

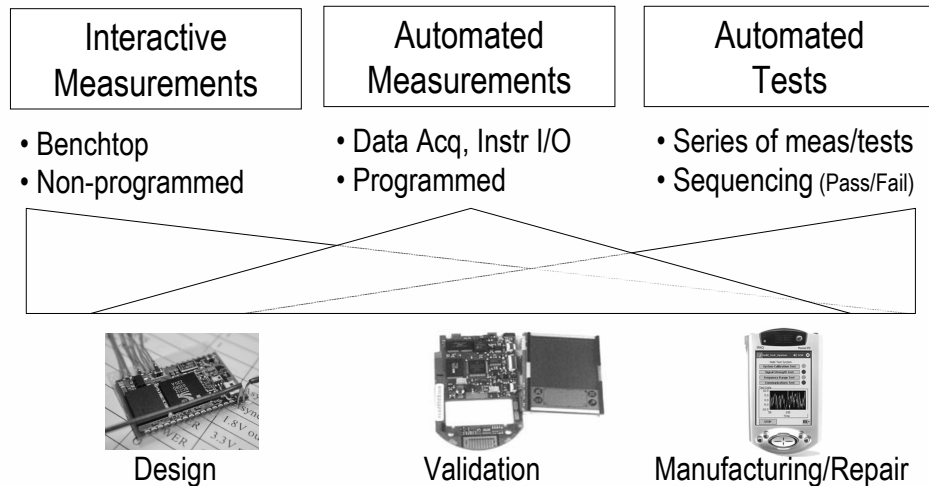
1.4 Managing Highly Automated Test Sequences with NI TestStand

Bridging Validation and Test with Common Software Technology



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

Automated Tests versus Interactive or Automated Measurements



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

As we have previously discussed, the nature and use of automated tests, measurements, and interactive measurements play key roles throughout the design cycle. Where interactive and automated measurements are primarily used in design and validation, automated tests dominate the manufacturing and repair space. This stems from the need for highly automated and efficient test execution for production where little to no human interaction is involved. These tests are typically more specific in the type of measurement taken, and once written, they are less interactive or versatile. The focus is more on the automation than on the measurement.

Identifying Test System Requirements

- **Technical requirements**

- **Specific tests** (frequency response, rise time, duty cycle, amplitude, and so on)
- **Execution flow** (single/multithreaded, branching, error handling, product family testing, multiple limit/configuration files, and so on)
- **Result logging** (custom reports, database, XML, “on the fly”, print unit failure ticket)
- **Test development** (develop tests using “X” language, reuse existing tests)
- **GUI** (pass/fail yield history, multiple views based on login, debug mode, error display, execution status, localization, embedded test measurement display)

- **Business requirements**

- **Reduce cost of ownership**
- **Provide greater test system flexibility**
- **Increase test code reuse**
- **Improve quality of tests**



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

Before developing your automated test system, several requirements need to be taken into account. Examining these forms of system requirements will emphasize the importance of a well architected software framework. Nearly every test project typically contains a set of technical requirements and business requirements or goals. Without a well defined software architecture, most test engineers will not have enough time to complete the project and will be forced to only focus on the technical requirements of the system. The business requirements are defined for good reason to reduce costs, increase scalability, and improve quality. Without addressing the business requirements in your overall test software architecture the effective lifetime of the test system, potential cost savings, increased scalability, and improved product quality will be sacrificed.

Balancing Test System Requirements

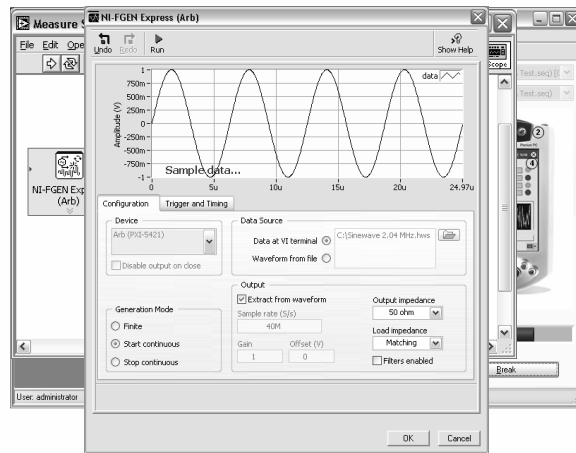
- Today's leading electronics companies focus on building a modular, reusable test framework
- Developing and maintaining test software without this framework is very expensive and time consuming
- Provides scalable solutions for use across multiple products, sites, and departments within an organization



