Boundary clarifying between integration and validation testing and engineering example

Mengqing Tanli[1,*], *Ying* Zhang[2], *Yulin* Wang[2], and *Yan* Jiang[2]

[1]Software School, University of South China, 421001 Hunan Hengyang, China
[2]Mechanical Engineering school, University of South China, 421001 Hunan Hengyang, China

Abstract. Partitioning of testing phases is a basic problem, and for clarifying the boundary between integration and validation testing, this paper discusses this issue from basic technological boundary and management and control aspect. Meanwhile, discussion about test certificate for validation testing is proposed for suggestion and reference. Without loss of applicability, an engineering application example is given, and how to construct test case respectively for integration testing and validation testing is briefly summarized according to engineering requirement.

1 Introduction and background

Currently, science and technology appears blooming and huge influence to human being, and itself emerges very interested phenomenon, which the interior become more and more refined and elaborate and various disciplines are mutually crossing and fusing. Similar phenomenon has occurred in software testing as a new discipline, on the one hand software testing technology is more and more refined and on the other hand thinking and method from many other disciplines has plugged into software testing [1-3]. For refining of software testing technology, figure 1 has illustrated this state in some extent.

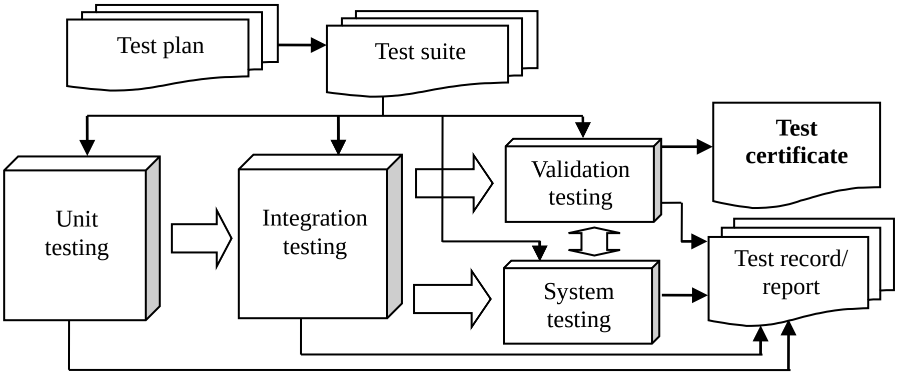

Fig. 1. Software testing phases and activity for product.

[*] Corresponding author: TLMQ-TEAM@163.com

Integration testing has been known a lot for software engineering in academia and industry practitioners, as the assembling for producing of mechanical and electronic product. However, validation testing is not so similar like "Verification File" for a construction. In fact, validation testing has been done by developer and test engineer in producing of software product, for example, the smoke test for a baseline product [4]. Today, software testing had been toward independent and specializing after the first software testing conference is opened in The University of North Carolina System and the IEEE paper referring testing data is delivered by John Good Enough and Susan Gerhart in 1972 [5].

However, the misunderstand that software testing is lonely for software producing process and software engineering must be avoided [6]. Factually, software testing has penetrated all phase of software producing process.

2 Basic technological boundary

In following description, we will discusses the basic technological boundary between integration testing and validation testing, it mainly refers three aspects: definition, subjective and content.

2.1 Basic definition

As we known, integration testing is defined as the synthetic testing which units and modules is combined together to judge whether it will work sucesussfully in terms of the specification. And integration testing is a testing activity while unit testing is finished.

However, validation testing is confirming testing of judging whether the software finished product is qualified and has been meet to release.

2.2 Main subjective

The subjective of integration testing is generally test whether assembling of units and modules can be run correctly and exactly.

But, the subjective of validation testing is usually confirm whether the whole software/system is qualified and can be leave the producer.

2.3 Typical content

Generally, integration testing is a mainly black-box testing, and its main content include two parts, that is, data testing and function/state testing [7]. For data testing, safety testing, usual & boundary value testing, and format & interface testing are three typical aspects. For function/state testing, it may be noticed that failure function/state should be be tested besides testing usual function/state. More details are shown in figure 2.

Correspondingly, the viewpoint of validation testing should be user and focus on the implementation of requirement of user. Here, we considered with actual experience, validation testing of software product should include following four processes: (1) Smoke test of baseline product, (2) Software configuration testing, (3) Business-based scenario testing, (4) Data-based scenario testing. Figure 3 has demonstrated the details of validation testing activity.

