

Close the gaps between system implementation and business requirements with Model Based Testing



Contents

1	Introduction	3
----------	---------------------	----------

2	What is Model Based Testing?	4
2.1	Why use models?	4
2.2	Why use model based testing for Financial Services?	5

3	Building a Model: What You Need to Know	6
3.1	Identify the best source for requirements	6
3.2	Select the right model	6
3.3	Consider structured analysis techniques	7
3.4	Design the model	7

4	Putting the Model to Work: Model Based Testing Tools	8
4.1	Capgemini's Model Based Testing Workbench	9

5	The Business Advantages of Model Based Testing	10
----------	---	-----------

6	Conclusion	11
----------	-------------------	-----------

1 Introduction

Today more than ever, the success or failure of a financial services institution is intricately linked to the quality of its information and systems. IT quality is achieved and maintained through software testing, and one of the most time-consuming and resource-intensive tasks of software testing is the creation of test suites.

Test scripts are used to verify that the final software implementation matches the original functional specification. Scripts are written to ensure that a new application will support the associated business processes and perform well enough to meet a project's business objectives, as well as the expectations of users and stakeholders. There is a direct correlation between the number of test scripts and the quality of the tested system or application. Using a test suite that covers all requirements and supports traceability can reduce defects and ensure all requirements are tested.

2 What is Model Based Testing?

Model based testing (MBT) is a methodology and IT toolset designed to accelerate the creation of test scripts. Under the current method, test professionals must manually create each test script and use case. This task requires both testing skills and domain expertise to build scripts and cases to test financial services applications.

By using MBT tools, financial institutions can automatically generate test cases from models to describe the application, object or system under test. Models and model components are more flexible and robust than manually created test assets. A single model can generate multiple test cases in far less time than it would take to write the same number of cases manually. Models can also be re-used to test other applications with similar functionality. Overall, using model based testing:

- Significantly reduces the time required to complete functional testing, bringing measurable cost reductions
- Provides increased test coverage, allowing test professionals to identify more defects

Depending on the experience and domain knowledge of test professionals, the time required to thoroughly test a newly implemented, core financial-industry-application system such as insurance claims processing or commercial lending can be reduced by as much as 30% through model based testing.¹

2.1. Why use models?

For test professionals, model based testing is a compelling methodology because the models can be used to:

- Precisely define functional requirements to better match user expectations;
- Reduce planning time by providing components of functional specification documents; and
- Automate test-case generation, creating of more test cases in less time.

¹ Based on Capgemini experience using model based testing on testing engagements for financial services institutions, compared to previous manual testing method in use.

The goal of model based testing is to create a generic set of model components that can be reused in every new project. Another approach, which is particularly helpful in a complex, financial-enterprise-IT environment, is to build model components specific to an application system which can be reused when incorporating business-process-improvement flows. Model based testing also lets you adapt models quickly for similar systems. For example, Capgemini created models to test a commercial lending system which was being launched in Bulgaria. The same models were quickly adapted to test the same application during the rollout to Albania and Romania.

2.2. Why use model based testing for Financial Services?

In Capgemini's *Financial Services World Quality Report*, only 16% of banking and capital markets respondents reported they reused 50% or more of testware. But all agreed that reuse provides the best opportunity for cost reductions. When testing financial services applications, most respondents felt domain expertise is increasingly prevalent and required among test professionals. They also reported a low adoption of tools for requirements definition and management, which in turn leads to higher testing costs since defects introduced in the requirements phase are the most costly to fix later in the lifecycle.

To address these challenges for banks, insurers and capital markets firms, Capgemini recommends a model based approach to testing financial systems. Model based testing brings rigor to requirements definition and control processes and engages testing teams earlier in the process to support requirements management and test case generation.

3 Building a Model: What You Need to Know

3.1. Identify the best source for requirements

Each system under test can be unique to the business process that it will support. So identifying the best approach to build the models for model based testing can be challenging. Functional requirements— like those derived from use cases—can serve as a starting point for the creation of the model.

