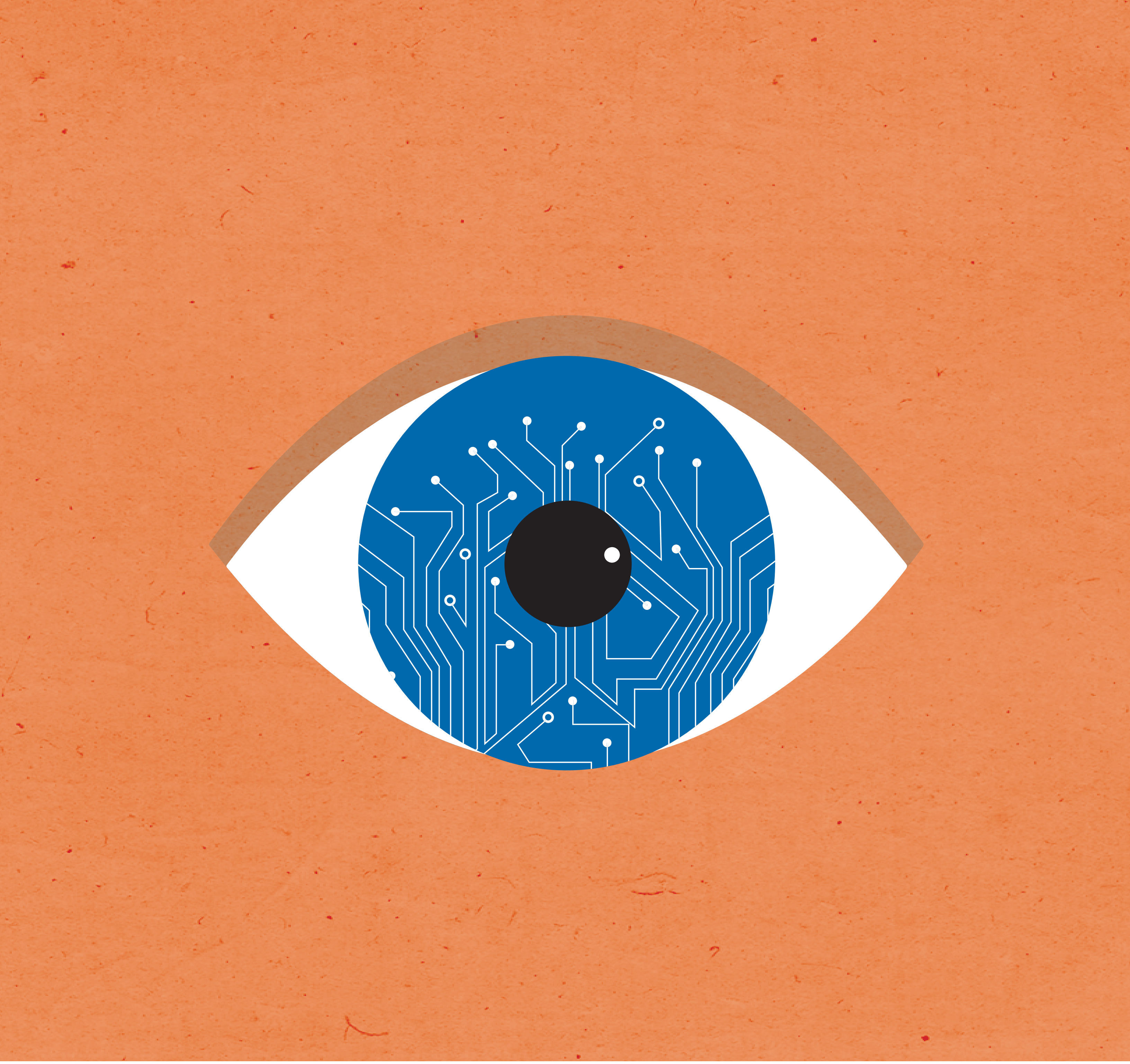
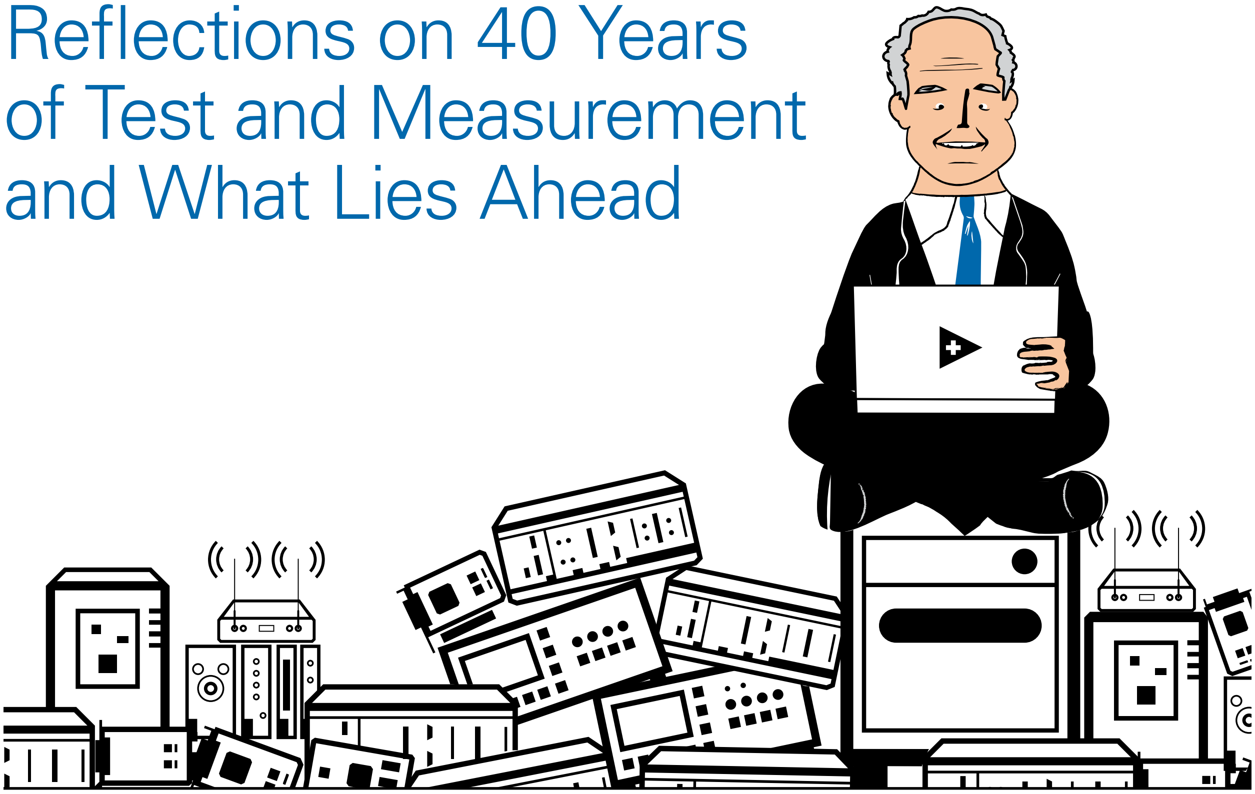


