Presenting a Method to Test the Designed Software with a Service-Oriented Approach Using International Software Testing Qualifications Board (ISTQB)

Ibrahim Azhdaripour  
MS student, Department of Computer Engineering, Islamic Azad University of Marvdasht  
Azhdari.ebrahim@gmail.com

Dr. Reza Javidan  
Associate Professor, Faculty of Information and Technology Engineering, Shiraz University of Technology, javidan@sutech.ac.ir

Abstract

Service-oriented architecture is a powerful architecture in the software engineering process that has many applications. Creating independence in operations; reusing these services increased the expansion capabilities and scalability of produced Software with service-oriented approach, and it is raised as dominant architectural design software. However, the lack of a comprehensive test structure for such a great architect was a disadvantage that caused problems in produced software with this architecture. In this research, a comprehensive plan is proposed to test the produced software with service-oriented approach using ISTQB test framework, which covers some shortcomings in this area. Removing some of unnecessary tests and adding some required tests for this method created a comprehensive framework for software testing with a service-oriented approach. By using this framework for testing GIS-based software, it can be concluded that applying this test framework can have a significant impact on increasing the quality of the software and satisfying the user and customer.

Keywords: service-oriented architecture, ISTQB test, GIS-based software.
1. Introduction
Many methods have been introduced for testing and developing software in software engineering processes that each has their own characteristics. Despite the recovery trend of software engineering methods and evolution of this process, there is no perfect way that can cover all needs in this field. However, some of these methods are welcomed more by those involved in software design and development process due to the relatively good performance and ease of use. One of the most popular methods in the field is service-oriented architecture.
Service-oriented architecture is welcomed by organizations and companies by separating software units and creating independent operating collections -in a reusable form- with the name of service instead of duplicate and similar applications. This architecture insists on loose coupling, interoperability, combining, and reusability of services as building blocks, and it can reduce the total information technology costs of organizations to an acceptable level.
However, this architecture has weaknesses like other architectures. One of these weaknesses in service-oriented architecture is that the test section is not considered as an important process. Although, some solutions are presented in this field, but none of them completely covers the service-oriented architecture. In this research, a comprehensive process will be provided for testing the software with a service-oriented approach based on the establishment of ISTQB framework.

2. Research literature and theory
Software test subject was raised to solve software problems. There are several models for this word, but none of them became international. Then, the idea of ISTQB was raised in 2002. ISTQB is a comprehensive and international standard framework for software testing. This test has essentially created a standardized way to measure the quality of software quality.

2.1. ISTQB objectives
The objectives of this test are summed in two overall cases, although each one can be expressed in more specific and more complete cases. The objectives are as follows:
1. Training tester and granting international certification in all valid countries (training)
2. Developing the software test content regularly and providing it to the world. This means that presenting methods and test techniques (research) is the ISTQB Certificate in three levels, which are the prerequisite for each other depending on the degree:
   1. Level of basics (3 days), it has one test.
   2. Advanced Course (15 days, 3 sub-course in 5 days: test management, test analysis and technical analysis test), it has three tests.
   3. Expertise course: a series of works must be produced and presented in an important area.

3. Service-oriented architecture testing method using ISTQB test framework
In this section, some details are described on service-oriented architecture test using ISTQB framework. In fact, to test the designed software with service-oriented approach, some changes need to be created in the context of service-oriented architecture using the ISTQB test framework.
to cover the needs of service-oriented architecture. For this reason, it is required to classify various tests in software to determine which of test is important for service-oriented architecture and should be included in the ISTQB test framework or which of them can be omitted, and which of them can be the alternative of other tests.

Here, it is assumed that the services, which are designed and built by service-oriented architecture, are not vital services that often are like this.

In the software world, two tests are generally conceivable: functional and non-functional testing. In functional test, different parts of the software and its modules are tested in terms of structure and functioning to do its task properly. However, in non-functional test, the overall performance of the software is examined from different directions. Functional testing is done in 3 steps and then, non-functional test is done. Functional testing includes 3 stages of unit testing, integration testing, and system testing. Non-functional test generally includes acceptability test.

Unit testing: testing small components of the software such as modules and functions
Integration testing: Testing some software units with each other so that the correctness of these modules and their proper function are considered in relation to each other
System testing: Testing the whole system comprehensively to determine whether it corresponded to the system functional requirements
Acceptability testing: Testing a system to determine how to achieve the performance requirements.

Functional tests, which are known as accuracy test in the framework of ISTQB test, are the important component each software test for all types of architectures. Therefore, there must be the functional section as the accuracy in the framework of the ISTQB test. There are a series of non-functional tests in ISTQB test framework, which should be removed or replaced for more compatibility with service-oriented architecture. In addition, some tests should also be added.

