Open Source in Test Automation

The Open Source Path to Effortless Automation

This white paper examines the impact of Open Source software on the Test & Measurement (T&M) industry and more broadly, the positive consequence of community-developed software for key T&M end-user segments. It examines how equipment OEMs and automated test developers benefit from the economics and velocity of Open Source software and a broadly participative development model & ecosystem. In particular, the white paper highlights the OpenTAP project, including background, motivation, project goals, technical scope (core and plugins) and the cross-ecosystem benefits of integration and participation.

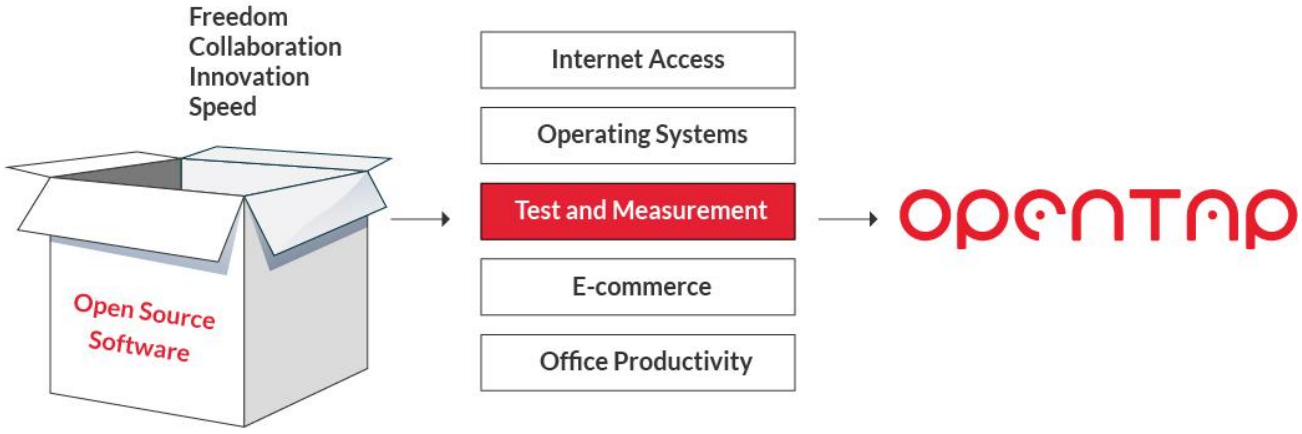


Table of Contents

Introduction

In May 2019, the OpenTAP project was jointly announced and launched by Keysight Technologies, Inc. and Nokia Corporation. In the highly proprietary test and measurement industry, the OpenTAP project is unique and potentially game-changing: it is released as Open Source when other test executives and enabling test software are typically closed and proprietary; by contrast, OpenTAP is designed to engage users and ecosystem partners, even competitors, in collaborative development of a core sequencer engine and plugins vs. creating branded technology silos; OpenTAP seeks to standardize testing and test plan development via open implementation vs. select adherence to paper-only (de jure) standards.

This white paper provides context and rationale for the launch of and investment in the OpenTAP project, describes the architecture of OpenTAP, and posits the re-invention of the test and measurement ecosystem with OpenTAP as a catalyst.

Situation analysis – The Test and Measurement Marketplace

The test and measurement industry constitutes a large and very diverse market, expected to grow from USD 27.7 billion in 2021 to USD 33.3 billion by 2026[1]. Market growth is predicated upon the creation of new devices needing test, with some areas growing at higher rates than the overall market, such as automotive & transportation, aerospace & defense, IT & telecommunications, education & government, semiconductor & electronics, industrial, and healthcare. Given the diversity of these segments and the divergent designs of devices under test, test and measurement is consequently a highly fragmented market, with a very long tail of vendors, sub-segments, and niche applications.

> There is a need to develop a platform-centric business model that effectively integrates hardware, software analytics, and services – Frost and Sullivan[2]

[1] Markets and Markets – Test and Measurement Equipment Market

[2] Frost and Sullivan – Asset Management and Integrated Service Delivery: Expansion of Traditional Business Models in the T&M Market

More than 150 vendor companies participate in this diverse market, with leaders boasting derived revenues ranging from one to three billion dollars, across multiple product lines, often developed and sold through multiple divisions. Some of those companies and their customers invest heavily in innovation and seek new ways to add value, even in the face of macro-economic challenges. Others are in retreat, resting upon past laurels, legacy brand equity and cash-cow revenues.

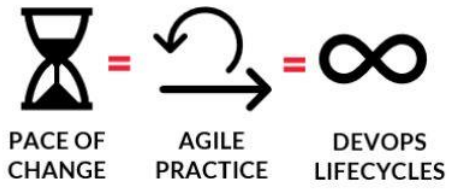
The complexity of test and measurement customer devices and their interfaces have steadily increased over the last 25 years and today automation is essential to rapid and comprehensive testing. As a result, investment in both commercial and in-house test development has gravitated toward software and interface development, software development tools, and packages for data management and analysis (test reporting). Users of test equipment across the many target segments utilize tools and equipment from diverse sources to test products with increasingly complex functionality.

Unfortunately, testing capability has been hard pressed to keep up with product complexity, placing ever-increasing stress on device manufacturers. This gap can be attributed to the hardware orientation of the industry versus the potential to close the gap with advances in software of all kinds, including Open Source.

