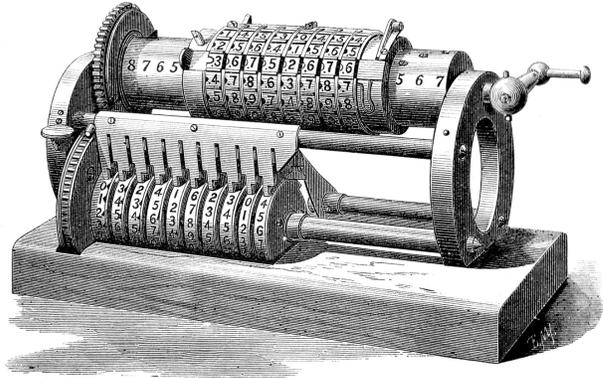


Table of contents:

- 3** History of Computing
- 4** Internet of “Things”
- 5** IOT Device Characteristic
- 6** IOT and Testing
- 8** What Will Change
- 9** What Will Remain
- 9** What It Means For A Tester
- 10** Conclusion

History of Computing

The history of computers—and in the twenty-first century, we’re far enough into the future that we can talk about the *history* of computers—contains numerous revolutions and evolutions that have not only caused fundamental shifts in thinking about computers and interactions with computers, but have also *quicken*ed the pace of change. Whether you consider the start of computing as the invention of the abacus several thousand years BC or the difference machine and algorithms developed by Charles Babbage and Ada Lovelace in the early nineteenth century, a computer history timeline grows ever more crowded to the right, particularly in the latter half of the twentieth century and beyond.

