

Test Methodologies

This document outlines a proposed approach to software quality assurance. It is divided into three sections:

1. **Quality Assurance Methodologies** – focuses on software quality assurance testing methodologies and outlines strategies for managing quality assurance functions.
2. **Leveraging Test Tools** – focuses on test tool customization methodology to leverage automated function testing tools like QuickTest Professional and other automated testing tools.
3. **Effectively Deploying Staff** – focuses on a methodology for effectively deploying staff on a QA project in order to achieve quality results at a reasonable overall cost.

Section 1: Quality Assurance Methodologies

Most major published testing methodologies focus on a “perfect world” solution, as if resources and time for testing is readily available. A better approach is to focus on achieving quality results, while also recognizing that quality assurance methodologies are pragmatic and practical for a customer’s needs.

As an example, our customers’ most common concern has always been that “there aren’t enough resources to do all the testing that needs to be done.” To deal with that concern, Telesis did extensive research into identifying the most critical areas for testing and developed a “Triage Methodology” which lets customers focus testing on areas that have the greatest potential for defects. This methodology is explained in another white paper, entitled *Triage Regression Testing Guide*. A complete list of the 13 methodologies Telesis has developed and deployed appears in the Appendix to this white paper.

Approximately 60-90% of software defects come from about 20% of the software, so focusing first on these areas is most productive. There is no magic here – if resources are limited, testing may still be more limited than desired – but the objective is to use available resources as effectively as possible within time and resource constraints.

Section 2: Leveraging Test Tools

A good test tool process must be tool independent so that it can be applied to a wide variety of automated functional tools, including the products offered by HP/Mercury, Rational and Compuware.

Functional test tools have long been on the market, but the abandonment rate for these products has been high. The reason for high abandonment is that the tool vendors have focused on making their products more programmatically robust, but have not provided an efficient and structured process for implementing the test tools in a manner that makes automated test scripts consistent, durable and easy to use and maintain.

For example, tool vendors typically place little emphasis on who will use the tool. Automated functional testing tools are intended to drive down the cost of testing. That means they should be easy to use by testers with non-programmatic skill sets, but vendors typically provide little or no information on how to use the tool effectively. Likewise, test tool vendors provide limited information on steps that can be taken to reduce the need and time required for updates to automated scripts as business needs change.

A structured approach to deploying functional test tools is necessary.

1. **Develop a customized approach to training.** Most vendors train on their automated tools using a standard classroom example. For example, HP/Mercury teaches everyone how to build a web-based airline reservation application.

But what if the application the customer wants to automate is a .NET application or a SAP based payroll system? It's always easier to learn with your own application. Customized training to customers using their own application in their test environment is the preferred approach. An additional benefit of this approach is that scripts created are "real" and can be used by a customer, while airline reservation scripts are tossed out after training is done.

2. **Build test scripts *without* programming.** Use a "less is more" approach to programmatic test scripting. The majority of tools today require programming skills to build the test scripts. It is very laborious and time consuming. Our test philosophy instead is to promote the utilization of the newest 5th generation functional test tools, like Telesis' product, BT3. These tools do not require any programming and instead will automatically generate the test script as you record normal business transactions. This results in test scripts that are much more durable, easier to read and maintain and ensures that the tool works effectively with the application under test.
3. **Develop strategies for driving down labor costs.** Many companies place the 4th generation tools in the hands of their most technical staff members. That raises labor costs and makes little sense. With 5th generation tools, there is definitely a need to have at least one highly skilled test tool technician. However, most changes are routine, such as updating an expected calculated price from \$305 to \$310. A task like that could easily be handled by a Business Analyst in a new 5th generation toolset.
4. **Create structured work processes.** Develop detailed sequences of events for testers to use to implement functional test tools in a structured manner. The

sequence walks the user through learning the application, creating application maps, and creating and documenting the test scripts.

5. **Leverage the use of shared objects.** The more advanced functional tools provide support for shared reusable objects. This offers the potential for substantial maintenance savings. For example, assume that a test tool user is updating the web front end for an on line clothing store. There may be a screen object on each page where the customer can enter quantity, color and size. Assume that screen object appears on 50 different screens. What happens when a decision is made to add additional fields to that screen object – such as a choice of fabrics?

The traditional approach would be to have someone go in and update 50 screens. Instead, create a shared object to support those 50 screens. If that shared object changes, a single update is made in one shared reusable component and all 50 screens are automatically updated from that one entry.

The best approach is to leverage test tools from a management perspective, rather than a traditional development strategy. Give priority to training, labor deployment strategies and emphasis on durability and ease of maintenance to maximize the ROI from the investment in automated testing tools.

Section 3: Effectively Deploying Staff

The most critical issue on many projects is deploying human resources efficiently and effectively. The cost of human resources is by far the biggest expense category on any project but often a human resource strategy is missing. You should customize your HR strategies to drive work down to the lowest cost resources that can do the project effectively.

