



## Beyond Brute Force: Testing Financial Software

---

### **Summary:**

*Functional testing of financial software requires a deep subject expertise and understanding of QA methods and processes. This article, describes the general approaches and methodology in financial testing.*

Development and testing of application software is heavily dependent upon the domain to which the software applies. To produce application software that the end user will find efficient and comfortable to use, software development process solicits subject matter expertise at various stages.

Let us consider large software development projects that follow formal processes and are carried out by structured teams with proper separation of roles. They are likely to employ development staff possessing domain-neutral technical expertise. This is especially true of large software development outsourcing vendors. Yet the needs of the end user are expressed in the concepts, terms and logic of the applied domain. Normally a special task force is charged with bridging this gap at software design stage, that is, software product manager and business analysts. The importance of domain analysis at this stage is well recognized and the methodology is described.

In much the same way, the subject matter knowledge is an important factor when considering testing efficiency (although this can seem like a less urgent problem to attend to / might not be as much appreciated as in the case with software design). Proficiency in general QA methodology and experience in testing alone might not make a sufficient qualification for application software projects especially in complex domains. A good alternative might be a specialized QA services vendor, which is focused on a particular domain. The better suited vendor will have a successful record of testing software similar in business content and functionality, business analysts who will immediately recognize and understand the concepts involved, an established process, resources and materials to train new teams in the subject area.



In this article we will consider testing software in the domain of financial markets. We will look at what qualifications are required of the QA staff and what specific test methods / techniques are used in their work. This analysis is built on our extensive experience in testing financial software in areas such as financial market data access, pricing of derivatives, risk management, portfolio management and technical analysis.

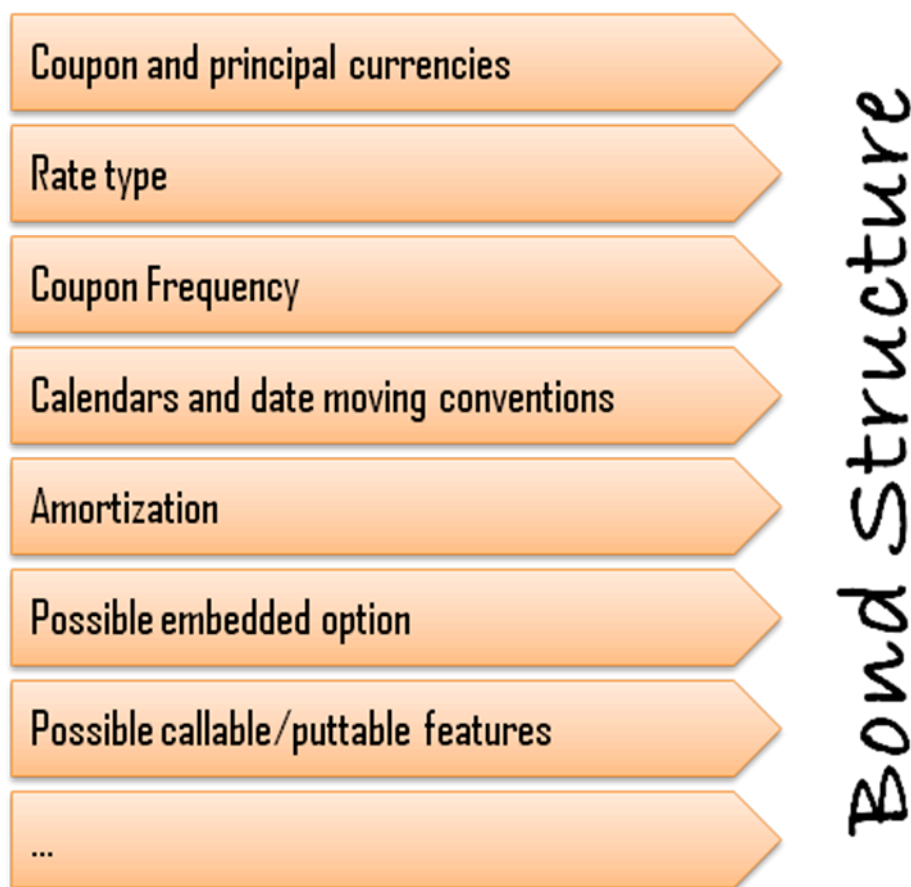
## ***Financial testing specifics and difficulties***

Before getting into methods and techniques, it helps to understand the major difficulties and setbacks encountered in testing financial software.