Test and Measurement today still primarily consumes Open Source rather than collaborating in its development.

Until recently, the T&M segment made relatively light use of Open Source software and development methods, especially when compared to other intelligent device designs and to enterprise software. And like other segments only beginning to embrace Open Source, T&M today primarily consumes Open Source code rather than collaborating in its development.

Influences and Key Trends



Among test and measurement customers, the areas of highest growth include Automotive, Communications (5G and IoT) and Medical. These application areas increasingly need to deliver larger numbers of lower cost devices with shorter product life-cycles. This dynamic creates an imperative to reduce the time and cost of test development and implementation. This trend also augments relative levels of investment from test equipment toward development of test software. According to Frost & Sullivan "It is not test equipment vs. test equipment anymore. No longer can test equipment focus on hardware alone. Instead, there is a potential focus on platform and ecosystem. There is a need to develop a platform-centric business model that effectively integrates hardware, software analytics, and services"[3].

Solutions to today's most pressing challenges in Test and Measurement lie in the creation of a broad ecosystem of interoperable tools, interfaces, and services. It therefore makes sense to emulate the approach that has facilitated the rapid emergence of many of the ecosystems in software technology today: leveraging Open Source software. Prominent examples of such facilitation include the buildout of the Internet and cloud computing, SDN and ICT orchestration, and, of course, Linux.

The Rise of Open Source

Background and Origin

Open Source software is frequently characterized as new phenomenon. In actuality its origins date back to the dawn of the IT industry. Beginning in the 1950s, computer manufacturers – IBM, Burroughs, UNIVAC, NCR, Control Data et al. – supplied hardware to customers together with operating systems and many applications in human-readable source form, to facilitate customization, maintenance, patching, etc. Only in the 1980s did it become standard practice for systems vendors and ISVs to supply software exclusively as precompiled binaries, opaque and inaccessible to developers and end-users.

[3] Frost and Sullivan

Also, during the 1980s, the Free Software movement emerged in response to developers' and academics' desire for the freedom to understand how software systems function, and to be able to repair and modify code for particular needs and circumstances (and not necessarily the "free beer" aspect of nominal acquisition cost). In the late 1990s, the term Open Source was coined, with the goal of emphasizing openness for business and to distance community-based development and distribution from Free Software, which had become a social ideology.

Today the software and hardware industries are dominated by Open Source software – the leading project site, GitHub, hosts over 200 million repositories and 65 million developers

Today, the computer software and hardware industries are dominated by Open Source software – the leading Open Source project hosting site, GitHub, boasts over 200 million project repositories and 65 million developers, with millions more hosted on GitLab and other services. Technology companies, from semiconductor suppliers to equipment manufacturers (OEMs, TEMs and NEPs), software vendors (ISVs) and service providers (carriers and ISPs) depend upon Open Source software to develop and deploy their wares and host their services. Increasingly, enterprise organizations and even mid-size businesses (SMB) also integrate and deploy Open Source software to support operations and serve their customers.

Some of the most popular and successful Open Source software includes Berkeley UNIX (underlying iOS and MacOS), Android (powering 72% of mobile devices worldwide[4]), the Apache and NGNX web servers, Cloud infrastructure (Cloud Foundry, CNCF (including Kubernetes and OpenStack) et al.), Eclipse-based development tools, and development languages C/C++ (GNU GCC project), Go, PHP, Python, Ruby, Swift and many others.

By far the most popular and successful Open Source project is Linux, the operating system kernel that powers over half of all intelligent devices (embedded systems and IoT), provides a kernel to Android and hosts almost all of the Cloud.

[4] Statcounter – Mobile Operating System Market Share Worldwide

Why Open Source?

Developers, systems vendors, and device manufacturers embrace Open Source for a range of motives:

- **Velocity –** Open Source projects evolve very rapidly due to collaboration and shared needs. Beyond the code itself, Open Source also facilitates development of ecosystems required to solve the large scale needs of information technology.
- **Flexibility –** Open Source software can be adapted easily to a range of domains and applications, frees users from the risks and limitations imposed by a single vendor, and be supported by community and third parties for much longer than typical commercial/closed product support.
- **Frictionless Distribution –** most Open Source licenses ensure the availability and unfettered circulation of program source code, facilitating evaluation, supporting proofs of concept, fostering community and in some cases, achieving the status of a de facto standard.
- **Cost –** acquiring Open Source software is usually free of charge and most often offers users lower total cost over the course of its deployment lifetime.
- **Peace of Mind –** many companies chose Open Source to avoid vendor lock-in and reduce long-term risk by relying on a community base of trained engineers. Adopting organizations always have the option to invest directly in the technology, to maintain and enhance it within their company (as a private fork) should that become necessary or advantageous, avoiding dependencies on outside resources.
- **Monetization Options –** while Open Source software can circulate freely, there are myriad methods to build businesses on and around Open Source software, including support, training, integration, productization, provision of services, and more.

