DEVELOP QUALITY CHARACTERISTICS BASED QUALITY EVALUATION PROCESS FOR READY TO USE SOFTWARE PRODUCTS

Daiju Kato[1] and Hiroshi Ishikawa[2]

[1]WingArc1st Inc., Tokyo, Japan
kato.d@wingarc.com
[2]Graduate School of System Design,
Tokyo Metropolitan University, Tokyo, Japan
ishikawa-hiroshi@tmu.ac.jp

ABSTRACT

The users who use ready to use software product had better get the products' quality information with classified by some kind of global standard metrics or technique for their evaluation. But many of those software products' companies don't provide the quality information because of the products are developed by their own development and evaluation process. But those users want to get quality information to reduce evaluation cost. Therefore, we develop our quality evaluation process with using quality characteristics of software external quality model on ISO/IEC 9126-1 for our software products. This evaluation process has feature of unsynchronized quality evaluation process toward development process. Also, the process starts from definition of classified quality requirement based on quality sub-characteristics at test planning process to quality analysis at test completion process under software development cycle. We provide precise quality data of our product for our users through this evaluation process and the process might be reduced cost of our customers' evaluation.

KEYWORDS

Evaluation process, Quality model, Quality analysis, Software development cycle, Ready to Use Software, ISO/IEC 9126, ISO/IEC 25051

1. INTRODUCTION

Many companies develop their own corporate IT systems to combine or be based on ready to use software product. They evaluate those software products to fit their systems and to determine those quality by themselves. This evaluation is very important for suitability and reliability of their systems, but however, many companies can't have enough cost and time for the evaluation. This concern becomes industry-wide issue and many countries have established products' certification programs based on ISO/IEC 25051[1]. For example, The Computer Software Association of Japan (CSAJ) has started PSQ Certification System [2]. According to CSAJ web site, this certification has responsibilities blow.

PSQ Certification System is the certification program conforming to international standards, which confirms and certifies that the documents and the functions of the software product are consistent. The system therefore evaluates descriptions including the product description (catalogue), user documentation (e.g. manuals etc.), and test along the document to ensure

David C. Wyld et al. (Eds) : ITCS, JSE, SIP, ARIA, NLP - 2016
pp. 09–21, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.60302

compliance using test report. Certification will then be issued to the authorized product and will receive the certification logo.

This kind of certification programs give positive effect to many users who plan to use certified software products. But the users need some evaluation costs to check whether the products have an adequate right quality and fit their corporate systems because the certified programs don't suggest the products' quality and what kinds of evaluation process are used. So, we believe our customers might reduce their evaluation cost if they get quality data to fit our customers' request.

Most of quality data as metrics are related to the development process. For example, bugs ratio, code review ratio, pass rate of test cases and test coverages are internal data under development projects and they are very difficult to be used key value indicator unless they are used d by the same development process. Those data are not suitable to standard scale for quality analysis without explanation of software development cycle.

It is necessary to solve this issue for providing useful quality information for our customers. So, our QA department had started to classify our products' quality by using quality characteristics which are specified in ISO/IEC 9126-1 [3]. But we realized that we have to spend much cost to analyse our quality using the quality characteristics after end of development projects. Also, classified quality by the characteristics doesn't fit basic development process, like V-model or agile model. Therefore, we considered to develop our own evaluation process using by quality characteristics. This process needs unsynchronized toward development process and covers from test planning process to completion process within software development cycle. Also, we finally need to provide detailed test reports analysed by quality characteristics.

2. BACKGROUND AND PROBLEM

Our company has developed and maintained several software products. But we don't have companies' standard development process and each development team decided development process to suite their development style. Each QA team needed to create testing process to fit each products. Unless using same development process, each QA teams had accumulated quality metrics data, as like bugs ratio or test cases density per source codes, to use judgement for product shipment. Those quality data depending on each development process and they were not used to compare our products' reliability. Also, it was definitely difficult to provide suitable quality information for our customers for their evaluation of our products. Quality model using quality characteristics specified in the ISO/IEC 9126-1 was some of key solutions to classify software quality. But it was very difficult to classify the result of our test case in each test types to fit quality characteristics. Therefore, we have started developing evaluation process to use quality characteristics.

3. DEVELOPING QUALITY EVALUATION PROCESS

After analysing our evaluation process to define by quality characteristics, we realize that we need to divide our evaluation process from development process. But complete splitting out from development process raised our test cost because of rework due to bugs. We have placed the point as a milestone which can synchronize development process and evaluation process to solve for this rework. We decide to use quality characteristics for our key value indicator.

