





Software Testing Levels in Internet of Things (IoT) Architecture

Teik-Boon Tan^(✉)  and Wai-Khuen Cheng^(✉) 

Universiti Tunku Abdul Rahman, Kampar, Malaysia
{tantb, chengwk}@utar.edu.my

Abstract. Testing the Internet of Things (IoT) solution is complex as it involves a diversification of implementation of smart objects that adopt a diverse and complex communication protocols. It is doubtful whether tests done in IoT solution have been adequately sufficient and scalable. This paper proposed a mapping of the IoT architecture to the conventional software test levels. The test levels shall provide a better view for tester to conduct tests based on different focus of the level.

Keywords: Internet of Things · Testing · Test levels · Challenges of IoT tests

1 Introduction

The applications of Internet of Things (IoT) with the Big Data trend is seen to be the trend for the next few years as more and smart objects are entering into the market with the expectation of 25 to 50 billions connected devices by the year 2025 [1]. One can say that IoT shall be providing huge potential for growth in development of applications from various fields. It could be for smart cities, smart farming, health tracking and management. Generally, IoT comprises of growing number of heterogeneous smart objects that are interconnected for data exchange using different protocols over the Internet [2]. The primary concept is to connect every “Things” to the Internet which can be implemented infrastructure of sensors that are deployed in buildings, external areas and carry within a body. The scales IoT implementation can be either very small or big scales. The diversification of IoT platform (implemented everywhere and different scales) has given rise on how testers are able to test sufficiently in such environment. At presence, testers tend to focus only on the local devices or the edges as the main test objects which can be regarded as insufficient to cover the communication of devices in large scale IoT solutions [3]. This paper outlines how the traditional software test levels can be adopted to provide views to test level in the IoT architecture. The representation shall be useful for tester to focus on the relevant aspect of the IoT implementations.

2 Background

Arguably IoT seems to be similar to the normal software in terms of functionality especially when functionality tests have to be conducted. IoT implementation however poses a problem of isolation of tests as there is a need to combine different hardware, software and network diversification. Network of IoT platform can include different protocols for data transmission and data exchange [4]. Testing is challenging in a large scale IoT implementation there shall be high amount of collaborative devices with requirements of complexed coordination due to the size [5]. Hence, there is need for proper levels of view of IoT platform in testing.

3 IoT Architecture

In this section, the general architecture of IoT is discussed. The architecture is a generic IoT architecture that have three main components namely Devices, Gateways and Communications and Application component [6]. Devices consist of the hardware that operates at variety of environment that can include indoor, outdoor, underwater and wearables. Most of these devices are tightly coupled to the platform implementation [7].

At middle layer shall be the gateways where the gateways shall be centre point of communication amongst the devices. The two common approaches of IoT communication shall be Constrained Application Protocol (CoAP) [8] and Message Queue Telemetry Transport (MQTT) [9]. The exchange of data shall involve communication between the sensors, sensors to gateways and gateways to gateways.

Application layer is actually concerning how to provide useful and accurate service to the ultimate users of the IoT system. The main focus shall be the data. The data shall be refined and further processed in accordance to the needs of the application. Data can be transferred to a Cloud Platform for further data analytic [10].

4 Software Test Levels

Software testing contains both verification (all specified requirements have been fulfilled) and validation processes (all the outcomes or test objects is complete). The processes is used in software testing for detecting bugs, for ensuring quality of products fulfilled and ensuring the application is for fit for its purpose [11].

Conventionally, the broad tests of software can be categorized into functional and non-functional tests. Software tests are further categorized to test levels to have better definition of software tests boundaries and test purposes. The tests levels are unit test, integration test, system test and acceptance test (see Fig. 1).

Unit is the component identifiable or testable parts of the program. It is the starting part of tests to be conducted in software testing involving codes. It is a technical test of the components therefore it is conducted by the developer of the unit. Unit testing is test to be done in isolation of other units. Isolation is achievable using the skeletal implementation of other related component using stubs and drivers.

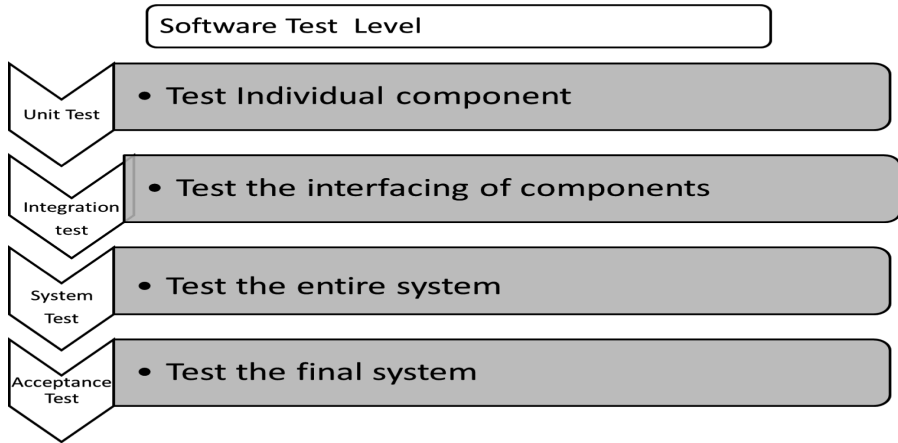


Fig. 1. Test levels of software.

