

Technical Trends and Challenges of Software Testing

TOSHIAKI KUROKAWA
MASATO SHINAGAWA
Affiliated Fellow

1 Introduction

Software tests are (herein referred to as software testing) commonly known as “a test to verify if there is no problem with software”. Software testing, in this sense, may seem to be similar to tests of various “artifacts (products)” such as ones to check the safety of confectioneries, whether batteries are produced as specified, or whether an aircraft has any problems concerning its flight. In fact, many problems have been revealed concerning inspections and testing of these products in recent years.

On the other hand, unlike such tangible products, software would have different types of problems because problems in software are hard to see by users until such problems actually are visualized by a series of defects. In the case of tangible products, you may notice some indication before the serious problem occurs. In other words, unless the certain conditions are tested for the defects, the software is considered and treated to be at normal state, or “no problem”. In a more extreme statement, it can be said that problems attributed to software are caused by insufficiency of its testing.

Recent software related problems are largely caused by lack of testing, and have actually brought social problems and financial losses. For example, there was a failure with the New Derivation Trading System, which deals with derivative products, of Tokyo Stock Exchange on February 8, 2008. Due to this problem, the trading system operation have been suspended until the 12th of the month. The reported reason of the failure was an “initialization error with the memory within the server, and no initialization were conducted under a certain condition”.^[1] While

this system was originally planned to be in effect from October 2007, testing showed that there were numerous problems. Therefore, the operation of the system was delayed for 3 months, and it started to be in use from January 15, 2008. The Tokyo Stock Exchange also had an accident in November 2005, in which their system failed and all stock exchanges were suspended.

In 2007, there was also an inconvenience with the automatic gate system at train stations in and around the Tokyo metropolis. There have been many reports and opinions on this issue.^[2-5] For example, a TV program reported that it was caused by a comparatively simple error (“+offset” was missing). In this case, the bug was not noticed until a certain condition is met. Once this condition occurred, it caused errors to all connected automatic gate machines. In addition, as this error affected the entire rail industry of the metropolis area, it was reported as a significant social problem with several million people suffered. Such events should not happen if some test were conducted to check these specific conditions of this accident. It was told that only such limited number of tests were conducted for big numbers as 1000, 5000 and 10000, so they missed the special case relating to the buffer which caused failures.

Due to the increasing complexity and proliferation of software, the importance of tests is increasing. However, the difficulties of testing are also increasing. There have been various attempts to solve these software testing problems. Recent trends in software testing have been frequently reported in technical community.^[6] For instance, a group for testing security software products named Anti-Malware Testing Standards Organization (AMTSO) was formed on February 5, 2008. As malware has become more diverse and products

has become more complex, the appropriate evaluation of the tests for current security software had become difficult. This organization is to improve testing methods and appeal the practical standards and guidelines to the world.

In this report, first, the meaning of software testing will be made clearer, especially for the so-called “embedded systems”, so that the testing becomes important for the technology related to the safety and security of citizens. Next, the testing technologies are sketched, and the emergence of companies and organizations specialized in software testing is reported. As the biggest issue of software testing, this paper reports the current treatments of software test designers and engineers, as well as the current state of their education. The development of a quality assurance industry based on software testing technology is discussed along with the evaluation of software quality and social/legal systems for software quality. We also need innovations to make all of these issues more commonly understood by general public. This paper summarizes that the efforts to overcome such challenges would contribute to create a “quality” industry in a long run.

2 Tests in over-all software development processes—differentiation from debugging –

In traditional software engineering, software testing was not emphasized much, compared to other parts, partly because people did not recognize the differences between software testing and program debugging as described below.

Software program failures itself may have been seen even in the early days of programming. Efforts to eliminate failures are called debugging since such failures in programs are called bugs. While software tests are related to debugging, they should be considered as a separate task from debugging. This idea was also emphasized in the “Art of Software Testing” by Myers (published in 1979), which is said to be the classic book on software testing.^[7]

The differences between debugging and software testing may be summarized as shown in Table 1. Although both debugging and testing have the common goal of quality advancement, debugging

aims to eliminate bugs of programs while testing is to identify any defects in the whole program including the system aspects where programs are used. Therefore, it should be noted that these two tasks are completely different in nature.

For example, even if there are no bugs in a program, defects may be found by testing. In other words, debugging is to identify whether a program satisfies the predetermined specifications, while testing is to identify any inconvenience for users of the program.

Here is an example outside areas from software. Suppose some users may put “waterproof” products into the ocean or bath with some chemicals put as bath agents. In these situations, the product needs to not only be waterproof against regular water, but also to have certain durability against salt or other chemicals. It is hardly said to be “waterproof” from the users’ viewpoint, that only the regular water is permitted.

