UNIT TESTING FOR INTERNET OF THINGS

Alin ZAMFIROIU [1*]
Daniel SAVU [2]
Andrei LEONTE [3]

ABSTRACT

The expansion of Internet of Things and migration to cloud computing, highlights the need for testing from the software engineering phase and each module that is part of them. This comes from the fact that each application, or module, or device is developed by a different provider, making software testing more challenging. We will evaluate the need of unit testing for Internet of Things from the point of view of a tester, especially a QA (quality assurance) tester. In the conclusion we will highlight the need of testing for IoT and further research.

KEYWORDS: *Testing, Unit Testing, Application, Internet of Things, IoT Architecture*

1. INTRODUCTION

Before going further into detail, we will focus on what we should know and understand about the concept of Internet of Things.

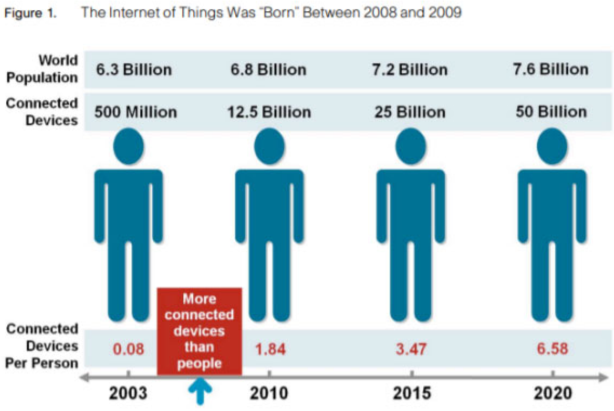
It is important to know the differences between the Internet and WWW (World Wide Web) – those two terms are usually used as synonyms. The Internet is the physical connection between one point and another, while the World Wide Web is the interface that makes the information to flow, being the layer, which is on top the Internet.

While the World Wide Web has continuously growing, the internet has been on a steady path since the beginning, until now, Internet of Things (IoT) is the evolution of the internet, and we are experiencing it right now. People are becoming more proactive than reactive only by having the possibility to be aware of the environment. We evolve as people, by communicating, so this is the next step of the digital world. Devices will evolve also by communication, being a part of the Internet of things.

[1*] corresponding author, senior researcher, National Institute for Research and Development in Informatics, lecturer PhD, The Bucharest University of Economic Studies, Bucharest, zamfiroiu@ici.ro

[2] assistant researcher, National Institute for Research and Development in Informatics, Bucharest, daniel.savu@ici.ro

[3] PhD student, University “Politehnica” of Bucharest, Bucharest, andrei.leonte2006@stud.etti.upb.ro



Source: Cisco IBSG, April 2011

Figure 1. The point in time when more things were connected than people via internet

The term of Internet of Things was first mentioned in 1999 by a member of RFID (Radio Frequency Identification Development) department, but the term has more value and meaning in the present. We can agree with the definition given by the Cisco Internet Business Solutions Group (IBSG), IoT is simply the point in time when more “things or objects” were connected to the Internet than people. [1]

Consider that around us is another world where billions of devices all interconnected over IP (internet protocol) networks. But not only devices and not only electronic objects that have a higher technological development, but also things such as furniture, clothing, trees and even also animals are connected [2], [3].

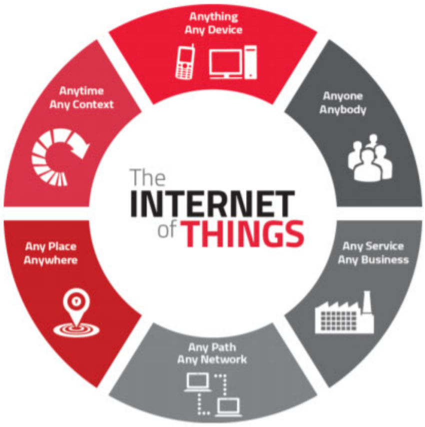


Figure 2. Internet of things

Internet of things is the new era of the internet, the evolution of internet. Objects use the power of internet in order to make them recognizable, being capable of obtaining information and further making context-based decision.

As the IoT evolves, it will unlock new possibilities for us as consumers making our day by day life easier, offering life-enhancing services and also will boost the productivity for enterprises. Addressing our lives as consumers, the connectivity provided by the Internet of Things, will converge into raising life quality by increasing the quality of security, health, education, energy.