Industry Segments Embracing Open Source

Device Software

Almost every type of business today uses and/or deploys Open Source software technology. Among the following industry segments in electronics test & measurement, there are numerous applications areas for Open Source software:

- **Aerospace and Defense –** radio test, low-volume-high-mix manufacturing automation, command and control, in-cabin entertainment, first-responder communications, signaling systems, airborne and ground-based drone control

- **Automotive –** IVI (in-vehicle infotainment), engine control, test/manufacturing, telemetry systems, ADAS and autonomous passenger vehicles and trucks, mass transportation, smart cities (traffic control and parking)

- **Energy –** fossil-fuel and renewable energy generation systems, microgrid control, wide-area grid monitoring

- **General Purpose Electronics –** consumer electronics and IoT: entertainment, gaming systems, home and industrial automation, medical equipment, test and instrumentation systems

- **Internet & Infrastructure –** core and edge routers, 5G/4G/3G base-station testing, access equipment, network gateways, web and streaming servers, NAS and storage systems, and other network appliances

- **Semiconductor Manufacturing –** silicon bring-up, wafer processing, foundry automation, chipset testing, design simulation and evaluation

- **Wireless Devices & Operators –** wireless handsets and tablets, baseband controllers/base-stations, network infrastructure

Enterprise Software

While Open Source software has become in device software, it took a bit longer for enterprise software stacks and compute loads to catch up. Today Open Source usage dominates enterprise data center and cloud deployments, with Open Source comprising 72% or more[5] of enterprise codebases for infrastructure and applications across financial services, energy, healthcare, e-commerce, and other segments, and in key enterprise technology areas like AI/ML, big data and cybersecurity.

Enterprise organizations not only consume Open Source software, for both internal and customer-facing applications, companies outside of tech have become major contributors to the projects behind the software used to run modern businesses, with Open Source playing a key role in DevOps and DevSecOps.

Licensing and Compliance

Among the many innovations associated with Open Source software, licensing is often overshadowed by feats of engineering and community organization. Open source observers and even practitioners are sometimes overwhelmed by the sheer number of licenses – those recognized by the Open Source Initiative (OSI) include popular and widely used licenses (Apache, MIT, Eclipse, GNU and Mozilla) as well as special-purpose licenses and other types, with a grand total of over one hundred named licenses and variants. Licensing innovation includes the notion of a "bare license" (vs. traditional contract-type agreements), the Copyleft (reciprocal disclosure), and leveraging of copyright law as the primary enforcement mechanism.

Open Source licensing is arguably a double-edged sword. Community voices attribute the ubiquity of Open Source software to the licenses that govern its distribution, modification and use: protections for code authors' intentions and intellectual property help motivate developers to create

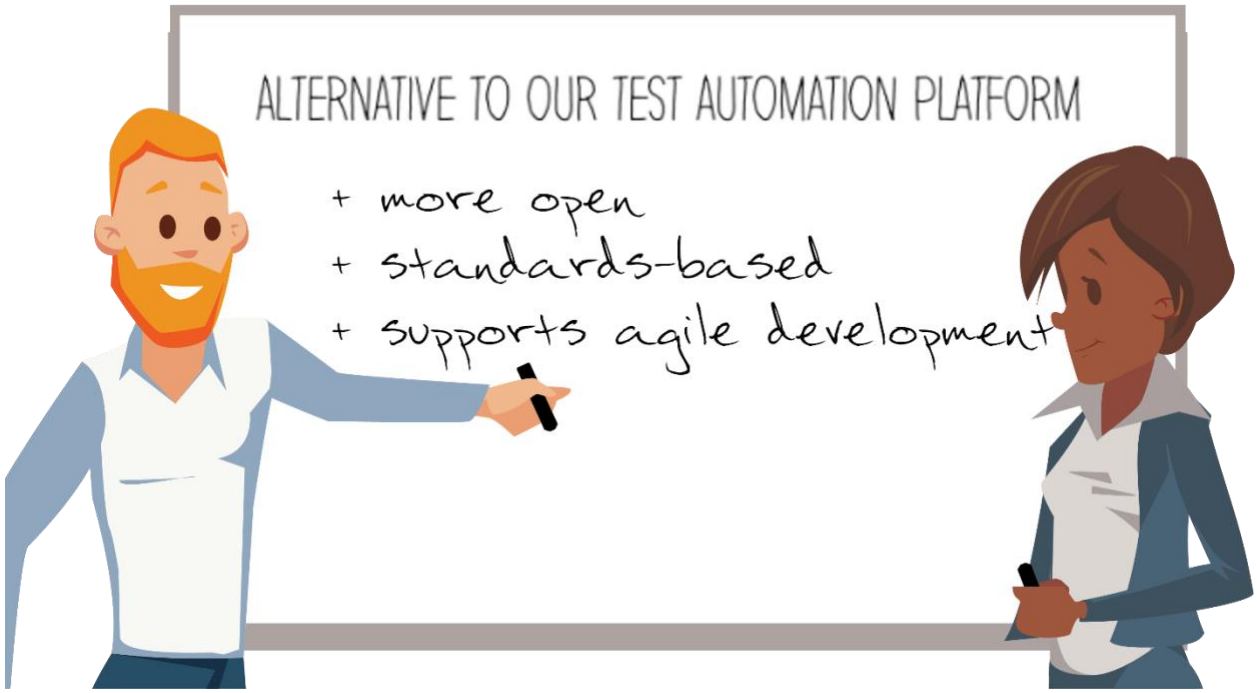
[5] Synopsys – 2022 Open Source Security and Risk Analysis Report

and contribute; requirements for disclosure help keep code flowing freely and inspire ongoing innovation.

