INTERNET OF THINGS

eGuide

TECHWELL™

TECHWELL™

The future is here, and it's connected to the internet. Almost everywhere you turn these days, there are internet-enabled devices, appliances, even toys. While all this new technology is intended to streamline and simplify our lives, the reality is, the Internet of Things depends on the quality of the software that powers it.

We've all heard the saying, "Garbage In; Garbage Out." Our challenge in this age of ultra-connectivity is to design, develop, and test our products to be highly reliable, functional, and user friendly. Anything less, and we lose the trust of our users and, ultimately, lose in the marketplace. This eGuide will help you navigate today's constantly changing IoT landscape.

In this Internet of Things eGuide

4 Fundamental Practices for IoT Software Development

The IoT enables devices designed to make our lives easier. But IoT products are only as good as the software behind them. Learn four practices you'll need to adopt when developing software for the IoT.

3 Steps to Nurture IoT Development and Testing

As more devices connect to the internet, QA must cultivate an understanding of the IoT and how to create software for these connected items. Find out the crucial three steps for IoT development and testing.

Accountability in Testing Embedded and IoT Software Systems

Take a look at the critical systems in the world today and you'll find software. We need to do testing from a risk-based perspective and be accountable to the public by acknowledging what is tested and what is not.

IoT and the Wisdom of Mobile

The IoT has taken the world by storm and is growing exponentially by the minute. Since mobile's been around so long, can what we learned from that revolution help us in this new connected age?

The Future of Cloud Connectivity in an IoT World

When new technologies are embraced and popularized, they usually fail sooner rather than later. The IoT, new architectures, and cloud systems will take time to develop and mature, finally providing calm, consistent conditions. How should you plan to fail?

DevOps Helps Enterprises Deliver Better, Faster Software for the IoT

As the world becomes more connected, it's changing the way we do things, especially in relation to software delivery. Software development for IoT applications presents obstacles concerning security, privacy, and unified standards. But we need look no further than DevOps to find the answers.

The Buzz on the Internet of Things

What industry insiders have to say about developing and testing for the Internet of Things.

Additional IoT Resources

Invaluable resources to keep you, your organization, and your practices at the leading edge of the IoT movement.

4 Fundamental Practices for IoT Software Development

By Lev Lesokhin

Today, if you're a technology leader and work for a public company without an Internet of Things (IoT) strategy, you can wave goodbye to your share price on its way down. IoT is no longer a nascent dream. By 2017, IDC analysts predict spending on IoT technology and services will exceed $7.3 trillion. [1] Global brands, such as Intel, already have announced significant changes to their business to focus on IoT, and as more devices "connect" the lines of autonomous provisioning, management and monitoring will continue to blur.

Is IoT Just a Fad?

Putting the hype aside, one of the most important conversations to emerge lately relates to the tactical elements of IoT. What do organizations need to address in development to make this a successful technological shift? Without precise execution, IoT could turn into a nightmare (remember the movie "Minority Report"?) Devices are getting smarter—talking to each other and cutting out the unreliable human elements, which result in higher quality and greater productivity. Embracing and successfully managing all of the technological complexity that comes with IoT are the most important steps toward its success.

> *Devices are getting smarter—talking to each other and cutting out the unreliable human elements, which result in higher quality and greater productivity.*

To twist the famous words stated in Forrest Gump, "Smarter hardware is as smarter software does." This implies that IoT products and services will only be as good as the software behind them. This isn't creating a new set of problems; software is already pervasive. Instead, the growing ubiquity of IoT just magnifies the potential impact of problematic software, and that is where the trouble begins.

Obstacles in Transitioning to IoT Development

First, many companies investing in IoT are not traditionally involved in computing. For example, today more than 50 percent of IoT activity is centered in industries such as manufacturing, government (smart city), and consumer products. [1] Some of these organizations suffer from a lack of proficiency in putting together such a dynamic software capability. This is often exacerbated by the fact that engineers, especially those with knowledge in connected device development, are in high-demand. The other issue is once an IoT competence is established, organizations then have to figure out how to manage the software. This results in a trifecta of problems. Even traditional embedded software developers haven't caught up with all the practices needed to develop Internet-connected applications. How will the rest of the current software industry manage?