Very often the tests performed do not specify expected results. Precise verification is time inefficient (particularly if all we want is a “sanity” check) and more often than not cripples the test coverage.

Business logic is not always clear just from working with the user interface of the software in question. The reason is that a *real* financial instrument which might be represented as a single input in an application, is in fact a complex object with a big set of properties which will have a significant impact on its analysis (an example is bond structure; see Fig. 1 Bond Structure).

**FIG. 1 Bond Structure**





Time required for testing can be difficult to estimate. One of the main factors here are the expected level of coverage and a matching choice of test methodology. Time efficiency and test coverage being a trade-off is likelier for financial tests than for any other tests. There are always several ways of testing, and the differences in effort are in a very broad range. Aiming for a better coverage can be somewhat like holding a naked call option position, with unlimited risk of loss “from zero to infinity”.

Another significant factor is the number of issues that is discovered during testing and the need to investigate them. In many cases, value-added analysis of seemingly incorrect behavior takes much more time than discovering such behavior. This is especially likely in automated tests. The qualification of the testers is an important moment here as well. This work cannot be done just by anyone, even if a very thoroughly spelled-out step-by-step procedure is provided.

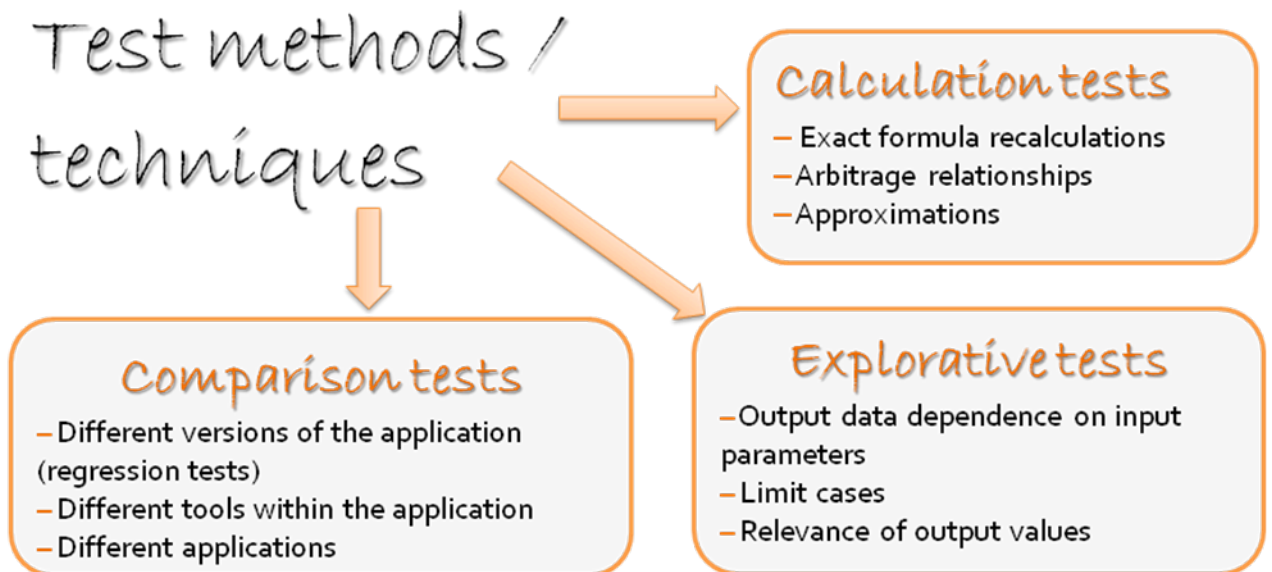
Even when a formalized test procedure is in place, explorative testing could still prove to be more productive given that it is performed by a subject matter expert. The reason is that due to a significant number of test conditions to be checked, the “brute force” method of trying to create a test for every single condition usually stops halfway. During explorative tests, specialists find error-prone conditions intuitively and their evaluation of result is intuitive rather than formal.