Integration test is to test the interfacing part of the units or components of the software. The units are joint or integrated part by part to test whether the integration contains any defects or not. The tester needs to decide on how these components are integrated. The approaches can be big-bang integration, bottom-up and top-down.

System test is to test the system as a whole in the similar environment of the actual system. The software is tested in accordance to the needs of software requirement specifications and software design specification. Tests shall be conducted at the development side from development perspective to ensure the software fulfils requirements defined. Standard tests are on the usage of the system in actual environment. Business processes and use cases shall be used as the basis for tests. Some non-functional requirements can be tested in this level.

Acceptance test is the validation process as to decide whether the software can be accepted by the user or client. Acceptance testing is primarily to establish confidence in the product and evaluate whether the software is fit to be used. It is from the perspective of the users. The tests therefore involved different angle of the software. It can be on contractual, legal, user acceptance, testing the software in multiple configurations and operational [12].

5 Test Levels for IoT Architecture and Challenges

IoT architecture as a matter fact can have similar software test levels.

With reference to the Fig. 2, IoT architecture can be broken into level similar to the software test levels. The same levels are applied but with different test items and test techniques to be employed as IoT is a type of solution involved multiple smart objects with diverse communication requirements and protocols.

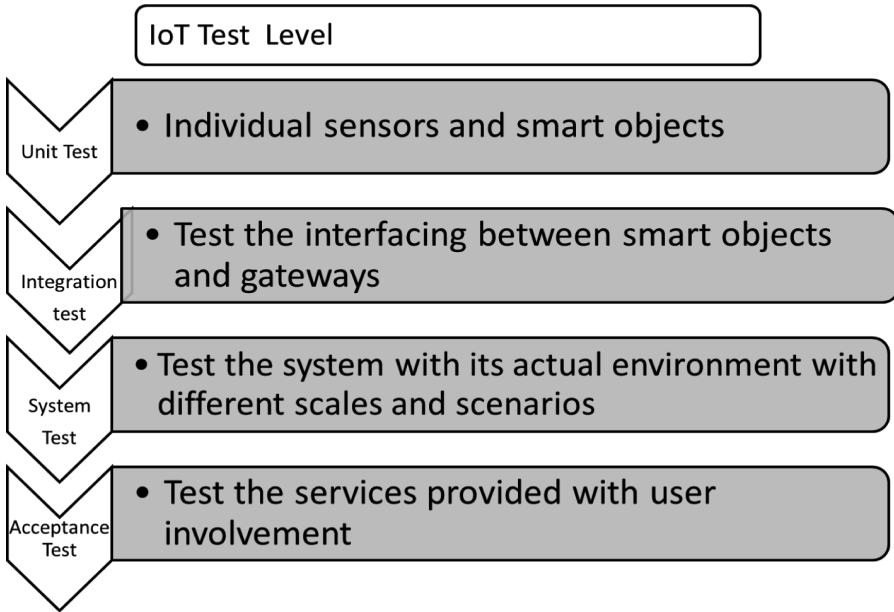


Fig. 2. Test levels of IoT.

5.1 Unit Test Level

At the component level, the identifiable units of IoT platform are actually the smart objects, sensors and gateways. In following the standard practice of Unit testing, each object in the IoT architecture is tested individually to ensure there are no defects or faults occurrence when tested. Units that are related to the unit under test needs to be replaced with skeletal units (with stubs or drivers) just provide feedbacks on data. Testers need to test whether sensors and smart objects are able to make measurement and able to exchange data. In addition, gateways are to be tested for data collections and aggregation.

Unit testing in IoT poses a challenge to tester as these units are constrained devices [7, 13] or they have limited resources. These devices are basically constrained low powered feature (sensors need to operate a few years), at times limited connectivity availability, low storage and hardware dependent.

5.2 Integration Test Level

In integration test tester needs to start joining or integrate component of things and gateway to test the interfaces induce any fault when integrated. The IoT platform consists of various networks devices and communication protocols [14]. The integration technique can be similar to the software integration test technique like top down or bottom up. Integration tests ought to cover failure in communication from the sensors, interfacing different types of sensors, capability of gateway to handle communication from multiple sensors and capability of gateways to handle different types of communication protocols (MQTT, CoAP, etc.).

Integration test poses challenges of combination of devices and network structure in testing. Tester needs to either simulate the environment for integration to be done or provide actual environment (test bed) for testing [14].