> *It may take a little bit more time to design and build robust software upfront, but secure software is more reliable and easier to maintain in the long run.*

Second, embedded software components have to interact safely with other Internet-facing components. Although an application can operate on secure subnets, access will be restricted to users of the same subnet. That may work for some business models, but it is not a viable solution for anyone who wants to access the global Internet. As a result, developers have to understand how these connected components will interact in order to ensure security, reliability, and efficiency. This is a common problem in IT but new in device software development.

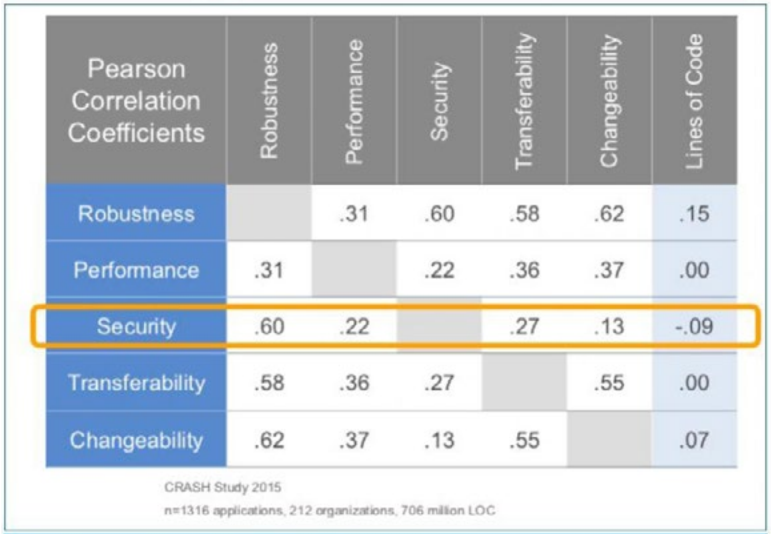
Third, IoT exposes developers to problems and capabilities that are, for the most part, already well known in traditional computing but how to counter them in a connected world is still relatively virgin territory. For example, enterprise and web developers are very familiar with the need for robust security against local and remote attacks. The notion of input validation as a first line of defense is well accepted in connected systems today. However, IoT development expands the scope of those concerns. Embedded, device, and mobile developers need to start considering security challenges such as input validation during development. Otherwise, it will be cost prohibitive to redesign onboard systems to include these defenses after they have shipped to customers.

What about Security and Product Quality?

One of the greatest concerns in IoT is security, and how software engineers address it will play a deeper role. Security needs to be tackled at the start of the design phase, making requirement tradeoffs as needed. During code construction, security needs to be baked into the process—making it more foundational than a mere "bolt on."

Due to the large amount of complex data being exchanged, manufacturers will have to build in privacy options that can be invoked at the expense of additional functionality. This comes back to designing in security during the requirements phase of software development.

Security is highly correlated to software robustness. It may take a little bit more time to design and build robust software upfront, but secure software is more reliable and easier to maintain in the long run. CRASH Report 2015 shows a high correlation between security and robustness as shown in table 1. [3]



Pearson Correlation Coefficients	Robustness	Performance	Security	Transferability	Changeability	Lines of Code
Robustness		.31	.60	.58	.62	.15
Performance	.31		.22	.36	.37	.00
Security	.60	.22		.27	.13	-.09
Transferability	.58	.36	.27		.55	.00
Changeability	.62	.37	.13	.55		.07

CRASH Study 2015

n=1316 applications, 212 organizations, 706 million LOC

Table 1: Correlation among software characteristics (CRASH Study 2015)

The CRASH Report results suggest that one third of security problems are also robustness problems—a finding that is borne out in our field experience with customers. In my view, there's a relationship between the security budget and the overall budget for software quality.

Despite software developers' best intentions, management is always looking for shortcuts. In the IoT ecosystem, first to market is a huge competitive driver, so this could mean that quality and dependability are sacrificed for speed to release. Device manufacturers have learned the hard way that putting "glitchy" software on devices is asking for trouble. Poorly written software continues to be one of the greatest safety issues today (e.g., Jeep Cherokee's car hacking fiasco of 2015).

Sometimes glitches happen because developers are under pressure to get products or software updates released. Consequently, they end up burning the midnight oil to fix bugs that should have been discovered during the development process. Third-party components help offload some of the burden, but with more complexities and upkeep in IoT, components are expected to be maintained and updated to address problems at a much faster pace.