Defects identified by debugging are attributed to cases that logic of an intended program was not achieved with the actual program, or there are mistakes in contents. On the other hand, system failures to be identified by software testing are inconvenience or risk for users so that the software should be designed and made to prevent such problems.

Buggy programs are written by the lack of skills of the programmer, and the improving individual programming skills is the responsibility of the programmer. If software testing were considered to be a process for debugging, programmers must be in charge of software testing, which is not a good practice. Putting programmers for testing may eventually lead to a higher risk of overseeing bugs undetected.

As an empirical rule in software engineering, there should be at least some bugs unfound for years in large programs with over ten-thousand lines. Even new bugs (sometimes even crucial ones) could be introduced during the program upgrading (or debugging) tasks. In lengthy and large-scale programs, it is important to conduct software testing from the viewpoint of fail-safe, that is to prevent the system from being damaged significantly or critically even though some bugs are remained and they are hit to do critically damaging function.

Table 1 : Difference of debugging and software test

	Debugging	Testing
Purpose	Eliminate bugs	Evaluate quality risks and guarantee an expected quality
Subject	Program	System (including software)
Work	Correct faults	Identify faults
Worker	Programmer	Test engineer
Start	After writing a program	A test plan is created from the phase of requirements specifications
End	(None = eternity)	Judgment of the person in charge of testing

Prepared by the STFC

3 Definition and contents of current software testing

There are still confusions in definitions of software testing. According to [8], software testing is commonly known as a test to evaluate and then guarantee the level of quality of software, or to verify if there are no risks. With this definition, software testing needs to be started from the upstream operations of software development.

According to the standard which stipulates the life cycle of software, JIS X 0160:1996 (software lifecycle process), the test is broken down into the 3 phases.^[9]

- [1] System and software qualification confirmation tests for the development process.
- [2] Operation tests for the operation process
- [3] Quality assurance process, verification process and validation process that include testing as part of the lifecycle supporting processes.

In addition, the standard for system lifecycles, JIS X 0170, which includes not only software but also hardware systems, addresses a task called software testing in the processes of verification and validation.^[10]

As stated in the Science and Technology Trends No.11, April 2004 “Toward the Improvement of Quality and Reliability in Information System Construction– A Study of “Business Rules” and Requirements Engineering in the Upstream Process”,^[11] quality assurance activities are imperative in the upstream process to enhance quality and reliability of a system, or in the “super upstream operations”, coined by the Software

Engineering Center (SEC) of the Information Processing Association (IPA), Japan. It is necessary to investigate the testability from the upstream operations of requirements specifications, not only to check if the users’ needs are met, but also to see how the tests will be effectively done. In other words, the quality assurance process should start from the point in the upstream process where demands and requirements are studied, where the person in charge of the test must participate from the start.

Moreover, as introduced in the Science and Technology Trends 2004 September “The Two Rationalities and Japan’s Software Engineering”,^[13] the methodology, known as Agile Development, of making a test program as the first step in program creation, has been promoted a new way of program development. Kent Beck, one of the advocates of Agile Development, has proposed a system called Test Driven Development (TDD), in which the software development itself is constructed around the testing.^[14] Furthermore, Hayashi and others have proposed a system where TDD is employed at the modeling stage before the actual program development.^[15]

Some says that the software testing process would be unnecessary if it is possible to create high-quality bug-free software. However, if user “demand” is taken into consideration in the first place, these demands will change depending on the time and environment. Therefore, it is important to understand that there is no such thing as “with absolutely no faults”.^[11] Secondly as history and experiences tell, it is very difficult to design and implement software with no faults or hidden bugs, even if the best developers/designers/programmers are gathered for the process. Consequently, it must

be understood that testing is an inevitable process.

Furthermore, software testing defined as a process of quality assurance or verification of risk-free quality is different from debugging so that it does not expect a perfect program without any failures of components. Rather, software testing should be effective to prevent critical damages on the whole system for users, by taking into consideration of potential failures of the program and its environments.

4 Importance of software tests for embedded system software

In particular, importance of software test is currently recognized in an area called embedded systems or embedded software. In this area, the users have no sense of using software, which is quite different from such areas of personal computer OS and word processors. Embedded software is embedded in a mechanical system, where the software supports the mechanical function.

In the past when computers were not easily affordable, embedded software system was only used for very special systems such as APOLLO aerospace ship or nuclear reactor control. However, with the decline of prices of computers and proliferation of microprocessors, the trend has changed to use the microprocessors for various machines, which also expanded the areas of embedded software usages. Currently, such embedded software would be used in most of mechanical products. For instance, an automobile has some dozens of microprocessors, and the total number of programs of its embedded software exceeds 10 million lines.^[16-18] Embedded systems are widely proliferated in a rapid pace. Its business scale in Japan reached 62 trillion yen, or 12% of the gross national products.^[19] According to this METI report, the number of employees of embedded system companies are more than 4.71 million, which accounts for 9% of the total workers in Japan.