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

Rather than defining a test system architecture around a specific product to be tested, today's electronic companies are focusing on building a standardized test framework based on a one-time investment to provide the scalability and extensibility for use across multiple products, multiple manufacturing sites, and within different organizations in their company. The modular test software architecture provided by NI TestStand test management software saves these companies millions of dollars in combined development costs and allows them to develop their test systems in a fraction of the time while reusing tests across the organization.

Flexible and Modular Test Strategy



Test Management

- Organize, control and execute test modules

Test Development

- Create modular test, measurement and automation routines

Measurement Services and Drivers

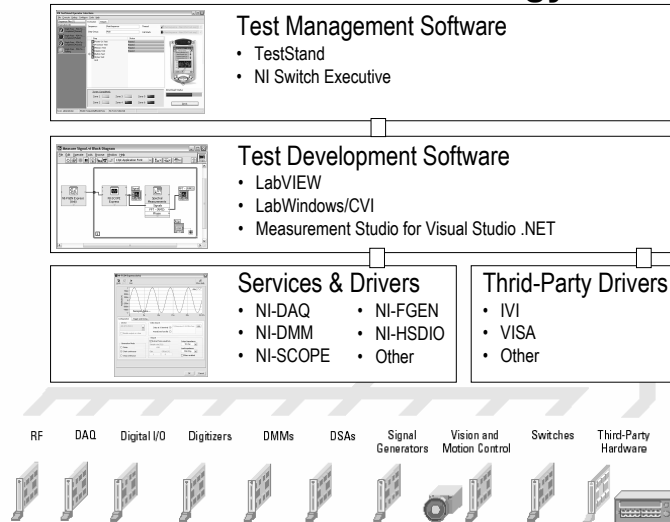
- Instrument and device setup
- Rapid prototyping and validation



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

The general components needed to architect a modular test system are shown above. A modular test software architecture consists of test management software, such as NI TestStand, to manage and execute the validation and manufacturing tests while communicating the test results back to the enterprise systems through built-in database integration. The test management software leverages popular application development environments (ADE) such as LabVIEW and LabWindows/CVI to create test modules that interface with test equipment through the use of measurement services which act as the driver interface to the equipment. Switch management software, such as NI Switch Executive, coordinates the configuration and automation of multiple switch devices through an interactive route editor and single API for controlling all of the switch devices in the system.

Flexible and Modular Test Strategy



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

The general components needed to architect a modular test system are shown above. A modular test software architecture consists of test management software, such as NI TestStand, to manage and execute the validation and manufacturing tests while communicating the test results back to the enterprise systems through built-in database integration. The test management software leverages popular application development environments (ADE) such as LabVIEW and LabWindows/CVI to create test modules that interface with test equipment through the use of measurement services which act as the driver interface to the equipment. Lastly, switch management software, such as NI Switch Executive, coordinates the configuration and automation of multiple switch devices through an interactive route editor and single API for controlling all of the switch devices in the system.

Role of Test Management Software

1. Eliminates programming of common test executive tasks
2. Reuses test code developed in any language
3. Simplifies updates and long-term maintenance
4. Accelerates test system development



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

Test management software is commissioned to perform four key roles. First and foremost, it is critical to eliminate programming of common test executive tasks for each test case written. This saves time and provides a framework that allows quicker development of test modules. The second is to leverage existing test code developed in other languages. Many man-hours have been spent developing test program sets and modules and being able to reuse that code is critical in reducing cost. The third key role is to simplify maintenance and updates. As your test system evolves over time, you will be required to add more functionality as well as adjust the current functionality of your test system. Having a test framework that handles the common tasks is the first step towards simplifying maintenance. The final role of test management software is to accelerate your test system development. The whole idea behind taking the time to develop a modular, reusable framework is to accelerate test system development. This can be accomplished by using the proper architecture and leveraging commercial-off-the-shelf test management software such as NI TestStand.