Non-functional testing includes the following:
Reliability test, portability test, resource usage test, usability test, ability test, exchangeability test, accessibility test, collaborative test, service quality violation test, regression test, test mode Production, test Based on the defect, test Based on the model, compatibility test, performance test, assessment of the authenticity and validity, Security test, safety test, maintenance ability test, risk-based test, conditions test, route test, and user interface test

Some non-functional tests that were expressed for greater coordination with service-oriented architecture should be removed, which will be referred below with their reasons.

- Reliability test: since, this test is more relevant to critical systems; it is not required in the service-oriented architecture.
- Ability test: In service-oriented architecture, this test is not considered as an important test, and this test is in other tests in an implicit form.
- Exchangeability test: since, this test will be very similar to the portability test, especially in service-oriented architecture, mentioning and doing one of these two in ISTQB test is sufficient.
- Accessibility test: performing this test in ISTQB test framework for testing on service-oriented architecture can be optional. Although, accessibility and simplicity will be
important for users, but since most software produced by a service-oriented approach are applicable in Web, and especially, Web services are one of the main cases in this architecture, accessibility in this architecture is implied. Therefore, mentioning and doing it seems to be necessary

- Testing based on the model: Although this test has a relatively high degree of importance, but it is not necessary to mention and to perform. In cases where the test is considered automatically for the purpose of statistical analysis, the test becomes more efficient.
- Safety tests: Services are usually not critical services. Thus, safety is not important in a service-oriented architecture.
- Risk Based Testing: As mentioned in terms of safety, it is not important due to not being critical.

3.1. Adding required tests to ISTQB test framework to better support the service-oriented architecture

In addition to changes in the previous section about ISTQB test framework (For more compatibility with service-oriented architecture), it is better to add some tests to achieve better results in service-oriented architecture.

One of the cases, which needs to be added in the service-oriented architecture test is SLA\(^1\) service level agreement test, in which the level of required service by the user that is mentioned in the contract be ensured. There may be circumstances, in which the service performance has a defect and it is not acceptable. These circumstances should be determined by SLA test. This test informs the service provider before releasing and it performs the necessary forecasting to deal with it. Violation of service may occur due to a combination of factors and conditions, including inputs, combining real and abstract services, network configuration and server load.

Another case that could prepare the ISTQB test framework for compliance with service-oriented architecture is service-based testing. In fact, each service is tested separately in terms of performance that makes this test to ensure the performance requirements of each service. Finally, if a service has a distance from the required specifications and expected output, then, applying it will be revised and decisions will be made based on that situation and the information that is available from that service.

The next case is using UDDI\(^2\) to send reports related to their testing and storage and to reuse the relevant results when needed. In this structure, a report will be sent to UDDI service registry after testing each service. This can help to increase the ability to manage remote testing. In addition, it creates concentration management on the test, which will improve the test process in the service-oriented architecture.

4. Implementation

The present section provides some reports on applying the proposed methods. This analysis is done on GIS-based software that supports service-oriented architecture for the development of

\(^1\) Service level agreement
\(^2\) Universal description, discovery and integration
geographic information system software and detailed maps of different places, towns, villages, nature, and in general, geography of the country.

Using powerful test frameworks such as ISTQB with the proposed reforms makes it possible for services to be implemented with high confidence, and each one has a high accuracy in the exchange with other services.

In this section, we perform functional and non-functional tests, which were discussed in the previous section on GIS-based software.

4.1. Functional tests

1. Unit testing: In this test, we need to have service implementation details. These details are not available or are difficult to reuse in the service. If the performance of a service is already approved and it does not require internal changes, thus unit testing is not necessary.

To perform this test on a GIS-based software, initially we identify the software functionality and consider each functionality as a service. Although in practice there may be some changes, but this impression is true in most cases. Some features may require several simultaneous service or some other in the common form of a service. Then, we perform the unit testing (if necessary) on each service. If the service does not have the desired efficiency, its reusing will be revised. Some of the identified capabilities in this software are as follows.

- Ability to view location and characteristics of all equipment related to the situation,
- Possibility of switching maneuver and displaying the results on the map,
- Using tracking tool in distribution network,
- Ability to view and record the history of various events such as maneuvers,
- Possibility of adding layers to the system design,
- Possibility of simple and advanced search,
- Possibility of recording daily information in distribution networks,
- Possibility of providing various reports,
- Possibility of providing statistical reports with graphs, map scaling, the ability of output in the form of file, map navigation tools and etc.

2. Integration testing: This test is necessary because the service must specify how they interact with each other. This test may waste much time because the number of services and evaluating the coordination of services in connection with other services is relatively long and time-consuming. To perform this test, initially service communications should be identified and services that are directly connected are tested to ensure about the integrity of the communication in services. For example, the relationship between two search services on the map and service of applying changes on the map must be recognized. In the case of combining these 2 services (for example, next search service) integration testing is done. Because, in the case of changing the point information of map, these changes be applied in next searches.