3 Further boundary for management and control

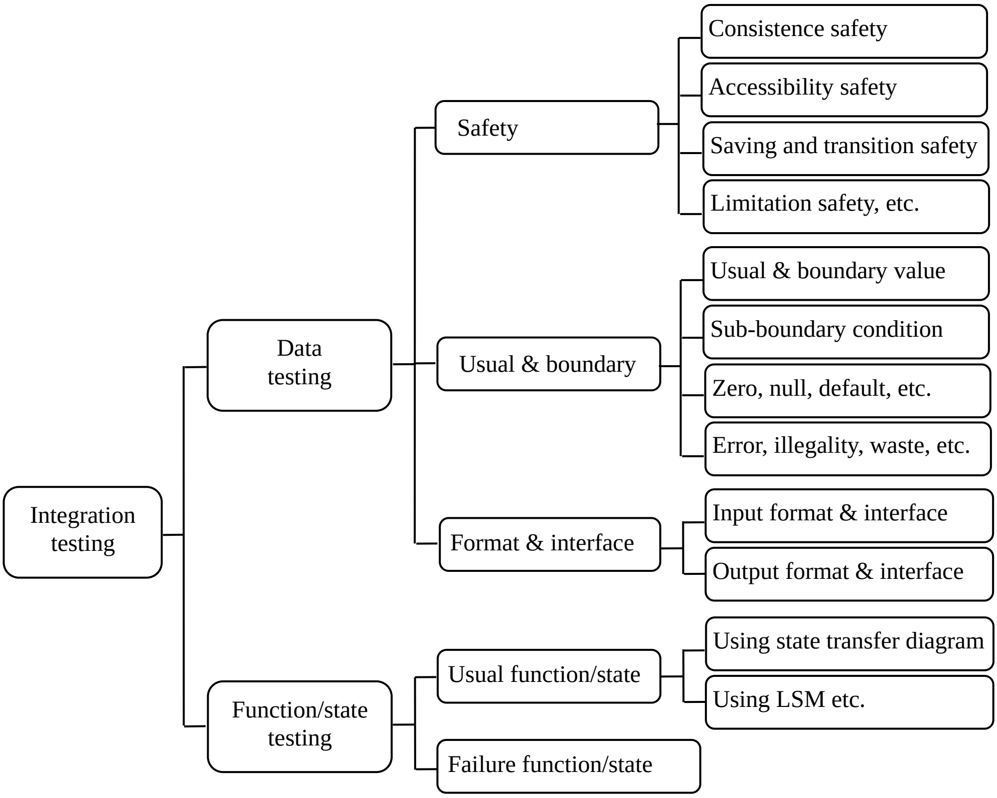

Fig. 2. Typical contents of integration testing activity

Aspects of management and control should be taken into account to clarify boundary between integration testing and validation testing, and there are three points to pay attention as follows.

3.1 Duty and task

The duty and task of integration testing is generally test whether assembling of units and modules can be run correctly and exactly and it focuses on the process of software producing.

But, the duty and task of validation testing is usually confirm whether the whole software/system can be released and leave the producer. As a consequence, it focuses on the result of software producing.

3.2 Administration and juristiction

For boundary clarification between integration testing and validation testing, administration and juristiction will be the most important deference

In software producing organization, integration testing may be operated by programmer and tester in programming department and within juristiction of developing department, however, validation testing must be executed by independent tester and within juristiction of testing department.

3.3 Influence extent

Because integration testing is operated by programmer and tester in the process of software producing, integration testing can mainly influence the internal part of software producing organization. However, validation testing will determine whether the whole software/system can be released and leave the producer, so the influence extent of validation testing will extend to user and customer.

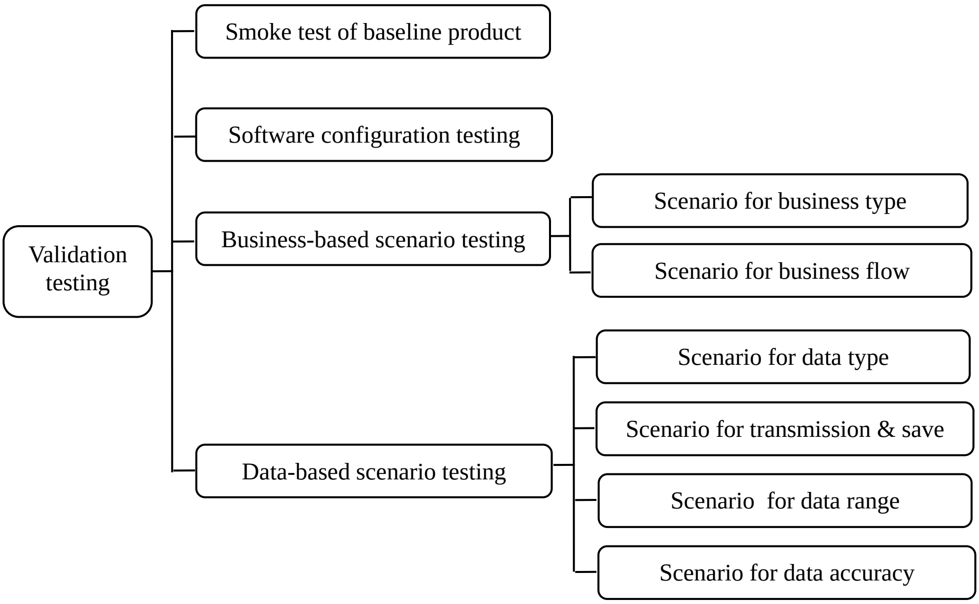
Validation testing
Smoke test of baseline product
Software configuration testing
Business-based scenario testing
Scenario for business type
Scenario for business flow
Data-based scenario testing
Scenario for data type
Scenario for transmission & save
Scenario for data range
Scenario for data accuracy