Automated Test Outlook 2017

Through the Eyes of NI's Cofounder: A Special Issue Guest-Edited by Dr. James Truchard

NATIONAL INSTRUMENTS™

Reflections on 40 Years of Test and Measurement and What Lies Ahead



As I approach the end of my 40-year career as CEO of National Instruments, I am reminded of the great progress and innovations the test and measurement industry has witnessed since 1976. We have gone from an industry driven by vacuum tube technology in the era of General Radio to a time when the transistor ruled with Hewlett Packard to today, when software truly is the instrument—a transition that NI helped shepherd. Moore's law has taken us for a wild, fast ride to say the least, and just when you think it's run its course, process innovations extend into new dimensions (literally) and push performance even further.

Just like the transistor, NI started from humble beginnings, but it has relentlessly focused on engineering great products and empowering world-changing innovation through our customers and platform technology. Allow me to reminisce on what the past 40 years have taught me and where I see this market heading as I shift into the next phase of my career.

"Do for Test and Measurement What the Spreadsheet Did for Financial Analysis"

When Jeff Kodosky, Bill Nowlin, and I started NI in 1976, we saw tremendous room for innovation in how engineers and scientists interacted with and built test and measurement equipment. We founded the company on the premise that there had to be a better way to serve the test and measurement needs that we, engineers and scientists, faced. We couldn't buy it off the shelf but at least we wouldn't have to build it from scratch.

The general purpose interface bus (GPIB, IEEE 488) was our gateway. Our vision, as stated in 1983, was to "do for test and measurement what the spreadsheet did for financial analysis." Stated today, the sentence loses some of its power, but think about the early '80s. At the time, the tools for financial analysis were "locked up" and too expensive for anyone without a big budget to access them. The early incarnations of spreadsheets turned this situation on its head, and that is exactly what we wanted to do. We wanted to make it so that any engineer or scientist could access the same tools or platform used by the R&D teams of the leading technology companies. It was a radically empowering view at the time and, in many ways, it still is.

"The Software is the Instrument"

While others might have seen GPIB as a hardware play, we recognized it for what it enabled in terms of software. As the PC industry evolved (as well as Apple's Mac, which we had a special affinity for given its graphical user interface), that GPIB cable made it easy to analyze and present data in a customized way for our customers' needs. They were no longer confined to the front panel of an instrument and their pencils and notepads for data acquisition. The opportunity to innovate then shifted to the software world, where programming languages needed instrument drivers for the connected boxes. Our strategy of writing and supporting those drivers offered a critical service that continues today as NI supports more than 10,000 drivers on the company's Instrument Driver Network.

But that world still left engineers and scientists with the burden of using tools designed for computer science to perform engineering, test, and measurement tasks. Our answer was twofold: LabWindows™/CVI, to offer engineering-specific tools in ANSI C programming, and LabVIEW, a graphical programming paradigm that took the way we think about solving a problem (in flowcharts and pictures) and turned it into compiled code. The story was simple: acquire, analyze, and present. Do it in software tools designed for a customer's use case that were easy to learn yet extremely powerful. We coined the phrase "The software is the instrument" to describe this approach, and seeing engineers and scientists save valuable time and get to results faster was all the market validation we ever needed.

Evolving With Moore's Law

People talk about Moore's law like it's about hardware, but computational hardware exists only to run software (and maybe firmware). Once we made test and measurement all about software, we had effectively enlisted Intel, Xilinx, and many other billion dollar companies in our R&D staff. With customers and partners building proficiency with our software tools, we just had to follow the chips to deliver increasing value to test and embedded systems. This has happened, so far, along two key dimensions: multicore processors and FPGAs.

Because LabVIEW is graphical and, therefore, not obviously sequential, it is tailor-made for parallel processing. LabVIEW users were among the first programmers to easily migrate from single-core processors to multiple threads and multiple cores and see almost instant speed improvements. Obviously, it's possible to take advantage of these trends with other languages, just like it's still possible to write highly efficient code in machine or assembly language, but why would you? The pace of change in modern electronics means you can't waste time doing by hand what a tool can easily do for you, and we hear that over and over from LabVIEW users.

That goes to an entirely different level with FPGAs. Some problems are just better solved in the highly parallel, deterministic world of silicon. But the toolchains and programming constructs were inaccessible to most mechanical engineers or medical researchers who were experts in their measurements and problems to solve (not digital design). We recognized this in the late 1990s with LabVIEW's graphical paradigm. We were on a quest to deliver the power of FPGAs to LabVIEW programmers, and we've done that. A quick look at our Engineering Impact Awards winners demonstrates the power of this technology: applications ranging from regenerating and restoring organ function damaged by disease or trauma to setting a world record in 5G wireless spectrum efficiency with massive MIMO.

