A NEW TEST PROCESS FOR SOA BY DEPLOYING ISTQB FRAMEWORK

Reza Kianyzadeh[1], Ramin Nassiri[2], Mohammad Malakooti[3]

[1]Department of Computer Engineering, Islamic Azad University, Dubai Branch, UAE,
Reza_OTMS@yahoo.com
[2]Department of Computer Engineering, Islamic Azad University, Tehran Central Branch, Iran,
r_nasiri@iauctb.ac.ir
[3]Department of Computer Engineering, Islamic Azad University, Dubai Branch, UAE,
malakooti@iau.ae

ABSTRACT

Today, using SOA is developed as efficient architecture. Despite the existing strengths in service-oriented architecture (SOA), the test section was not considered as important process. Although, some solutions are presented in this regard, none of them covered SOA test completely. In this paper, we attempt to present a comprehensive process for SOA test based on Deploying ISTQB framework. ISTQB is a comprehensive framework and international standard to test the software. This process is consisting of some sub-process each consisting of special activities and can be used as comprehensive process as a solution to SOA test.

KEYWORDS: service-oriented architecture (SOA); SOA testing; software testing; software test process

1 INTRODUCTION

Various definitions are presented for SOA but one of the most public definitions for SOA is as:

SOA is a solution to design distributed systems presenting software functions in the form of service [4]. The key difference of this architecture is replacing system modules with the services with high flexibility and connection ability. It is vivid that flexible properties of this architecture facilitated the rapid and easy development of the systems but the dynamic property of SOA challenged the current techniques of the test and obliged to use various methods to test it [3]. This

dynamics in the SOA structure makes difficult the selection of a definite process for the test. Because we want to apply a definite process consisting of tangible activities on SOA. As the dynamics is hidden in the SOA, the test process should consider the dynamic properties of SOA. As with its hierarchy nature, it can define the test stages dynamically as for the properties of SOA. In other words, comprehensive process of SOA not only should define the test stages for a dynamic version of SOA, it should easily define the test stages for various versions of applying the service-oriented concept.

Regarding the design of software test processes, various activities are done until now but none of them privately designed the comprehensive test process of SOA. For example [3], [4], [5], [6],[7], [8], [9],[10], [12] investigated an aspect of the test problems in SOA and in some examples presented their proposed process to solve it but none of them presented a comprehensive process for SOA test. Other studies investigated one of the above items, ISTQB is a comprehensive framework and international standard to test the software defining software test generally and independent of any software structure. In the following we will attempt by deploying test process in ISTQB framework and private properties of SOA achieve a comprehensive process for SOA test.

At first in section 2, ISTQB framework is investigated, then in section 3 test process is developed in ISTQB framework to present a test process in SOA. Finally, in section 4, the conclusion and future works are stated.

2 ISTQB FRAMEWORK

As it was said, ISTQB is an international comprehensive framework to test the software. Despite its comprehensiveness, it investigates the test generally and is not dependent upon the structure of software. This issue has this advantage for ISTQB that

it can be use as a reference to test each model of software but due to generality it can not deal with all the details achieved of various architectures of software. ISTQB investigates the test aspects in the following domain:

- Layout of the test in system life cycle
- Test process
- Test management
- Test risks
- Test techniques

As our discussion about the test process issue we don't enter other aspects but test process is affected by life cycle, the life cycle is investigated.

2.1 Test in software life cycle

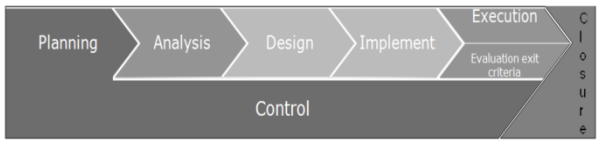

Figure 1. Test process in ISTQB

Software test can not be considered as a separated activity of system development [2] otherwise we should pay considerable cost for the existing problems in the system development. ISTQB proposes V-model to test in system life cycle and at first it introduced system development levels based on ISO-IEC 12027 and then the test was proposed equal to each of the levels.