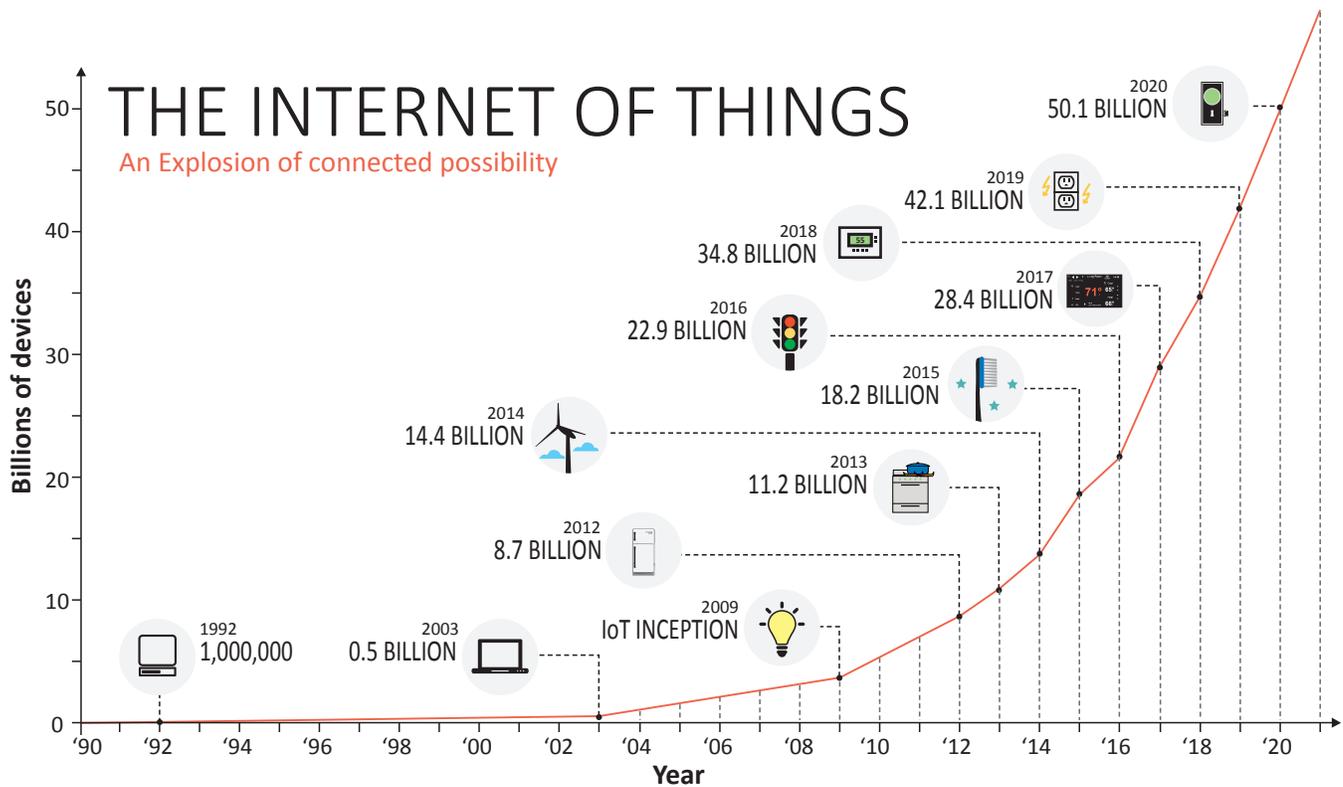


The timeline would highlight:

1940s and 1950s	The work of Alan Turing Grace Hopper’s spearheading of the UNIVAC project, the development of COBOL and the A compiler in the middle of the twentieth century
1960s	The development of ARPANET
1980s	The personal computer revolution and the platform-independence of MS-DOS
1980s and 1990s	Consumer-based networks such as Prodigy, America Online, and CompuServe
1990s	Marc Andreessen’s development of the Mosaic Web browser and the rise of the World Wide Web Amazon and the rise of online commerce at the end of the 20 th century and into the 21 st century
2007 and 2008	Introduction of the iPhone in 2007 and Android phones in 2008 leading to the rise of mobile programming and computer apps in peoples’ pockets

Each development changed technology, some evolutionary and some revolutionary. In most cases, the changes broadened the reach, making it possible for more people to use the computers and for the computers to interact with each other in new ways. Each change also required a shift in thinking about technology and in approaches to writing and testing software.

The personal computer revolution and rise of the Internet placed a wealth of knowledge and computing into the hands of users, and the rise of mobile apps have led to specialization as users have picked very specialized programs to handle discrete needs. The new evolution—or revolution—in computing technology is the proliferation of smart devices and apps.



Internet of “Things”

Smart devices extend this specialization beyond mere smart phones.

Smart devices include:



Implanted medical devices that monitor metrics of a patient’s health—or a wrist-worn fitness meter



Sensors in automobiles that monitor tire pressure and display a light on the dashboard when the tire pressures vary



Consumer appliances that alert owners via text message when a load of laundry is finished or when the user needs milk

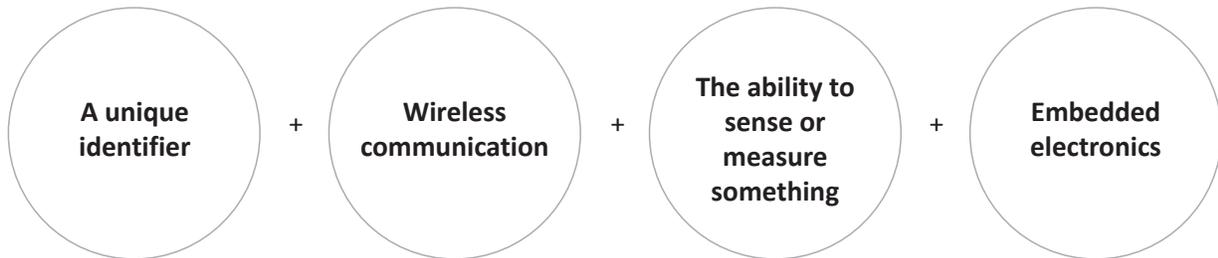


Flow control systems in wastewater treatment plants that determine water volume and act accordingly

Unlike smart phones and their apps, smart devices in many cases don’t rely on a human’s input or report their findings to a human. Some smart devices receive their information from other smart devices and act on their own according to their programming. It goes far beyond simply receiving information; the smart devices also analyze the data they receive and make decisions.