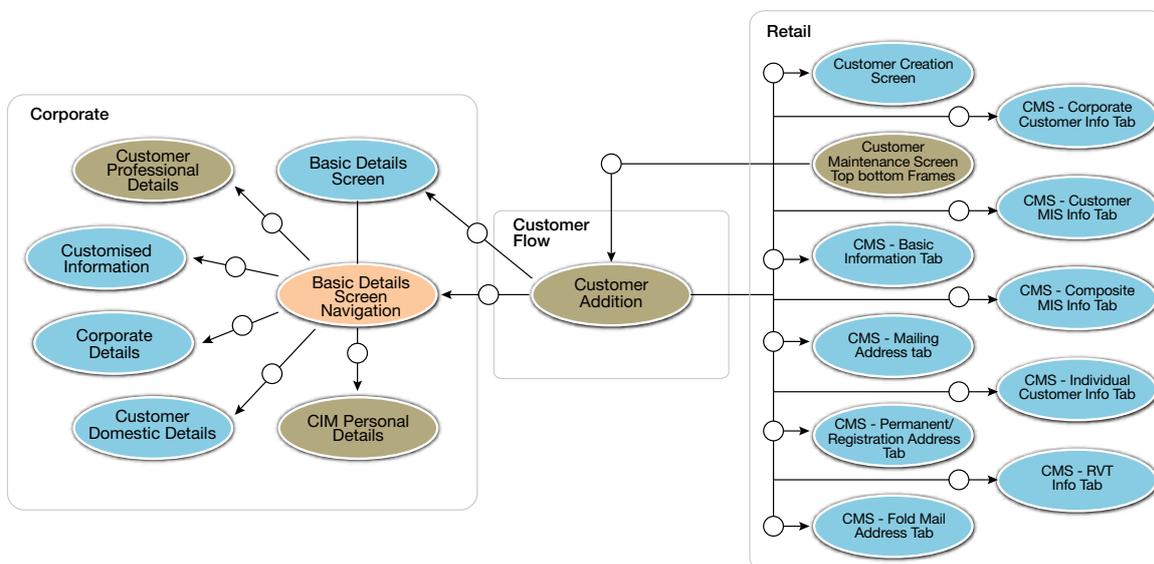
3.2. Select the right model

Most MBT tools support the development of models using unified modeling language (UML). When building models, test professionals can use a software-application-development model or create a separate test model.

3.2.1. The case for a separate test model

Using a separate test model offers some advantages. A test model is a unique model developed specifically for the MBT project. Therefore, the model design does not have to accommodate the needs of other project stakeholders: software analysts, designers or developers. It is purely used for testing.

Exhibit 1: Example of a separate test model



This approach has some benefits:

- The test model is more targeted than a development model. For example, test models use test data and Boolean logic operators;
- Errors that may be present in the development model will not be replicated in the test model, allowing testers to find defects created during the software-design phase.

3.2.2. The case for using a development model

The advantage of using a software-development model is time savings. If no separate model is created for testing, testing can begin sooner. Some MBT tools allow the use of other models like pseudo code, decision tables, Markov chains or models specially designed for the tool.

Why use TMap®?

- Allows test professionals to correlate test strategy with business risks
- Provides a structure for test strategy estimation and planning
- Supports early defect detection and reusable test asset creation
- Strengthens transparency and management for test processes

Guidelines for the TMap® Model Development Process

- Review requirements to develop a high level approach
- Identify the TMap® test design techniques to use
- Prepare a technique checklist
- Create the model components
- Identify the business flows to be modelled
- Integrate the model components while creating a model
- Verify the model
- Simulate the model
- Generate the test cases and scenarios from the model

3.3. Consider structured analysis techniques

By building a model of a system in a way that supports structured analysis, the desired behavior of the system—given specified inputs—can be tracked. For the testing team, a structured approach is very important since test-case scenarios are comprised of sequences of interactions with the system; each scenario having a pre-defined “correct” processing outcome. Efficient and comprehensive functional testing—exercising all possible combinations of user-system interactions—is the goal.