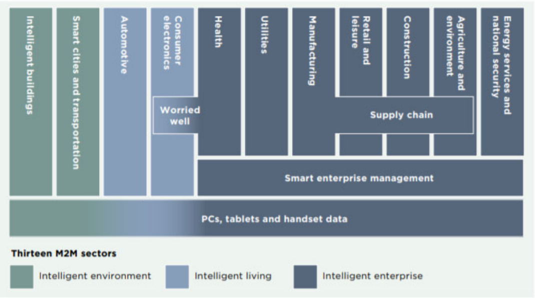


Figure 3. Fields that will be impacted by IoT expansion

The Internet of Things is currently in the spotlight right now. Your devices are now giving you pieces of advice on how to improve your life, your watch is advising you on your fitness and what to eat. Our smartphones can open doors for us and lock them if we forgot, but we have not thought yet of testing each of these systems that we interact with. We should consider ways of testing and analyze methods of testing Internet of Things devices.

Challenges related to IoT environment go beyond devices, hardware and sensor and relate with big data which is the real issue for testers.

The high amount of sensors interactions is making it a challenge for testers in order to be able to create the environment for testing [4]. Even considering that the hardware and protocols are tested by the hardware developers it becomes harder to understand the whole application intelligence [5].

Having a point of view, we will assume the role of tester from the QA (quality assurance) perspective, even though QA eventually means going through the whole stages of testing. We should consider testing IoT devices before they reach maturity, taking into consideration the fact that these devices are transmitting huge amount of data, part of which can be considered as critical. In the IoT world, sensors, devices and applications form an ecosystem, if we apply the testing thinking about the QA convergence of hardware and software, it is not enough, the simple validation of functionality of each system individually is not sufficient in a complex architecture [6], [7].

As we mentioned taking each system individually is still not what we expect to be enough; a working system may not collaborate with all the systems that is interacting with. For example, a shipment tracking device will need to be able to communicate with different back-ends and will need specific algorithms in order to assure the delivery respecting the defined parameters.

2. IOT TESTING AREAS

Cognizant and collaborators [8] defined the following types of tests to be performed within an IoT system:

- *Connectivity testing:* aims at testing the wireless signal to determine what happens when the connection is poor or when there are several devices that are trying to communicate.
- *Security testing*: has the objectives: confidentiality, authorization and authentication.
- *Functionality testing*: validates the correct operation of the IoT system.
- *Performance testing*: validates communication and computing capabilities. Stress testing is done to determine the number of simultaneous connections that can be supported by a particular device.
- *Exploratory testing*: knowns as *user experience tests*.
- *Compatibility testing*: verifies the correct operation with respect to different protocols and configurations.

The important areas for testing the IoT systems are presented in Figure 4.