IOT Device Characteristic

Dr. John Barrett of the Cork Institute of Technology in Ireland identifies four characteristics of smart devices:

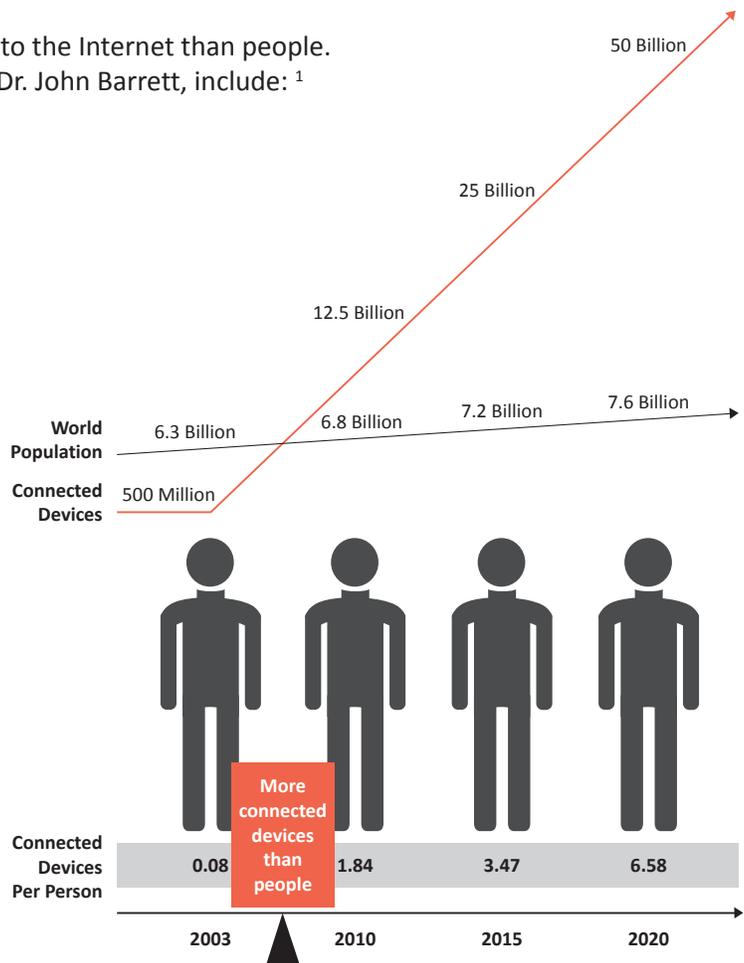


A confluence of events is making this smart device possible now. The electronics are becoming small enough, powerful enough, inexpensive enough to enable widespread adoption, but the Internet infrastructure has ripened as well. Because each smart device needs a unique identifier, it requires an Internet Protocol (IP) address. The recent adoption of IPv6 has increased the number of available identifiers from 4.3 billion addresses to 3.4×10^{38} addresses. The address space expansion means that every atom on Earth could receive its own address with enough left over for over 100 other planets.

Already, more smart devices are connected to the Internet than people. The applications of smart devices, according to Dr. John Barrett, include: ¹

- ① Learn things
- ② Monitor things
- ③ Search things
- ④ Manage things (like cities and traffic)
- ⑤ Control things
- ⑥ Play with things

Experts predict that the Internet of Things (IoT), with that many possible applications and probably more unimagined ones, will have a huge economic impact. ² The McKinsey Global Institute predicts \$2.7 trillion to \$6.2 trillion annually by 2025. Gartner anticipates 26 billion devices on the Internet by 2020, but ABI Research goes further and foresees 30 billion wireless devices connected to the Internet by 2020. These big numbers signal big changes in the Internet as we know it as well as development and testing.



IOT and Testing

So what impact will the Internet of Things have on software testing? What will change? Just as the introduction of smart phones and mobile apps brought new concerns to testing (including touchscreen gestures, location awareness, and orientation concerns), testing the smart devices in the Internet of Things will require some recalibration as well. Whereas tests that we run against computer software involve keyboard and mouse input, smart device testing must account for the data received from the device's sensors as well as user input through tapping and typing. Testing "in the wild" will be required.

For example, consider testing a wrist-mounted fitness tracker. Its user wears the tracking device around the wrist, and the device uses its sensors to pick up the user's body metrics, such as pulse rate and temperature as well as location, elevation, and movement. The smart device takes this input and calculates things such as steps, flights of stairs, and calories burned based on configuration settings. The user can view a simplified version of the data on the device itself, but the device also syncs to a mobile device and possibly to the cloud for detailed analysis. The user will wear the device in all sorts of conditions and place it under all sorts of stresses. That's far more complicated than typing the user's name and address into a Web form, and it requires far more complicated testing.