The Four Fundamental Practices in IoT Software Development

To meet demands, avoid pitfalls, and achieve success in the growing IoT marketplace, organizations need to adopt four important practices for IoT software development: review, assessment, responsibility, and advocacy.

Review: Proper code review and repetitive testing need to be a priority. Manufacturers must communicate this message to software engineering teams and call for stricter software quality measures. The high complexity of IoT applications leaves software susceptible to security lapses and software quality failure. One bad transaction between an application, a sensor, and a hardware device can cause complete system failure. Organizations just can't afford that.

For some developers, this environment is quite familiar, especially if they are running mission critical systems for a utilities provider or a bank. However, ordinary app and device software developers may find themselves engaging in the same degree of structural quality analysis and code review required to develop airline autopilot systems.

Assessment: Continuous deployment in the connected world is business as usual. Updates occur non-stop and are often pushed multiple times a day. The quality assurance burden on the software that interacts with IoT devices is greater than ever. If the software isn't continuously monitored and the code evaluated, failure is almost guaranteed.

Responsibility: Management must take responsibility for quality assurance. Any manufacturer that doesn't have a set of analytics to track its software risk—be it reliability, security, or performance—is negligent in its responsibility to customers and other stakeholders. Management needs to lead by example and communicate the direct link between software quality and security. It's in their best interest, too, since security vulnerabilities caused by poor coding or system architectural decisions can be some of the most expensive to correct.

Advocacy: In addition to measurement and analytics, a cultural shift to include education needs to occur. Developers and management collectively need to spread the word in the community about standards. Significant strides have been made in creating initiatives for manufacturers and IT departments to consistently measure the quality of their software.

In 2015, the Object Management Group (OMG) approved a set of global standards proposed by the Consortium for IT Software Quality (CISQ) to help companies quantify and meet specific goals for software quality. [4]

> *Sexy consumer applications (predictive coffee makers, self-driving cars) and cyber attacks (DDOS assaults launching from refrigerators) still dominate the IoT headlines, but the deeper business value is starting to emerge.*

CISQ was formed as a special interest group of OMG to create standards for automating measures of software size and quality attributes (e.g., security) at the source code level. While software quality standards have existed for a long time, the traditional ISO standards only specified the categories of what should be measured—not how to actually measure it.

What CISQ has done is to define in detail how these characteristics should be measured. These measures were designed for use by IT organizations, IT service providers, and software vendors in contracting, developing, testing, accepting, and deploying software applications. CISQ's measurement standards include security, reliability, performance, and maintainability. This allows businesses to certify the quality of codebases and IoT networks.

Succeeding with IoT

Sexy consumer applications (predictive coffee makers, self-driving cars) and cyber attacks (DDOS assaults launching from refrigerators) still dominate the IoT headlines, but the deeper business value is starting to emerge. The McKinsey Global Institute report points out that mature IoT systems will take the guesswork out of product development by gathering data about how products, including capital goods, actually function and how they are used, rather than relying on customer focus groups. [5]

That is certainly a game changer requiring development teams to reduce risks in the software they engineer now. Understanding the importance of a secure architecture foundation and insisting that developers comply with industry standards will be the first line of defense.

After that, the rest is up to you.

References

1. *IDC, Worldwide Internet of Things Predictions, 2015*

2. *Intel Corporation. "Intel Announces Restructuring Initiative to Accelerate Transformation."* Intel Corporation *(news release). April 19, 2016.* https://newsroom.intel.com/news-releases/news-release-intel-announces-restructuring.

3. *Bill Curtis, Lev Lesokhin, Alexandra Szynkarski, and Stanislas Duthoit. The CAST Research on Application Software Health (CRASH) Report 2014–2015." Cast Research Labs.* http://www.castsoftware.com/resources/resource/cast-research-labs/thank-you/2015_cast_crash-full-report.

4. *CISQ. "Code Quality Standards."* Consortium for IT Software Quality. http://it-cisq.org/standards.

