WHITE PAPER | JUNE 2016

How Will the Internet of Things Affect Testers?

In a series of articles, Paul Gerrard, a testing guru and consultant, discusses a range of testing topics. In this article, Paul talks how the Internet of Things (IoT) which he says is "a whole new ball game" for testers and testing. Paul introduces an architecture which consists of seven layers and discusses the scope of the IoT and the range of issues that it creates for testing.

Paul Gerrard
Gerrard Consulting

Sponsored by ca technologies

Introduction

In this article series, I want to explore how the IoT—also known as the Internet of Everything (IoE)—will affect testing and testers.

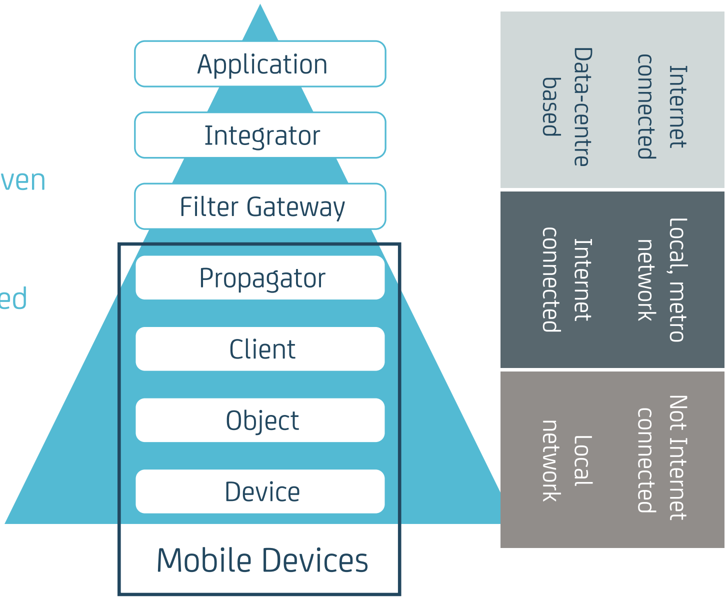
It seems like everything will become "smart" eventually. In parallel with domestic applications, industries such as retail, manufacturing, transport, agriculture and telecommunications are embracing the IoT with enthusiasm. Needless to say, government and the military are pressing on with research and, of course, the emerging phenomena—smart cities—will watch over or control our daily lives.

This article references a series of articles I have written [1, 2, 3] and draws some conclusions about the impact on testing and testers.

A Layered Model of the IoE

In the same way that standards for networking devices at the periphery are evolving, there is no standard or conventional architecture for the IoE. However, some patterns are emerging and, as might be expected, the architecture can be split into logical layers.

Seven-Layer Architecture



Not every system has seven separate layers.

Layers are often combined into one object/system.

This seven-layer model is an amalgam of several styles of schematic that have been published. This one is device and product neutral and it is not aligned with any particular technology. In terms of scale and perspective, these layers look best drawn as a hierarchy with the device layer at the base. The seven layers are described in [2]. Regardless of implementation, the layered model should help you understand the architecture of your system.

The Risks of Failure

The seven-layer architecture might help you to understand the function of each component in an IoE implementation. I've used the layered model to create a list of what I would call risk patterns [2].

But there are also societal or personal risks that are being aired in the media and that we need to pay some attention to. These may or may not exist for your application—but if they do, they're likely to be unique to your project. Here are the main contenders:

- **Social/personal risks**: Security and privacy dominate.
- **Complexity**: Interactions between devices may be unpredicted, unforeseen and unknown.
- **Privacy**: Data collection is pervasive but invisible and largely out of our control.
- **Abuse**: The IoT brings benefits but the pervasiveness of networks and data invites criminal activity.
- **Corporate security**: Previously secure company systems are now being connected to much larger, insecure networks.

In contexts such as transportation systems, manufacturing production lines, TV stations, energy generation/distribution and smart cities, the potential for hacking, disruption and terrorism is unbounded. The security industry has a lot to learn and more to do. The IoT is a whole new ball game.

The IoE brings new levels of complexity and scale. The non-functional risks are reasonably well-known and we know how to address them. What's new is the need to do functional testing and simulation at scale.