Our evaluation processes have three process, test planning, test management process for monitoring and controlling verification and test completion process. Verification period has several test levels which we call test stages. We define test planning period as alpha stage, test verification period as beta stages and test completion period as RC stage and those stages are timely continuous, “Figure 1”.

But beta stages as verification period doesn’t need to keep turn like waterfall model. Each beta stage has started from acceptance test for entrance criteria of test levels and QA managers can decide whether test teams start the test level or not.

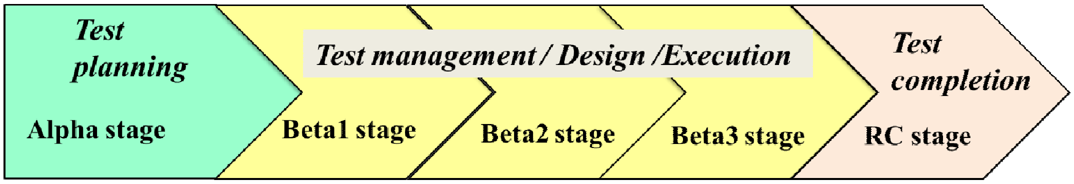

Figure 1. Brief of evaluation process

3.1. Test planning process

Test planning process includes developing master test plan (MTP) and level test plan (LTP). MTP and LTP are finally authorised by stakeholders. QA managers write MTP with some of the test baseds like product requirement plan, marketing requirement documents and product vision documents. We use test document template which is specified in IEEE 829 [4]. QA managers determine test strategy for their project and their think of test approach.

After test strategy is established, QA managers focus on definition of quality requirement for shipment of our target software product. The quality is classified by each quality sub-characteristic, “Table 1”.

Table 1. Example of definition of quality requirement

Characteristics	**Sub-characteristics**	**Target quality**
Functionality	*Suitability*	• The development requirement of new features have been verified. • Migration from past version are possible and compatibility is collateral.
	Accuracy	• The results of new functional requirements are correct and have been verified. • Each product definition files developed by past version are worked properly. • All functions under English and Chinese environments have same quality as Japanese environment. • The result of scenario test has no problem under the estimated users operations.
	Interoperability	• The results of combination of other WingArc products have no problem.
	Security	• No vulnerability is found including used OSS components and web interface..
Reliability	*Maturity*	• Each test level has analysis action for test coverage and turn-around time to fix bugs. • Each test level has quality improvement action derived from quality analysis of previous test level. • All bugs are verified at RC stage.

	Fault Tolerance	• Operations have perfect continuous even though one node is stopped under clustering environment. • Migration program is keeping on running under some of definition files have some errors.
	Recoverability	• Past version of definition files can be saved even though the files have some errors.
Usability	*Understand-ability*	• There is no difference between operation steps in users' manual and actual operation. • Localized UI provide same quality as Japanese environment.
	Learnability	• Sample files and tutorial manual can be used self-study materials.
	Operability	• Scenario test includes behavior and procedures which can be aware of target customers' operations.
Efficiency	*Time Behavior*	• Performance is less or equal to maximum 3% than past version. • Concurrent multi-access test is performed with no error. • There is no high CPU load or I/O load condition under some functional operations.
	Resource Utilization	• There is no memory leak under usual operations. • Memory and I/O resources are effectively used.
Maintainability	*Analyzability*	• Functionality of debug logs are implemented and the logs can be used for error analyzing.
	Changeability	• External API is easily extended.
	Stability	• The result of Load test have no error.
	Testability	• Testing layers are prepared between client and server communications.
Portability	*Adaptability*	• Product is perfectly running under supported environment.
	Installability	• Product is easily installed under supported environment.
	Co-Existence	• Product can be used with antivirus software and backup software.
	replaceability	• Product is easily replaced from past version.

QA managers define test level to accomplish the definition of quality requirement. Each test level also has quality requirement classified by quality sub-characteristics and each test level has several test types to fit the requirement and results of each test types become the evidence for each target quality classified by quality sub-characteristics, "Figure 2".

QA managers keep controlling cost of testing for functionality characteristics doesn't exceed 70% in total cost of all testing when QA managers develop for planning of test types. This reason is that verification is not focused on functional test and concentrate on quality balance. Reference value of 70% is composed of our statistical data of past projects.

There are two way to fit test types to each test level. One way is that QA managers list up test types first and looks for dependency of each test type. After defining the dependency, QA managers are mapping each test type to suitable test level referred by milestone, defined synchronized point to development process.

