given the opportunity to identify and inform the testing engineers about ambiguities or differences in business processes.

Certain situations illustrate the value of collaboration with Business Users. For example, within similar types of business (i.e. within investment banks or within stock exchanges) certain business processes may be known by similar names when, in fact, they differ substantially from each other; or QA engineers may have difficulty understanding that an application supplied by an external vendor has been substantially customised multiple times to the suit specific requirements of individual customers. QA engineers cannot easily do their work of studying the particular functionality of these different business processes without input from Business Users.

Collaboration with Business Users is also important for calculation algorithms. Although industry-wide standards are usually followed, the specifics of an individual market or agreements with a particular client may require firms to apply different calculation rules or valuation models. Here, test cases are used to find discrepancies between the expected result and the result calculated by the system; QA engineers reveal the assumptions underlying the pricing model and the impact from different points of view and Business Users make a decision.

Risk Management is an area where Business Users perform multiple, complex, analytical tasks. These tasks need to be assessed to determine whether automation will significantly improve testing quality. In order to do this, it is important to understand Business Users' expectations and this requires close collaboration between test automation experts and business analysts.

Contacts

Alyona Lamash, Head of Risk Management Practice
alyona.lamash@exactprosystems.com
+7 (495) 640 2460 / +7 (917) 522 5552

Iosif Itkin, Managing Director
iosif.itkin@exactprosystems.com
+7 (495) 640 2460 / +7 (915) 333 5593