The business scale is something, but the implication for the quality of the product is more important which greatly affect the quality of life of general public. From this viewpoint, embedded system software currently gains a lot

of attention. Any failure of embedded software could lead a significant problem on users, as stated in the example of automatic ticket gate machine failure which was far more than expected by the responsible parties. Failures of mobile phones would require a tremendous amount of costs for recall and replacement of defect products. Embedded software testing becomes more important than that of enterprise information system. Some companies have emerged with the expertise of embedded software testing and are increasing their sales.

The designer of embedded systems should have expertise of both hardware and software. Failures of the system may be caused by both hardware and software so that it is necessary to test both, not either one.

The popularization of Internet and integration of various corporate information systems have a dangerous implication that even conventional information systems to affect various aspects in our society. In other words, corporate systems come to be treated as important as embedded systems for general public. Even one system failure of a company could lead a serious inconvenience to our society. An example is software viruses. There is a possibility that all people are affected severely by destructive viruses with the failure that an individual or a company fails to take anti-virus measures.

5 Recent trend of software test technologies

There are various types of software tests available thanks to the accumulation of its research and development activities. These are categorized depending on testing environment, testing objectives and purposes, testing methods, and the means of testing. Nomenclatures and definitions for the testing types may slightly vary in the textbooks listed in Reference.^[20-22] Readers can consult software test technologies and methods described in these textbooks and references. This report describes recent topics.

5-1 Links between test process and software development process

Traditionally, software testing are considered to

be a V-shaped model as shown in Figure 1, because the test is conducted after software development. Consequently, failures are detected and corrected only after software development. Therefore, defects in the upper stream cannot be remedied until the end point of testing, which led an increase of workload for correction.

In the testing phase, traditional focus have been the increase of numbers of tests, which may not bring effective results to solve problems in

software. One of the software testing methods developed in Japan is HAYST Method.^[23] This is a method based on a scientific test plan method, which is to to implement effective testing in advance.

Quality engineering tells that quality should be incorporated in the upstream process. This means that upstream process needs testing for quality. Recently, the W-shaped model as shown in Figure 2 supersedes the traditional V-shaped model. The

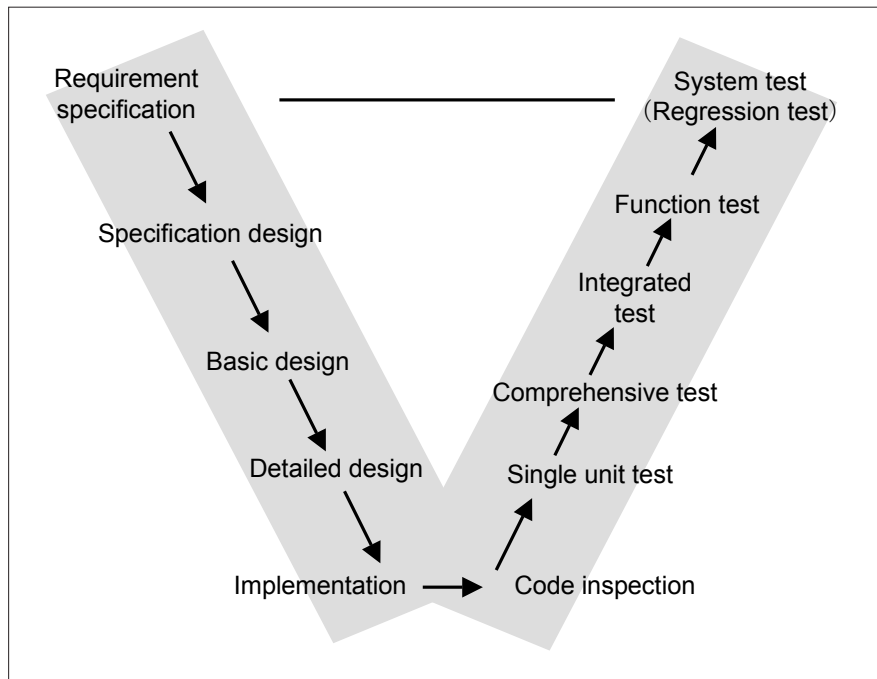


Figure 1 : V-shaped model of development and test (conventional)

