Advanced Risk-Based Test Reporting

－A consideration of test monitoring and reporting in Risk-Based Testing－

Atsushi Nagata
System Quality Department, PSG PGSD, Sony Corporation,
Kanagawa, Japan
atsushi.nagata@jp.sony.com

Abstract

I had a case that applied Risk-based Testing but could not show how much the result of risk analysis affected testing, or how much the level of risk changed after testing. Test monitoring and summary reports are very important artifacts, because these reports would provide the level of quality and impact to the release judgment. In this article, I propose a way of monitoring and reporting of Risk-based Testing and consider applied case studies.

1. Introduction

1.1 Risk-based testing and risk management process

Risk-based testing is a risk management process that conforms to test process. ISO31000:2009 [1] provides Risk management process as in Figure 1.

Figure1. Risk Management Process: ISO31000:2009

Risk Management helps decision makers make informed choices. Risk management can help prioritize actions and distinguish among alternative courses of action [1]. The objective of risk management is to support decision making. Software testing has same idea. The stakeholders recognize the test team as a trusted team to make useful information with provided resources, to customize the information appropriately and to give it precisely and in a timely trusted way [2]. Testers detect defects during test execution and issue incident reports to the stakeholders. When the design team fixes the defect, the quality is improved. On the other hand, it is also very important for decision makers that testers deploy the current level of quality at release decision. Figure 2 shows the relationship between risk-based testing and risk management. Risk identification is to examine the expected defects and their expected phenomena and effects. Risk analysis is a process to analyze the risk items and to judge the level of risk. The level of risk is indicated as RPN (Risk Priority Number) that is usually the result of multiplying Likelihood and Impact. The result of risk analysis should be agreed with not only the test team but also the stakeholders. A process of risk mitigation in ISTQB consists of test process as in Figure 3. We could confirm that the level of risk is mitigated by passing the test. In case of detecting the defects, it is mitigated by fixing the defect and confirmed by testing or by other risk treatment activities.

1.2 Fallacies of Risk-Based Testing

Risk-based testing sometimes receives the argument that it is testing by cutting corners.

In this case they list up the test cases and put level of each risk by risk analysis.
And they decide the line of cutting corners so that they do not initiate the test cases under the line.
We do not think this is risk-based testing. The test should be designed very carefully with the level of risk in mind.

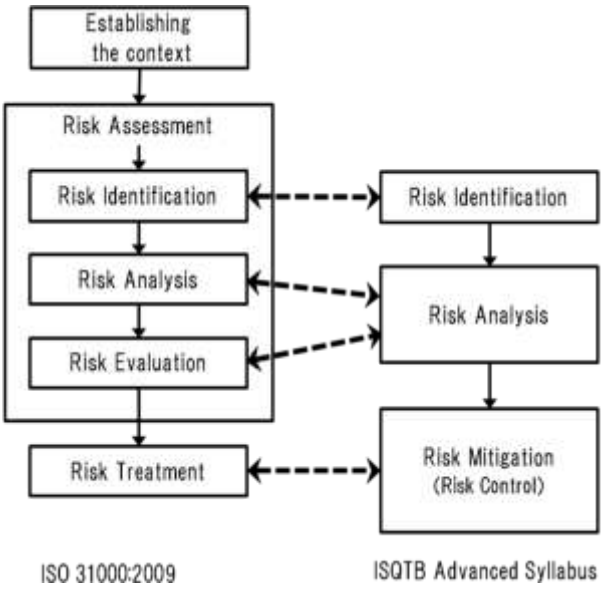

Figure 2. Risk management process and Risk-based testing

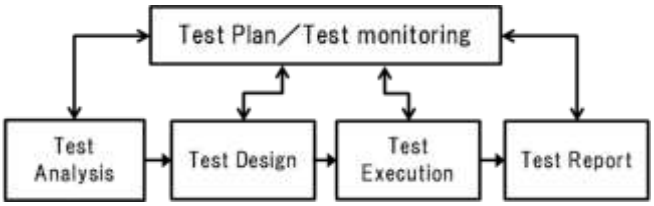

Figure 3 Risk Mitigation (Risk Control)

Another opinion is that risk-based testing is one of many testing methods. They indentify the expected impact to customer with the failure of the related defect and risk-based testing is used to mitigate the risk early in the testing period. We think that risk-based testing is not a method of testing but is a testing approach that covers the whole testing process as in Figure 3. We should be concerned with the risk from before test analysis through test reporting. These fallacies are discussed clearly on the blog of Rex Black [3].