Compliance requirements of more restrictive Open Source licenses, especially stipulations around disclosure and reciprocity, have at times challenged adoption, especially from legally conservative corporations and legacy proprietary interests. Until about 2015, many organizations found it simpler to ban Open Source from products and services than to accommodate the associated license requirements. Today, driven by the broad value proposition of Open Source, and availability of quality, liberally-licensed software components[6], the vast majority of companies have made their peace with Open Source licenses.

Open Source Program Office

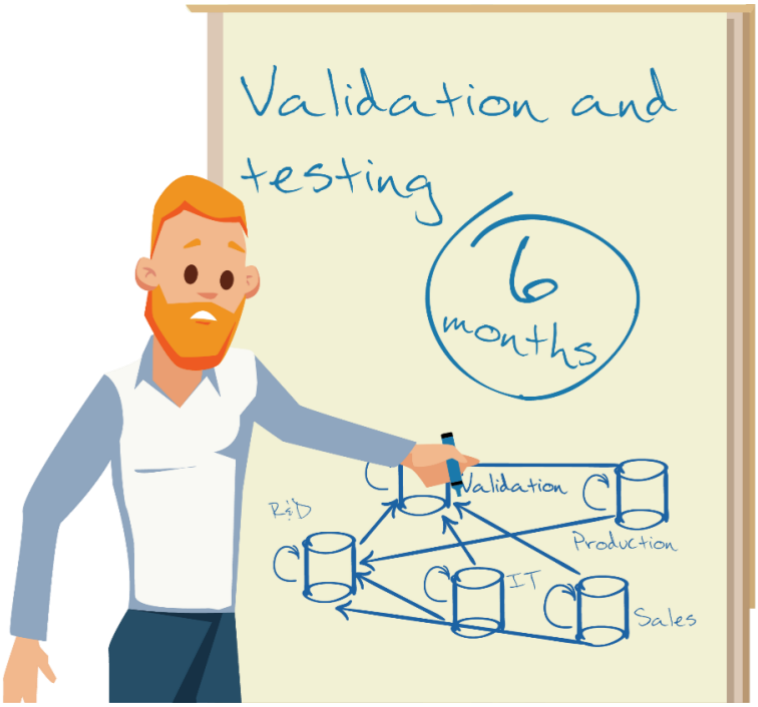
Organizations using, integrating, and deploying Open Source software typically traverse a path that starts with consumption and can evolve to deploying Open Source in their products and services, contributing to Open Source projects, and even becoming leaders of Open Source project communities.



[6] E.g., under APLv2, BSD, MIT and other similar licenses

Many companies start by establishing an ad hoc review board, focused on vetting Open Source licenses and use cases, comprised of representatives from engineering, legal, product management and stakeholders. As usage and integration increases, companies often assign an Open Source program officer to lead management and adoption efforts. Larger organizations will establish a permanent program office, with dedicated resources to coordinate technology selection and approval, implement Open Source policies and processes, mitigate IP and security risks, help choose and deploy Open Source management tools, promote adoption across the company, coordinate inner source[7] programs and enhance returns from investments in Open Source.

Open Source in Test and Measurement



The Test and Measurement industry has historically centered around sales of test instruments – highly proprietary products consisting of specialized hardware running custom embedded software. With the development of VISA (Virtual Instrumentation Software Architecture) in the 1990s, the coupling of test instruments with various kinds of test automation software took off. VISA interface software itself is owned and licensed by the IVI Foundation, but most implementations have been developed and sold under proprietary licenses by instrument vendors.

The availability of VISA interfaces increased the usefulness of test development tools, in theory enabling interoperation across instruments multiple vendors. In practice, however, nearly all VISA-branded tools were proprietary, tightly linked to product portfolios. As such, VISA did not catalyze

[7] Inner Source – employing Open Source collaborative development techniques within and across a (closed) organization

the development of an open multi-vendor test ecosystem. Instead, as confirmed by a recent survey[8], 91% of companies still use software tools built in-house for testing and verification.

The Impact of Open Source on Test and Measurement

To date, the impact of Open Source software on test and measurement has been minor. Its use has been limited to interfaces to specific proprietary products or software elements deployed to Open Source environments, especially to Linux. As a result, Open Source has not significantly facilitated integration, interoperability, reuse, or establishment of an ecosystem of development tools and supporting applications, as has been the case in more software-centric industry segments.

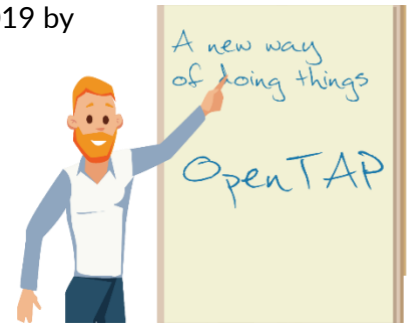
> The influence of Open Source has been limited in an industry dominated by vendors who depend on protecting the advantages of their unique hardware and embedded software investments.

It is not surprising that the influence of Open Source has been limited in an industry dominated by instrument vendors whose businesses depend on protecting the advantages of their unique hardware and embedded software investments. However, as the balance of user investment moves towards test software development, integration and analysis, the proven ability of Open Source software to facilitate innovation and interoperability will come to the foreground.