A Software-Centric Approach to Hardware Design

When you think about software as uniquely as we have, it's easy to think differently about hardware, too. Modular, PC-based plug-in boards were a natural by-product. Make the hardware as lightweight and cost-effective as possible (no dedicated screens, power supplies, fixed buttons/knobs, and so on) and focus on ADCs, DACs, signal conditioning, and data movement. I have yet to see a test and measurement vendor design a user interface better than a customer, for any specific task, that makes the customer more productive. Even the best front panels on box instruments are cluttered with unused buttons or menu structures. Many of our hardware products have size constraints dictated by the I/O connector. Can it get more efficient than that?

The reality is our strategy is more than just efficient; it's right. Take the new Vector Signal Transceiver (VST), which combines an RF analyzer, RF generator, parallel and serial digital interfaces, and high-performance signal processing into a 2-slot PXI module. This product delivers industry-leading bandwidth (1 GHz), amazing RF performance, and scalability for MIMO applications for one reason: software. We moved as many technical problems into the FPGA as we could, and Moore's law (along with Xilinx) delivered a vehicle capable of handling the computation. We, in turn, passed the keys to that vehicle over to our customers by allowing them to customize that FPGA with LabVIEW. From 5G cellular technology development to automotive radar and driver assist algorithm development to reductions in the cost of Internet of Things (IoT) devices, the VST and LabVIEW are helping customers achieve goals conventional instruments quite simply prevent them from obtaining.

The Future

We are seeing glimpses of the future everywhere we look. A modern factory features what we call "cyber-physical systems," which combine software-centric computing technology with electromechanical systems and human operators to improve safety, efficiency, and cost structures. The acquire, analyze, and present concept is still valid, but we've added "sense, compute, and connect" as a parallel flow for IoT devices. Wireless technology in general is pervasive. We've been saying this a while, but if you aren't an RF engineer today, you will be. And the more you connect things, the more you'd be crazy not to take advantage of the data you can collect for billions of sensor nodes. For us, this is Big Analog Data™ solutions, and it's the richest set of data in the world. NI customers are acquiring terabytes and terabytes of it every day.

> "All engineers, scientists, and vendors need to embrace new approaches like this to foster the innovation that will ultimately address the grand engineering challenges of our time."
>
> —James Truchard, PhD, President, CEO, and Cofounder, National Instruments

But even as our capabilities become more advanced and the scale of the problems we try to solve grows vaster, the tools we use must be easier to navigate. Just as machine language migrated to assembly and to object-oriented, other paradigms, including graphical dataflow programming, are critical to offer the right level of abstraction. The multirate diagram in our LabVIEW Communications System Design Suite is a great example; no single software tool delivered the productivity needed to prototype 5G algorithms until we were bold enough to tackle multiple models of computation inside a single flow that could deploy directly to hardware.

No great innovation will be done alone. The best platforms we use today are effective because they've fostered an ecosystem. Our software-centric approach at NI spawned a partner network of more than 1,000 companies and 300,000 active LabVIEW users. The rise of mobile devices and "apps" is possible only because of a healthy ecosystem built on developer-friendly platforms. Team-based development, code sharing, and community support soon will no longer be novel or best in class. They will be expected.

In Closing

It would be impossible to have witnessed what I've witnessed in our industry for the past 40 years and not be excited about where all of these technologies and trends are leading us. My advice to any new engineer is simple: develop a vision for the future and pursue it with intensity. And, at the end of the day, don't be afraid to have fun.

Thank you for 40 great years. I believe the following five selections from the Automated Test Outlook archive are as true today as the day they were published initially. May they inform your vision for the future and bring you and your organization prosperity and success.

Table of Contents

Editor's Note: I love seeing test leaders and organizations turn their "necessary cost centers" into strategic assets to improve profitability, time to market, and product quality. Having led, and lived through, multiple organizational inflection points, I can attest that it's both hard work and worth it, personally and professionally. Take advantage of others' insights and experiences via the global communities of test engineering leaders we host throughout the year: the Test Leadership Forum, regional advisory councils, and online LinkedIn group. You'll be amazed at what you can learn from your colleagues, both inside and outside your industry.

Optimizing Test Organizations

In tough economic conditions, companies are more diligently looking for opportunities to gain a competitive advantage while growing revenue, profits, and customer loyalty. This has led to a strong adoption of business improvement strategies such as Six Sigma, Lean Manufacturing, Capability Maturity Model Integration (CMMI), and Agile Product Development. Additionally, companies will elevate and strategically take advantage of a support function within an organization as a marketplace differentiator.

For example, the role of information technology (IT) has changed dramatically over the last two decades. IT was originally a support function that provided standard computing applications, data storage, and routine task automation. In leading organizations, IT can now streamline critical line-of-business processes and help executives make real-time decisions at the core of a company's business. The strategic importance of IT was confirmed by the *Chief Information Officer* (CIO) magazine 2010 State of the CIO Survey, which revealed that 70 percent of CIOs are now members of their companies' executive committees.

Similar to IT, product testing has been historically viewed as a support function during the product development and manufacturing process—just a necessary cost center. Hence, many companies invest at much higher rates in other areas of "strategic" value such as product development and sales enablement. This leaves the test organization fragmented, outmatched to meet business requirements, and outdated with old technologies and test methodologies that often create bottlenecks for their organizations. However, as research has shown, test is critical because it validates a product's performance, reduces development time, increases quality and reliability, and lowers return rates. By catching defects earlier in product development and collecting the data to improve a design or process, test delivers tremendous value to the organization.