2.2 Test process

ISTQB divides test process as the following stages (Fig. 1):

- Test planning and control
- Test analysis and design
- Test Implementation
- Test evaluation
- Test closure

Logically, the above activities (except control) are done consecutively but based on the project can be done parallel and repetitious [13]. In the following we introduced each briefly.

Test planning and control: In this stage, the test strategy is selected and defines how this strategy is used in testing. After the planning here in test control, we define how

we proceeded based on the plan. It is obvious that test control continues parallel with other activities and from the beginning of test process to the end of test process.

Analysis and designing the test: In this stage, we should design the test cases. Indeed, in this stage, the general goals of test are turned into tangible plans and the test infrastructures and tools are determined.

Test Implementation: In this stage, the test cases are applied and execute.

Test evaluation: In this stage, we evaluate to what extent, the test was successful in detecting the errors.

Test finish: In this stage by finishing the test and delivering the results, we analyze the extracted data of test.

3 TEST PROCESS IN SOA

To make the software testing successful, it should be considered along the life cycle of software [13]. To achieve a test process in SOA based on ISTQB, at first based on ISTQB we test in SOA life cycle.

3.1 The test in SOA life cycle

As it was said, ISTQB proposes V-model to test in software life cycle, on the other hand OASIS (Organization for the Advancement of Structured Information Standards) determines the following conditions for different kinds of tests in SOA [2].

a. Being made based on the needs, policies and the required characteristics selected by owners of each system.

b. It delivers the main requirements to the users.

By adding SOA governance test to V-model (referring to the general policies and standards that govern the design, build and implementation of an SOA solution), we present a special V-model for SOA testing.

In the other hand the definition being presented by ISTQB of the system consisting of some systems is as following: The system of systems is a system of some collected and independent systems presenting a service for a common aim [13].

Test levels for these systems in ISTQB are as (Fig. 2):

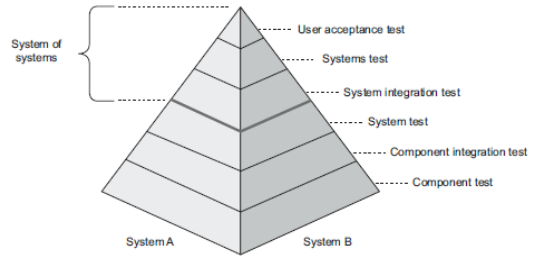

Figure 2. The test levels in combinational systems [11]

It is clear that SOA is used in the environments in which some systems communicate with each other, thus the presented model in the above figure should be observed in our proposed V-Model. Briefly, we can say that at first each of the systems from partial to total tested and then total system is tested from partial to total.

Thus, we should consider the partial to total levels of the SOA systems and by adding governance test to V-Model, we have our proposed V-Model as (Fig. 3).

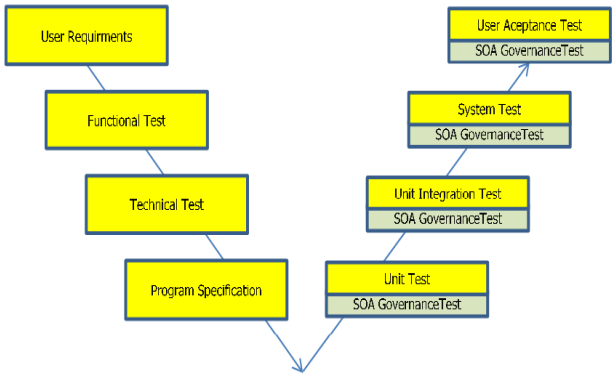

Figure 3. The proposed V-model for SOA testing

Here each of test levels is explained briefly:

Unit test:

The main aim of unit test in SOA is the investigation of the accuracy of the act of components or services and as these properties are shown in SOA governance, we considered them parallel to governing test. In this test, the smallest part of testing section in a system is tested separately of other sections to define whether it acts as expected or not. All the components before being collected in a service are tested separately in this level. In unit test, all the components should be tested abstractly to define that are determined based on the standards of organization in SOA and

probable errors are defined in efficiency and security issue.