Structured test design techniques, such as those comprising Capgemini’s proprietary TMap® testing management methodology, are proven to be effective in the model-development process. The current version of the TMap® methodology—TMap NEXT®—supports the process-focused description that is fundamental to test-model development and incorporates the business objectives for the software project as guidance for the testing process. Incorporating four features essential to defining models for model based testing, TMap NEXT®:

- is based on business driven test management;
- describes a structured test process;
- includes a complete test-management tool box;
- can be adapted to any test practice.

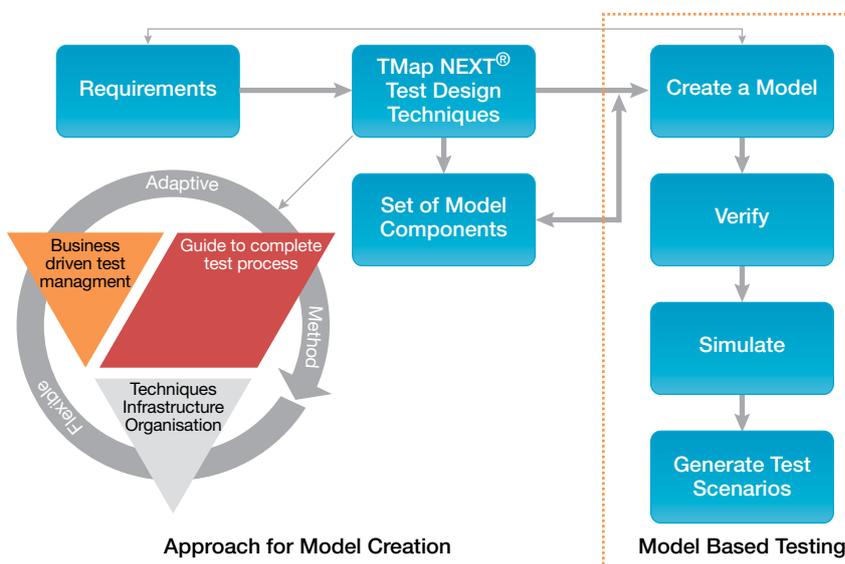
3.4. Design the model

By using the TMap® test design technique checklist, models can be created quickly and efficiently using pre-packaged components. TMap® can help test professionals:

- Prepare a set of generalized model components and packages;
- Integrate or reuse generalized components and packages when building models for new requirements;
- Identify common operations for systems under test which can form a set of model components specific to the product. These model components can be integrated and reused when building models for the specific requirements.

By applying these testing techniques, test professionals can develop a set of generic packages to test standard functions such as Windows compliance, screen validations, specific field tests or business rules. These generic packages can be customized for the products or applications in use based on company-specific requirements.

Exhibit 2: Capgemini’s approach for model creation using TMap®



4 Putting the Model to Work: Model Based Testing Tools

A model based testing tool can automatically generate multiple test cases from a single model and can include traceability to and from different requirements. The MBT tool generates test cases by using paths through the system on which the test cases are based. These test paths are the same that are used in test design techniques. When generating test cases, some applications can automatically check for consistency. If any cases are identified as inconsistent, errors must be fixed in the test model.

Using formal unified modeling language (UML) model test data helps to generate automated test cases. For financial institutions, using proper test data for models provides a balance between the need to test systems using relevant financial data while providing auditability and traceability to meet customer data privacy regulations.

Some tools have plug-ins for various automated test execution tools. Among the tools on the market that can be used for MBT²:

- HP Quality Center
- IBM Rational Testing Tools
- Smartesting Test Designer
- VectorCAST/Cover
- Conformiq Qtronic
- All4Tec MaTeLo (Markov Test Logic)
- Microsoft Spec Explorer

Model based testing calls for a collaborative approach where business analysts create models and quality assurance analysts create test cases using a one click approach. The cases are integrated into a test management tool which helps ensures requirements and test cases can be exported for better traceability and automation using leading industry tools.