An emerging trend for electronics manufacturing companies is using product test for competitive differentiation. This has resulted in elevating the test engineering function from a cost center to a strategic asset. This shift was confirmed by a recent global NI survey of test engineering leaders who said their top goal over the next one to two years is to reorganize their test organization structures for increased efficiency. This strategic realignment reduces the cost of quality and impacts a company's financials by getting better products to market faster. Research has revealed that "optimized" is the ideal maturity level—when a test engineering organization provides a centralized test strategy that spans the product life cycle. This optimized organization develops standardized test architectures with strong reuse components, enables dynamic resource utilization, and provides systematic enterprise data management and analysis that result in company-level business impact.

Companies making this transformation must commit to a long-term strategy because, according to NI

COMMITTING TO A LONG-TERM PHASED APPROACH

Transforming a test organization into a strategic asset requires commitment to a long-term phased approach.

research, it generally takes three to five years to realize the full benefit. A company must have a disciplined and innovative investment strategy to transform the test organization through the four maturity levels: ad-hoc, reactive, proactive, and optimized. Each level includes people, process, and technology elements. The right people are required to develop and maintain the cohesive test strategy. Process improvements are required to streamline test development and reuse throughout product development. And finally, tracking and incorporating the latest technologies are required to improve system performance while lowering cost.

When companies implement changes to process, people, or technology, they are sometimes tempted to bypass transition projects because they believe they can attain a higher level of maturity more quickly. However, before an organization can achieve an optimized level, it must first reach the proactive level in each major competency area: enterprise alignment, business planning, deployment life cycle, system development, and test technologies and architectures.

An organization steadily builds a foundation for strategic transformation by sticking to a sequential approach and identifying short-term initiatives that help the company improve its maturity level and that map to annual operating objectives. And as the foundation gets built, test productivity and asset utilization increase, paying dividends on the original investment. This phased approach enables organizations to realize benefits early on—after the completion of just one or two projects. Examples of these transition projects include the following:

- Standardized Test Architecture/Process (Ad-Hoc->Reactive)—Adopting standardized software and hardware architectures and test methodologies improves productivity with faster test code development and increased test asset utilization.
- Test Total Cost of Ownership (TCO) Financial Model (Reactive->Proactive)—Creating a TCO financial model for test helps companies calculate business productivity metrics and financial metrics (return on investment, payback period, net present value, internal rate of return, and so on) for test improvement initiatives.
- Enterprise Test Data Management (Proactive->Optimized)—Developing a comprehensive test data infrastructure that spans across sites with universal access improves real-time decision making.

This transformation requires a shift from only supporting ongoing operations to developing innovation-based initiatives alongside ongoing operations. The test industry is still early in its transformation. Using the IT industry as an external benchmark, IBM published in its 2010 global technology outlook that highly efficient companies that strategically transformed their IT organizations spend only 60 percent of their IT budgets for ongoing operations, leaving 40 percent for new and innovative initiatives, compared to other organizations with an 85/15 split in their legacy business models. Similarly for test, leading companies gain a competitive edge by keeping their test organizations agile and matching the level of innovation leveraged in other strategic departments.

When test engineering organizations become strategic assets, they create standard test platforms, develop valuable test-based intellectual property, deliver a more productive workforce while lowering operating costs, and align with the business objectives by continually contributing to better product margins, quality, and time to market.

Editor's Note: Three years after this article ran in the 2010 Automated Test Outlook, NI introduced the Vector Signal Transceiver, a PXI module that revolutionized RF instrumentation and created a new class of software-designed instruments that users can reprogram. At first, others in the industry called it "cute" and dismissed the notion that users would want to own the functionality of their instruments at that level. But the VST became the most successful hardware product from NI to date and redefined the future of instrumentation. If your organization isn't considering software-designed instrumentation yet, I strongly recommend it.

Reconfigurable Instrumentation

Software-defined instrumentation, also known as virtual instrumentation, is based on a modular architecture that enables a high degree of reconfigurability. Software-defined instruments consist of modular acquisition/generation hardware whose functionality is characterized through user-defined software running on a host multicore processor. This basic model is ideal for most automated test applications in use today, but new technologies and test methodologies on the horizon are creating the need to push the reconfigurability down to the hardware to achieve required performance. One example of this is testing a modern RF receiver, where coding/decoding, modulation/demodulation, packing/unpacking, and other data-intensive tasks may need to occur inside a clock cycle of the device under test (DUT). In these cases, the software defined architecture needs to be flexible enough to incorporate user-programmable hardware—often a field-programmable gate array (FPGA)—to place the necessary intelligence inside the instrument. User-programmable instruments create an architecture where data can be acted upon in real time on the FPGA and/or processed centrally by the host processor (see figure). FPGAs are a key enabling technology because they combine the best parts of ASICs and processor-based systems. At the highest level, FPGAs are reprogrammable silicon chips. Using prebuilt logic blocks and programmable routing resources, engineers can configure these chips to implement custom hardware functionality. They can develop digital computing tasks in software and compile them down to a configuration file or bit stream that programs the FPGA components. In addition, FPGAs are completely reconfigurable and instantly take on a new personality when recompiled with a different configuration of circuitry.