Prepared by the STFC

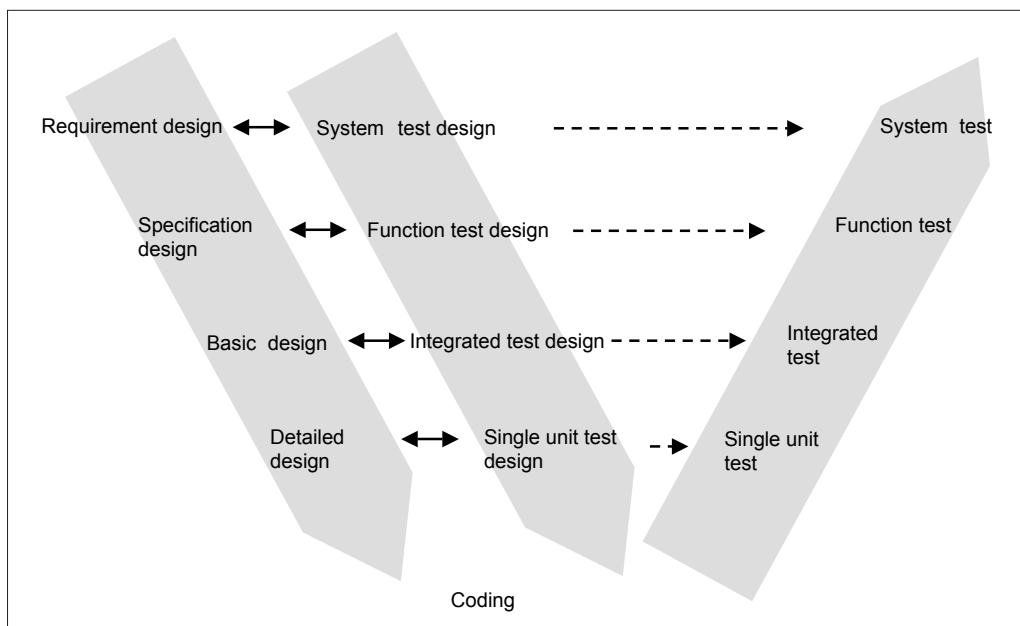


Figure 2 :The "W"-shaped model for software development and tests (the most recent model)

Prepared by the STFC

two large downward arrows in the left of Figure 2 represent a waterfall development of traditional software development, which was referred as a linear development in “Information and Communications Technology and Shiso – Shiso as a Capability for Science and Technology”.^[24] However, horizontal arrows indicate that test designs will be conducted with the development in each phase. Even the pre-development requirements phase will be accompanied the system test design and testability consideration. While the system test itself will be conducted in the final phase, the test design and testability investigation may correct failures in the upstream.

The W-shaped model creates a test program in line with the program design in the same manner of Agile Development Method. However, while Agile Method requires programmers who write a program would conduct the test, the W-shaped model assigns test engineers, not programmers to do the test.

Some idealist claims that in the future, special quality software could be produced which could avoid testing. However, it is impossible to omit a test for any products or systems today, and most likely even in the future we still need testing. Moreover, software is growing further complicated and larger in scale, and it becomes more difficult to create a perfect flawless program. Hence, it is crucial to manage software risks by the effective tests of the existing and under development software. If we extend the principle of “testing accompanies with development” to the principle of “test is created before development, and development should proceed to satisfy the test”, it will produce the framework of the test driven development methodology (TDD) described in Chapter 3.

Since the software is invisible, it would be effective to apply requirement testability and TDD for the software development. These methods can be applied for software in a wider sense, such as laws and regulations. Industrial efforts for “visualization” and “performance evaluation” can be an indication for the effectiveness of these testability/test-driven approach.

5-2 Static test

Software tests employ static and dynamic

techniques. Dynamic test is to execute a program and analyze the results. The static test is to analyze a program text without executing a program. “Test” section in “Guide to the Software Engineering Body of Knowledge”^[25] refers to dynamic techniques, while the static testing is categorized in “Verification and Validation” section. However, other software test textbooks include the static test in the part of testing. The static test includes human-based activities such as “review” and “inspection” as well as computer-based analysis and data processing such as “metrics”, “coding rule inspection”, “static analysis” and “model checking”. While the dynamic test can be done without special knowledge, the static test can only be performed by specialists. Its advantage lies in its prompt feedback to the development team because it is done in the midst of development process rather than after the development. Static test can improve development process and human abilities.

In terms of quality improvement of system development, it is necessary and efficient to ensure the quality in the upstream process.^[11] This is also true for software testing. A method called “model checking” now gains a keen attention in a design phase.^[26] The model checking is especially useful for the time critical system which is difficult with dynamic testing technique. The model checking does not test software itself but build a model of the system and test the model using a verifier. The system including hardware and software is represented as a state transition machine with temporal logic. There are skeptical opinions whether it is appropriate to include model checking in the software test technology, Not many software testing textbooks include model checking. While this technology has a long history as an automatic Theorem Proving or mechanical proving method, various tools have been developed recently and effectively used. SPIN^[27] is the most widely used tool.

5-3 Automation of software test

In general, the workload of software test would increase exponentially with software size. This is because when the number of conditions increases, the number of necessary tests also increases. To overcome this issue, it is natural to adopt the automation of tests with the accompanying quality

and efficiency. The following methods are available for software test automation.