5. *Manyika, James, Michael Chui, Peter Bisson, Jonathan Woetzel, Richard Dobbs, Jacques Bughin, and Dan Aharon. "The Internet of Things: Mapping the Value Beyond the Hype."* McKinsey Global Institute report . *June 2015.*

3 Steps to Nurture IoT Development and Testing

by Sanjay Zalavadia

The Internet of Things is a trend that many businesses have seen as “edging over the horizon,” but the truth is that IoT is already here. More devices are being imbued with the ability to connect to the Internet and engage in machine-to-machine communication. This no longer only includes smartphones and wearables; it’s extending to ordinary objects like fridges, cars, and toasters. With many new opportunities available, QA management must cultivate an understanding of the IoT and how to create software for these connected items. Here are three steps to nurture IoT development and testing.

1. Have the user in mind.

In regards to IoT testing, user experience is top priority. After all, if a driver tries to use GPS capabilities while on the road and software has defects, the app will likely be abandoned. Organizations not only have to gain a good understanding of what their object is capable of and what apps would pair well with it, but also how users will actually leverage the item. In an interview with StickyMinds, Zephyr CTO Shailesh Mangal noted that test subjects will be essential to the quality and functionality of an IoT app. Requirements will be different depending on if a person is testing a health device or a pet is trialing a bio-chip. By keeping the user at the forefront, developers and testers will have guidance to complete tasks and aim the app directly at the audience.

2. Prepare for challenges.

Although determining a user is beneficial, there are a number of other difficulties that teams are likely to run into when creating IoT software. For this reason, it’s critical to prepare for and understand these challenges now. ReadWrite contributor Alex Brisbourne noted that connectivity, Internet infrastructure, and privacy are still main concerns when nurturing IoT testing environments. Each object has its own protocols, making communication a tricky goal to reach, especially in areas where the Internet infrastructure isn’t up to par. The differing standards also make it much harder for quality assurance teams to guarantee security and protect sensitive information on IoT devices.

3. Combine new and old testing concepts.

For IoT, teams cannot solely rely on new testing methods. They must also use their traditional practices to cover all their bases. This means implementing things like risk-based, exploratory, and unit testing. Although these have all been used for years, they will still be valuable in an IoT project. In addition to this, organizations may find it beneficial to use test management tools to handle the variety of scripts and track progress across projects. *LogiGear Magazine* contributor Jon Hagar noted that these solutions can help teams test better and increase the likelihood of identifying errors early. This functionality, along with other tools such as automation, will be significant assets in producing quality apps and supporting software throughout its lifecycle.

IoT development and testing is a considerable challenge that is still being tackled. As more standards fall into place, teams can use these steps to nurture an IoT development and testing environment.

Accountability in Testing Embedded and IoT Software Systems

by Jon Hagar

I usually write about testing, particularly in the internet of things and embedded environments. However, recent terror attacks and cyber crimes have made me stop and think about what we in the software testing world do—or maybe *should* do—to protect our systems. I hope this piece makes you stop and think also.

Years ago, I did some consulting for the Nuclear Regulatory Commission (NRC) with the US government. I was the young member of a group of very senior researchers and practitioners who considered what the NRC should require when companies place digital devices into nuclear systems.

Historically, nuclear systems were mainly analog electrical and mechanical, but the NRC knew this was changing with the introduction of digital devices with embedded software. The experts were concerned that the insertion of software into nuclear systems would cause problems when there were bugs. We made recommendations, and embedded software devices became successfully used in nuclear plants. At that point, we were not worried about the connection of devices to the internet, which barely existed.

Then, Code Took Over the World

Take a look at the critical systems in the world today and you'll find software. From water, power, and utilities to nuclear plants, factories, and cars, pretty much everything has become integrated with digital devices and the internet. For a long time, embedded software didn't really have to worry about connections to the outside world; these connections were nonexistent or minimal, often via a secondary connection. Then the internet of things, or IoT, changed all that.

For the last few years, much of the professional software engineering and testing world has become well aware of the security threats to embedded and IoT devices. By this point, the public has also heard of cars getting hacked, threats to the infrastructure, and digital weapons such as Stuxnet. Again, the IoT is changing our world.

Testing Needs to Grow Up

While I sense awareness and people saying, "Yes, yes, testing and quality of the IoT is important," I do not see the real action and accountability that I feel is needed. The public and even the general IoT industry seems willing to wait until something really bad happens, like cyberterrorism targeting a nuclear plant or major infrastructure. If this does happen (maybe I should say *when* this happens), there will be a great cry for better quality, engineering, and testing—concepts that have been preached all too often through the evolving years of software.