The other way is that QA managers define quality of each test level referred by milestone and look for the test types to meet. We choose the way whether development process is clear or not. After mapping test types to each test level, QA managers define quality characteristics for each test level like "Table 2".

QA managers also write organization of test teams, test bases, verification schedule, product and project risk, training plan, policy of criteria, policy of test development and used metrics from development process, to MTP referred by IEEE 829 template.

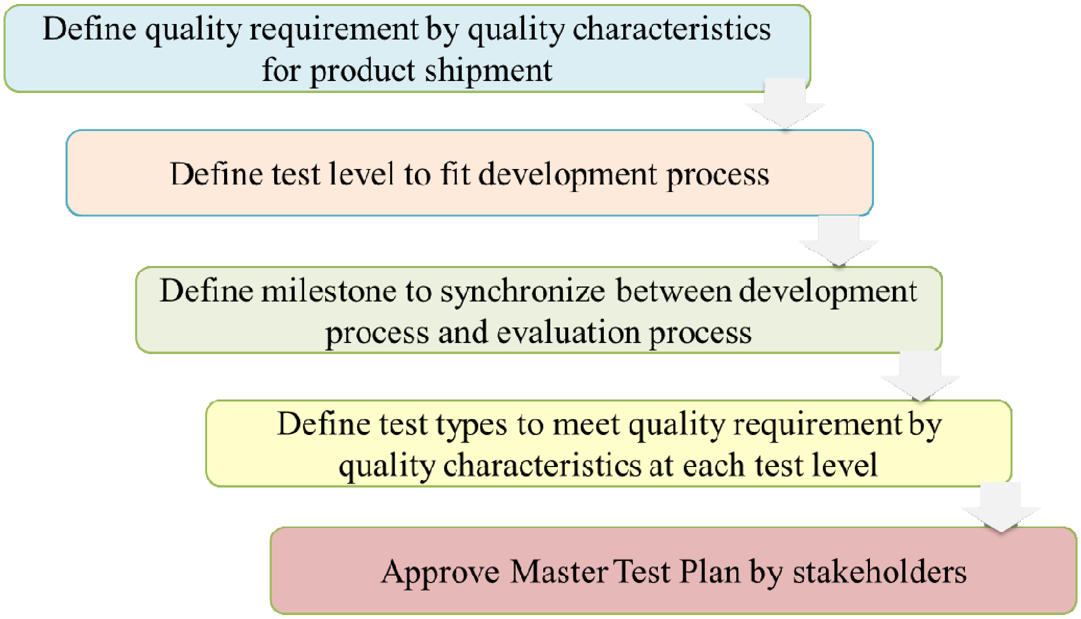
Define quality requirement by quality characteristics for product shipment
Define test level to fit development process
Define milestone to synchronize between development process and evaluation process
Define test types to meet quality requirement by quality characteristics at each test level
Approve Master Test Plan by stakeholders

Figure 1. Outline of MTP developing process

Table 2. Mapping of quality characteristics to test levels

Stage	**Quality characteristics**					
	Functionality	*Reliability*	*Usability*	*Efficiency*	*Maintain-ability*	*Portability*
Beta1	◎	○	○	○		
Beta2	◎	○	◎	○	○	
Beta3	◎	◎	◎	◎	○	○
RC	◎	◎	◎	◎	○	○

After writing up MTP, QA managers or QA leaders start writing LTP. Basically LTP in our evaluation process is only acceptance test plan for each test level. Policy of criteria for acceptance test is defined in the MTP. This acceptance test is entrance criteria whether we can start the test level or not.

While QA managers write the MTP and LTP, QA teams start test designs and analyse software product by test types. QA managers manage test designs with test management process.

3.2. Acceptance test for test level

Acceptance test is entrance criteria for each test level. QA managers select less than 10% of test cases from all test types with using stratification method and minimum set of all test types operated in the test level, “Table 3”.

Our evaluation process defines period of acceptance test is within one week including all test execution and test analysis. This rule means that we cannot spend much time if acceptance test is failed.

According to this acceptance test, QA teams can reduce much rework because QA teams realise the situation of software quality under development project. QA teams can request quality improvement to the development with static testing if result of the acceptance test doesn’t reach the criteria. QA managers can plan and evaluate several times of acceptance test unless they couldn’t pass the criteria. QA managers can change criteria policy for the acceptance test under negotiation with development teams.