For example, in the implementation of automated 5th generation testing tools, you can focus on training quality assurance analysts to use the tools rather than developers.

Within your Quality Analyst teams, focus on customizing training, so that one or two “expert level” QA staff members understand the testing tool in depth while other team members, who will only make maintenance changes to scripts, will only require a basic skill level. If your resources are constrained, or you do not have the expertise in house, consider outsourcing your testing to a test facility. The time and resources you may save and the expertise you can gain may well justify the investment.

APPENDIX

This Appendix contains a list of 13 detailed test methodologies that Telesis has developed in working with its clients. Telesis also helps its clients design quality assurance strategies that fit with the client's management processes and mesh the individual QA processes into a cohesive quality assurance process. The summary schematic attached to this report is an example of a strategic management quality assurance process developed by Telesis and customized for the unique needs of a client.

1. **TTCP** (Telesis Test Tool Customization Process)
TTCP is a testing methodology that allows testers to develop automated test scripts in an efficient and thorough manner. TTCP is distinguished by an emphasis on creating automated test scripts that are more reliable, reusable and repeatable by use of a structured process for customization of automated test scripts. TTCP has now been incorporated into a functional test tool product called Business Technology Test Tool (BT3). In client implementations, BT3 has achieved no less than an **85% ROI** based upon resource time invested. BT3 is over **seven times faster** than traditional functional test tools programmatic approaches.
2. **TTM** (Telesis Test Methodology)
A set of practices and standards for managing a testing facility. TTM has a proven record of producing high lab productivity at low unit costs (costs per transaction).
3. **TFTA** (Telesis FlexTeam Approach)
TFTA builds off TTM, but allows more flexibility for business needs. TFTA creates a test facility that can handle routine business transactions when production volume is high, or switch to software testing when the priority is getting out the next version. TFTA is structured so that the switch from test to production can be done on very short notice – usually on a same day basis if that's what is needed.
4. **TIVR** (Telesis Interactive Voice Response testing)
A six phase structured and automated process for testing voice response applications, whether DMTF or tone.
5. **TDCM** (Telesis Data Conversion Methodology)
A step-by-step process for converting data formats or structures. This is of particular value to clients who are changing their data formats or merging different applications. TDCM is segmented by phases with tasks for each phase clearly identified.

Unlike many data conversion methodologies that focus only on the data, TDCM also covers the “domino impact” of data conversion. For example, it may be easy to convert an account number from a 9 digit field to a 12 digit field, but TDCM also covers the consequences of that type of change, such as revisions of web screens and changes to the layout of printed documents.

6. **TPTP** (Telesis Performance Testing Process)
A structured approach to performance and stress testing. Unlike traditional performance testing, TPTP also covers queue management functions, background noise, memory leakage and the implications of peak volumes on application cleanup activities.
7. **TTA** (Telesis Triage Approach)
A risk managed approach to developing test scripts that facilitates the process of defining the most critical test conditions that need to be tested and reduces your test beds to a reasonable size. The TTA approach focuses on four components:
 - a. **High Volume Activity**
 - b. **High Defect Rate**
 - c. **High Recovery Costs**
 - d. **New Features**

A more detailed description of this methodology is set forth in the white paper entitled *Triage Regression Testing Guide*.

8. **TRDP** (Telesis Requirements Development Process)
A detailed Q&A approach to developing and evaluating business requirements. The process is detailed and includes a 30+ page manual for users. The manual can be used by business representatives and does not require development or QA skills.
9. **TT** (Telesis Templates)
A comprehensive set of QA templates designed to be efficient and thorough. All of the templates have been extensively tested and refined in actual use. Testing templates in the overall package include test plans, test cases (two variations), test conditions, risk analysis, signoff and QA process assessment form.
10. **TDG** (Telesis Discovery Guides)
Discovery Guides provide a comprehensive set of questions for doing a self assessment of existing QA processes. The guides cover the test environment, applications and tools, test planning & execution, version releases and test results analysis. Customized versions for web, client/server and mainframe applications are also available.
11. **TSD** (Telesis Software Dictionary)
An easy to read 11-page document that provides detailed definitions of common quality assurance terminology.
12. **TDTG** (Telesis Development Testing Guidelines)

An easy to use and concise document that outlines testing guidelines for use by developers.

13. UserOne

A structured focus group methodology that can be used to get user feedback on applications in a disciplined and measurable approach. UserOne translates the information into metric format that defines the user satisfaction level and whether satisfaction levels are going up or down. The process has been used over 100 times across the country with excellent results.

Note:

The methodologies listed can be customized for the unique needs of an individual client. Customization adjusts the methodology based on the client's application, the complexity of that application, risk tolerance, company terminology and other factors. Refinement and expansion of the methodologies is ongoing.