Until then, testing is, as James Bach once said, a bit of a Peter Pan. We do a lot of things that might or might not add value. If there is a failure, we can say, oops, the programmer messed up too, and the requirements are unclear, and the combinations are impossible to fully test. Because of that, testing never really needs to "grow up."

It has been thirty-five years since I started working at a major defense contractor. My role in the community is such that I get to do what I think needs to be done instead of pursuing a paycheck. I am coming to believe I am to be a bit of a bellwether, raising the warning before that really bad thing does happen—such as seeing that car control systems could get hacked while driving, and expanding that train of thought to caution against the potential for major power outages or nuclear meltdowns.

We Must Do Better

Testers, engineers, and companies must do a better job *now* and be accountable to our users and the general public, which includes our families. Accountability in the software devices we produce can't be addressed solely by standards or in our legal systems. Most of those systems seem to protect the producers of software, not the consumers. When we create an embedded or IoT device, particularly for industrial use, we need to do testing from a risk-based perspective where we acknowledge what is tested, what is not, what qualities the device has, and any implications of not testing pieces or parts or the whole. That might be happening already, but the customers don't know; they are simply trusting our judgment.

> *Accountability in the software devices we produce can't be addressed solely by standards or in our legal systems.*

Imagine a world where the test strategy is documented and clearly communicated to the customer and stakeholders, in a way that they can understand the tradeoffs between testing, cost, and time to market. Instead of choosing what product they want to buy and use based only on brand or price, they get an added dimension of test strategy.

I define tester accountability as the obligation of an individual or organization to account for its activities, accept responsibility for them, and disclose the results in a transparent manner. There are no standards telling us to be accountable for our software testing. Testers who have been involved in legal actions regarding software failures learn that software can be liable or answerable to the courts. Why are we waiting to be accountable until it's mandated?

We need to be clear about the qualities of embedded and IoT systems, and software in general. What we engineer affects others, so accountability is essential. The hackers and other bad guys out there will eventually take advantage of our lack of accountability if we don't do our jobs. It's time for testers—and testing itself—to grow up.

IoT and the Wisdom of Mobile

By Steven Winter

Industrial revolutions come and go. Their impact is measured over time by the progress their disruptions bring, and typically, they are measured in how we reacted and evolved due to that disruption. Personal computing, the Internet, and mobile devices are all prime examples of such massively disruptive forces, and now the Internet of Things (IoT) has taken the world by storm with its tentacles of connectivity growing exponentially by the minute. Sound a bit dramatic? Well, it's not. With over 12 billion—yes, I said "billion"—devices connected today and 5000 more wired up by the time you're done reading this article, the hyper-connectivity of IoT is here with its promise of dramatically increasing the quality of our lives.

Rough estimates give the IoT impact an economic impact north of $11 trillion per year by 2025. This is relatively the same amount of fiscal value mobile will have by then, but mobile's been around the block a heck of a lot longer. So, to say IoT is the greater disruptor is an understatement.

Since mobile's been around so long, what have we learned from that revolution that can help us in this new connected age of IoT? A great deal, when you break it down! Pretty much all of the big aspects of mobile development and testing can be applied to IoT. From firmware on the device, to the network protocols and APIs they use to communicate, the security of those connections, and the data transferred to and from the device—all are existing technologies used in mobile. Meaning, we already have the knowledge to be prepared for and thrive in a hyper-connected world. That's the good part of the message.

> *Pretty much all of the big aspects of mobile development and testing can be applied to IoT.*

The challenging part is in managing the scale, managing and maintaining the software on the devices, managing the increasing demands on APIs and big data management, and managing the increased security threats that this hyper-connectivity will create. To that, security, data management, and test automation become even greater aspects of IoT that absolutely must be mastered to not just keep pace but, if you do it right, stay ahead of the curve.

Managing security, alone, is shaping up to be a big IoT market differentiator as the key threats research tells us. When breaking it down to its most simple form, you have more access points to a network, which generates potentially mass amounts of data about you. Device fragmentation, over the air software updates, third party API integrations, big data, and all of it networked are the big challenges we've seen from mobile, and all are present in the world of IoT but at greater scale.

With the big friction points being security, data management, API management, and software and firmware updates to the connected devices, we see that we have the knowledge to thrive in the new IoT world! Don't let the hype scare you away from this new reality. If you've been on top of your mobile game, which most of the industrialized world has for some time, then you have what it takes.