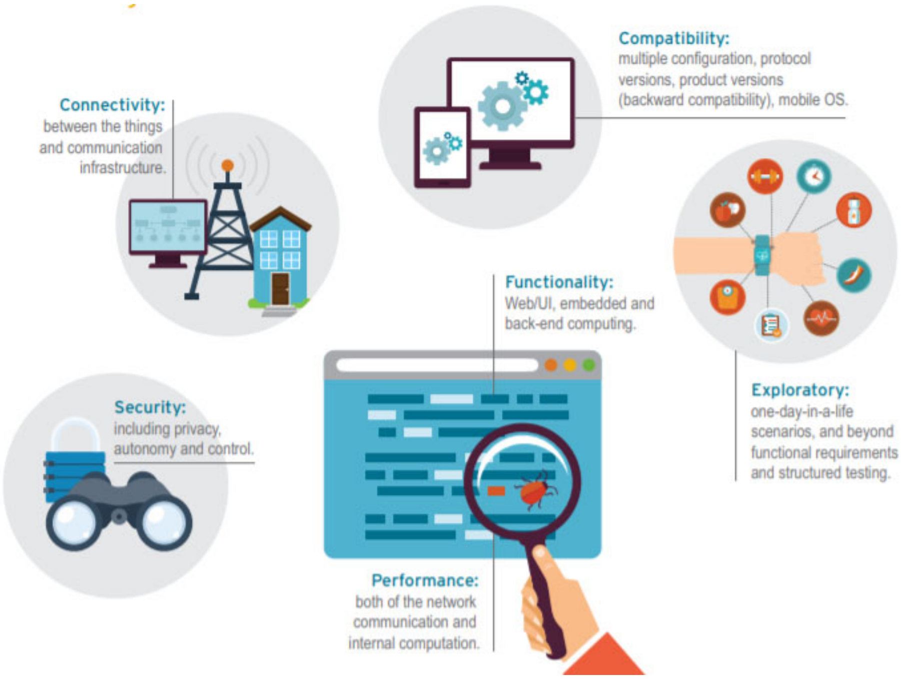

Compatibility:
multiple configuration, protocol versions, product versions (backward compatibility), mobile OS.
Connectivity:
between the things and communication infrastructure.
Functionality:
Web/UI, embedded and back-end computing.
Exploratory:
one-day-in-a-life scenarios, and beyond functional requirements and structured testing.
Security:
including privacy, autonomy and control.
Performance:
both of the network communication and internal computation.

Figure 4. IoT testing areas

There are some reports on IoT testing procedures [8], [9], [10]. Most reports focus on performance testing [11], [12], [13], IoT resource emulation and IoT testbeds deployments [14], [15].

2.1. Connectivity

The main objective of connectivity testing is to ensure the connection between the objects and the communication infrastructure. Seamless connectivity is an element of critical importance in an IoT network. All devices must always be connected to each other and to other systems, such as IoT servers.

The success of an IoT system depends on how well the devices and the hub are connected. Even a split-second connection loss may lead to inaccurate data, which will make the system unusable. Thus:

- the device must be permanently connected to the hub even if the hub is in the sleeping / power saving mode;
- the device must regularly send ping messages to ensure the connection is in place.

As for the information exchanged between users and devices, IoT products implement APIs (Application Program Interfaces). These interfaces are programs responsible for receiving a message and for replying to it through a new message. There are currently several tools available for testing APIs. These tools can simulate a message sent / received by a device, which helps validate information accuracy.

Connectivity testing is done both online and offline. The online test analyzes the connection between devices and applications, data transfer and network security. The offline test analyzes what happens when the network is not available. For vital applications, devices such as pacemakers or health monitors must work continuously, regardless of the state of the network. The device must have the ability to store and process the data collected while offline, and then transmit it when the network connection is restored.

Often the network connection is intermittent or uncertain. Thus, if the connection is unexpectedly lost, it is important for the user to be sure that their data is saved and stored correctly and that they are provided when the connection is restored.

Devices have to communicate with each other. Attention should be paid to the different methods of communication and information exchanged between devices and users. During connectivity tests, the context in which the devices (network type, signal strength, weather conditions, etc.) will be used should be considered and it must be checked whether the device is operating under these conditions.

2.2. Security

Security access is granted to authorized access to protected data and unauthorized access is restricted.

Security testing verifies the security of information, confidentiality and reliability of the system in order to guarantee the quality of the IoT environment. Because the connected devices store delicate information, security testing ensures the correctness of the steps taken to ensure safety and confidentiality.

It is important to validate the user through authentication and to have data privacy checks as part of the security tests.

Security testing covers confidentiality, autonomy, control, and protection against espionage. Appropriate security and penetration tests are essential because weak security measures can lead to loss of sensitive personal information. In the case of IoT devices, in addition to theft of private information, cybercriminals can also attack home security systems or in-vehicle systems to cause accidents.

According to a study by Hewlett Packard, 70% of existing IoT devices are vulnerable to security issues for the following reasons: the lack of data encryption, the lack of minimum password requirements, and unparalleled access to the user interface.

The most common security issues reported include: confidentiality issues, insufficient authorization, lack of encryption of the transport, insecure Web interface, inadequate software protection.

With billions of built-in sensors, it is essential to address data privacy and security issues within the IoT ecosystem. The different types of security testing requirements are: identification and authentication, data protection, data encryption, data storage security in the local and remote cloud. Avoid unauthorized access to devices or information.

Security testing is often ignored because of market pressure on companies to continuously launch new products. Another reason why this type of testing is neglected is due to the lack of understanding of security tests by IoT object manufacturers.

There are two main types of security testing:

- **Static tests:** Perform either manually or through code-examining tools. The objectives of these tests are: analyzing the programming language / code developed for the device, identifying whether the programmer has complied with best coding practices and the code is not breaching security;
- **Dynamic tests:** The device is checked during normal operation. The tools used are looking for authentication problems, simulate hacking attacks, indicate invalid device memory usage, etc.

Security issues cover various security breach scenarios where the IoT device is used as a weak point of access to the network, possible security breaches leading to user harm or violation of confidentiality.