Thus selecting the right staff for financial testing is of outmost importance. As you might guess, it is difficult and expensive to attract financial experts to work in software testing. A good alternative is to hire individuals with suitable education, background and skills and train them intensively on the job. Higher education in mathematics and an ability to learn quickly are mandatory requirements. Programming skills add a significant advantage.

## Test methods/techniques

Test methods employed in financial testing can be roughly divided into three categories – from easy to complex - *Calculation Tests*, *Comparison Tests* and *Explorative Tests* (see Fig.2).

**Fig. 2 Test methods/techniques**





## ***Calculation Tests***

Methods of *calculation tests* group require recalculation of values using known functions and formulas. Using given inputs we can calculate the values of cash flows, NPV of the future cash flows, European option prices, etc.

For some financial values mathematically formalized relationships exist and it is easier to verify these relationship that to do a complete straightforward calculations of output values. For example, we can consider the term “arbitrage relationship”. It means that we expect the prices of related financial instruments to be balanced so that there is no way to make money on the market without risk exposure. Once again, the idea here is that there is a relationship between the prices of related instruments that is easier to verify than to recalculate the prices.

In general, calculation tests provide reliable test results, but can only be applied to a limited set of the simplest analytics. They are not practical with more complex analytics in which case they tend to take significant time to implement and also require good development documentation in cases when formulas used are not textbook stuff. Whenever they are applicable, these tests can be easily formalized and documented and can be executed by a less qualified staff.

## ***Explorative Tests***

*Explorative tests* require checking for patterns that should be true for the financial values under test and do not require any calculations. For example we can check that a certain input field influences only appropriate output values or we can check that output values movements depend on input parameters, The monotonic property of option price with respect to volatility can be verified in this manner. Monotonic implies that the price is “always increasing” or “always decreasing”. Vanilla option premium will increase when volatility increases. Calculations of option value for different exercise types can be verified using, for example, the following pattern: European call value is less than American call value for non-dividend equity option whereas European put value should equal American put value.

Another way to verify results of a financial application without calculations is to use irregular or limit input data. For example, you can enter negative volatility value, and see that an error message occurs, or enter a very high volatility value and check the pricing algorithm convergence.

Finally, relevancy of output values can be verified by looking at sign and order of the calculated value.

All these tests are quite quick to perform, and they have a certain chance of finding an issue if there is something wrong in the major way with how the application calculates. It is a good technique for touch tests because it is quick. It also provides a wider coverage, but less precision than recalculation tests. Designing that is writing down, such tests or performing them requires a good qualification and deep understanding of the meaning of financial analytics.



**Comparison Tests**

Comparison testing can be used when different applications are available which perform similar calculations. It would obviously be logical if they produced consistent results, i.e. identical solutions to identical problems.

Another case of using this test method is when regression testing against an older version of the same application is considered practical, or when one application is built on top of the other.

It is also used if different tools or models within one application can be compared or, output of different modules within one application can be compared.

There are a few things to be aware of when considering comparison testing. First of all, it must be possible to configure input parameters in the same way for the compared applications or modules. There can be dozens of both user-driven and system-driven input parameters, which will need to be synchronized.

The method itself does look quite simple, which is true as long as no differences are found in comparison. But that is rarely the case. The analysis of differences can be quite painstaking and requires a good qualification. The tester must know both applications well. Comparison testing of output values against an older version is a method of choice in regression testing.