Consider this sample test case:

- ① **Test Case Name:** Health Fitness Device v3.2 - 3 hour high-activity utilization in sub-freezing temperature
- ② **Test Case Objective:** Verify accurate sensory functionality and data reading over extended (3 hour) high-activity (jogging/running) in sub-freezing temperature (less than 32 degrees)
- ③ **Test Subject:** 185 lb., 6-ft male capable of jogging/running for 3 hours in sub-freezing temperature with no more than a 1-minute break (to gather test verification data) once every 30 minutes
- ④ **Pre-conditions:** 1. Health fitness monitor device is programmed, specific to model, for:
 - a. Gender - male
 - b. Weight – 185 lbs.
 - c. Height – 6 ft.

Step Number	Action	Expected Result
1	Put correctly pre-programmed device on test subject's (device wearer) wrist ensuring device is exposed (not covered by any clothing) and in contact with skin to ensure monitoring capabilities	
2	Manually take test subject's temperature (using standard thermometer), pulse (using a pulse oximeter) and blood pressure (using blood pressure monitor). Compare to temperature, pulse, and blood pressure readings of health fitness device.	Readings of health fitness device must be within .5% of manual readings.
3	Verify health fitness device activity readings – steps, speed, and direction.	Steps should display 0. Distance should read 0 miles. Speed should reflect 0 mph. Direction should reflect null.
4	Have test subject begin jogging, maintaining a 6 - 8 minute mile pace for 30 minutes. Maintain proximity (within 15 feet) to test subject using motorized vehicle and readings of motorized vehicle. Verify activity readings of device (while test subject is still jogging).	Distance, speed and direction reflected on health fitness device should match vehicle's readings exactly. Calculate number of steps based on test subjects height, expected stride, and distance. Should match steps displayed on health fitness device within .5%.
5	After test subject has jogged for 30 minutes have stop for no more than 1 minute to manually take test subject's temperature (using standard thermometer), pulse (using a pulse oximeter) and blood pressure (using blood pressure monitor). Compare to temperature, pulse, and blood pressure readings of health fitness device.	Readings of health fitness device must be within .5% of manual readings.
6	Repeat steps #4 and #5 for 5 additional cycles having test subject perform activity for 30 minutes. Verify activity readings while test subject is active. Have test subject stop for 1 minute for bodily readings (temperature, pulse, and blood pressure.)	

What Will Change

The testing process itself must change to reflect the new realities of interconnected, specific smart devices. Testers and quality assurance professionals will have to account for the data input from accelerometers, gyroscopes, ambient light sensors, altimeters, GPS locators, and myriad other types of sensors and to test the boundaries of that input.

As mobile testers have discovered, smart device testers will have to test their devices in the field and to imagine real-world scenarios to test outside of the lab, in the messy world where network connectivity can be spotty, where location detection is imperfect, and where step sizes can vary between short decorative steps in a parkway to the terraced rock leading to Machu Picchu. In addition to vetting the sensors in the smart device, testers will gain insight into the user experience. For example, do the numbers display large enough and bright enough to read when you're jogging at night? Is the display too bright for the user to sleep while wearing the device?

Smart device testers will need to move beyond conceptualizing user scenarios on paper and make more extensive use of test subjects. If one is not a six foot tall, one hundred and eighty pound male, one cannot reasonably simulate this in a test lab.

Testers will have to find and recruit different people to test smart devices, such as:

- ① Dedicated runners to test fitness trackers
- ② Patients to test implanted devices
- ③ Drivers to test automobile monitors



In addition to test subjects, smart device testers will need to coordinate with subject matter experts who know how to implant medical devices as well as other experts in the industries using the smart device. When testing a fitness tracker, testers will need someone to take the test subject's temperature and check the test subject's heart rate. When testing a smart device performing chemical analysis, testers will need to rely on chemists who can synthesize the chemicals to use in testing.

Although automation plays an important role in testing computer applications, it might prove far less effective in testing smart devices. Testers might simulate certain environments or automate some actions, such as running or jogging, the beating of a heart, and the rolling of a tire, but it's far more difficult to 'automate' these physical events than interaction with a Web browser. Additionally, especially when concerning medical devices, government regulating authorities might not approve of automated tests instead of the real thing.