By acceptance test as this entrance criteria for test level, QA teams can execute several beta stages at same time even though QA teams don’t finish prior beta stages. For example, our QA teams can execute beta1 stage and beta2 stage simultaneously unless both acceptance tests are passed. This acceptance test as entrance criteria helps to reduce test cost.

Table 3. Example of acceptance test

User Type	**Test Types**	**Guarantee of sub-characteristics**
Beta2 Stage		
New	New requirement test	Suitability Accuracy Interoperability Security Fault Tolerance Recoverability Resource Utilization
Upgrade	Compatibility test	Suitability Accuracy Interoperability Security Maturity Fault Tolerance Recoverability Resource Utilization
New/Upgrade	Performance test	Time Behavior Resource Utilization
Beta3 Stage		
New/Upgrade	Mobile function test	Suitability Accuracy Interoperability
New/Upgrade	Secinario test	Suitability Accuracy Interoperability Security Maturity Fault Tolerance Recoverability

		Understandability Learnability Operability Stability
New/Upgrade	Quality inprovement test at beta2	Suitability Accuracy Security Maturity Stability
New/Upgrade	Performance and load test	Time Behavior Resource Utilization Stability
New/Upgrade	Platform test	Suitability Accuracy Interoperability Operability

3.3. Test management process as monitoring and controlling verification

Once the acceptance test is passed, QA teams execute all of test types as planned for each test level. Each test types become tied to quality sub-characteristics performed at each test level. For example, performance test ensures of Efficiency-Time Behaviour.

In test designing, QA teams develop test viewpoint for each test type. Test viewpoint is bird's-eye view representation to analyse test. We develop test viewpoints by design base or requirement base and we define quality sub-characteristics to all test viewpoints. So, all test cases derived from test viewpoint also have relationship with some quality sub-characteristics.

QA managers check progress of test execution to each test type. Also, they manage bugs founded by those test types. Those bugs can also mapping to quality sub-characteristics because test cases have definition of quality sub-characteristics, "Table 4". QA managers can easily analyse what the kind of sub-characteristics is weak.

Table 4. Example of found bugs at beta2 test stage

Characteristics	Sub-Characteristics	Total test cases	Total Bugs
Functionality	Suitability	848	7
	Accuracy	9662	383
	Security	97	2
Reliability	Maturity	38	19
Usability	Understandability	1543	60
Efficiency	Time Behavior	1414	8
	Resource Utilization	1414	8
Portability	Installability	332	6
	Replaceability	1513	55

Basically, definition of quality classified by quality characteristics is qualitative analysis, but, QA managers define pass rate and test density for each test type derived from development process. QA manages can analyse quality with both qualitative analysis and quantitative analysis.

Basically our test management process is no difference as usual test management process even though quality characteristics are key value indicator. QA managers focus on progress of both test design and test execution compared with plan and real progress.

3.4. Testing for 'Quality in Use'

Quality model which is specified in ISO/IEC 9126 defines 'Quality in Use', "Figure 3". This quality characteristics need ready to use software for operation and sustain. Lack of this quality rises total maintenance cost even though internal software quality is mature. But testing for 'Quality in Use' is difficult for ready to use software. One reason is our software customers use our products with other software and their own system, therefore, we can't estimate benefit of our product easily. Another reason is characteristics at 'Quality in Use' is completely difficult to fit within development project.

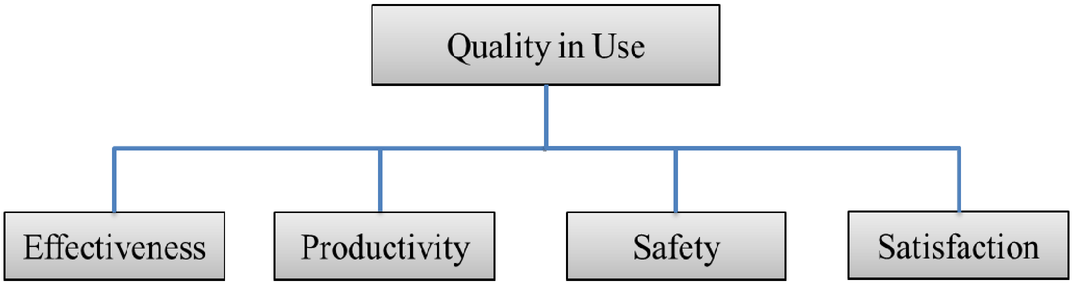

Figure 3. Definition of 'Quality in Use'

Our QA teams evaluate quality characteristics on 'Quality in Use' with scenario testing and our own test method. We define method of 'Manual based Testing', "Figure 4". The method of manual based testing is not testing for user manuals or some kind of documentations. The testing method below has three approaches and their methods evaluate conjunction of software and user manuals.