2.3. Functionality

Functionality testing includes end-to-end testing of the IoT ecosystem to ensure that the system generates the desired results and behaviors according to business requirements. Functionality testing targets Web sites, user interface and back-end. The purpose of these tests is to verify the application's functions to see if they meet all the functional requirements.

It analyzes customer requirements and how the consumer wants the output, based on IoT application-specific inputs. Functionality testing is one of the most important methods for any software project and will continue to be extremely important as the Internet of Things expands, requiring powerful test management solutions.

The strategy for functional testing of IoT must start by creating virtual devices that can simulate real-world environments and connectivity. One example is the Nest Home

Simulator Testing Tool that works with Nest products so that it can quickly and easily test system events and sensor status. IoT companies should consider using a virtual environment for functional testing of their products.

There are a considerable number of challenges and obstacles that developers and researchers have to face in the testing of IoT products. Therefore, it is important that they focus on the main components of the products. Similar functionality testing principles apply to both IoT automation products and IoT Web sites. As a first step, basic components that require functional testing need to be identified. The identified components must be tested on both local and mobile devices. The key factor for functional testing of IoT is a real-life simulation environment using real devices or simulators to test these core components.

Modes of functional testing of IoT devices may differ from product to product as the Internet of Things market expands. There are a significant number of challenges and obstacles that manufacturing companies have to face during the testing process. That is why it is important to plan ahead and create the necessary tools to simulate the real-life environments. In addition, innovations are needed to ensure the quality and security of IoT products.

There are cases of negative or positive testing. In case of positive testing, the application is verified based on valid input data. Negative testing is performed to make it clear that the application does not work when invalid input data is provided. When conducting test cases, staff involved should consider aspects such as body movements, voice commands, and sensor usage.

2.4. Performance

Performance testing aims to test the behavior of IoT devices along the network, to test internal capabilities of embedded systems and network communication. Performance testing allows you to evaluate the promptness of a communications network and internal computing capabilities of the embedded software system. It must be checked that data is correctly transmitted and stored, even when an unexpected service interruption occurs.

The primary objective of this type of test is to determine the relationship between the object and the software with which it interacts and to standardize the association between them.

Performance testing validates the hardware and software components of a device based on multiple test cases. It evaluates whether an application can handle the projected increase in user traffic, data volume, frequency of transactions, etc., therefore addressing scalability issues. Compatibility should be validated by analyzing interactions between sensors in order to ensure effectiveness in a real IoT environment.

Evaluators should consider factors such as network bandwidth, latency, packet loss, number of competing users, etc. as these factors influence performance significantly. For example, a physical object may not respond to a user's order.

In order to avoid connection problems that can affect device performance, tests with different types of networks and data streams can be performed. Network activity needs to be analyzed in detail, paying particular attention to the speed of data transfer from one network layer to another. It is also necessary to verify that data is transmitted and stored correctly, even in the event of an unexpected service interruption.

In order to evaluate the overall performance of the IoT application and to validate the response time for different load rates, the code needs to be optimized and various scenarios such as battery discharge, memory reduction, switching between different networks must be followed.

Testing the performance of IoT devices involves the following aspects:

- each authenticated device within the range must be able to connect to the hub;
- the device must be able to send any amount of data to the hub (as required);
- if the data sent by the device exceeds a predefined quantity, data transfer should only be initiated after confirmation received from the hub;
- the device must be able to send data, even in case of power supply problems. These issues need to be resolved as soon as possible.

Performance testing targets three main levels:

- system level: processing, analysis, database;
- application level;
- network and gateway level: testing technologies such as Wi-Fi, Bluetooth, Z-Wave, RFID, NFC, protocols such as HTTP, CoAP, MQTT and other specific IoT protocols.

Tests can estimate the built-in software capacity, the reliability and stability of communications networks under certain operating conditions.

2.5. Exploratory

Exploratory testing is the process of investigating an application through logically-created but ad-hoc tests that allow you to study and understand the applications, features, and operations of an application.

Exploratory testing is conducted in an empirical system where it is possible to study and evaluate in a different way than through predefined test procedures.

Exploratory testing of software included in the Internet of Things combines the logical-cyber world with the physical world.

The success of any software, including those of the Internet of Things, is determined by its users. Even an IoT application that meets all requirements can be a failure if it does not gain the trust of the target audience. In this context, exploratory tests are important because they allow you to determine how an application behaves when it is used by the end user.