- Automatic generation of input load such as load tests on the Internet
- Automation of tests by test script languages such as TTCN-3
- Automatic generation of test cases using UTP (UML Testing Profile) etc.

On the other hand, there are critical opinions against the introduction of automation tools.^[28] This is because the ad-hoc automation or random introduction of test tools produced confusions. In order to succeed in automating a system, an adequate and enough preparation is needed such as to develop human resource of skilled engineers and to nurture the necessary environments to produce successful results.

6 | *Software test industry*

Increased importance and workload of software tests has let some companies to outsource the testing activities to other companies. This is mainly because it is required (1) to decrease the cost related to testing, (2) to adjust manpower in a limited time frame and (3) to get specialized knowledge and skills in testing from outside. In particular, the number of test required for recent embedded systems has become enormous, so companies tend to rely further on outsourcing. With this circumstances, IT Verification Industry Association (IVIA)^[29] was established in Japan in 2005, which has 47 member companies as of March 2008. IVIA has divisions such as Technology, Standardization, Education/Training and Alliance etc., and especially puts their efforts on the skill certifications for test engineers. According to IVIA, the companies in test industries reached around 1000, and 100million yen sales. They view this industry has a strong potential of growth.

While it is true in software development in general in Japan that test engineers of large companies has more skills than those who are individual contractors or of small and mid-sized companies, large organizations tend to treat tests as lower class than planning, development, manufacturing or sales so that newly-hired or

temporary/extra workers are considered to be enough for testing. Moreover, even a test period could be significantly shortened to compensate the delay of development, which represents a distorted situation that the local politics gains precedence over the fulfillment of testing activities which has an overall importance for quality. This is another reason to bring the birth of test industry.

This movement of independent test industry happens not only in Japan but in India where software industries is booming as a major national business. For instance, an Indian company specialized in software testing named STAG has their office in Japan and the United States. Infosys, the largest software company in India has Independent Validation Center (IVC). This is an individual organization within the company with 3000 employees and rapidly growing. IVC has Chinese Walls for software that is developed by Infosys.

These testing companies and organizations have increased because the software usage has increased and the companies who try to use the software are not able to afford personnel to test the software. In the current circumstance where new technologies are introduced every month while software gets more complicated, it is more effective to employ specialized companies or organization to do the testing job in terms of cost, work period and quality. Infosys IVC is capable of maintaining a knowledge base to evaluate requirement specification according to its development schedule, work items and expected results. Specialized companies are able to use their experiences based on many actual cases, so that their know-how could be utilized by outside companies.

7 | *Education of test engineers and designers*

The needs of highly skilled IT personnel have been discussed in many occasions for the past few years. However, these discussions have not come to the need for personnel for software test yet.^[18]

In fact, the needs of software test specialists have been discussed by concerned parties for years. However, very little attempt has been conducted by universities or colleges to have a special course on software testing. In the reference^[21] published in

1999, no student has got software test course and the situation was regarded as hopeless.

In Japan, IVIA, the software test company association as mentioned earlier, and other Non Profit Organizations have been providing a certification for software test skills. One of the most representative organizations is Association of Test EngineerRing (ASTER) established in April 2006. ASTER holds an annual software test symposium called JaSST to encourage technical and human development in the field of software testing.

There is a world-wide organization named International Software Testing Qualifications Board (ISTQB) supporting skill certification for software test engineers.^[31] According to its website, 39 member countries are participating including Japan as of February 2008. ISTQB's member organization, called Japan Software Testing Qualifications Board (JSTQB) provides certification for software test engineers.^[32] The website of JSTQB shows the data about the number of certified engineers in 20 member countries and regions. It shows that there are about 24,000 certified engineers in the world and more than 1000 in Japan. Information-Technology Promotion Agency, Japan (IPA) has developed ITSS (IT skill standard) and ETSS (embedded software skill standard). ITSS "systematically summarizes abilities/skills required to provide various IT related services, and to provide a scale (common framework) effective to measure the level of performance of IT service professionals in the academia and industries". ETSS aims to "enhance the human ability for embedded software development and to develop mechanisms to achieve human resource development and exploitation for embedded software development".

ITSS was originally developed in 2002, and has "testing skill" in its skill item for system, database, network and other function skills required for IT specialists. For application specialists, "testing skill" is found among the required methodology for software engineering. However, there is no job category for testing specialists. In the job category of consultant or IT architect, no testing skill is included. Only the various testing related skill items are listed.

Meanwhile, ETSS developed in 2005 lists "test

engineer" as a job category, because test engineers are much more important for embedded software. ETSS specifies required skills for test engineers.