The Future of Cloud Connectivity in an IoT World

by Steven Woodward

Imagine your company has just completed development of a great application solution that leverages new Internet of Things (IoT) components.

The application has been rigorously tested in a comprehensive manner, meeting and exceeding all functionality, performance, and security tests. The associated connectivity points are based on state-of-the-art software-defined networking and network function virtualization frameworks.

It gets rolled into production and, voila! Multiple outages and unsatisfactory customer experiences. Does this sound familiar?

When new technologies are embraced and operationalized, they usually fail sooner rather than later. The IoT, new architectures, and cloud systems are developing into perfect storms that will take time to develop and move on to maturity, eventually providing calm, consistent conditions.

Based on historical data, we know that as new technology is introduced, learning curves are required, mistakes are made, and defects appear out of nowhere. Now, let's think about additional security risks and threats that are more often the major targets on the radar. Moving forward, the risks intensify further. Fortunately, technical staff are inherent optimists (something that usually surprises the business), so in their minds, everything is fine until things go wrong.

Does this mean we should never use new technology?

Of course not. However, it does mean that developers and testers must be allocated adequate budgets, schedules, and tools to plan for failure so that risks can be mitigated and benefits realized.

Remember, I said the technical staff are optimists, so the reality is that they will underestimate what is really required. The technical community must clearly communicate the risks (financial, legal, and credibility-wise) and the multiple points of potential failure or security vulnerabilities so that stakeholders can make informed decisions.

The IoT, cloud, and connected world we will live in require tremendous abilities to recognize all the end-to-end solution components and the integration challenges that accompany them. Each component is on the critical path, including, of course, networking and connectivity. Therefore, planning for component failure regardless of root cause has tremendous benefits, especially as several new sets of disruptive technology rolls into production.

Planning for failure is the key to successful complex IoT cloud deployments.

DevOps Helps Enterprises Deliver Better, Faster Software for the IoT

By Anders Wallgren

Nowadays, just about every business is a software business. Software is considered the main driver for innovation and disruption across markets. To keep up with the competition, large enterprises need to continually deliver better software, faster.

Along with the emergence of the software-driven business, we've also seen the Internet of Things (IoT) grow and influence our everyday lives through smart homes, automated features in cars, and more. There are already billions of IoT devices in use, and those numbers are only projected to increase: Gartner predicts that by 2020, there will be more than 26 billion IoT devices deployed.

As the world becomes more connected, it's changing the way we do things, especially in relation to software delivery. We have become accustomed to connectivity from anywhere. We have environments that will automatically optimize based on comfort or cost, we can access continuous reporting, and we expect software updates over the air.

When we examine the vast amount of software needed for a connected, IoT-driven world, we also see a lot of challenges. The IoT software delivery pipeline is complex and difficult. For starters, software development for IoT applications presents obstacles concerning security, privacy, and unified standards.

Additionally, each IoT product is composed of at least three separate application components: the software embedded in the device, the back-end service, and the mobile application for the end-user's controls. Each component is developed by a different team, using different technologies and practices, and is deployed to a different stack or target. All these variables make the integration of separate pipelines and the coordination of software updates for IoT problematic, to say the least.

In addition to muddled development teams and disparate practices, we have to figure out a way to combat skyrocketing infrastructure costs, visibility and compliance concerns, and quality issues. Not only do customers want software fast, but they want value at speed. Continuous delivery of IoT services is essential for success. How do we successfully connect all these different pieces to deliver software?

> *Most large enterprises that are delivering fast, high-quality software are somewhere along their own personal DevOps journeys.*

The good news is that we need look no further than DevOps to find the answers. Most large enterprises that are delivering fast, high-quality software are somewhere along their own personal DevOps journeys. These organizations have a lot of things in common, such as shared goals across teams, a can-do culture, measurability, visibility, continuous improvement, smaller cycle times, and automation.

When the right patterns, tools, and processes fall into place, software development speeds up and quality reigns. Suddenly, disparate teams experience end-to-end visibility and traceability, accelerated delivery, and predictable quality.