Exploratory testing is a test type that is performed from the user's perspective. Considering from this perspective developing software, staff involved in testing can have a better picture of how the application behaves in real terms. This is particularly important because not only the ability of IoT applications to communicate and interact with a wide range of devices can be evaluated, but also their ability to significantly improve end-user living conditions.

Internet of Things devices can be affected by new types of software errors that require new testing approaches. Staff involved in testing activities need to know and use new exploratory testing models to tackle both physical and cyber worlds.

Exploratory tests help developers learn more about the functionality of a program and find out if the requirements were correct and fully understood.

2.6. Compatibility

It is an essential step in IoT testing that evaluates the interaction between IoT software and various intelligent devices, platforms, network layers and operating systems. It aims to guarantee the scalability and security of data exchanges and ensure the compatibility of communications protocols.

Considering the complexity of the IoT system architecture, compatibility testing appears to be a pressing necessity. This type of testing must always be done in the real world and not in the virtual environment.

Because IoT systems include a multitude of devices, sensors, protocols and platforms that are constantly updated, the number of possible combinations is extremely high. There are a lot of devices that can be connected via IoT systems. These devices have a very diverse software and hardware configuration. Therefore, it is essential that compatibility testing involves a comprehensive test matrix to cope with the complex architecture of Internet of Things.

At a technical level there is a wide range of quality attributes: compatibility, installation and use of resources. These must be checked to provide objective test results to the client.

Also, by testing compatibility, the way the different devices communicate with each other and with the digital environment is evaluated. Various validation options, such as hardware and encryption compatibility checking, and compliance with security standards from device level to network layer are being performed within this type of testing.

Testing features such as: operating system versions, browser types and versions, generations of devices, communication (e.g., Bluetooth).

Testing compatibility verifies whether the functions are working correctly in different configurations, combinations of device versions, protocols versions, mobile devices, and mobile operating system versions.

3. UNIT TESTING

You may not hear about testing in the IT field, but the only reason you may not know what is testing is not being part of software engineering industry. It may not be given the importance that it deserves, but its purpose of checking if a piece of software corresponds with what it should do on the paper remains very important in the development process. It is basically the acceptance which determines that software is working at the quality level required.

Testing may be seen from different perspectives. Functional testing validates that the main functions fulfill the requirements; system testing validates the most common usage paths before releasing; performance testing is approaching a different angle, by testing the system under certain conditions of load and stress.

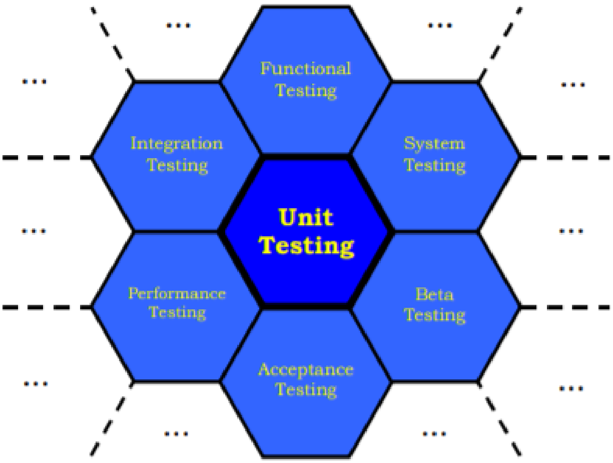


Figure 5. Unit testing is in the center of all testing techniques

Unit testing, on the other hand, is testing that each part, even the smallest unit will correspond with the defined documentation and APIs. Having the certainty that all it is running and correspond after the unit testing, we can go further and validate the software. Taking into consideration this information we can state the unit testing is the basis for any other forms of testing.

As is unit testing is actually testing an application, a function or a class. Usually unit testing is automated

From our perspective when we talk about IoT devices, unit testing should start at the level of interconnecting of different objects / devices.

4. UNIT TESTING FOR IOT SYSTEMS

If we consider an IoT architecture, by applying the principles of testing, and considering its steps, unit testing represents the device testing but from different perspectives such as: performance, security, compatibility and usability of each component [15], [16].

To be easier to understand unit testing in the IoT scenario we will split the environment into two main categories, the device interaction layer and the user interaction layer.

From the perspective of device interaction there are some required elements that need to be present in testing:

Interoperability. Devices must have the possibility to work among other devices, other developments and implementations.