Unit integration test:

In unit integration test, it is investigated that after the combination of components, whether each of the components acts accurately in relation to other components or not. As the properties of components combination are used in SOA governance, this test is considered in parallel with governance test.

System test:

The system test investigates whether the system act in line with the expected policies in that business or not and as organization expected policies and we considered this test in parallel with the governance test.

User acceptance test:

The name of this test defines that this test is done to show that the final system meet the demands of the user, the system is made based on the need of the user or not. As the main needs of the user are used in the SOA governance, we considered this test in parallel with the SOA governance test.

3.2 An example of SOA

Before we present the proposed process to test in SOA, an example of using SOA is presented to present each of the process stages with mentioning an example as tangible. The selection of existing systems in cargo transportation cycle is a good example of using SOA because the cargo are transferred in geographical regions between the origin and destination and in this cycle, various systems are involved with their data processing and due to the existing variety in these systems, the SOA is a good model for these systems. Thus, we consider the existing systems in cargo transportation cycle (Fig. 5) that as service-oriented exchange the data in the form of giving services.

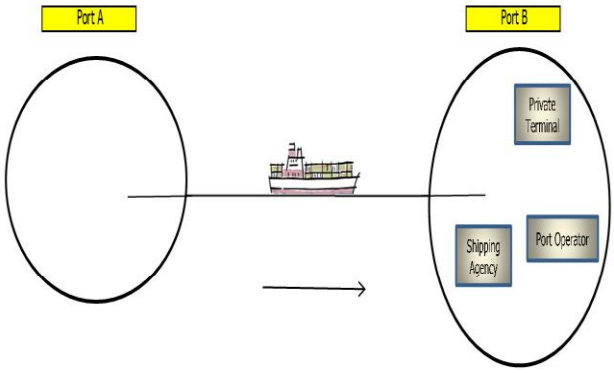

Figure 4. Simple cargo transportation cycle

As is shown in Fig. 4, it is assumed that shipping company in port *A* after taking administrative stages, loaded the cargo in metal boxes called container and sent to destination *B*. For simplicity we assume that each container of load is dedicated to an cargo owner being loaded from origin *A* by the *shipper* and is delivered in destination *B* to the consignee. As various vessels are arrived or departure in destination *B*, the control of the port is as the responsibility of port operator doing the discharging and movement of the container. After discharging the containers and cargo delivery to the consignee, the empty container is returned to private terminal to load another new cargo and enter to export cycle. The communication figure of involving systems in the above example is shown in Fig. 5.

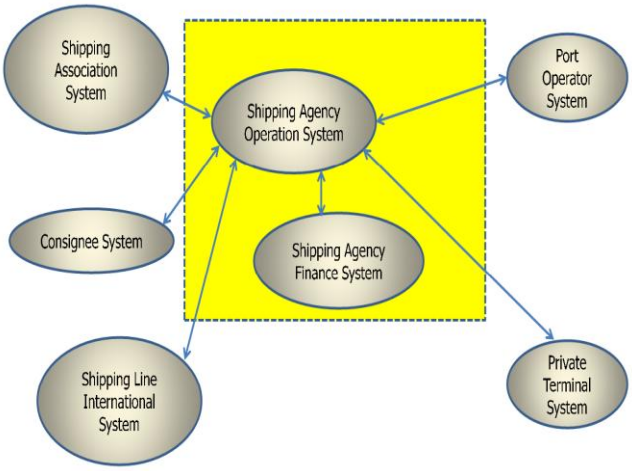

Figure 5. The systems involved in simple transportation cycle

Some of the services presented in the above system are in accordance with Table 1.