3. System testing: some flaws and defects may occur in the software system without completion of this test, and this defect may not be known previously. In addition, since some of these services do not have a direct contact with each other, this test is necessary. To perform this test, the fields that the software is made with this purpose must be recognized, and different modes of the test must be produced. For example, one of the fields of producing the studied software is the ability to link with the analysis software.
such as CYME and DLGSILENT. To perform system testing, we should link the studied software to the analyst software. Then, we should send some information from the studied software to the analyst software. The conducted analysis must be consistent with the sent information to ensure the accuracy of the link that has been established.
| Test Tool | Static techniques | Dynamic techniques | Non-functional testing | Testing design and planning | Test run | Testing control and planning | SLA testing | Return testing | Non-basic requirements testing | Service testing after SLA changes | Evaluation of the implementation of service in line with the organization's policies | User acceptance testing | Security testing | Performance testing | Service composition testing | Service relation testing | Component testing | Total Points |
|-----------|-------------------|-------------------|-----------------------|---------------------------|---------|---------------------------|------------|-----------------|-----------------------------|-------------------------------|----------------------------------|---------------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|
| Total     | 14                | -4                | 27                    | 0                         | 0       | 0                         | 1          | 1               | 1                           | 1                             | 0                                | 1                   | 1               | 1               | 1              | 1              | 1              | 1              | 1              |
| January   | -1                | 0                 | 1                     | 0                         | 0       | 0                         | 0          | 0               | 0                           | 0                             | 0                                | 0                   | 0               | 0               | 0              | 0              | 0              | 0              | 0              |
| 2016      | 1                 | 1                 | 1                     | 1                         | 1       | 1                         | 1          | 1               | 1                           | 1                             | 1                                | 1                   | 1               | 1               | 1              | 1              | 1              | 1              | 1              |

4.2. Non-functional tests

These tests are very important. Because works that are done and all software that are written are for users and users' satisfaction is the most important software engineering issue. Based on this test, we will be sure that the software that has been produced could act reasonably to be accepted by the user. By conducting some of non-functional tests on the studied software, we conclude that the studied software should have the following functionality to be accepted by the user.
Simplicity of using the system, easy access to features required by users, lack of user fatigue from working with the software, strength in terms of graphics, simplicity of working with it, quick learning of the software, lack of consecutive errors in the system, readability of reports, good quality and good coloring of maps, advanced searches with high precision and etc.

5. Evaluation of Results

In this section, some of the implementation results of the proposed method and its comparing with other proposed methods (which are designed in the field of designed tests with service-oriented approach) are mentioned. By developing the STQB test framework, we reached a comprehensive framework for service-oriented architecture. To assess the provided solutions on service-oriented architecture, it is enough to see how much each of them have covered areas of the framework. To do this work, we compare 9 provided papers in the field of service-oriented architecture in the form of the proposed framework. For this purpose, we consider three sets of ratings for each of test fields as follows:

Score -1 is for the case that the provided solution has not considered that test field.
Score 0 is for the case that the provided solution has considered that test field incompletely.
Score 1 is for the case that the provided solution has that test field completely.

Table 1. Comparison of service-oriented architecture test papers in the form of the proposed framework

As seen in the table above, the test aspects, which are shown in the red color, have achieved the lowest rating. This means that they have allocated the lowest study to themselves, which respectively, are:

- User acceptance testing
- Service testing after SLA changes
- Evaluation of the implementation of the service in line with the organization’s policies
- Security testing
- Estimating the duration and timing of testing
- Testing risks

In the proposed method, each of these cases is expressed explicitly or implicitly within other test cases. In addition, each of items in the table is taken from at least two papers from previous works. Therefore, the options in the table are the cases, which are necessary in service-oriented architecture testing. According to all the mentioned issues, the proposed method is the most comprehensive method among other methods.

One of the problems of this framework is spending more time for testing. Here, there are two answers for this issue:

More tests are conducted here, but due to the overlap between some tests and conducting some test even after project delivery (E.g. testing after SLA changes), and conducting them in different levels of software development, this test will not need much time for the produced software rather than other frameworks, which are proposed.
The second cases is that whatever the test cost and time be more, lower time and costs will be spent for support and reform that evokes the notion of foresight in the software. This principle is an important principle in the design and manufacturing software and considering it can have a good guarantee for the software user-friendly and higher durable.

6. Conclusion
This research has proposed a comprehensive method to test produced software with service-oriented approach. In addition, no comprehensive test was suggested to include various aspects of this type of software. This issue causes the lack of a specified reference to test this software, which is considered as a weakness for this software. With the increasing development of this software, the need for a comprehensive framework to test the software is essential. With the growth and development of produced software with service-oriented approach, it is necessary to implement a comprehensive test for the software. In this research, a method was proposed using ISTQB test framework that completely covers the software testing requirements generated by the service-oriented approach. This testing method can be applied for all phases of software production and development with service-oriented approach, and it can be present at all stages. In addition, this study has focused on GIS-based software on how the proposed test can be applied on it and what results are expected. According to obtained information, applying the proposed framework on this software can reduce the error in the map and service providers and satisfy the users.
References