Fig. 3. Usual contents of validation testing activity

4 Discussion of test certificate for validation testing [5,6]

Now, the test certificate is lack for software product, not only on-line, but also offline. Generally, before all kinds of product leave the producer, it must be passed by finishing test and guaranteed with a test certificate. Why not for software product?

4.1 Role and function of test certificate

At first, test certificate can be constructed as a part of quality assurance system of software product [10] like a qualified mark. Of curse, like other products, test certificate or qualified certificate should become a certifying mark [4] for software product. By which, users will know that this software had been tested with rigid specification and testing measurement and this product can be used in safety.

Additionally, the test certificate can also be identification of the producer and its product, especially with authority of government license in some extent.

4.2 Type and format of test certificate

For issuing type of test certificate or qualified certificate, three types can be generally used for factual software product. (1) Postmark/stamp type, this type is fitted to small software product, and its shape and format may be set up in terms of software producer respectively, and items should include tester code. (2) Card type, this type is usually used in material object of general software product. The shape and format of this type may be set up in terms of software producer respectively, but items should include: a.Brand, b. Name, c. Grade, d.Tester code, e.Standard code, f. Other comment. (3) File type, this type can be applied in large scale software product, in particular, it is fitted to safety-critical software/system. The shape and detail format of this type must be obliged to specified standard, and items should include: a.Brand, b. Name, c. Product standard code, d. Grade, e. Tester code, f. Safety standard code, g. Other comment.

4.3 Condition of issuing and presentation of test certificate

Before issuing and presentation of test certificate for a software product, following conditions must be met: (1) Business-based and data-based scenario testing for validation testing is passed [4]. (2) Software configuration testing is passed [12]. (3) System testing required is passed [12]. (4) α testing - if necessary - is passed [7].

4.4 General procedure of issuing test certificate

The issuing and presentation of test certificate must be done by independent test department for any software producer, and the duty must be responsible by a particular people. As a result, following requirements must be Confirmed: (1) All testing contents are passed. (2) All items of software product are correct and consistent. (3) Tester information is right. (4) Special disposing is aware and controlled by manager.

4.5 Usage of test certificate

- Being able to leave the software producer.
- Important material to appraise safety responsibility accident.
- Main basis and accordance of government supervising.

5 Engineering example focusing distinguish of test design [7-10]

In order to give an actual demonstration of boundary clarifying between integration testing and validation testing, here, we discusses the deference of construction of test case [8,9] each other with the examples of PQMS2 (Product Quality Monitoring Software 2.0).

5.1 Test design of integration testing based on improving efficiency

Construction of test case must synthetically consider quality, progress and cost, but the progress should be main point for integration testing. Of course, how to decrease the execution time of test suite is a main aspect, and we have proposed a grey-box approach to do it. About grey-box approach, details may refer [11]. Table 1 has demonstrated the result of applying grey-box approach to improve efficiency for integration testing in PQMS2.

Table 1. Efficiency improvement of applying grey-box approach.

	According to testing number	According to testing time	
		Execution time/min	Efficiency improvement
Theoretically	333%	88.00	438%
Factually	-	34.16	398%

5.2 Test design of validation testing based on assuring quality

As a consequence, construction of test case for validation testing should be consider two aspects as follows.

(1) Construction based on actual business scenario

In order to assure the software product quality, test case of validation testing should be conducted based on actual business scenario [14]. Table 2 has shown the distribution of business scenario of test case construction in smoke test of validation testing in PQMS2.

Table 2. Construction based on actual business scenario.