[8] Dimensional Research for Keysight

Introducing OpenTAP – Open Source for Test Automation

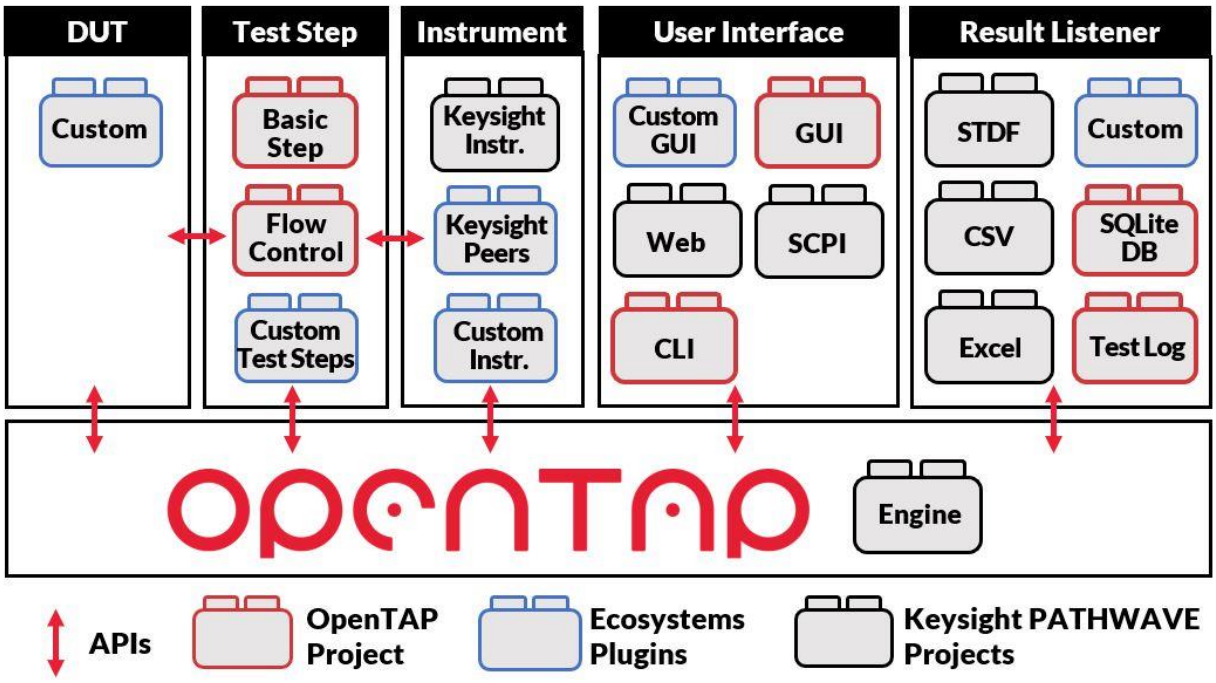
The OpenTAP Project



OpenTAP is an Open Source initiative launched in the first half of 2019 by Keysight and Nokia. The goal of OpenTAP is to transform test automation from a largely home-grown endeavor to a standardized, reusable activity that speeds development and adds innovative capabilities. OpenTAP leverages Open Source collaboration as a proven strategy to create broadly interoperable and reusable technology available to all equipment and software tools vendors, integrators and users in test and measurement.

Project Origins – the Keysight TAP Engine

TAP (as it was originally called) was conceived by a team in Keysight Labs as a generalized architecture for delivering test automation solutions using Keysight instruments. OpenTAP derives from these internal efforts at Keysight.



OpenTAP System Architecture

OpenTAP provides an Open Source, full-function test sequencer distributed under the Mozilla Public License 2.0 (MPLv2). OpenTAP consists of a sequencer (the OpenTAP Engine), well-defined APIs, a plug-in facility, and a set of reusable plug-in modules integrated together in a simple, scalable architecture.

As interest in the project grew, the effort was structured inside Keysight as an inner-source project (internal Open Source within a company). Multiple product groups within Keysight collaborated, tested and added plugin capabilities to the project. As a result, the OpenTAP architecture and software has been employed in over fifty Keysight products and solutions such as:

- Cyber Security: SA8710A Automotive Cybersecurity Penetration Test Platform
- 5G Base Station Conformance: P7000A Base Station Measurement Automation Solution Software
- 5G Device Benchmarking: S8710A Device Benchmarking Toolset
- Massively Parallel Board Test: i7090 Massively Parallel Board Test System

In these and other application areas, OpenTAP technology and its usefulness for creating new solutions have been amply proven.

TAP grew beyond inner source when Nokia requested access to the platform to build its own custom plugins and interfaces to its existing test user interfaces. The TAP team in Keysight Labs collaborated closely with Nokia to realize the same development acceleration demonstrated repeatedly within Keysight.

Nokia, already accustomed to working with Open Source software from its mobile and infrastructure businesses (as is the whole networking and telecom industry), began to advocate for an Open Source release of the TAP platform.

> Open source provides multiple opportunities to interact with test automation users.

Launching an Open Source software project was an unfamiliar undertaking for Keysight. It was not clear how giving away a valuable platform technology would benefit business. To better understand the opportunity, the company sought expertise from other industry segments already using Open Source strategies successfully.