Standards. Devices and sensor must respect the standard and be validated that they are working conform to the established standard for their category regarding communications protocols and quality. This is the point where the devices need to be tested before putting them on the market.

Security: It has become a sensible matter of speaking. Security must be assured on the regarding data protection, authentication, data encryption, cloud storage and back-up [17],[18]

The user interaction layer is where the device and the end-user communicate. This may vary on the level of experience of the user. Test areas may be:

Network capability: Devices must be able to communicate, so testing different network modes is a must. Also, it is necessary that the communication methods are able to fulfill the user needs and be aware of the energy consumption as well.

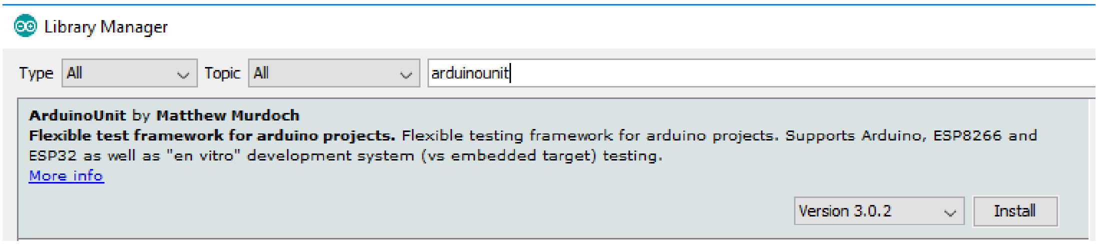
Usability: Of course, that the user knowledge influences the level of usage of a device, but the response of a device must be in certain parameters in order to offer the end user the capability to understand the interaction between different machines / devices.

Services and back-end development: User interface and interaction may be the key of a system to work properly, but the IoT system as a whole must have a complex analytical engine to ensure the user experience at higher level [19], [20], [21].

When we talk about IoT unit testing these should be the areas that a tester should cover for each device in the ecosystem. The software engineering it will always be covered by the manufacturers, so the 'unit' from the IoT perspectives is the device / object itself that should be tested, but in the corresponding architecture that will be part of.

If we have an system based on sensors with Arduino board, we can use ArduinoUnit to develop unit tests [22].

To use this framework it is necessary to have Arduino IDE and from Menu->Sketch->Include Library->Manage Libraries to install ArduinoUnit, Figure 6.



ArduinoUnit by **Matthew Murdoch**
Flexible test framework for arduino projects. Flexible testing framework for arduino projects. Supports Arduino, ESP8266 and ESP32 as well as "en vitro" development system (vs embedded target) testing.

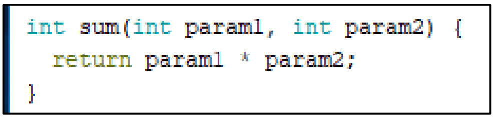
Figure 6. Install ArduinoUnit library

To use in a program the Arduino Unit we have to import the h file:

```
#include<ArduinoUnit.h>
```

And after to create test for our methods.

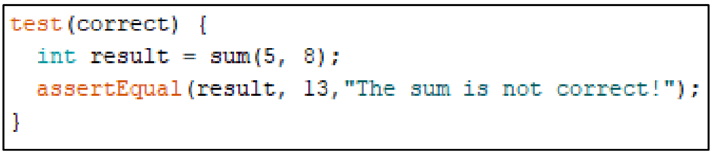
We suppose that we have a method that calculate the sum of two numbers, Figure 7.



```
int sum(int param1, int param2) {
  return param1 * param2;
}
```

Figure 7. The tested method

To create the right test, from the Right-BICEP principle, for this method we implement the test from the Figure 8.



```
test(correct) {
  int result = sum(5, 8);
  assertEqual(result, 13,"The sum is not correct!");
}
```

Figure 8. The Right test for the sum() method

In the loop method what we have to do to run the implemented tests is to call the run method from the test:

```
Test::run();
```

5. CONCLUSION

A basic IoT system will borrow some testing techniques from the software engineering field in order to validate the applications. For an IoT ecosystem the unit testing it will be linked to testing each device functionality regarding its ecosystem that will be supposed to work within. Consider this the focus of testers will have to be on the following:

- performance testing regarding network capabilities, ensuring the level of communication;
- security level;
- intercompatibility;
- exploratory testing for the user experience.