	Median	XAve	X	NP	P	C	U	Sum	Percent
Purchase inspection	1	2	-	-	-	-	-	3	27.2%
Process inspection	3	-	-	-	1	1	1	6	54.6%
Finish inspection	-	-	1	1	-	-	-	2	18.2%

As shown in table 2, three test cases for control chart are used in purchase inspection and it occupies 27.2%. Six test cases for control chart are applied in process inspection and it is 54.6%. Two test cases for control chart are utilized and it has ratio of 18.2%. This result of percent has perfectly revealed the actual distribution status.

(2) Construction based on actual data scenario

Data is the base of software testing, and construction of test case of validation testing should be based on actual data scenario [15,16]. Table 3 is actual scenario distribution of data input in constructing of test case of "Smoke Test".

Table 3. Construction based on actual scenario for data input mode and type.

	Manually input	Input appling digital gauge	Input from saving data	Input with Notepad of Windows
Median chart	60	70	60	40
XAve chart	-	40	90	-
X chart	-	12	-	-
NP chart	20	-	-	-
P chart	-	-	-	16
C chart	-	-	-	12
U chart	-	-	-	12
SUM	80	122	150	80

Form table 3, the number of manually input data is 80, and the number of input data appling digital gauge is 122, while the number of input data from by saving is 150, but input data with Notepad of Windows only has 80 bathes. This kind of distribution is fitted to application status of actual factory.

Consequently, Table 4 is the actual scenario distribution of data value in constructing of test case of “Smoke Test”for validation testing in PQMS2.

Table 4. Construction based on actual scenario for data value.

	Data accuracy			Data dimension	
	0.001	0.01	0.1	10^{-2}	10^{-1}
Distribution status	★	★	★	★	★

Table 4 implies that construction of test case in validation testing has distributed for all kind of reasonable values.

In a word, construction of test suite for integration testing should be based on improving efficiency, and applying grey-box testing technique is good way for decreasing the execution time. Consequently, construction of test suite for validation testing should focus on assuring quality, and constructing of test case should be based on actual business scenario including business kind and business flow, while constructing should be considered for actual data scenario including data input/output, data types, data accuracy, data value, etc.

6 Conclusion

Basic technological boundary for clarifing between integration testing and validation testing should include basic definition, main subjective, typical content. However, key distinguishing is that integration testing is a process judgement and validation testing is a concluded confirming. For engineering application, there are deferences each other especially in test design, construction of test suite for integration testing should focus on improving efficiency, and construction of test suite for validation testing should pay attention to assuring quality.

References

1. B. W. Boehm. *Software Engineering Economics* (Upper: Prentice-Hall, 1981)
2. Per Runeson, J. Empirical Software Engineering, 14, 2 (2009)
3. Jeremy S. Bradbury, James R. Cordy, Juergen Dingel, Proc. the ACM SIGPLAN-SIGSOFT Work on Program Analysis for Software Tools and Engineering. (Portugal: Lisbon, 2005)
4. Mengqing TANLI, Ying ZHANG, Yan Jiang, et.al., Proc. 2021 International Conference on Communication Engineering and Logistics Management. (China: Shanghai, 2021)
5. Fu Bing, *Course of Software Testing Technology* (Tsinghua University Press, Beijing, 2014)
6. HyunSook Do, Gregg Rothermel, Sebastian Elbaum, Proc. the 2004 International symposium on Empirical Software Engineering. (USA, 2004)
7. Ron Patton, *Software engineering* (Pearson Education Inc., USA, 2006)
8. XU Yuan-yuan, SOFTWARE, 36, 9 (2015)
9. TANLI Meng-qing, ZHANG Ying, WANG Yu-lin, SOFTWARE, 41, 9 (2020)
10. MENGQING TANLI, YAN JIANG, YULIN WANG, et al., Proc. 2020 the 10th International Workshop on Computer Science and Engineering (China: Shanghai, 2020)
11. Mengqing TANLI**,** Ying Zhang**,** Yulin Wang, et.al., Proc. 2021 3rd International Conference on Artificial Intelligence and Computer Science. (China: Beijing, 2021)
12. TANLI Meng-qing, ZHANG Ying, WANG Yu-lin, SOFTWARE, 41, 11 (2020)
13. Li Fan, *Software Testing Technology* (Mechanical Industry Press, Beijing, 2016)
14. Dan Tang, Mengqing TanLi, Yan Jiang, Proc. the 4th Annual Int. Conf.on Information System and Artificial Intelligence, (China: Changsha, 2020)
15. Elfriede, Dustin, *Effective Software Testing: 50 Specific Ways to Improve Your Testing* (Electric Press, China, 2004)
16. Chen Zhanhua, Wireless Electron Engineering, 29, 2 (2005)