The Scope of IoE Testing

The range of concerns that the IoE brings is wider than ever before. Not all IoT systems will be huge, complex and expensive but all provide a different technical and risk profile than we are used to. Here are the main dimensions of scope for the IoT tester:

- **Hardware-level testing**: The lowest level devices are sophisticated, but perform simple functions; most will be performed by manufacturers.
- **Scale**: A wearable application might be simple in its architecture, but could be scaled to a user base of millions.
- **Object- and server-level functionality**: Most functional testing will take place at the level of local hubs, aggregators and data-centre-based servers. Architectures will range from simple Web apps to systems with dozens of subsystems.
- **Mobile objects**: These move in and out of the range of networks and roam across networks. Environmental conditions, the sources of data and device location affect behaviour. Power, interference, network strength, roaming and jamming issues will all have an effect.
- **Moving networks**: Some systems (for example, connected cars) carry their own local network. A network that moves will encounter other networks that interfere or may introduce a rogue or insecure network into range and pose security problems.
- **Network security risks are at many levels**: Rogue devices use your network and eavesdrop or inject fake data. Rogue access points hijack your users' connections and data. Vulnerabilities exist at all levels in your architecture and are prone to attack.

- **Device registration, provisioning, failure and security**: Initial device registration and provisioning are failure-prone. Devices are prone to power failures, snow, heat, cold, vandals, animals, thieves and so on. Power-down, power-up and automated authentication, configuration and registration processes may need to be tested.
- **Collaboration confusion**: Moving devices (for example, cars, again) collaborate in complex ways and in large numbers. But accidents happen, drivers change their minds, car park spaces will become available and unavailable randomly, and so the optimisation algorithm must cope with rapidly changing situations. At the same time, these services must not confuse users.
- **Integration at every level**: Simple and complex flows of data and control—end to end.
- **Big data—logistics**: Substantial data storage services will be part of the system to be tested.
- **Big data—analysis and visualization**: Data science and visualization are likely to be in the scope of the IoT. This includes timely, accurate and consistent data as well as filtering, merging, integration and reconciliations.
- **Personal and corporate privacy**: Hackers and crooks are one threat, but your own government may be seen as another potential villain.
- **Wearables and embedded**: Wearable and human-embedded devices provide new and unique challenges.
- **Everything connected**: The time will come when all of the devices used in a hospital, hotel and factory, for example, will be connected.

The range of issues we need to consider in testing the IoE has increased and the scale of the testing required has increased too.

Functional Testing at Scale

When we test the functionality of components higher in the architecture, particularly at the integrator and application level, we might have to simulate thousands or millions of devices in the field. The numbers of combinations and permutations may be beyond computation or prediction. Our simulations will repeatedly generate scenarios to be tested, record the outcomes and might replay the simulations for later study.

The higher level components must be testable. We'll need facilities like exception handlers, utilities that inject data, capture and reproduce or replay scenarios. Cem Kaner has written quite a lot about what he calls "High Volume Automated Testing" [4]. This is a good starting point.

These techniques could also be called big data testing. We'll need to find data that fits our purpose. We'll need to generate, tag, edit and seed data so we can trace its usage. We'll need tools to monitor the use of tagged data and the ability to reconcile data from collection, storage, use and disposal. We'll need new test visualization tools to support diagnostics and debugging.

This is a volumes game. Individual tests may or may not be important, but we'll spend a lot of time dealing with large scale outcomes, visualizations and decision making.

Test Environments, Testing in the Field

In a simple case, a test lab for a home environment management product could be set up in any office as the scope of the local network is confined to a single household. In the case of an urban environmental management system that monitors air pollution levels across a city, for example, the sensor data capture could be simulated in a lab. However we would expect to have to pilot the service in a real city environment to calibrate the sensors, data aggregation and integration processes to obtain meaningful data visualizations.

Tool Support

Although there will always be a need and opportunity to do manual testing, a much larger proportion of testing than we are used to will have to be performed by tools, which will need to execute a very large numbers of tests. The challenge is not that we need tools to execute tests; the challenge will be, "how do we design the hundreds, thousands or millions of tests that we need to feed the tools?"