The Buzz on the Internet of Things

INSIGHT FROM AROUND THE INDUSTRY

On Digital Transformation

"Don't panic. Digital is going to change things, but now, those challenges are going to be the same as what you did before. You'll still be using classification trees; the difference is you'll be applying them in a sense of testing an IoT platform or a new banking API. You still have to use your skills."

» Jonathon Wright

On Security

"As scary and as ominous as that is, it's a known quantity. We've been dealing with these threats from when PCs became a thing, and they connected from the fancy thing called 'the Internet,' and then the mobilization of all that connectivity. The challenge around IoT and security is you have scale, and you have that many more entry points to do bad things."

» Steven Winter

On Mobile Driving the Iot

"Mobile is still the dominant controlling factor for the IoT world. I do think that over the next five to six years, IoT will just be a flood of data, but still I think you're going to be using mobile to drive a lot of that."

» Jason Arbon

On Lack of Standardization

"It's very much the Wild West out there. There's very little standardization, and one of the big promises of IoT is that all of your devices are going to work together and talk together and function like a seamless whole. For that to work, then obviously, these things need to talk to each other, and that's really hard when there's still so much fragmentation."

» Kevin Rohling

On the Future of the Iot

"The internet as we know it will absorb what people are currently thinking of as the Internet of Everything. That is, [the internet] is going to be ten times bigger, one hundred times bigger. Who knows? I like the idea of it becoming just 'the internet.' In the future, the internet will accommodate all these things that are connected.

» Paul Gerrard

On Defining the IoT

"As these devices are getting cheaper, easier to build, and there's more innovation happening, it's going to keep evolving and there's going to keep being new devices that just fall under this blanket. You can't really nail it down with one or two specific things. It's just anything that's going to be connected to other devices or to the cloud."

» Chris Beauchamp

> *The internet as we know it will absorb what people are currently thinking of as the Internet of Everything.*

On the Ubiquity of the IoT

"Look at all the computers that we carry around with us: laptops, tablets, phones. Our televisions are now computers; they're not just glass tubes anymore. Our thermostats are computers, our smoke detectors are computers, so it's really down to the point where everything is going to be on the internet, everything is going to be connected. The ice bucket in my hotel room is probably connected to the internet somehow."

» Anders Wallgren

On Testing the IoT

"Testing the Internet of Things is not all that different from testing software. It's just you have a thing you've opened up to the environment. You have to understand all these pieces, and a lot more. But it's still the ones and zeros. The input; the output."

» Jane Fraser

Additional Resources

MORE INFORMATION FOR SOFTWARE PROFESSIONALS

STICKYMINDS™ A TECHWELL COMMUNITY

StickyMinds is home to thousands of software testing resources, including articles, *Better Software* magazine articles, conference presentations, and interviews with industry notables.

CLICK HERE

NARROW YOUR SEARCH TO A SPECIFIC TYPE OF RESOURCE:

StickyMinds Articles

StickyMinds articles cover a wide range of software testing topics including testing for the IoT, test automation, test management, test design techniques, agile testing, test process improvement, test tools, and much more. **Click here** to read IoT articles on StickyMinds.

Better Software Magazine Articles

Better Software magazine is a digital quarterly filled with expert analysis, how-to articles, and real-world case studies covering all aspects of software development. **Click here** to join StickyMinds and access *Better Software* magazine articles about the Internet of Things

TechWell Conference Presentations

Couldn't make it to a STAR software testing conference? TechWell conference presentations are available to StickyMinds members soon after a conference ends. **Click here** to join StickyMinds and access conference presentations related to the IoT.

Interviews

Each year, TechWell interviews dozens of software professionals including well-known thought leaders, seasoned practitioners, and respected conference speakers. **Click here** to read, listen to, and watch interviews with Internet of Things experts.

TechWell Software Conferences

STAR CONFERENCES TECHWELL EVENTS

The STAR conferences focus exclusively on software testing and quality improvement, covering the spectrum from established practices to emerging trends.

Better Software CONFERENCES TECHWELL EVENTS

Learn what you need to build better software now. Better Software Conferences keep you current on the entire development lifecycle.

Agile Dev CONFERENCES TECHWELL EVENTS

Whether you're new to the agile process, or you're experienced and ready to take your team to the next level, our hands-on, in-depth workshops have you covered.

DevOps CONFERENCES TECHWELL EVENTS

Learn how the practice of DevOps brings cross-functional stakeholders together to deliver software with greater speed and agility.