Japan Electronics College has established a new course called Software Test Design Course in April 2008 to cope with the recently increased demand of software test specialists. This 2-year course provides curriculum for test engineering. According to its website, the course addresses: testing method, testing environment development skill, test management skill, performance evaluation skill, quality management, idea development skill, embedded system, and case studies. The pre-requisite skills and knowledge are also handled including computer in general, programming, and networking skills. The college says that many companies are interested in employing the graduates so that the students need not worry about their employment. However, there is a concern if high school students could have a favorable impression on the word "test".

Looking at the education overseas, for instance, IT companies in India are famous on their employee training. In the human development program of IVC in Infosys, they aim at not only testing skills but also other skills in technology, quality, process, application and even behavioral abilities such as leadership and communication skills. They also provide career opportunities for test engineers to become the board member in testing divisions.

Product tests are not just a series of trial use. To conduct a test will require special knowledge about the product and the user. Test engineers must precisely understand the product perspective with its strategy and value proposition, which is equivalent to understanding of the status, strategy and value of the organization for the product. That is, test designers/engineers should share the sense of issues that the corporate management have.

In this sense, those who are capable of leading software testing are very valuable, so that there might be a discussion whether it is appropriate to have them just to handle testing. Human resources related to software testing should play a significant role to evaluate various risks related to quality of products as well as to ensure the "quality of organization". Likewise, it will be necessary to foster software testing engineer/disigner/manager

in the advanced education in universities and graduate schools. Enterprises are also required to promote the personnel who are specialized in software testing to higher states/positions.

8 | Software risk management and its future

Software test specialists are required to have communication skills to talk to development engineers to avoid any unwanted defects in quality of products, as well as specific test skills. Moreover, from the viewpoint of risk management, it is required to have management skills to consider the influence of inconvenience due to the society in large. Nonetheless, some companies are still putting beginners on the testing job considering that they can do the test without any special skills. Software are becoming further complicated and the whole world is relying more heavily on software. Therefore, even though software test technologies will advance, our concerns on serious software accidents will continue to exist and to grow. Considering the significance of risk management, it would be necessary to create a social framework to this issue. This is also related to the needs of compliance in recent years.^[33-38] Hence, here is a proposal for software risk management in a wider perspective.

8-1 *Knowledge sharing through the establishment of accident investigation committee*

In case when any failure of software system occurs that could affect the society, the accident investigation committee analyze the software system thoroughly to prevent similar accidents, and our society and engineering community can learn these lessons to advance the software systems. There would be several possibilities how to form such committees. One would be to consist of contractor and owner/orderer. The other is to have outside organization where the concerned party would not participate in. Their mission is not to determine who is responsible but only to analyze why the accidents have occurred. In the current government, local or central, procurement operation, most are divided up according to work phase, so that multiple contractors are participated

to make a whole system. If any accident occurs, various parties would be involved, and this will add the complexity to handle the problems.

8-2 *Software test audit by the third party*

Software systems that may affect the society should require audit by the outside third parties other than the owner/orderer and contractor. This auditor will compile the results with the software test and put the documents in the third party archive. Once any inconvenience occurs to the system, any people to solve the problem will immediately access the archived record and act any adequate operations. Such organizations are required to have the capability of comprehensive quality evaluation and risk management including extensive software testing capabilities.

8-3 *Legal systems to handle software failures*

The current product liability laws (PL Laws) would not apply to software. However, as there are requests to apply PL laws even for software because accidents could occur by the failure of software of the system, there is a possibility that the law will be enforced for product liability concerning software. In this case, it is desirable to have laws and regulations to minimize any damages of accidents. This may handle - the current software license agreement (End User License Agreement, EULA) - which claims entire product liability to the producer. One issue would be how to control potential damages due to the modification by users or outside parties, while promoting the technical development.

8-4 *Study on terminology, description and representation, and basic knowledge of software*

Software related accidents and problems may only be attributed to software specialist, and the organization on the whole may not understand the issue. This may lead to a neglect of software quality because it is not perceived as the issue for whole the stakeholders. In order to avoid this situation and to share the successful results of e-Japan strategy by all citizens in Japan, it is necessary to make efforts to promote researches on software terminology, description and representation so that general people and managers of organization can

easily understand software related accidents and issues. Activities for training and proliferation for software to general public would also be required.

These kinds of activities described from 8-1 to 8-4 have potentials to bring an opportunity to review the quality of Japanese systems including products made in Japan, and to be an activity to widely provide quality related services not only to a company but also to the society, which is “industrialization of quality”, so to speak. If software test is treated from such viewpoints, investment in software test personnel including researchers, designers, engineers and managers will not only enhance the industrial competitiveness of Japan but also to create a new knowledge industry to contribute to the safety and security of Japan and the rest of the world.

9 Conclusion

As software tests have been conducted as a final process in conventional software development, some are mixed up with debugging, which caused problems. Now, in recently years, software test as a quality assurance and risk management activity gains a lot more attention and appreciation.