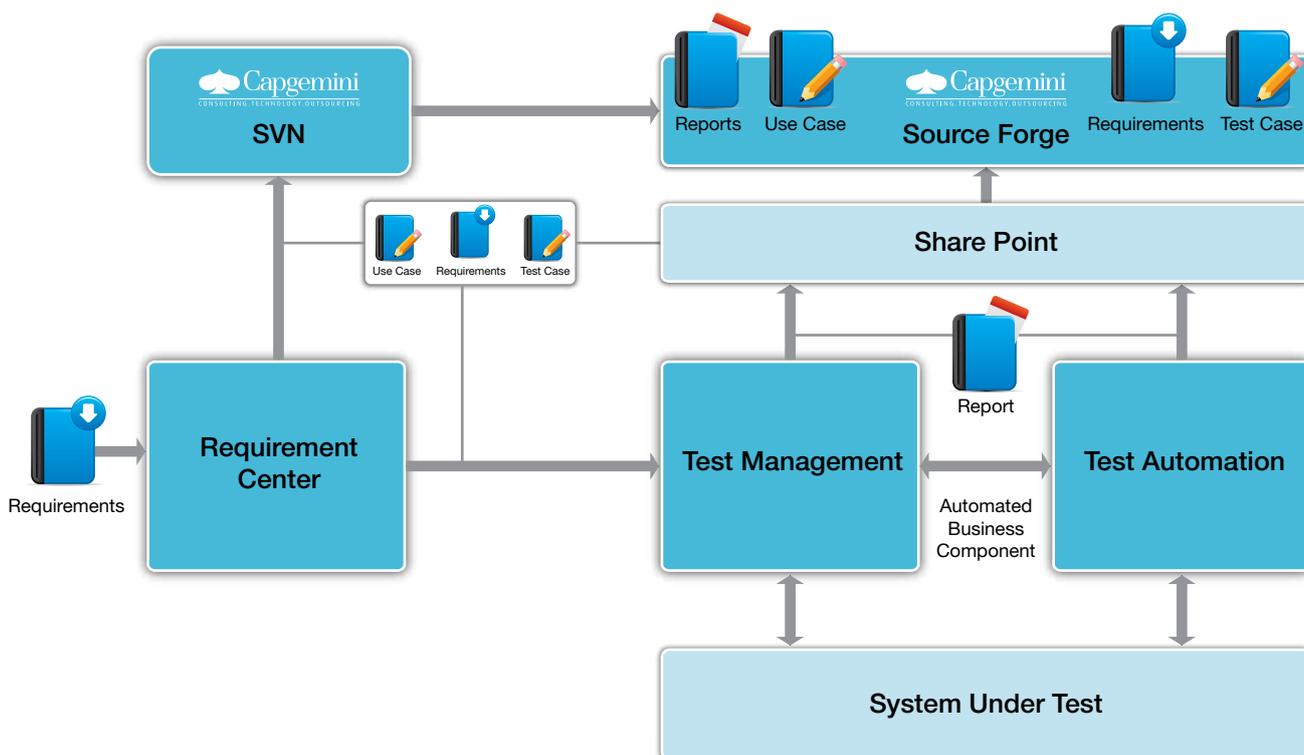
² All products or company names mentioned in this document are trademarks or registered trademarks of their respective owners.

4.1. Capgemini's Model Based Testing Workbench

Capgemini created a Model Based Testing Workbench to address the challenges of testing commercial applications for financial services. The Workbench lets test professionals build test cases efficiently and effectively by using predefined accelerators and tools. We've used this approach on major engagements across the world to help Capgemini's financial services clients lower costs, reduce time to market, manage scope and raise quality for commercial off the shelf implementations. Our Workbench:

- Uncovers gaps and supports better communication to ensure requirements are complete and understood by all stakeholders
- Automatically generates test scenarios which can be run through standard test management tools
- Integrates with industry-leading HP and IBM solutions
- Automatically creates bi-directional traceability
- Supports easy implementation of change requests

Exhibit 3: How it Works: Capgemini's Workbench Model



5 The Business Advantages of Model Based Testing

Since model based testing generates test cases from a requirements-based model, it is easy to re-generate the test cases when requirements change for the software under test. After a requirements change, test professionals can adjust the model and generate new test cases. This enables the dynamic creation of test cases, a tremendous time saving advantage over traditional methods where tests must be recreated manually. Financial services institutions can invest in building test models once and reap the benefits over the whole lifetime of the application.