Keysight believes that growing the entire T&M industry "pie" by addressing the challenges of test software development with Open Source offered results as good or better than working only to increase the Keysight "slice." Keysight recognized that Open Source provides the company with multiple opportunities to interact with test automation users along a "ladder of engagement", helping existing users of "home grown" system benefit from a standards-based platform, with the opportunity to provide advanced commercial solutions and services to meet more advanced requirements.

With this new style of customer engagement in mind, Keysight decided to package and release OpenTAP as an Open Source project.

Who Needs OpenTAP?

OpenTAP serves several constituencies with use cases across the test and measurement ecosystem. OpenTAP is of primary interest for users looking to migrate from legacy/homegrown systems; it provides an equally apt option for use in commercial test systems, as well as in green fields applications – entirely new test automation projects.

Legacy / Homegrown Systems – the majority use-case, comprising approximately 85% of the test and measurement marketplace[9]. OpenTAP will draw interest from organizations that

- rely on ad hoc test executives which face challenges to interoperability and maintainability
- understand that building and maintaining their own test executive is not a value-added endeavor
- seek the freedom to innovate in product and service design without getting bogged down in the nuts and bolts of building a test management engine
- want to take advantage of an ecosystem of ready-to-use plug-in modules to speed development
- want to streamline and standardize existing practices

Commercial Test Systems– users and integrators of legacy commercial test executives, where

- vendor and technology lock-in adds cost and limits options from siloed commercial products
- shrink-wrap testing software does not accompany a rapidly evolving marketplace

[9] Electronics World – Can homegrown test software keep up with modern projects?

Greenfield Applications – start-ups new projects at existing companies facing

- budget constraints on all software build and test expenditures
- time-to-market / time-to-revenue pressures

> OpenTAP is a proven platform for quickly building test applications through a standardized architecture, tools, and reusable code elements.

OpenTAP and the Test and Measurement Ecosystem

OpenTAP is a proven platform for quickly building test applications through a standardized architecture, tools, and reusable code elements. Except for the core TAP engine and utilities, the OpenTAP architecture is built out as plugins, which may be licensed in any way their authors choose. An open plugin approach is ideal for building a rich ecosystem that supports collaboration through shared code, but also supports incorporation and integration of commercially licensed technologies.

As of Q1 2022, the OpenTAP ecosystem includes the following hardware manufacturers, software vendors, system integrators, research labs, universities and other organizations:

To learn more about the OpenTAP partner ecosystem and to meet its latest participants, visit https://news.opentap.io/opentap-community-ecosystem-members/.

Plug-ins – Types and Applications

The easiest way to understand the OpenTAP ecosystem is by exploring the various types of plugins and the technical and business opportunities they afford.

Type	Supplier/Contributor	Commentary
DUT Plugins	OEMs, Integrators	Devices under test represent a broad and varied set of use cases, with each DUT potentially requiring unique tests and test "harnesses" for different device types, configurations, connectivity, etc.
Test Step Plugins	T&M vendors, OEMs, Integrators	The OpenTAP project includes basic test step plugins and flow control and enables sharing of new ones from ecosystem participants and within development groups.
Instrument Plugins	T&M vendors, Integrators, OEMs	With OpenTAP as an emerging open standard, instrumentation vendors can foster greater interoperability by offering plug-ins for new and existing equipment in their product lines, as Keysight is doing to support the OpenTAP launch. Integrators and OEM TestOps staff can also enable legacy and custom instruments and test benches.
User Interface Plugins	T&M vendors, ISVs, OEMs	CLI and simple GUI plugins accompany the core OpenTAP project; Keysight offers a professional-quality GUI for creating test plans; other open and proprietary UI plugs are in progress from ecosystem participants
Result Listener Plugins	T&M Vendors, ISVs	Test results are marshalled by the OpenTAP kernel and filtered / presented by plugins into familiar document formats, (CSV, Excel spreadsheet, etc.), input into SQL and other types of data bases, or consumed as raw data, based upon plugins from Keysight and other T&M suppliers and software vendors.

OpenTAP Project Licensing

The OpenTAP code itself is released under the Mozilla Public License version 2.0, chosen because it imposes no significant impediments to test and measurement users, while still encourages contribution of modifications to the core “upstream”, back to the OpenTAP community.

Importantly for many companies, this license and the OpenTAP architecture, allow development, use and commercialization of proprietary plugins, without loss of control over intellectual property.