5.3 System Test Level

System test level means that the IoT solution to be tested as a whole system. It is tests to be executed before the actual deployment at the actual site. The system needs to be very similar or the same as the real implementation. The test is to be done by tester from the perspective of developer.

Due to the nature of heterogeneous nature of IoT and the complexity it is appropriate to test or simulate the IoT application before deployment. However, tests conducted are small scales and tend to be localized with only partial integration of devices and gateways [5], Preparing environment that can be used in modelling and testing IoT applications is hard due especially in the large scale IoT scenarios. Tester can only rely on good simulation tools and techniques for scalability.

Tester can rely on IoT simulation tools for preparing the tests before deployment. However not all simulators are suitable to model all supported architecture and components like sensors and gateways. Testbeds environment provided is expensive, hard to use and hard to maintain where there is change of setting [15].

5.4 Acceptance Test Level

Acceptance Test is on whether the service provided by system is acceptable to ultimate users. One of the test that is suitable for IoT platform is user acceptance test. User shall determine the IoT solution is able to provide useful and fitting service to them.

There are large quantities of data in IoT platform as multiple sensors are deployed to get measurement of things at the deployed sites. Acceptance test in IoT shall be to test the data intensive functionalities of the system. Acceptance testing therefore shall cover data profile development from aggregation of data, analysis of data and application to domains.

The challenge of acceptance test is on quality of service of data transmission and important metrics to be used for analytics [16]. Tester needs to ensure quality of connections and measurements as sensor nodes can be dispersed and ever changing in behavior.

6 Conclusion

The key of good testing is always test in isolation. Isolation is able to help tester to trace and identify which device, component, interface or sub-part causes failures in the system. The software test level from unit test, integration test, system test to acceptance test, separate the focus of part of the software that needs to be tested. Test level approach can be applicable to IoT architecture as well. Testers however need to be aware what shall be tested in each level of the IoT test level. In the unit test level the focus is on individual functionality of the sensors, smart objects and gateways. Secondly in the integration test

level the primary concern is to test integration the IoT devices part by part and it is more a localize test of sub-components. The communication and exchanges of data are the main tests to be conducted. Thirdly, the system test is on the testing the entire IoT solution. Unlike the integration test, the system test requirement requires a more complex test that could involve large scale testing. Simulation tools can be adopted to simulate the environment. The fourth level is the acceptance test in which a user centric approach shall be adopted. The ultimate aim is to ensure the user is able to use and apply the data collected to a particular domain. The proposed test level is to allow tester to determine what to be tested in IoT and there are still challenges to be solved.

References

1. Evans, D.: The Internet of Things - how the next evolution of the internet is changing everything, CISCO white paper (2011)
2. Rosenkranz, P., Wählisch, M., Baccelli, E., Ortmann, L.: A distributed test system architecture for open-source IoT software. In: Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems - IoT-Sys 2015 (2015)
3. Hagar, J.D.: Software test architectures and advanced support environments for IoT. In: IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Vasteras, pp. 252–256 (2018)
4. Gardašević, G., et al.: The IoT architectural framework, design issues and application domains. *Wirel. Pers. Commun.* **92**, 127–148 (2017)
5. Kim, H., et al.: IoT-TaaS: towards a prospective IoT testing framework. *IEEE Access* **6**, 15480–15493 (2018)
6. Ojie, E., Pereira, E.: Exploring dependability issues in IoT applications. In: Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing (2017)
7. Marinissen, E.J., et al.: IoT: source of test challenges. In: 21th IEEE European Test Symposium (ETS), Amsterdam, pp. 1–10 (2016)
8. Constrained Application Protocol. <http://coap.technology>
9. Message Queue Transport. <http://mqtt.org/>
10. Kanstrén, T., Mäkelä, J., Karhula, P.: Architectures and experiences in testing IoT communications. In: IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Vasteras, pp. 98–103 (2018)
11. Myers, G.: *The Art of Software Testing*, 2nd edn. (2004)
12. Bourque, P., Fairley, R.E.: *Guide to the Software Engineering - Body of Knowledge* (2014)
13. Bormann, A.K.C., Ersue, M.: Terminology for constrained node networks. Internet Engineering Task Force (IETF), Informational 2070-1721 (2014)
14. D'Angelo, G., Ferretti, S., Ghini, V.: Simulation of the Internet of Things. In: International Conference on High Performance Computing & Simulation (HPCS), Innsbruck, pp. 1–8 (2016)
15. Rosenkranz, P., Wählisch, M., Baccelli, E., Ortmann, L.: A distributed test system architecture for open-source IoT software. In: Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems, pp. 43–48 (2015)
16. Kanstrén, T.: Experiences in testing and analysing data intensive systems. In: IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Prague, pp. 589–590 (2017)