This is because embedded software system has become widely use, though which has led unexpected troubles by unexpected use by users. Some troubles have had caused a significant influence on our daily lives. These problems cannot be solved only by debugging, but by conducting various software test both in and around the system where every possible cases are covered. Technical development and system and environmental maintenance should be conducted to create a quality and reliable software, along with the promotion of software test development, fostering its specialists. In fact, due to both the importance and difficulty of software test, new companies and organizations specialized in software test have emerged and their business is steadily growing.

These are “industrialization of quality” of software so that it should be emphasized that investment in software test researchers, designers, engineers and managers would not only enhance the industrial competitiveness of Japan but also to create a new knowledge industry contributing to the safety and security of the country and the

world.

Acknowledgment

We would like to extend our gratitude for providing invaluable information in this project to the following people: Professor Zengo Furukawa, Reliability Information System Engineering Department, Faculty of Engineering, Kagawa University, Lecturer Yasuharu Nishi, Faculty of Electronics and Communication, The University of Electro-Communications, Takahiko Kosuge Principal Researcher, Open Source Education System, Japan Electronics College Dr. Masashiro Tsuruho, Director of Software Engineering Center (SEC), Information-Technology Promotion Agency, Japan (IPA), Hiroshi Monden, Project Leader of Embedded Software Project, SEC, IPA, Dr. Masayuki Hirayama, Researcher, Embedded Software Project, SEC, IPA, Yurie Ito, Director of Technical Operation, Office of Business Planning, Japan Computer Emergency Response Team Coordination Center (JPCERT), Takayoshi Shiigi, Senior Analyst, JPCERT, Kazuya Togashi Chief System Architect, Office of Business Planning, JPCERT, Hirohisa Furuta, Information Security Analyst, Group Manager, Information Coordination Group, JPCERT, Kenji Onishi, Senior Consultant, Enginnering Solution Division, Mamezo Corporation, Toru Matsuoya, President and CEO of Debug Engineering Research Institute, Dr. Juichi Takahashi, Distinguished Engineer, Software Assurance Dept. Platform Technology Division, TV Business Group, Sony Corporation, Kiyotaka Asai, President and CEO of Veriserve Corporation, Nobuhito Katsumata, Manager, and Katanori Sasaki, Manager, Veriserve Corporation.

References

- [1] Nikkei ITpro, Tokyo Stock Exchange New Derivative Selling/Buying System Restored, due to initialization error of the memory : <http://itpro.nikkeibp.co.jp/article/NEWS/20080212/293526/> (Japanese)
- [2] Mainichi Newspaper, Train Station Ticket Gate Machine Trouble: confusion of commuters in metropolis, October 12, 2007No.42, Special Article 1, 2004 (Japanese)
- [3] Asahi Newspaper, A mistake of a text led a big trouble: Ticket Gate Machine Trouble in