Table 1.Some example services in transportation System

NO.	Service producer	Service consumer	Service description
1	Operator system of origin port	Operation system of destination shipping agency	The list of loaded containers on the vessel departure from the origin
2	Operation system of destination shipping agency	Port operator system in destination port	The list of loaded containers on the arriving vessel
3	Operation system of destination shipping agency	Consignee system	The last status for the cargo of consignee
4	Private Terminal system	Shipping agency system of destination port	The daily list of the arrive containers
5	Operation system of destination shipping agency	consignee system	Issuing the invoice for transportation services
6	Operation system of destination shipping agency	Financial system of shipping agency in destination port	Issuing accounting document of daily invoices

3.3 The proposed process for SOA testing

After defining the proposed V-model to test SOA, we can achieve test process in SOA by developing test process in ISTQB.

- As it was said, the test process stages in ISTQB are test planning and control, test analysis and design, test Implementation, test evaluation and test closure and those activities defined as:

3.3.1. SOA test planning: In this stage, in a definite planning we should determine how to do the test. As is shown in Fig. 6, it can be said that this activity is consisting of some sub-activities and each of the sub-activities are defined for service-oriented test as:

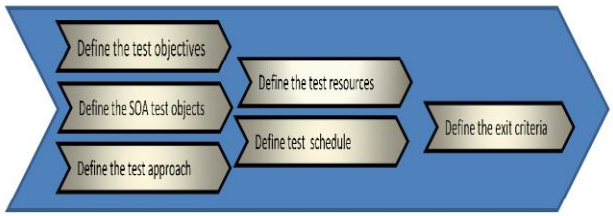

Figure 6. Test planning

Define the test objectives: The objectives of test should be defined as to show to what extent the project estimates the policies of project owners. To do this, the test objectives should have the following characteristics:

1- To determine the services can be presented to what extent and with what quality, for example one of the objectives of test in transportation system is being ensured of daily establishing of service row 4, Table 1.

2- To determine the project to what extent is based on the policies in governance. For example, in transportation system, security policies require that the information of the status of the cargo of owner is transferred as coded. Thus, another aim of the test can be being ensured of code of the service information of row 3 of Table 1.

Define the SOA test objects: The test objects in SOA are services, WSDL, SOAP,BPEL and the systems participating in SOA.

Define the test approach: Based on proposed V-model (Fig. 3) combined with the governance with bottom to top view.

Define the test resources: Here we define how much human resources and facilities are needed for the test.

Define test schedule: Here we define when the proposed V-model (Fig. 3) test steps starts and finishes.

Define the exit criteria: The test end point after the test is the final level of proposed V-model (Fig. 3).

3.3.2 SOA test analysis and design

In this stage, we define the test cases and test conditions. These test cases and conditions should be designed based on test goals. Indeed, in this stage, we attempt

before test implementation design the test. The sub activities of this stage are shown in Fig. 7.

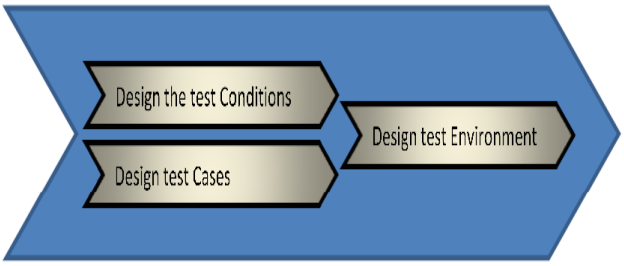

Figure 7. Test analysis and design

Design the test conditions: In this activity, we define the aspects or system attributes being tested. These attributes should be considered in all the system development levels considered in proposed V-model (Fig. 3). The most important properties being considered in the services are: service functional properties, service non-functional properties, service security properties, service performance and service interoperability but as the investigation of all the test condition is not possible, we should prioritize them.

Design test cases: The test cases are the scenarios consisting of special parameters and expected results and by them the test conditions are evaluated. Some examples are shown in Table 2.