- Define four actors referred by persona for developing product and use cases
- Classify all functionalities by actors
- List up view points for reading and searching manuals by actors

We defined four actors: administrators, developers, power users and general users. Also, we estimate some of functionalities used by specific actors and general users can't distinguish between individual system and ready to use software. So, we verify whether each manuals are suitable for actors using by developing test cases. Using this method, we can check the quality whether concept of functionalities and user documentations are matched completely. We also define each use cases testing by actors.

We estimate unclear descriptions at manuals connect to lack of the quality, satisfaction characteristics. Confusing and vague functionalities meets lack of productivity characteristics. Also, Confusing manual is difficult to operate for administrators, it is considered as lower quality.

We can evaluate and guarantee quality characteristics at 'Quality in Use' by use case based scenario testing and manual based testing.

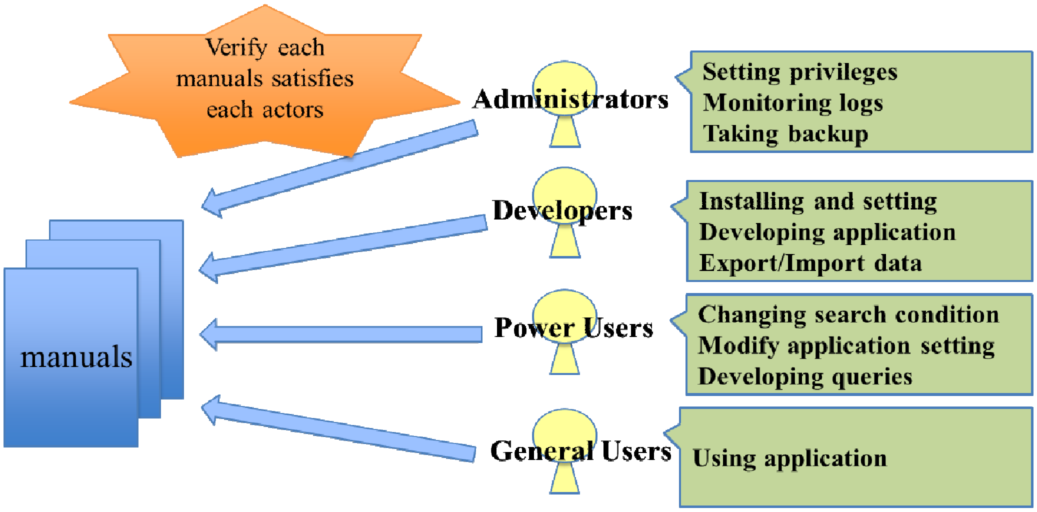

Figure 4. Concept of manual based testing method

3.5. Test completion process

A managers determine test level completion from result of each test type and verifies whether cause of all bugs found at the test level are identified or not. QA managers can authorize completion of the test level when all condition of test completion defined by MTP are verified.

QA manager can add extra test for quality improvement if QA managers judge QA team need to find remaining bugs.

At RC stage, QA managers judge whether all quality definition written in MTP is completed entirely or not. QA managers can declare end of all test.

Same as test management process, test completion process is no difference as usual process except we deal quality characteristics with analysis method.

3.6. Quality report for customers

After end of all test, QA managers write test report. Test report is summarized result of all test. QA managers write view of quality analysis classified by quality characteristics, as like bugs rate, “Figure 5” and test cases rate, ”Figure 6”. QA managers describe the reason whether the project is achieved the quality standard of our company for product releasement. The standard of our company has four rank starting from B to AAA as like automobile car safety integrity. For example, our quality standard of AAA needs that bugs rate is less than average of 3% through evaluation process, test density(the number of test cases per kilo step of source codes) than 50, and all quality characteristics are guaranteed with several test types.

QA managers need to describe product quality covered with both qualitative analysis and quantitative analysis. All test results and data, especially performance test, multi-concurrent access testing and load verification, and results of acceptance test at each test level is put at appendix section. Those test results become evidence guaranteed to each quality characteristics. Our test report is finally authorised by stakeholders and becomes complete evidence of project. We also provide this test reports to our partners. The partners can easily judge our products’ quality and get variety of quality data at appendix section. Also, the partners can provide those quality data to their customers who use or evaluate our developed products.