> "The ability to customize the measurement hardware itself represents yet another milestone in the path toward a completely software-defined test system. In 10 years, we will wonder how we ever programmed test systems effectively without this capability."
>
> —Mike Santori, Business and Technology Fellow, National Instruments

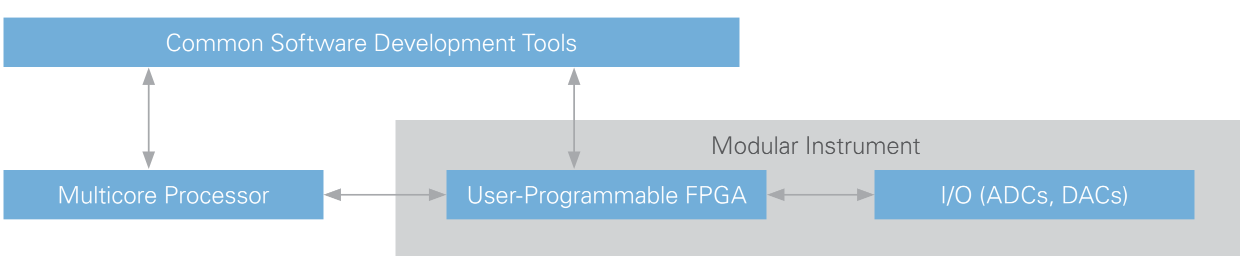
Beyond being user-programmable, FPGAs offer hardware-timed execution speed as well as high determinism and reliability. They are truly parallel so different processing operations do not have to compete for the same resources. Each independent processing task has its own dedicated section of the chip, and each task can function autonomously without any influence from other logic blocks. As a result, adding more processing does not affect the performance of another part of the application.

While FPGAs have been used inside instruments for over a decade, test engineers were seldom given access to embed their own algorithms on them. To be useful in a software-defined instrumentation context, FPGAs must be reprogrammable by the engineer in software; in other words, they should be used to push software programmability down into the hardware itself. In the past, FPGA technology was available only to engineers with a deep understanding of digital hardware design software, such as hardware description languages like Verilog or VHDL, which use low-level syntax to describe hardware behavior. Most test engineers do not have expertise in these tools. However, the rise of high-level design tools is changing the rules of FPGA programming, with new technologies that convert graphical block diagrams or even C code into digital hardware circuitry. These system-level tools that abstract the details of FPGA programming can bridge this gap.

Clearly, there are advantages to performing different types of processing on a host processor versus an FPGA. For example, an FPGA is generally well-suited for inline analysis such as simple decimations on point-to-point I/O, whereas complex modulation might achieve better performance running on a multicore processor due to the large amount of floating-point calculations required. The ideal solution for developing a software-defined test system is a single graphical system design development environment that provides the ability to quickly partition the processing on the host or an FPGA to see which offers superior performance.

This new software-defined architecture can meet application challenges that are impossible to solve with traditional methods such as the previous example that requires real-time decision making by the host to properly test the device. Instead, engineers can fully deploy the intelligence to the FPGA embedded on the instrument for pass/fail guidance. This is often the only way to supply the intense timing and determinism required by the DUT. Examples of this type of device include RFID tags, memory, microcontrollers, and engine control units (ECUs). For some applications, engineers also perform the communication over a protocol—wireless or wired—which requires a significant layer of coding and decoding before making a decision.

Reconfigurable instruments will continue to find more mainstream applications as test engineers continue to look for creative ways to reduce test time and system cost. Take, for example, a digitizer that has an FPGA inline with an analog-to-digital converter. An engineer can deploy functions to the FPGA such as filtering, peak detection, fast Fourier transforms (FFTs), or custom triggering. Not all data is created equal, but an FPGA-based digitizer can make quick decisions on which data is worthless and can be discarded and which data has value. This can ultimately reduce measurement time substantially. Test engineers in the military and aerospace industry have been early adopters of FPGA-based instrumentation through their synthetic instrumentation initiatives, but this technology also has potential for telecommunications, automotive, medical device, and consumer electronics applications.



Reconfigurable instruments provide a Host + FPGA configuration that delivers both performance and flexibility.

Editor's Note: For the past several years, I've used the bagpipe tuners in the iOS App Store to demonstrate the power of vibrant ecosystems. Like Apple, NI has an ecosystem. It's built on our open LabVIEW platform and clearly defined APIs and hardware specifications. With ecosystems, users don't have to start from scratch unless they want to. Critical to the health and productivity of an engineering platform, a vibrant ecosystem delivers an order of magnitude more value faster than any test vendor's R&D department ever could. Understand the ecosystems surrounding your test systems and embrace them.

Software-Centric Ecosystems

The transition under way in mobile devices offers insight into an important trend for test and measurement: the power of a software-centric ecosystem. Early-model mobile telephones were built to make calls first and later send text messages, but the capabilities were almost completely defined by the vendor. Once the software on these devices was opened up to the user, capability ranging from music players to cameras to email quickly followed. But the effectiveness of the transition was more than just an open software experience. Apple, and later Google, built robust ecosystems around their products and created a community of developers for "apps" that accelerated usefulness.

The inherent openness and community concept for mobile phones arguably could have been fostered by mobile phone providers themselves, but in this case it was Apple and Google that worked on software environments first and deployed hardware second. By exposing an appropriate level of customization to users or third-party developers, they succeeded in changing the way consumers view their mobile phones.

This same concept is making an impact on the test and measurement industry. Communities of developers and integrators, building on standard software platforms, are using commercial off-the-shelf technology to extend the functionality of complex hardware into applications previously impossible. The level of productivity and collaboration delivered by software-centric ecosystems will have a profound effect on test system design over the next three to five years.

Ecosystems Defined

In his book *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems,* James F. Moore defines a business ecosystem in the following way: "An economic community supported by a foundation of interacting organizations and individuals—the organisms of the business world. The economic community produces goods and services of value to customers, who are themselves members of the ecosystem. The member organisms also include suppliers, lead producers, competitors, and other stakeholders. Over time, they coevolve their capabilities and roles, and tend to align themselves with the directions set by one or more central companies."