Table 2.The test case and conditions of some sample tests in transportation system

NO.	test example in transportation system	Test condition	Test case
1	The test to be ensured of sending the data in accordance in row 1 of Table 1(Be sure that the list of containers of other vessel are not sent by mistake)	Non-functional properties	**Input:** message, vessel name **Expected result:** the information is in accordance with the request
2	The test to be ensure of the systems are ready to use (all the existing systems in Table 1)	Functional properties	**Input:** target system name, system ready message **Expected result:** approval response
3	The test to check the security of messages (each of the message of Table 1)	Security	**Input:** message **Expected result:** The message is made based on security pattern
4	The test of balance of accounting document issued in row 6 of Table 1	Functional properties	**Input:** message **Expected result:** The document is balanced.
5	The test of similarity of basic common information code (each of	Non-functional properties	**Input**: the tables of basic information of systems

	the message of Table 1)		**Expected result:** The tables are similar
6	The test of customers middle code in other systems (each of the messages of Table 1)	Non-functional properties	**Input**: the tables of systems **Expected result:** The tables are similar.
7	Policies combination test (each of the messages of Table 1)	interoperability	**Input:** all policies combination case **Expected result:** the combination of policies is possible.
8	The test of sink of the systems and being ensured that data are update (all the systems in Table 1)	Performance	**Input:** The systems inventory report **Expected result:**The reports are consistent
9	The test to be ensure of re-update in case of the lack of required data (all the existing systems in Table1)	Performance	**Input:** check of sending data conditions **Expected result:** The lacks are updated
10	Sending the data to the operator of port at allowed time, at least 6 hours before the entrance of vessel (service row 2, Table 1)	Performance	**Input:** message, the arrival time of the vessel **Expected result:** The data are sent at allowable time.
11	The test to be ensured that there is no typing error of the user in message data (each messages of Table 1)	Non-functional properties	**Input:** message, list of updateable data **Expected result:** the data are correct
12	The test to be ensured that there is no mistake in the number of containers (each of the messages of Table 1)	Functional properties	**Input:** message, the number of exiting container **Expected result:** The number of containers is correct.
13	The test to be ensured that there is no keywords in XML messages (each of the messages of Table 1)	Performance	**Input:** message, keywords for the systems **Expected result:** keywords are not in the message
14	The test to be ensured of delivering the report of the latest update to the user (all the systems in Table 1)	Performance	**Input:** The systems inventory reports **Expected result:** The reports are update

Design test environment: In this stage, the infrastructures and required probable tools are defined to test the services. For example,

if we want to evaluate the standardization of the messages based on XML structure, we need private test tools.

3.3.3 SOA test implementation

In this stage, we implement the test cases which have designed in the previous stage and based on the results we take decision to more tests. The activities of this stage are shown in Fig. 8.

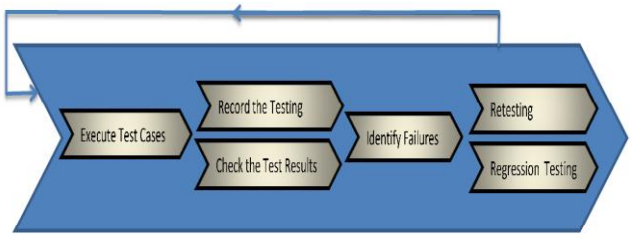

Figure 8. Test implementation

Execute test cases: In this stage, we execute the designed test cases to evaluate the system properties. It is clear that the more we design more scenarios for the test, the results of test will be more exact. The selection of scenarios for the services execution should be such that test all the system levels are integrated with governance.

Record the testing: To do this, a database table is considered to store the test results and its columns are the test level, the object or test service, scenario, pass or fail, test conditions and test case then after performing each of the test cases, a row is added to the mentioned table.

Check the test results: It is required to investigate the completed table after performing the test cases.

Identify Failures: To do this we selected the table rows that their pass/fail column is filled with fail and after attempting to repair the system and remove the errors, we update the database again.

Retesting: After removing the error, we do the test again to be ensured the error is removed.