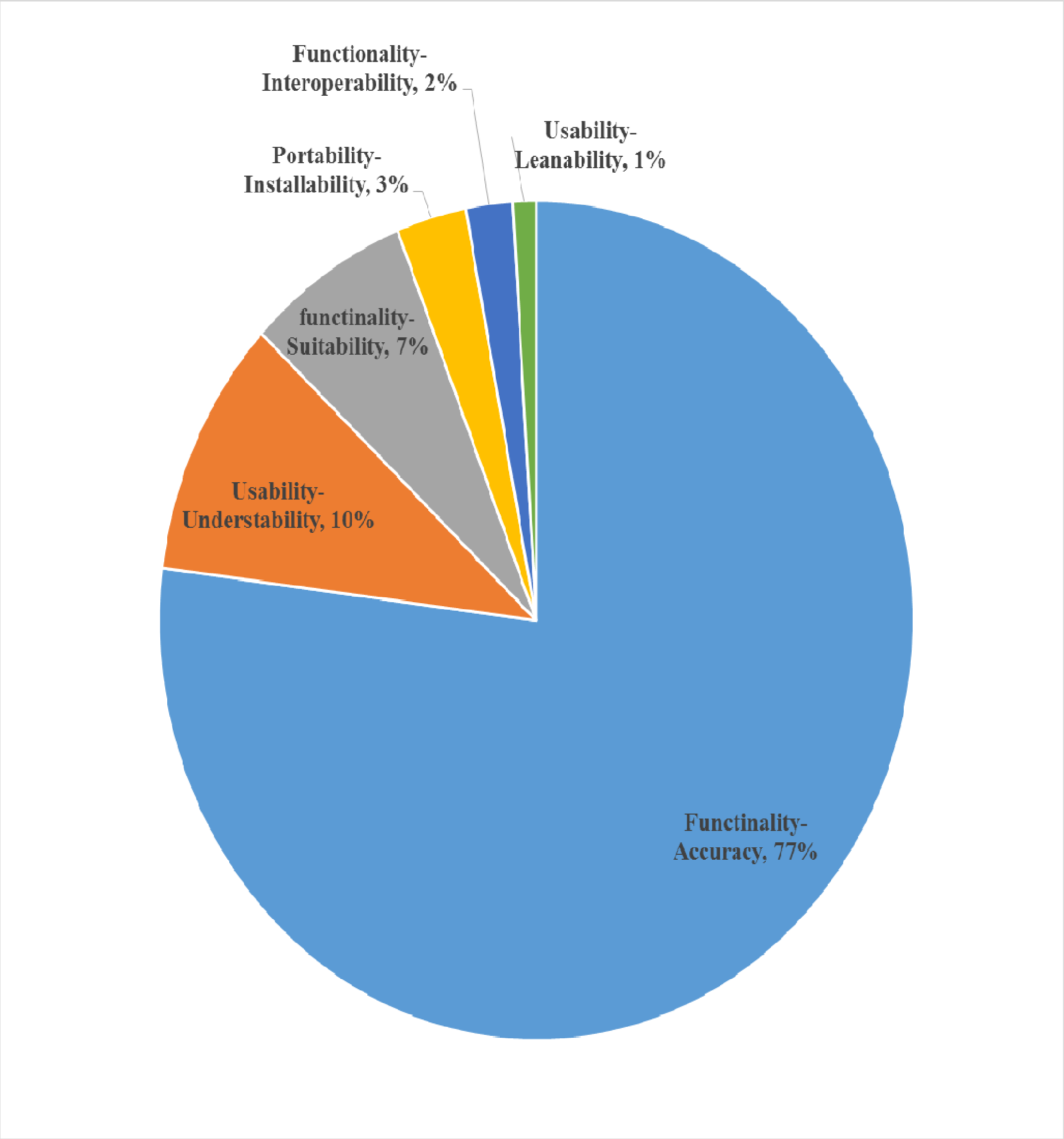

Figure 5. Bug rates classification by quality sub-characteristics

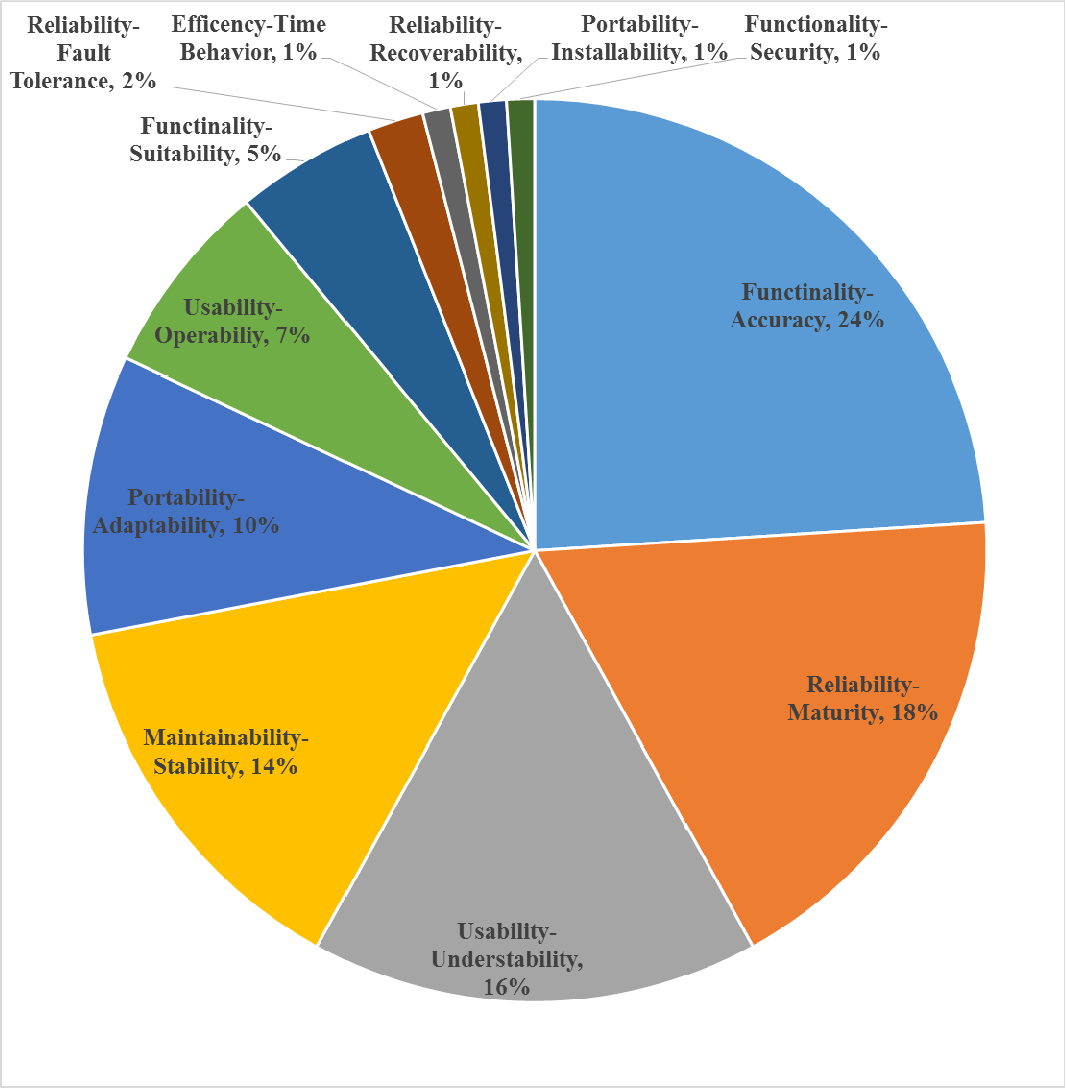

Figure 6. Test cases ratio classification by quality sub-characteristics

This approach of providing test reports with analysed quality characteristics for customers meets a part of concept of ISO/IEC 25051. According to this standard, customers can know risk, functional quality, performance, test results, existed bugs information before they buy or use ready to use software products. Our test report fits this think of the standard.

Our products have already get approval of PSQ certification by our test reports in Japan. We believe our evaluation process and our test reports have much benefit for our customers.

4. Effort And Benefit of Our Evaluation Process

Our evaluation process can fit any kind of development process unless it is development process for ready to use software product. We have already adapted on our evaluation process to 7 development projects to release our software products. We can use the process both major release and minor release without troubles.

The process has traceability with quality characteristics for key value indicator between MTP with test reports. This process is unsynchronized to development process, therefore, it is easily to add the process to usual development process, as like V-model, scrum process or agile process. We think other companies which develop software products can use our evaluation process with their own development process.

With using our evaluation process, we can reduce bugs found by our customers after releasement of our products and also total evaluation cost. We can estimate our evaluation cost by writing MTP and LTP in detail. QA teams can also develop test cases and execute the cases smoothly referred to MTP. QA teams reduce rework according to lack of quality.