OpenTAP Project Coordinates

Project Site: OpenTAP.io

Packages: packages.opentap.io

Gitlab Repository: github.com/opentap/opentap

Documentation: doc.opentap.io

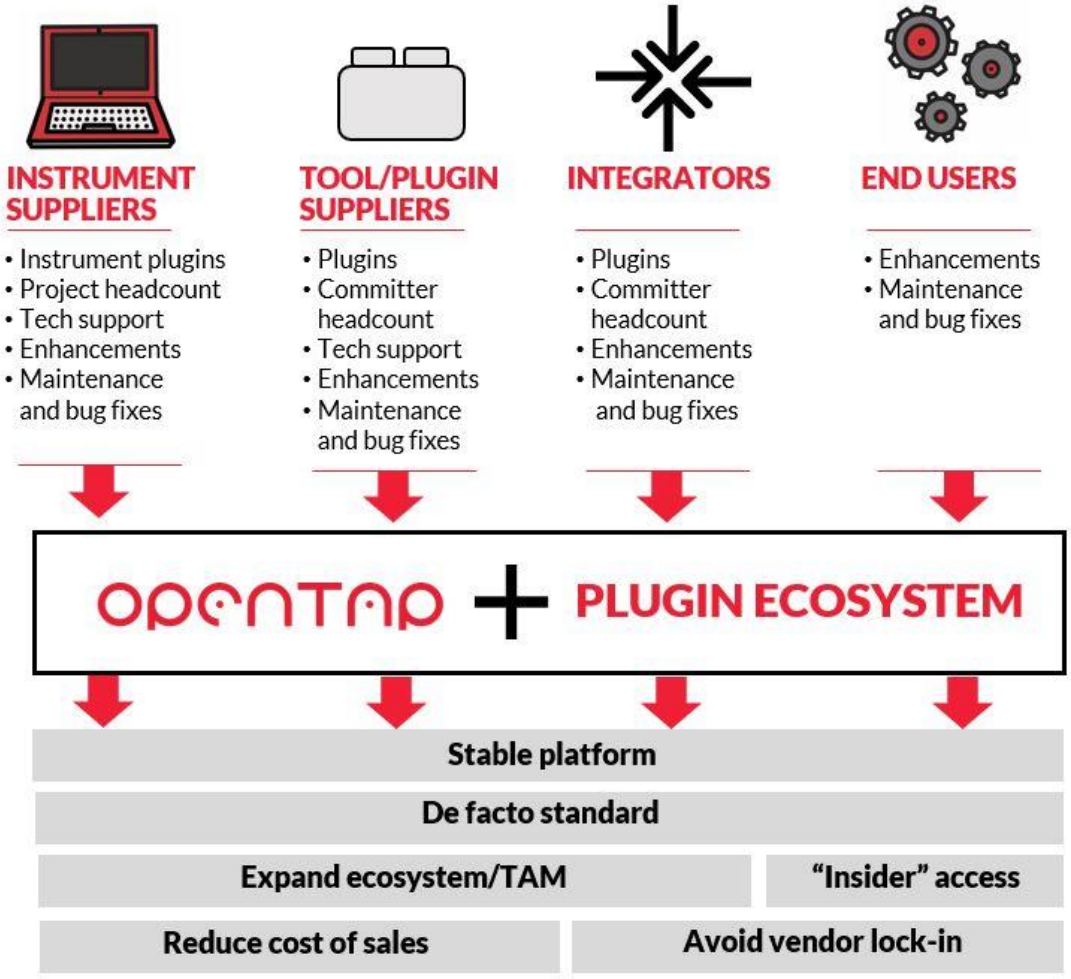
Community Forum: forum.opentap.io

Package Repository

The OpenTAP project features an exchange for publishing, collaborating on and promoting plugins.

The exchange provides instant access to

- Open source plugins, including those that are part of the core OpenTAP project and community contributions
- Commercial plugins (open and closed) from Keysight and other third-parties



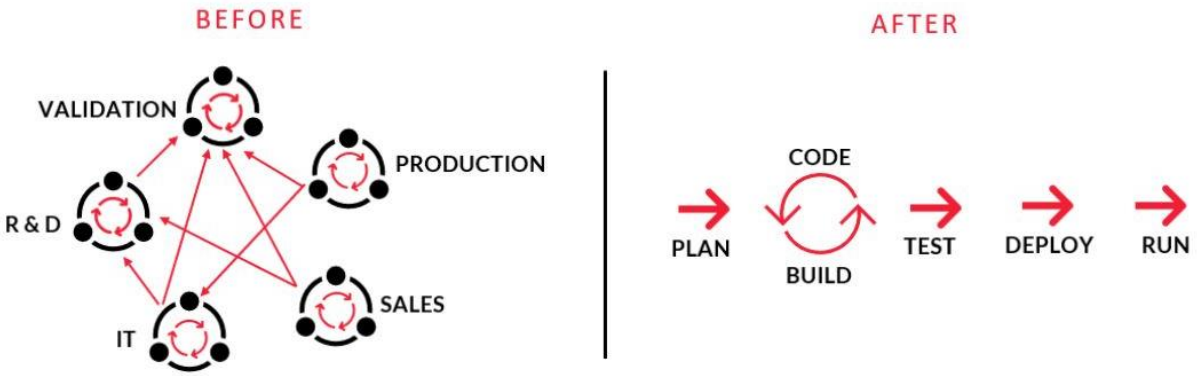
The OpenTAP Vision for Test and Measurement

Keysight and the rest of the OpenTAP community believe that the OpenTAP platform offers the test and measurement industry an efficient way to address the multiplying challenges of test software development. OpenTAP provides an Open Source platform that can grow and evolve to meet the incredibly diverse needs of test and measurement users and suppliers.

Key OpenTAP attributes include:

- a vendor-neutral, standard architecture derived from proven, widely-deployed technology
- an architecture and code framework that encourages good coding practices, e.g., separation of functionalities, clean abstraction, and well-defined interfaces
- support for reusable plugins to handle functions specific to DUTs, test planning, instruments, and logging/display/export
- OEM-friendly MPLv2 licensing of the core project with plugins licensed under either Open Source or commercial licenses

A clear separation of functions in a standard architecture allows all types of test and measurement users and ecosystem participants to create and integrate new test applications faster through software reuse and a broad array of off-the-shelf plugin elements. The project founders believe that only an Open Source approach can adequately address the needs of the 85%+ majority of companies that still rely on in-house test development tools and software.