Regression testing: After being ensured of the error removed, we investigate other parts

that after removing the previous errors, no problem is created for other parts.

3.3.4 The evaluation and finishing of SOA test

In this stage, we evaluate the reports which has extracted from the previous stages and based on the comparing with the expected levels in SOA governance and service level agreements (SLA), we determine the success of the project with SOA and finish the test. The activities of this stage are shown in Fig. 9.

Figure 9. The evaluation and finishing the test

Comparing test results and expectations: In this activity, the results of the test are compared with the expectations in the test objectives.

Reporting: In this stage we present the test report.

3.3.5 **SOA test control**

As it was said, in test control, we investigate to what extent we proceed in the test according to the plan and we take decision to continue, modify or stop test process. It is clear that test control continues parallel to other activities from the beginning of test process to the end of test process.

4. CONCLUSION

In this paper it was attempted that by giving a good model to test SOA and developing the test process in ISTQB international framework, achieve a new process and this process can be used as a reference for the testers to test the projects under SOA and by the steps presented for test process, we save the time and increase the test quality. In the future papers, we try to apply the proposed process for OTMS project as shipping and international transportation software being designed based on SOA. And design test cases by tangible examples. This leads into

the exact investigation of this process, proof its applied nature and doing probable modifications on proposed process to test SOA.

5 REFERENCES

[1] D. Graham, E. Veenendaal, I. Evans and R.Black, "Foundations of Software Testing ISTQB Certification", Patrick Bond, 2008.

[2] K. Laskey, P. Brown, J. Estefan, F. McCabe, D. Thornton, "Reference Architecture Foundation for Service Oriented Architecture Version 1.0" OASIS(Organization for the Advancement of Structured Information Standards), Julay 2011.

[3] G. Canfora, M. Di Penta, " SOA: Testing and Self-Checking", Research Centre on Software Technology - University of Sannio Palazzo ex Poste, Via Traiano - 82100 Benevento, Italy, 2008.

[4] A. Bertolino, A. polini, "SOA Test Governance: enabling service integration testing across organization and technology borders", IEEE, 978-0-7695-3571-2/09, pp. 277- 286 , 2009.

[5] G. Canfora, M. Di Penta, "Service Oriented Architectures Testing: A Survey" RCOST - Research Centre on Software Technology University of Sannio Palazzo ex Poste, Via Traiano 82100 Benevento, Italy, 2009.

[6] S. Wieczorek, A. Stefanescu, "Service Integration: A Soft Spot in the SOA Testing Stack", IEEE, 978-1-4244-5665-9/09, pp. 211- 216, 2009.

[7] T. Parveen, S. Tilley "A Research Agenda for Testing SOA-Based Systems", IEEE International Systems Conference, Montreal, Canada, 2009.

[8] H. Liu , Z. Li, J. Zhu, H. Tan, H. Huang, "A Unified Test Framework for Continuous Integration Testing of SOA Solutions", IEEE International Conference, IBM China Research Lab Beijing, China, 2009.

[9] H. Yoon, E. Mi Ji, B. Choi, "Generating Test Requirements for the Service Connections based on the Layers of SOA", IEEE 0-7695-3127-X/08 $25.00, p.p348-355, 2008.

[10] L. Ribarov1, I. Manova, S. Ilieva, "Testing in a Service-Oriented World", Proceedings of International Confrance on Information Technologies, Bulgaria, September 2007.

[11] S. Bruning, S. Weibleder, M. Malek, "A Fult Taxonomy For SOA", IEEE 15 30-2059/07 $25.00, Germany, 2007.

[12] W. K. Chan, S. C. Cheung, Karl. R. P.H. Leung, "Towards a Metamorphic Testing

Methodology for Service-Oriented Software Applications", IEEE 1550-6002/05 $20.00, 2005.

[13] R. Black, "Guide to the ISTQB Advanced Certification as an Advanced Test Manager", Vol.1, Rocky Nook, USA, 2009.