National Instruments TestStand is a ready-to-run test management environment for organizing, controlling, and executing your automated prototype, validation, and manufacturing test systems. Quickly build your test sequences with NI TestStand by incorporating tests written in LabVIEW, LabWindows/CVI, Visual C/C++, Visual Basic .NET, and C#. Built on a high-speed, multithreaded execution engine, TestStand delivers the performance to meet your most rigorous test throughput requirements.

TestStand is also completely customizable, so you can modify it to match your specific needs, including customizing the operator interface, generating custom reports, and modifying sequence execution requirements. Using TestStand, you can focus your engineering efforts on unique test requirements, while TestStand manages the common sequencing, execution, and reporting tasks for you.

Eliminate Common Test Executive Tasks

Test System Components:

Operations *repeated* for each device tested:

- Operator interfaces
- User management
- Unit Under Test (UUT) tracking
- Test flow control
- Archiving results
- Test reports

Operations *different* for each device tested:

- Calibration
- Configuring instruments
- Data acquisition
- Measurements
- Analyzing results
- Test strategies

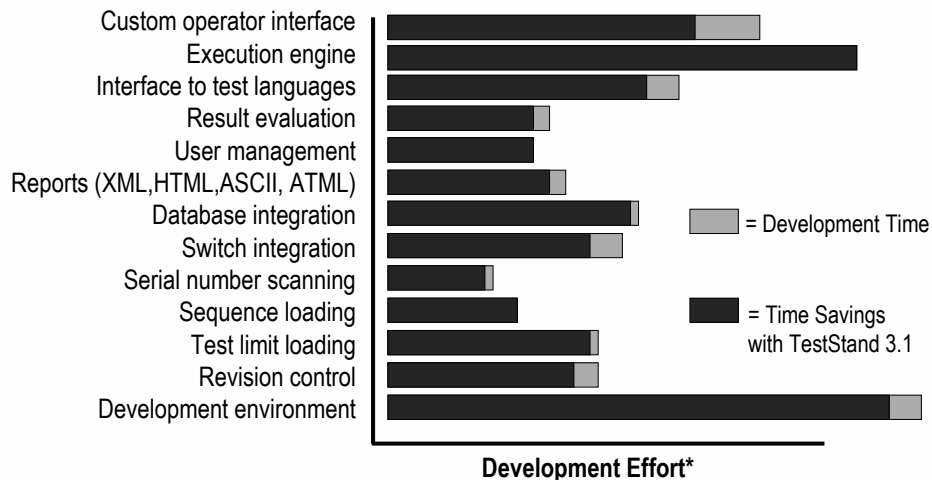


Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

A test management framework is designed to streamline development of your automated test systems. There are many components required for building an automated test system. Some components are specific to the device being tested while others are repeated for every device tested.

TestStand provides an off-the-shelf, customizable test management environment that delivers all of the common components you need along with seamless integration with your test development languages for automating your specific test operations. This provides increased reusability of your test code, faster development, and decreased maintenance efforts over the lifetime of your systems.

Quicker Development Using TestStand Framework



*Production Test Management Software – Build or Buy?: A Financial Comparison Using TestStand white paper



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

Building and maintaining in-house test executive software can be daunting. The operator interface, sequence development environment, database integration, and interface with various test development languages are just a few of the factors you must consider when determining the amount of development time and cost associated with building your own in-house test executive.

Fortunately, TestStand provides many off-the-shelf features and a development framework that helps you build your test systems many times faster. TestStand is also designed with your unique requirements in mind for custom operator interfaces, custom reports, and custom tests. The bottom line: TestStand saves you time and money by providing the core functionality you would otherwise have to rely on someone else within your organization to design, code, maintain, and support.

The relative development times are referenced in the *Production Test Management Software—Build or Buy?: A Financial Comparison Using TestStand* white paper. For more details refer to the white paper.