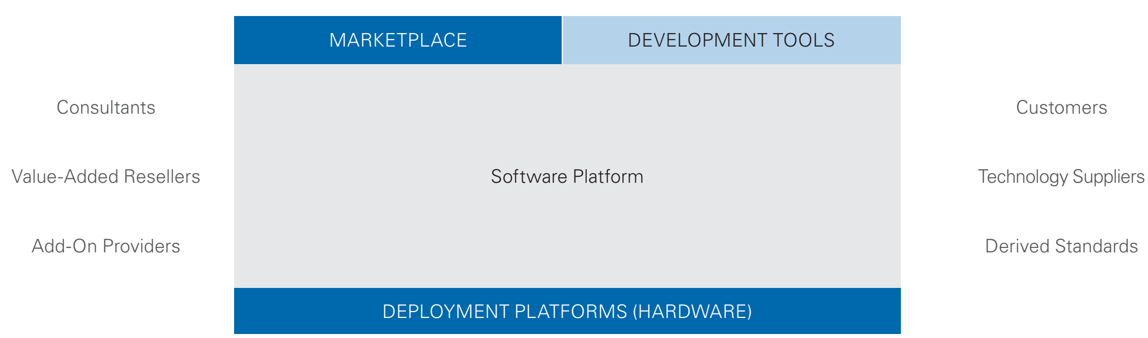
For test and measurement, cross-industry collaboration is nothing new. Active industry groups such as the IVI Foundation, PXI Systems Alliance, and LXI Consortium have been bringing industry players together for decades but often with key gaps as outlined in Moore's description. With active participation in these groups now including software-specific, hardware-specific, and joint hardware/software vendors, the focus on enabling interoperability for proprietary architectures and ease of use for open architectures is fostering business ecosystems. The most successful examples of current ecosystems in this industry, though, are rooted in software. LabVIEW is an example of application software made more valuable through its ecosystem. Significant numbers of engineers have been trained on LabVIEW and developed add-ons suitable for private application needs as well as others through commercial vehicles like the LabVIEW Tools Network. System integrators in the NI Alliance Partner Network as well as LabVIEW Consultants work to deploy this ecosystem. With every additional supplier, producer, competitor, or other stakeholder, the value of the software to each user grows.

Ecosystems in Open and Proprietary Software/Hardware Architectures

An extremely useful ecosystem standardizes the way we communicate with instruments—Interchangeable Virtual Instrument (IVI) drivers. By offering a common means of communicating to similar instruments across multiple vendors at the application programming interface level, the IVI Foundation reduced the learning curve for users and the development cycle for vendors. This opened the door for third parties to create drivers, aggregation websites to house them (like IDNet on ni.com), and abstraction layers to be created on top of them. With well-architected hardware abstraction layers, technology insertion for systems designed to last decades became not only possible but routine. The ecosystem fostered by standardization was crucial in achieving this, and it continues to grow with the recent ratification of native Microsoft .NET implementations for IVI in the past few years. When programming FPGAs in applications like inline signal processing or DUT control, most test engineers practically require hardware and software from a single vendor to achieve the abstraction necessary to meet their skill levels. When these solutions are delivered in the context of a software-centric business ecosystem, the platform can retain as much user flexibility as a disparate or interchangeable hardware/software approach. For example, the FPGA programming capability of the LabVIEW reconfigurable I/O (RIO) architecture can incorporate third-party VHDL or Xilinx CORE Generator IP inside the LabVIEW system design toolchain. The LabVIEW Tools Network helps users exchange sample projects and compiled code to support different application spaces among users and vendors in automated test. This ecosystem opens the doors of FPGA programming to nontraditional automated test spaces and offers the IP necessary to be successful. Without a software-centric ecosystem, many viable open platforms have struggled. The xTCA platforms have seen adoption in telecommunication infrastructure and interest from the high-energy physics community, but they have failed to develop a strong ecosystem in automated test. The multiple form factor, communication bus, and software options presented by the platform have delayed or complicated adoption by leading vendors. While efforts to rein in those options and improve them for automated test are under way in the AXIe Consortium, success or failure will be dictated by the use of a software-centric ecosystem.

The Future of Ecosystems in Automated Test

Over the next three to five years, automated test systems will become more software-centric and ecosystems will have more impact on the value users derive from these platforms. The previous examples of instrument communication and abstracted FPGA programming are just the beginning for automated test ecosystems. As software vendors take greater advantage of their ecosystems and leverage commercialization models for third-party IP, the scenario unfolding for mobile devices will have a transformative effect on the test and measurement industry.



As software platforms develop ecosystems that grow with each additional customer, supplier, add-on provider, and so on, they become more valuable to each user. Software-centric ecosystems will make a large impact on the value that engineers derive from software-based test platforms.

Editor's Note: The media tends to focus on the consumer Internet of Things, but thinking of a test system as an IoT device presents additional opportunities. On the small scale, test organizations can optimize the performance of their test hardware assets. On a larger scale, the insights from managed test systems can improve yield, quality, productivity, uptime, and much more. A great example is how large semiconductor manufacturers use real-time data to optimize their processes, and this trend will only increase as test systems become smarter than the devices they're testing.

Managed Test Systems

As Moore's law continues to influence the performance and complexity of test systems, the need for robust system management capabilities is increasingly apparent. Test managers responsible for maintaining the uptime of a test system are looking for improved management features in their test equipment. Simply defined, manageability comprises the set of features that support the ability to identify and supervise a computing system. Borrowing from a rich heritage established in the information technology (IT) industry, manageability features enhance a test system's ability to perform its primary task (testing and measuring) by ensuring the components of the system are up to date, healthy, and meeting performance expectations.

In the same way that IT administrators rely on manageability features to efficiently maintain client and server computing assets in a corporate environment, test engineers and operators will benefit from manageability features when developing, deploying, and supporting the operation of test systems.

Elements and Operating Modes of Managed Test Systems

Managed test systems are composed of the system infrastructure, peripherals, and hardware and software elements that manage them, including management consoles and APIs. For example, management console software, such as NI Measurement & Automation Explorer (MAX), can run directly on the test system being managed or be executed remotely via a network on a separate computer. In both cases, the management console issues configuration, calibration, platform monitoring, and deployment requests on behalf of the test engineer or operator managing the system, and the managed system fulfills those requests. In addition to vendor-provided management consoles, users can define their own or integrate manageability features directly into test applications using APIs. With these standard elements, manageability features can operate in two distinct modes: in-band and out-of-band.