2. Problems of Risk-Based Testing

We have had some problems through practical experience of Risk-based testing as follows.

2.1 We could not identify risks appropriately or explain them

This problem is out of scope of this paper because it is so important that it calls for more discussion in the next step.

2.2 We could only analyze the risks

We have some cases which stopped at analysis of risks because we could not understand the relationship between the result of the analysis and test cases. In such cases the test summary reports did not describe the change of level of risks or residual risks. As a result, the test engineers could not understand the benefit and also their stakeholders could not understand the risk-based test.

2.3 The test team does not make agreement of the result of risk analysis with stakeholders

The advanced level syllabus of ISTQB states, “The risk analysis process should include some way of reaching consensus or, in the worst case, establishing through dictate and agreed upon level of risk. Otherwise, risk levels cannot be used as a guide for risk mitigation activities.”[4] Basically, the value of risks is different between the test team and the stakeholders. The agreement of the result of risk analysis makes consensus of the value of risks across the test team and the stakeholders. Otherwise, we would miss some risks in the process of risk identification. It would be difficult for the test team to decide the level of risk by themselves since they would not have enough information of business issues and/or the detail of the software architecture. Although they could not have enough confidence in the adequacy of level of risk, they would use the result of risk analysis for testing without the evaluation of stakeholders. Such lack of consensus across the stakeholders would make the stakeholders unable to understand the approach of test activities. The test team could not receive the resources or system under test on time. There is a principle of risk-base testing, “test cases in higher risk should be tested earlier.” If they could not get consensus across their design team, they could not get the high risk functions earlier, or they could not test these on schedule. If they could not get the consensus across their decision makers, the decision makers could not make the appropriate delivery decision using the report from the test team.

2.4 We could not get the traceability between the result of risk analysis and the test cases

If you do not have a test design process (so you do not have a test design document), you might write down only a test procedure document. For some test teams, the only derivatives delivered are a test procedure document or a test summary report. Without a test design process, you would not understand what the test engineers considered or what they intended to design using the level of risks. We would miss some test cases and not find the missing ones by review, because the test cases or test procedure document is so large and descriptions are a consistent pattern. Therefore the link between risks and test procedure document would be unclear. This of course makes risk-based results reporting impossible.

2.5 Stakeholders could not understand the effectiveness of risk-based testing

The risk-based testing is using risk as a value of quality. The test team shall deliver the appropriate reports to their stakeholders using the risk, so it is much easier to share the level of quality. This is the effectiveness of risk-based testing.

3 Counter measures

3.1 Chart of residual risks

The direct scopes of this paper are 2.2 and 2.3. I propose the residual risk graph as in Figure 4. I use it for monitoring of test progress and as one of the test results in the test summary report.

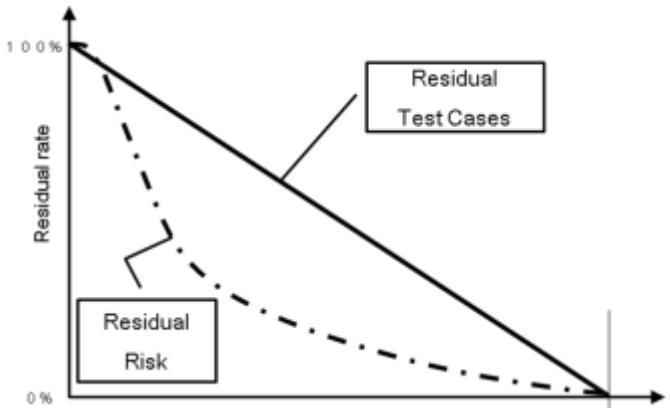

Figure 4. Residual Risk Graph

The vertical axis shows the residual ratio. The horizontal axis shows the execution time of testing. The solid line shows the residual ratio of the test cases. The dotted line shows the residual risk over time. The residual

ratio of the test cases means the number of residual test cases divided by the total number of test cases. The residual risk means the level of residual risks divided by the level of all risks. The calculation of the risks is as followings,

Rr : Residual risks :
Ro : Original risk priority number
Rm: Mitigated risk priority number
Tr : number of Residual test cases
Tt : Total number of test cases.