The devices now appearing all have their nuances and complications and will experience unexpected events. Even a simple system could encounter thousands of scenarios. Systems are getting more complex and these need testing. It's only going one way.

Test Analytics, Visualisation and Decision Making

I will expand on test analytics in another article in this series. If your production systems are platforms for experimentation, then the code that serves marketers can also serve developers and testers. Think of your analytics code as your sensor network. DevOps processes are things, too.

Performance Testing and Test Data

Performance testing in an IoT world is not much different from our current experience. What may make life harder is when the data that is captured by sensors must be coherent. For example, the messages received from cars in a city must match a physical location to be meaningful. A randomly generated set of location coordinates will not do. We'll need trusted data sets from real world operation or utilities that can generate meaningful and trustworthy data for testing.

The Future for Testing and Testers

It seems to me that high levels of test automation are bound to be required to make the IoE a reality. Automation will not make testing easy; it will make testing possible.

Test automation in a DevOps environment is a source of data for analysis just like production systems, so analysis techniques and tools are another growing area.

Should testers learn how to write code? I have a simple answer—yes. Now it's possible that your job does not require it, but the trend in the U.S. and Europe is for job ads to specify coding skills and other technical capabilities. More and more, you'll be required to write your own utilities, download, configure and adapt open source tools or create automated tests.

The IoT and digital trends are pushing testing to the left. It seems like every company is pursuing what is commonly called a "shift-left" approach. The activities, or rather, the thinking activities of testing are being moved earlier in the development process. My advice is to embrace it.

Summary

The IoT increases testing in scale, diversity and complexity.

Testing low-level components or subsystem testing is pretty much the same as before. But at the system level, we'll need simulation methods and high volume test automation. The tools we need may not yet exist, so you may find you have to create your own until the tools suppliers catch up.

High volume test automation requires test models, test-data generators and automatic oracles. Modelling, simulation, analytics, visualisation and tool-supported decision making will become important capabilities of test architects and testing teams. Testers will have to learn how to create better test models and how to use them with more technical modelling and simulation tools.

Creating trustworthy test environments and meaningful test data will cause big headaches (as always).

Large-scale test environments in the lab and environments in the field will be required and the boundaries between experimentation in production and testing in the lab will become blurred. Test analytics derived from DevOps processes will become a critical discipline in testing the IoT.

About the Author

Paul Gerrard is a consultant, teacher, author, webmaster, developer, tester, conference speaker, rowing coach and a publisher. He has conducted consulting assignments in all aspects of software testing and quality assurance, specialising in test assurance. He has presented keynote talks and tutorials at testing conferences across Europe, the USA, Australia, South Africa and occasionally won awards for them.

Educated at the universities of Oxford and Imperial College London, in 2010, Paul won the Eurostar European Testing excellence Award and, in 2013, won The European Software Testing Awards (TESTA) Lifetime Achievement Award.

In 2002, Paul wrote “Risk-Based E-Business Testing” with Neil Thompson. Paul wrote “The Tester's Pocketbook” in 2009. Paul co-authored “The Business Story Pocketbook” with Susan Windsor in 2011 and wrote “Lean Python” in 2014.

In 2014, Paul was the Programme Chair for the EuroSTAR Conference in Dublin.

He is Principal of Gerrard Consulting Limited, Director of TestOpera Limited and is the host of the Test Management Forum.

Mail: paul@gerrardconsulting.com
Twitter: @paul_gerrard
Web: gerrardconsulting.com

For more information visit **Develop & Test** with CA Technologies.

Connect with CA Technologies at ca.com

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments. Learn more at **ca.com**.

References

1 The Internet of Everything—What is it and how will it affect you?, Paul Gerrard, http://gerrardconsulting.com/sites/default/files/IoEWhatIsIt2.pdf

2 Internet of Everything—Architecture and Risks, Paul Gerrard, http://gerrardconsulting.com/sites/default/files/IoEArchitectureRisks.pdf

3 Internet of Everything—Test Strategy, Paul Gerrard, http://gerrardconsulting.com/sites/default/files/IoETestStrategy.pdf

4 An Overview of High Volume Automated Testing, Cem Kaner, http://kaner.com/?p=278