In-band management uses the primary computing resources, including the system controller's main CPU, network interface, and operating system, to manage the system. In addition to running the test application, the system controller runs software to enable manageability features, including management consoles and supporting infrastructure. In this way, in-band management can support a rich set of manageability features while the system is operating in the "fully on" state. If the system controller is powered off, unprovisioned, or not operating normally because of a failure, out-of-band management is required.

Out-of-band management can be particularly useful for those diagnosing a system that has failed. While rare today, more test equipment is incorporating these features by using dedicated computing resources, including a secondary management processor, network interface, and operating system, to manage the test system independently of the system controller's computing resources. For example, if the system controller is unable to boot normally because it has experienced a hard drive failure, out-of-band management can be used to remotely power the system on and execute diagnostics on the hard drive, allowing for remote analysis to determine the cause of the failure. Further, because out-of-band management does not require the use of the system controller's computing resources, the system controller can remain fully dedicated to executing the application. This is particularly important for applications that are sensitive to disruptions in CPU or data bus usage, including real-time and high-throughput measurements.

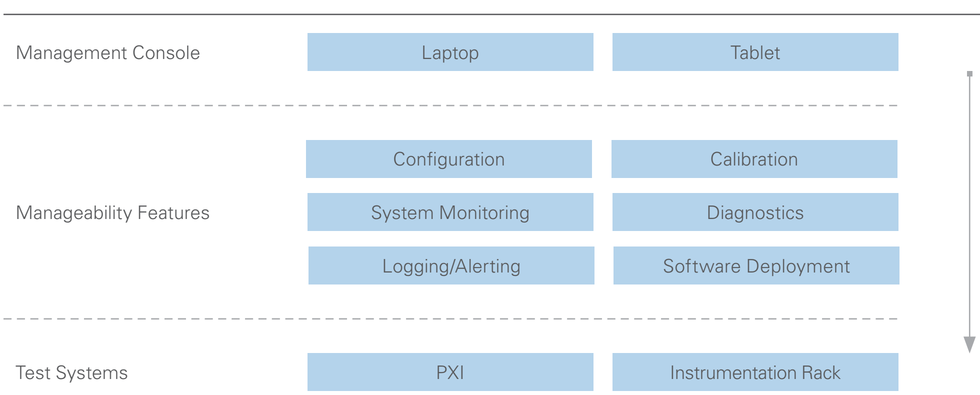
Trends in Managed Test Systems

As modular instrumentation platforms continue to displace traditional box instruments, the need for asset management capabilities is increasingly important. Because modular test systems separate the system into components (system controllers, chassis, and instruments), the number of assets to be managed naturally increases. By knowing which test assets are being used and how they are being applied, test managers can lower costs by maximizing the use of available equipment. In a validation lab, for example, it is critical that the location and operational state of all assets are known so that components not actively being used can be redeployed in other test systems. The same applies to high-volume production test environments but on a much larger scale.

Increasingly complex measurement devices are also driving the need for comprehensive manageability support, particularly in platform monitoring and control. Modern modular instruments, especially RF instruments, offer unprecedented measurement flexibility and speed by taking full advantage of the power and cooling capabilities of the modular platforms that support them. Test system designers can maximize the long-term reliability, usability, and measurement accuracy of these systems by selecting platform elements that use monitoring and control features. For example, by monitoring the cooling requirements of the instruments in a chassis, a chassis can optimize its fan speeds to minimize acoustics. This is especially important in an environment where noise must be minimized such as a validation lab. Further, measurement accuracy is optimized when an instrument is operating as close as possible to its calibrated temperature. By monitoring the temperature of an instrument, a chassis can precisely control its fans so that the instrument can maintain a steady temperature at or near its calibrated value to ensure the integrity and repeatability of the measurement.

Benefits of a Managed Test System

Test managers can significantly benefit from improved manageability features, which lower the test system's integration risks by ensuring that issues can be diagnosed and resolved efficiently, especially for large and complex testers and testers in remote locations. Additional benefits include minimizing a test system's "time to value" by ensuring that initial and subsequent test station deployments can be managed in a fast and repeatable manner. Finally, manageability features lower the total cost of ownership of a test system by enabling the ability to proactively monitor and diagnose problems as well as convert unplanned outages into planned outages. Just as manageability features helped drive the transformation of the IT and telecom industries, they will play an increasing role in test systems in the years to come.



Manageability features enhance a test system's ability to perform its primary task by ensuring the components of the system are up to date, healthy, and meeting performance expectations.

Editor's Note: If you're in the automotive industry, you can't use test capabilities designed looking in the "rearview mirror." As a test equipment manufacturer and a Tesla owner, I've had a front-row seat to the promise and challenge of autonomous vehicles, from the thrill of receiving new features via over-the-air software updates to discussions with automotive engineers on the challenges of meeting safety regulations. If you're affected by technology convergence, you'll find the NI platform and ecosystem uniquely capable of addressing these unsolved problems.

Driven by Necessity

In the aerospace and defense industry, reducing release cycles and preventing program delays have become increasingly difficult. In automotive, consumer demands are driving up test complexity and introducing new costs in areas like infotainment. In response, test managers must find affordable ways to incorporate RF testing for wireless signals and machine vision testing for assisted parking to meet the widening I/O spread of test coverage.