Quality assurance will definitively shift its focus from active testing of the device to a planning and management role, where the tester plans the tests, coordinates the test subjects and subject experts, and analyzes the test results—and to manage the defects and their resolutions.

What Will Remain

Just as certain aspects of testing are changing, so too, are there aspects that will remain the same. In the best of situations, quality assurance teams are already performing detailed planning work as part of the testing process; however, in the smart device world, this becomes absolutely critical. Testers with complicated field testing will need to plan ahead to capture all of their needs before they need them. This includes the designing and developing tests, managing test assets such as test release and test cycles, test requirements, test cases and test scripts, test sets (test cases with common test objectives executed together), and defects. Testers will also have to trace and associate releases and cycles, requirements, test cases and scripts, test executions, and defects. The complexity and variability of the testing make this planning crucial.

What It Means For A Tester

The testing landscape will change for Internet of Things smart devices. The variety of devices will bring flood users (and testers) with options in hardware, operating systems, deployment tools, release tools, and more. Testers who have done Web testing already have a feel for this, as the same Web application in different browsers proved maddening at times. Multiply this complexity by imagining testing every single browser on every single operating system in every form factor in a new world of surging permutations and combinations. Testers will need to adjust to manage testing an overwhelming number of combinations.

Because the smart devices communicate wirelessly and possibly on a limited set of hardware, securing the device will present a new set of challenges for designers and developers--and the task of hunting for holes in the security layers of these devices will challenge testers in new ways.

Conclusion

The Internet of Things and the rise of the smart devices, whether it ultimately proves to be revolutionary or evolutionary, will require some testing practices to evolve as well. Planning and design in the beginning of the project will continue to prove crucial to success. The testing itself will prove more complex and will rely more on testing in the field and the use of test subjects, but this shift will provide fresh and meaningful insights into the smart device under test.

References

- 1: Dr. John Barrett, Head of Academic Studies at the Nimbus Centre for Embedded Systems Research at Cork Institute of Technology (CIT) and Group Director of the Centre's Smart Systems Integration Research Group.
- 2: QAI Worldwide, Essentials of Software Testing CBOOK

Author Bio.

Bob Crews, President and Co-founder of Checkpoint Technologies, is an I.T. consultant and trainer with over twenty-six years of I.T. experience in the complete Software Development Lifecycle including development, requirements management, and software testing. He has consulted and trained for over 280 different organizations in areas such as effectively using automated testing solutions, strategic test planning, implementing automated frameworks, risk analysis, and developing practices which ensure the maximum return-on-investment with automated solutions. Bob has presented at numerous conferences and user groups throughout the world including QAI QUEST, EuroStar (Copenhagen), HP Software Universe, TestCongress (London) and LatinStar (Mexico City).



Bob Crews
President
Checkpoint Technologies

Email: bcrews@checkpointtech.com
Direct: (813) 818-8324 ext. 101

Co Author Bio.

Shailesh Mangal is the technology visionary at Zephyr. He has been developing and testing enterprise and web applications for more than 20 years. With an emphasis on quality, Shailesh has developed applications using Flex, LCDS, AIR, Java, Ruby, Groovy, etc.

His main areas of interest include object-oriented design, system and cloud architectures, data dynamism, real-time search and testing. Shailesh holds a Masters in IT from The University of Texas and a Bachelors in Engineering from India. He has yet to meet a technology he doesn't want to play with!



Shailesh Mangal
CTO & VP of Engineering
Zephyr

Email: shailesh.mangal@getzephyr.com
Phone: 510-400-8656



Checkpoint Technologies, Inc provides expert services and best-of-breed software solutions with a focus in Quality Assurance and software testing. We enable organizations to deliver higher quality applications to market faster. Checkpoint Technologies is Software Quality. Assured. For more information, please visit www.checkpointtech.com.



Zephyr is a leading provider of quality management solutions, powering intelligent DevTestOps for more than 11,000 global customers across 100 countries. Project teams and enterprises of all sizes use Zephyr's products to enable continuous testing throughout their entire software delivery pipeline to release higher quality software, faster. Zephyr is headquartered in San Jose, CA with offices in King of Prussia, PA, Europe and India. For more information, please visit www.getzephyr.com.