Benefits to Industry and Ecosystem

OpenTAP confers a range of technical and commercial benefits upon users and suppliers across the test and measurement ecosystem:

- Helps tame complexity inherent in test and measurement from need to address the universe of DUTs with a long tail of commercial and RYO solutions
- Enables multi-vendor test plans and sharing/re-use of test plans with and across organizations (community)
- Offers multiple paths for migration from legacy test infrastructure to new projects
- Reduces or eliminates external lock-in to incumbent suppliers and internal lock-in to legacy homegrown test beds
- Facilitates support for new test equipment and DUT-specific test harnesses
- Speeds time-to-production by streamlining test planning and execution
- Commoditizes low-level test executive functionality with shared technology

Establishing an Industry Test Automation Standard

In multiple technology domains, Open Source has been highly successful in establishing de facto standards based on ubiquitous adoption of a shared implementation (vs. de jure or paper standards which can take years to create and approve and where multiple vendors build incompatible implementations). Examples of successful de facto standardization include Mozilla (originally Netscape) and HTML; the Apache web server and HTTP; Open Source compilers and runtimes for C, C++, Java, PHP and Python; and myriad other technologies spanning the gamut of industries.

By providing OpenTAP as Open Source and building community bridges across the test and measurement ecosystem, OpenTAP aims to provide a comparable standard that will foster interoperability and better support organizations building and testing hardware and software.

Getting Started with OpenTAP

OpenTAP is built with simplicity, scalability, and speed in mind, and is based on an extensible architecture. Its design and packaging support quick adoption and minimal time to achieve useful results. The documentation, code and additional packages are all available at opentap.io.

The OpenTAP SDK includes installation packages for Windows and Linux. In addition to the pure Open Source SDK, a Developer System, including a graphical user interface and results viewing

capability, is also available from Keysight call PathWave Test Automation. Other development tools include Visual Studio integration, Python support, result viewer and a timing analyzer.

The greatest power of OpenTAP lies in its extensible plugin architecture. All of the essential plugins to get started, including a Demonstration package, are provided on the project website. Other advanced plugins, such as REST-API, HTML5, spreadsheet and SQL database are also available, and the list is continually growing.

Most importantly, the installations, developer tools and plugins are all well documented, enabling you can get results fast. And if you need help, both community and commercial support are available.

Conclusion

The test and measurement industry has been, and for the most part still is defined by proprietary solutions. Device software developers needing test executives, test plan creation, instrument interfacing, test execution, results delivery/analytics and other functions historically turned to a long tail of commercial vendors. Often, OEMs launched skunkworks projects as homegrown test infrastructure (and subsequently had to maintain it). Test and measurement is not unique in this bifurcated scenario: the multi-billion dollar embedded systems industry faced the same structure and accompanying software platform challenges until the arrival of Linux and other Open Source platform software at the beginning of the last decade. And just like embedded systems, test and measurement is ripe for the adoption of Open Source solutions.

Glossary

The following acronyms and other terms are used throughout this white paper:

ADAS – Advanced Driver Assistance System

APL – Apache Public License

AI – Artificial Intelligence

AMX – Automated Measurement Expert

API – Application Programming Interface

BSD – Berkeley Software Distribution (license)

CLI – Command Line Interface

CNCF – Cloud Native Computing Foundation

CSV – Comma Separated Values

DevOps – practices combining software development (Dev) and IT operations (Ops)

DevSecOps – DevOps + Security (Sec)

DUT – Device Under Test

FOSS – Free and Open Source Software

GNU – Gnu's Not UNIX (family of licenses)

GPIB – General Purpose Instrument Bus

GUI – Graphical User Interface

HTML – Hypertext Markup Language

HTTP – Hypertext Transport Protocol

ICT – Information and Communications Technology

IoT – Internet of Things

IP – Intellectual Property

ISP – Internet Service Provider

ISV – Independent Software Vendor

IVI – In-Vehicle Infotainment

MIT – Massachusetts Institute of Technology (license)

ML – Machine Learning

MPL – Mozilla Public License

NAS – Network Attached Storage

NEP – Network Equipment Provider

OEM – Original Equipment Manufacturer

OSI – Open Source Initiative

OSS – Open Source Software

REST – Representational State Transfer

RYO – Roll-Your-Own

SDK – Software Development Kit

SDN – Software Defined Network(ing)

SMB – Small or Medium-sized Business

SPICE – Simulator Program with Integrated Circuit Emphasis

SQL – Sequential Query Language

STDF – Shared Test Data Format

TAP – Test Automation Platform

TEM – Telecommunications Equipment Manufacturer

TestOps – Connected, agile design and test

TSC – Technical Steering Committee

UI – User Interface

VISA – Virtual Instrumentation Software Architecture

Revision History

Date	Version	Description
18may2022	5	Minor errata, edits to section titles, styles
17may2022	4	Errata and updates from g-galante
28mar2022	3	Updated title, changed project path from GitLab to Github, added Revision History
07jul2021	2	Integrated graphics from LightSource, published to OpenTAP.io