When the system under test adds new features, test professionals can incrementally add them to the model. If issues arise with the quality of the system, test professionals can improve the model and expand test coverage. If new people are introduced to the testing team, the test model serves as a training guide to jump start productivity.

When creating test models, questions often arise about the quality of the functional requirements. These questions merely highlight any defects in requirements earlier in the project lifecycle. To solve such issue, test professionals can present them to the designer or analyst of those requirements.

In traditional testing projects, questions on requirements would normally come up during the manual specification of test cases, so any defects found at this stage may already be added to the code or prototype. With model based testing, these discoveries can be found during the preparation phase while building models using functional requirements. This helps the software testing team 'shift left' in the application development process.

Model based testing brings other advantages to the business:

- **Supports reuse.** Reusable repository of model and model components provides proven productivity gains and time savings.
- **Decreases rework and errors.** By eliminating ambiguous, unclear, and incomplete requirements through up-front definition and validation, basic flows can be verified and errors minimized.
- **Provides structured, traceable change management.** Model based testing makes it easier to manage changes in the project by merely updating the model.
- **Increases test case effectiveness.** Test case generated by model based testing techniques such as Capgemini's proven TMap® approach are more effective at providing optimum coverage and quality.
- **Reduces total test preparation time.** Reusable models and model components help minimize test preparation time by allowing hundreds of test cases to be created with one click.
- **Ensures full coverage for validation.** Checklists assure all validation aspects. For each field, a model component checks all possible conditions across all test cases so there is no chance of missing a validation check while building a new model. For example, once a model component is built to validate credit card numbers, every test case created with the model will have that validation check built in.
- **Lowers efforts for integration testing.** Since it's easier to integrate functional or software components into the model, integration testing can be accelerated.

6 Conclusion

Model based testing, a methodology developed to automate the functional-testing phase of a systems implementation project, can generate significant software quality assurance benefits. In addition, it can allow financial services institutions to reduce testing time by as much as 30% over traditional testing methods.

By creating models specific to IT or business-application environments, test professionals can clarify business requirement definitions and automate the generation of test cases to thoroughly check system functionality and identify system errors that may frustrate all users—internal stakeholders and financial services customers alike. Once created, the model based testing application model can be reused to support future testing of the target applications, including changes to support business process improvements or new systems software releases.

For financial services institutions, the use of model based testing can simplify the testing of complex, core financial systems by building a library of models and model components that can be reused across different divisions or regions that have customized versions of the same system.



About Capgemini and the Collaborative Business Experience

Capgemini, one of the world's foremost providers of consulting, technology and outsourcing services, enables its clients to transform and perform through technologies.

Capgemini provides its clients with insights and capabilities that boost their freedom to achieve superior results through a unique way of working, the Collaborative Business Experience™.

The Group relies on its global delivery model called Rightshore®, which aims to get the right balance of the best talent from multiple locations, working as one team to create and deliver the optimum solution for clients.

Present in 40 countries, Capgemini reported 2010 global revenues of EUR 8.7 billion and employs around 112,000 people worldwide.

Capgemini's Global Financial Services Business Unit brings deep industry experience, innovative service offerings and next generation global delivery to serve the financial services industry.

With a network of 18,000 professionals serving over 900 clients worldwide, Capgemini collaborates with leading banks, insurers and capital market companies to deliver business and IT solutions and thought leadership which create tangible value.

For more information please visit www.capgemini.com/financialservices

Capgemini Financial Services

6400 Shafer Court,
Rosemont, Illinois 60018
Phone (847) 384 6100, Fax: (847) 384 0500