The progress made by the Internet of Things in last couple of years and the investments made in infrastructure speak for the fact that this is the future of the internet. As the functionalities and multiple domain usage of the IoT in the consumer and even enterprise markets, should gear up the QA and tester teams and keep up with the digitization. The skill and training on the testing part will make a huge importance on how the IoT is expanding its world of interconnectable devices. Blending the Internet of things into enterprises will make the tester to have more developed skill in order to overtake traditional functional testing and think about integration testing of software, big data and the components of IoT ecosystems.

Taking into consideration all of the above and the fact the testing for the Internet of Things is still limited, there are some areas that should be taken into consideration when we talk about IoT testing.

- interoperability testing;
- testing the Internet of Things ecosystem under limited connection;
- techniques for standardization of platforms and possibility of configuration;

Another important step that should be explored will be automation of integration testing, but is hardly to be possible until IoT will reach its maturity.

ACKNOWLEDGEMENT

This paper presents results obtained within the PN-III-P1-1.2-PCCDI-2017-0272 ATLAS project ("Hub inovativ pentru tehnologii avansate de securitate cibernetică / Innovative Hub for Advanced Cyber Security Technologies "), financed by UEFISCDI through the PN III –"Dezvoltarea sistemului national de cercetare-dezvoltare", PN-III-P1-1.2-PCCDI-2017-1 program.

REFERENCES

[1] Keyur K Patel, Sunil M Patel, Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges, 2016

[2] GSMA, URL: https://www.gsma.com/iot/wp-content/uploads/2014/08/cl_iot_wp_07_14.pdf, 2014

[3] Dave Evans, The Internet of Things How the Next Evolution of the Internet Is Changing Everything, April 2011

[4] URL: https://blog.testlodge.com/testing-the-internet-of-things/, 2017

[5] Patrick Kua, Unit Testing, URL: https://www.thekua.com/publications/AppsUnitTesting.pdf

[6] URL: https://www.360logica.com/blog/internet-things-iot-testing-challenges-considerations/

[7] Kelly Hill, Testing the internet of things: making the IoT work

[8] Cognizant (2016). The internet of things: Qa unleashed. https://www.cognizant.com/InsightsWhitepapers/the-internet-of-things-qa-unleashed-codex1233.pdf, 2016.

[9] Bloem, J. (2016). IoTMap - Testing in an IoT environment. Sogeti Publisher.

[10] RCR-Wireless (2016). Testing the internet of things: Making the iot work. https://www.2j-antennae.com/files/1479994838.pdf

[11] Lunardi, W. T., de Matos, E., Tiburski, R., Amaral, L. A., Marczak, S., and Hessel, F. Context-based search engine for industrial IoT: Discovery, search, selection, and usage of devices. In 2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA), pages 1–8, 2015.

[12] Thangavel, D., Ma, X., Valera, A., Tan, H. X., and Tan, C. K. Y. Performance evaluation of mqtt and coap via a common middleware. In Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on, pages 1–6, 2014.

[13] Vandikas, K. and Tsiatsis, V. Performance evaluation of an iot platform. Proceedings - 2014 8th International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST 2014, pages 141–146, 2014.

[14] Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., and Watteyne, T. FIT IoT-LAB: A large scale open experimental IoT testbed. In IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings, pages 459–464, 2016.

[15] Sanchez, L., Munoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., and Pfisterer, D. SmartSantander: IoT experimentation over a smart city testbed. Computer Networks, 61:217–238, 2014.

[16] Esquiagola, J., Costa, L., Calcina, P., Fedrecheski, G. and Zuffo, M. Performance Testing of an Internet of Things Platform. In Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017), 309-314, ISBN: 978-989-758-245-5, 2017.

[17] Kiruthika, J., Khaddaj, S. Software Quality Issues and Challenges of Internet of Things. In 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), IEEE, 176-179, 2015.

[18] Marinissen, E. J., Zorian, Y., Konijnenburg, M., Huang, C. T., Hsieh, P. H., Cockburn, P. and Verbauwhede, I. Iot: Source of test challenges. In 2016 21th IEEE Europe an Test Symposium (ETS) , IEEE, 1-10, 2016.

[19] Xu, T., Wendt, J. B. and Potkonjak, M. Security of IoT systems: Design challenges and opportunities. In Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design, IEEE, 417-423, 2014.

[20] Sicari, S., Rizzardi, A., Grieco, L. A. and Coen - Porisini, A. Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, 146-164, 2015.

[21] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376, 2015.

[24] https://github.com/mmurdoch/arduinounit