Develop Simple Modules Inserted in a Common Test Framework

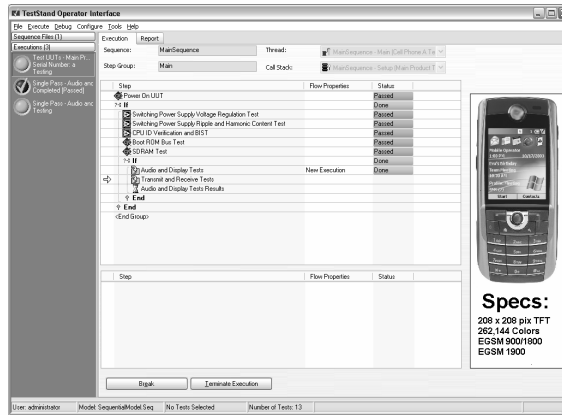


Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

The illustration above shows the system-level tasks performed by your test management software on the right side while the UUT-level tasks defined as test steps in the sequence file are shown on the left side. By leveraging TestStand to manage system-level tasks, maintenance on these tasks is much easier as they are logically organized and implemented in one location (even if they are executed multiple times in different locations during your sequence execution). Also, by developing simple, function specific test modules, you can create additional test sequences in TestStand that reuse these test modules.

Demo – Test Management with TestStand

- Graphical sequence editor environment
- Automate tests written in any language
- Multithreaded or parallel sequence execution
- ASCII, HTML/Web, XML, ATML report generation
- Access, Oracle, SQL Server and other database connectivity



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

This demo shows some key advantages of developing a test system using TestStand as the framework. Executing the “Test UUTs” entry point allows you to enter the serial number for your product test and ultimately allows you to test more than a single product. The test system will dynamically load a specific test sequence for a specific product based on the serial number entered. If you start your serial number with “A” it performs testing on the Smart Phone. If you start your serial number with “B” it tests the Flip Phone. If you enter a serial number that does not start with these two letters, it performs the PDA test.

1. **Common Customized Operator Interface**—The demo is a LabVIEW executable customized from the TestStand shipping LabVIEW OI. It is a single interface that can be used to test 3 different products (2 different cell phones and a PDA). You will notice that the product graphic on the left will dynamically change depending on the product and that the operator and execution view remain consistent
2. **Execution Flow Control (If-Then-Else, Looping, and so on)**—During execution of the three product tests, the sequence contains conditional and looping flow control. These are used to run arbitrary tests depending on the results of the specific flow control statements. These structures give you flexibility and power to define pass/fail strategies and build intelligence into your system.
3. **Asynchronous Parallel Execution**—During the two cell phone tests, there are two sequence calls to test specific functions on each cell phone. The first is an “Audio and Display Test” and the second is a “Transmit and Receive Test”. Both these tests execute concurrently in parallel and are displayed in the two different execution views.
4. **Dynamic Sequence Loading**—As mentioned above, TestStand was configured to read the serial number and dynamically select a sequence based on the value of the serial number. This demonstrates the versatility in terms of testing different products with the same test system. This is often the case in organizations producing more than one product.
5. **HTML Reporting**—TestStand is capable of reporting results in ASCII text, HTML, XML and even in the new ATML standard reporting. This reporting is completely customizable and uses a schema to create a report using the test results obtained in TestStand. This particular report is in HTML.
6. **Property Loading a Sequence for Testing Multiple Products**—Typically when testing products from the same family, the tests and sequence used are very similar if not identical. The only difference is typically the test parameters and step properties. In this case, we are running a standard cell phone test but we are preloading the specific test parameters and properties for an individual test.

Summary

- Test software architecture has major impact on overall effective lifetime of your test systems
- Focus on building a modular test software framework rather than a product-specific application
- Take advantage of test management software architectures that are available for your use



Automated Test Summit | 2005
AN INDUSTRY EVENT HOSTED BY NATIONAL INSTRUMENTS

In summary, focusing on developing a modular test software framework for your test software architecture you are able to balance your technical requirements with your business requirements. Leveraging a commercial-off-the-shelf test software framework such as TestStand can also greatly reduce your development time as the framework already provides the modularity and the structures and features required to build a complete test system.