The benefit of our evaluation process can easily provide quality analysis and metrics data based on quality characteristics. Our customers who receive our test reports can judge our products' quality fairly to fit their corporate systems. We believe our test reports reduce cost of customers' evaluation and validation whether our software products are suitable for their systems.

5. LIMITATION

Our evaluation process use quality characteristics based on ISO/IEC 9126-1. ISO/IEC 25051 is based on ISO/IEC 25000 series, SQuaRE [5]. Therefore, Compatibility or Security characteristics on ISO/IEC 25010 [6] are not fit our evaluation process. We need to change our quality characteristics to support ISO/IEC 25010.

6. CONCLUSION AND FUTURE WORK

Our evaluation process starts how to provide quality data of our software products with easy understanding and fairly to our customers. We want our customers to reduce much cost and time for evaluation whether our software products fit their corporate systems. We use quality characteristics at ISO/IEC 9126-1 to solve this issue and develop unsynchronized evaluation process toward development process from test planning process for MTP to writing test reports. Our evaluation process uses quality characteristics for key value indicator and acceptance test for entrance criteria to verify progress of quality at each test level. By definition of test level, our evaluation process is unsynchronized toward development process and the process has flexibility to meet variety of development process. We believe many companies can use our way easily to provide quality information by their test reports.

We have already started to implement quality characteristics based on ISO/IEC 25010 to satisfy ISO/IEC 25051. According to ISO/IEC 25010, "Quality in use model" is defined with sub-characteristics. We need to research mapping the characteristics at 'Quality in Use' to our use cases testing. Also, we need to improve manual based testing and define metrics for the testing. Approach of the testing method can improve quality both software and users manuals, but we believe much of unclear test description in the method.

We have also developed detailed process to comply ISO/IEC/IEEE 29119-2 [7] and also test documents are based on ISO/IEC/IEEE 29119-3 [8]. Our internal evaluation process, as like test planning process and monitoring and controlling process to comply this software test process.

After our evaluation process supports those new standard, as like SQuaRE and ISO/IEC/IEEE 29119, our process becomes key solution to comply ISO/IEC 25051.

REFERENCES

[1] ISO/IEC 25051:2014, Software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing

[2] PSQ Certification System, The Computer Software Association of Japan (CSAJ), http://www.psq-japan.com/

[3] ISO/IEC 9126-1, Software engineering -- Product quality -- Part 1: Quality model

[4] IEEE 829, IEEE Standard for Software and System Test Documentation

[5] ISO/IEC 25000, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE

[6] ISO/IEC 25010, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

[7] ISO/IEC/IEEE 29119-2, Software and systems engineering -- Software testing -- Part 2: Test processes

[8] ISO/IEC/IEEE 29119-3, Software and systems engineering -- Software testing -- Part 3: Test documentation

[9] M. Azuma, (1996), "Software Products Evaluation System: Quality Model, Metrics and Process - International Standards and Japanese Practice", Inf. & Software Tech., Elsevier, Vol.38, No. 3.

[10] M. Azuma, (2000), "QUALITY IN USE; Its Concept, Metrics and Methodology", Proceedings 2nd WCSQ 2000

[11] M. Azuma, (2011), "The impact of ICT evolution and application explosion on software quality: a solution by ISO/IEC 250nn square series of standards", WoSQ'11 Proceedings of the 8th international workshop on Software quality, Pages 1-2

[12] S. Ishikawa, T.Komiyama, (2013), "Systems and Software Quality: 6. Establishment of the International Standard based Software Quality Certification Scheme", Journal of Information Processing, the Information Processing Society of Japan, in Japanese.

AUTHORS

Daiju Kato is Quality Assurance Director at WingArc1st Inc. He is now a doctor course student of Tokyo Metropolitan University. His research interest includes quality assurance, test automation, and evaluation process and database technology. He has authored some books related with software testing and database skill in Japan.

Hiroshi Ishikawa received the B.S. and Ph.D degrees in Information Science from the University of Tokyo. After working for Fujitsu Laboratories and being a full professor of Shizuoka University, he is now a full professor of Tokyo Metropolitan University from April, 2013. His research interests include database, data mining, and social big data. He has published actively in international, refereed journals and conferences, such as ACM TODS, IEEE TKDE, VLDB, IEEE ICDE, and ACM SIGSPATIAL. He has authored some books, Social Big Data Mining (CRC Press). He is fellows of IPSJ and IEICE (The Institute of Electronics, Information and Communication Engineers) and members of ACM and IEEE.