Though industry regulations provide a guide to ensure safety in embedded electronics, compliance with these regulations requires the thorough testing of embedded software across an exhaustive range of real-world scenarios. Developing and testing embedded software with an emphasis on quality can strain the balance of business needs such as short time to market, low test cost, and the ability to meet the technical requirements driven by customer demand for new features and product differentiation. All embedded system manufacturers face similar demands, but they cannot sacrifice quality when it comes to safety-critical applications. Organizations that can evolve their development strategies to incorporate advanced hardware-in-the-loop (HIL) testing can reduce spending on quality-related problems, improve their market perception, and, most importantly, ensure customer safety.

HIL test becomes even more valuable as the need to offload test time in the field or the test cell intensifies with the addition of functionalities to controllers and the increase in test cases.

HIL Test Helps Meet Safety and Business Needs

Complying with safety standards requires an understanding of all potential health risks and hazards as well as the capability to rigorously test those scenarios. HIL testing meets many of these growing test needs at a lower cost and in a shorter time frame than physical tests and field tests. With this method, companies dynamically simulate real-world environments using mathematical models to provide closed-loop feedback to the controller being tested. HIL test becomes even more valuable as the need to offload test time in the field or the test cell intensifies with the addition of functionalities to controllers and the increase in test cases. Hybrid electric vehicle motor controllers are establishing new levels of functionality by managing safe power control between an internal combustion engine and an electric motor. While designing Subaru's first hybrid electric vehicle, the Subaru XV Crosstrek, engineers at Fuji Heavy Industries needed to deliver complete test coverage of their innovative powertrain technology.

Subaru Uses FPGAs for Greater Safety and Reliability

Testing the hybrid motor controller required advanced test tools and new methodologies to provide high-quality software within the engineers' timeline. Subaru chose to use FPGA technology to meet its high-performance needs and verify a wide range of tests. For instance, when the vehicle slipped on ice, the controller had to recognize the loss of traction and provide the appropriate response to the hybrid powertrain. Re-creating these conditions on the proving grounds inconsistently yielded accurate data, and traditional processors for HIL could not accurately simulate the fidelity and speed required of an electric motor model.

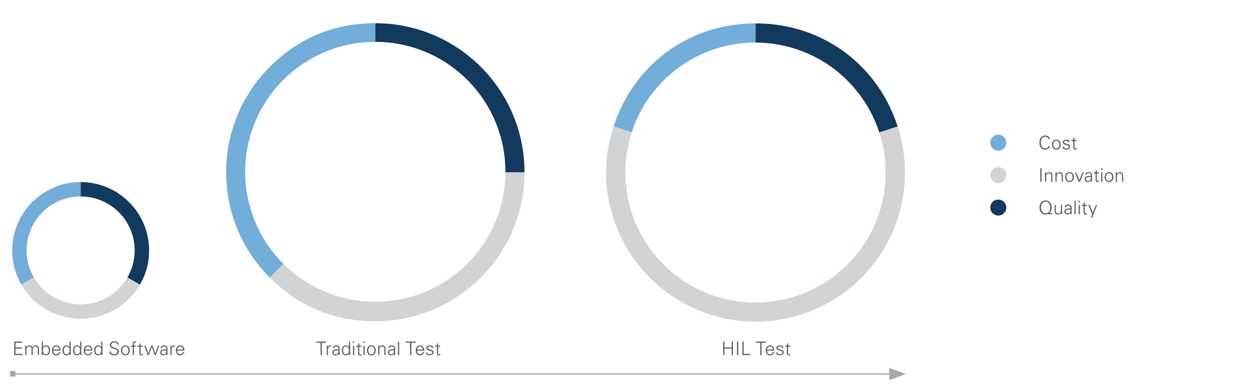
Using open and flexible FPGA modules, which significantly reduced communication time by collocating the processing node and I/O node, Subaru engineers offloaded taxing calculations and performed HIL tests on their system for corner cases such as traction loss on ice to provide greater safety and reliability. With the open architecture, they programmed their system to use a high-fidelity JMAG-RT model and achieve the 1.2 μs simulation rate required to accurately simulate the safety handling of an electric motor. The ability to move more field tests into the lab resulted in a 20X reduction in test time, so the engineers did not have to compromise innovative technology, shorter time to market, and lower test cost to achieve high-quality software. Subaru's HIL testing platform provided cheaper, more comprehensive, and faster testing than physical testing.

Scalable Test Platforms Offer Affordability While Ensuring Safety

Embedded software design and test teams must continue to find new ways to use this practice to ensure quality and make consumer safety a priority without sacrificing release schedules. HIL testing is mostly entrusted to only a specific test team, but developers have also been performing manual stimulus testing known as knob-box testing for quick functionality checks. This restricted form of testing allows them to spoof the controller by manually changing a limited number of channels. However, many functionality defects are still found in the later stages of HIL testing, or even in the field, which cost developers more resolution time. With higher levels of automation and easily repeatable test scenarios, developers can discover more of these functionality defects so that test engineers can focus on identifying performance and integration-based defects. Full-rack HIL test systems are not necessary for this application. Instead organizations must build scalable test platforms to provide an affordable solution across varying capabilities.

As increasing embedded controller capability drives further innovation, safety regulations will be honed to ensure even greater user safety. To keep up with feature demand while preserving the quality of the overall system, test capabilities will need to grow accordingly. Simply adding more test bandwidth will not scale with overhead; test managers need to adopt advanced HIL test technology and new techniques. This ensures that as industry regulations help guide system engineering teams toward higher levels of safety for more advanced products, test platforms can still meet critical cost and time requirements.

NEXT-GENERATION DEVELOPMENT



HIL solutions help drive down test costs without sacrificing the growing quality requirements inherent with new innovations.

National Instruments
US Corporate Headquarters
11500 N Mopac Expwy, Austin, TX 78759-3504
T: 512 683 0100 F: 512 683 9300 info@ni.com

ni.com/global—International Branch Offices
ni.com/ato

100%
From well-managed forests
FSC® C000000
www.fsc.org

351409G-01 27514