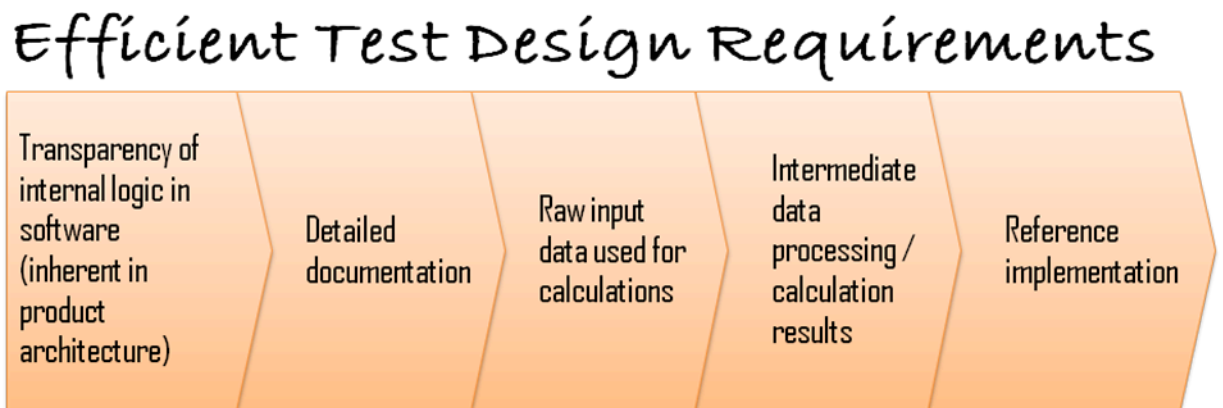
Sometimes comparison testing is a mandatory approach or an approach of choice e.g. in case when consistency with another application is a business requirement.

As a summary, the direct factors in choosing this or that method are time constraints, requirements to coverage and reliability of results, as well as staff qualification.

**Factors in efficient test design**

After a test method is defined we fall into the test design stage. This is where specifics of financial testing come into play and reveal the important factors in efficient test design (See Fig. 3)

Fig. 3 Efficient test design requirements



When we talk about raw input data, it is obvious that ideally the whole universe of data, which can be used in the application, must be available to the test team. This allows us to vary the input data in each test pass, which is great for test coverage.

Let's cite as an example our work on a new application whose purpose is to relate the cost of equity trade to a benchmark calculated from historical data recorded on a



reference market exchange. The reference data is a selection of exchange trades for a configurable period of time, filtered by a certain criteria specific to the market. A basic choice for a test designer is to either lock the tester down to a specific expected benchmark value for a specific exchange and time period, or to instruct the tester to select a random data sample with required properties and recalculate the expected benchmark. In the latter case, each pass of the test uses a unique data sample and hence increases the test coverage.

## Tests automation

Automation of tests is common practice for time and costs savings and has its own specifics when implemented in financial software testing. The big issue that we are faced with from the start is where to get reference implementation, which is worth trusting and convenient to work with. Though it is usually possible to recalculate all financial formulas using standard Excel functions, this approach requires a lot of time and leads to problems when verifying individual tester's recalculations, because complexity of formulas is high. Another approach is to verify the application calculations using previous versions of the product or other products of the same kind. The first approach (i.e. using previous version) is an approach that is commonly used for automated regression testing. The second one suggests searching for an authoritative enough application. However, a tester may not be able to configure this application in the same way as the application under test, because of differences in business logic.

Another difference is related to real-time data. The structure of calculator input data changes with time. Recalculations should take this aspect into account and adjust structure of input in accordance with real-time changes.

Financial automated test gives opportunity to cover most possible test scenarios. However, it is important that automated financial testing does not completely remove human factor from the testing process after scripts for automated tests are written. Manual retesting of failed tests and thorough investigation still require significant time due to inherent complexity of issues at hand. This allows us to make sense of "Failed" results and increase the efficiency of future automated testing.

Finally, it should be noted that the process of automation scenario creation is in itself extremely beneficial. When faced with the task of creating an efficient test, the writer tries so many scenarios and gets such a deep understanding of software, that many business logic issues reveal themselves in the process.

## Training

Having a well-prepped and trained staff is of outmost importance. In our experience mathematically educated staff is required. Newcomers without financial background should start with a standard program based on authoritative financial reference books, such as the ones by Hull and Fabozzi. Usually it takes approximately half a year for a person with mathematical education and basic computer skills (programming background is a plus but not necessary) without financial background to feel more or less confident in a financial project. For more experienced QA staff seminars on more advanced topics should take place. Highly qualified or end user consultant on a part-time basis will help a great deal, as well as cross-training on other similar projects in-house to give the staff a broader outlook. In-house experience in other departments



available to share is desirable. Encouraging the staff to pass exams for recognized financial certificates, such as FRM and CFA, is also extremely beneficial.