- Tokyo Area, October 28, 2007 (Japanese)
- [4] Teiichi Aruga, Another trouble of train station ticket gate machine, no lesson was learned from the past, Future of IT industry, NikkeiNet, October 16, 2007: http://it.nikkei.co.jp/business/column/aruga_gyokai.aspx?n=MMIT0z000016102007 (Japanese)
- [5] Teiichi Aruga, “Suica” troubles at train station ticket gate, Accountability of JR, Future of IT industry : http://it.nikkei.co.jp/business/column/aruga_gyokai.aspx?n=MMIT0z000004122006 (Japanese)
- [6] Naohiko FUKAYA, Zengo FURUKAWA, Yasuharu NISHI, special issue: New Trend of Software Test, Journal of IPSJ, Vol 49, No.2 126-173 (Japanese)
- [7] Glenford Myers, The Art of Testing, First Issue in 1979, Second Edition in 2004.
- [8] SQuBOK Committee, Software Test Quality Knowledge system Guide – SQuBOK Guide, Ohmusha, 2007 (Japanese)
- [9] JIS X 0160: 1996, Software Life Cycle Process, Japan Standards Association, 1996 (Japanese)
- [10] JIS X0170 : 2004, System Life Cycle Process, Japan Standards Association, 2004 (Japanese)
- [11] Toshiaki KUROKAWA, Toward the Improvement of Quality and Reliability in Information Systems Construction – A Study of “Business Rules” and Requirements Engineering in the Upstream Process –, Science & Technology Trends – Quarterly Review, No.11, April 2003, pp.19-29
- [12] Software Engineering Center, Information-Technology Promotion Agency, Securing a required quality with involvement of corporate management – Key of IT business from the upstream process. SEC BOOKS, Ohmusha, 2005 (Japanese)
- [13] Susumu HAYASHI, Toshiaki KUROKAWA, The Two Rationalities and Japan’s Software Engineering, Science and Technology Trends - Quarterly Review, No. 14, January 2005, pp.41-53
- [14] Kent Beck, Test Driven Development: By Example, Addison-Wesley Professional, 2002
- [15] Susumu Hayashi, Pan YiBing, Masami SATO, Kenji MORI, SUL Sejeon and Shusuke HARUNA. Test Driven Development of UML Models with SMART modeling system. <<UML>> 2004 – The Unified Modeling Language, Lecture Notes in Comp. Sci., No. 3273, Springer-Verlag. pp.395-409, 2004
- [16] Katsuhiko KAJI, New phase of IT policy and expectation to SEC, SEC Forum 2006, 2006 : https://sec.jpa.go.jp/download/dl.php?filename=event/2006/20060612/secforum2006_0_kazi.pdf (Japanese)
- [17] Industrial Structure Committee IT Economy Group, IT Service Software Subcommittee Interim Report, June 2006 : <http://www.meti.go.jp/press/20060613010/torimatome-hontai-set.pdf> (Japanese)
- [18] Toru YAMASHITA, Proposals for Advanced IT Human Resources, NHK Shuppan, 2007 (Japanese)
- [19] Ministry of Economy, Trade and Industry, Information Policy Agency, Information Policy Unit Information Processing Unit Information Process Promotion Section, 2007 Embedded Software Industrial Activity Research Report, June 2007 (Japanese)
- [20] Kenji ONISHI et al., JSTQB textbook, JSTQB certification test engineer, Foundation Level Test, Shoeisha, 2007 (Japanese)
- [21] Cem Kaner, Jack Falk, Hung Quoc Nguyen, Testing Computer Software, 2nd Edition, John Wiley & Sons, 1999
- [22] Kaner, Cem, James Bach, Bret Pettichord (Eds.) Lessons Learned in Software Testing, John Wiley&Sons, 2002
- [23] Masataka Yoshizawa, Koichi Akiyama, Taro Sengoku, Software Test HAYST Method Basics, How to Improve Quality and Productivity, Nikkagiren, 2007 (Japanese)
- [24] Susumu Hayashi, Information and Communications Technology and Shiso – Shiso as a Capability for Science and Technology, Science and Technology Trend Quarterly Review, No. 23, April 2007, pp.11-22
- [25] Alain Abran and James W. Moore (Executive Editors), Pierre Bomrque and Robert Dupuis (Editors), Guide to the Engineering Body of

- Knowledge, 2004 Version, SWEBOK, IEEE Computer Society and IEEE,2004
- [26] National Institute of Advanced Industrial Science and Technology, Model Inspection to learn in 4 Days, NTS, 2006 (Japanese)
- [27] ON -THE -FLY, LTL MODEL CHECKING with SPIN: <http://spinroot.com/spin/whatispin.html>
- [28] Toshikazu TAKAHASHI, Chishiki Zero kara Manabu Software Test, Software Test to learn for Beginners, Shoeisha, 2005 (Japanese)
- [29] IT Verification Industry Association(IVIA) website: <http://www.ivia.gr.jp/index.html> (Japanese)
- [31] International Testing Qualifications Board(ISTQB) website: <http://www.istqb.org/>
- [32] Japan Testing Qualifications Board (JSTQB) website: <http://jstqb.jp/index.html> (Japanese)
- [33] Hisamichi OKAMURA, Kaisha no Naibu Tosei, Corporate Compliance, Nikkei Shinbun Shuppansha, 2007 (Japanese)
- [34] Financial Services Agency, SOX Laws and Financial Services Act and its Enforced Regulations, February 2007 (Japanese)
- [35] Financial Services Agency, Financial Statement Compliance Evaluation and Supervision Standard related to Financial Statement, February 2007 (Japanese)
- [36] Financial Services Agency, Financial Statement Compliance Evaluation and Implementation Standards concerning Supervision, February 2007 (Japanese)
- [37] Ministry of Economy, Industry and Trade, System Management Standard Supplemental Information (IT Governance Guidance concerning Financial Statement), February 2006 (Japanese)
- [38] Ministry of Economy, Industry and Trade, System Management Standard Supplemental Information (IT Governance Guidance concerning Financial Statement) Annex, December 2006 (Japanese)



Toshiaki KUROKAWA
 Affiliated Fellow, NISTEP
 CSK Holdings Corporation, CSK Fellow
<http://www.csk.com/index.html>

Worked for Toshiba and IBM before joining CSK. Specialized in programming language, object-oriented programming and meta data standardization. Interested in the upstream process of system development, services sciences, science and technology community, and cloud computing.



Masato SHINAGAWA
 Affiliated Fellow, NISTEP
 Academic Professor, Hosei University Information Technology Research Center

1967- 2000, MPT (Ministry of Post & Telecommunications)
 2003-2006, SI business
 Interested in correlation between natural science and social science.

(Original Japanese version: published in April 2008)