The residual risk Rr is calculated from expression (1)

$$Rr = \Sigma((Ro \times Tr) \quad \cdots\cdots (1)$$

Total level of risk Rt is

$$Rt = \Sigma(Ro \times Tt) \quad \cdots\cdots (2)$$

Therefore the ratio of residual risk : Rrr is

$$Rrr = Rr / Rt \quad \cdots\cdots\cdots (3)$$

Figure 4 shows that the residual risk is reduced earlier than the residual test cases. It is the ideal case of risk-based testing. The ideal case means the test is tested on one of analytical risk-based strategy, “the higher the risk, the earlier the test coverage”. On the other hand, the actual result would be as in Figure 5.

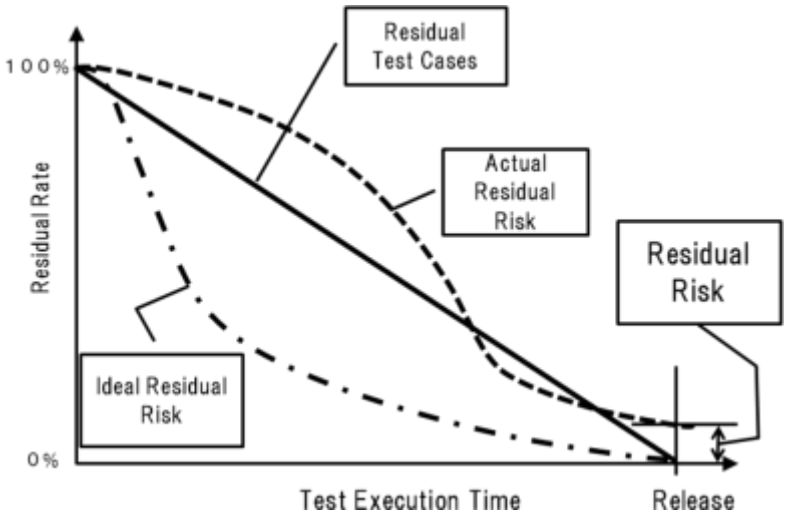

Table 5. Actual Residual Risk pattern

For example, the design team could not deliver the planned functions on time into this testing. The test team would have to change the order of test cases, issue the lower risk test cases than planned. Therefore the level of residual risk is not reduced, but defects would be found. In the end, some defects remain at the release date, which means the residual risk remains in the code. The point is that the result of risk should be agreed between test team and their stakeholders from the beginning. This graph shows that the tests confirm that the risk is mitigated through the test execution. In other words, this test monitoring shows the effective testing to confirm the risk mitigation. Since the residual risk is agreed with stakeholders, they could understand how serious the impact is to their customer immediately and they share the image of the quality. Therefore they could make appropriate and immediate decision and agreements to treat the residual risk, e.g. the decision to release the software, mitigation plans, or priorities of action for the risk treatments. We can manage the risk by the proposed idea as PDCA loop as follows: agreement of risk analysis (Plan), testing that is designed by the result of risk analysis (Do), check the risk status (valuation of level of risk) by test monitoring and test result using the residual risk graph (Check), execute the reactive testing strategy

when we find additional risks or change the risk level, and execute the judgment of the quality to release the product (Act).

3.2 Revised residual risk graph: when testing cannot change the impact of risk

I had a chance to get the consultation of Rex Black and his review for this approach of risk-based testing. He commented on one problem about the expression of level of risk. In figure 4 the level of residual risk will be zero percent at the release day. It means the risk is gone if the test is passed. But Rex insisted that should not be zero. When the test cases associated with the risks are passed, we can consider that the likelihood would be reduced from the result of the risk analysis. On the other hand the impact of the risks would not be changed. We determine the likelihood is 3 and the impact is 4, using likelihood and impact ranging from 1-5 for each, 1 being lowest and 5 being highest. When we finish the tests associated with this risk and pass them, we can consider the likelihood becomes 1. The General testing principles in ISTQB foundation level syllabus states, principle 1 - testing shows presence of defects: Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness [5]. So the testing can reduce the likelihood but cannot reduce the impact of the level of risk. Therefore the lowest value of level of risk should not be zero but the sum of the level of risk as in Figure 6. The expression of residual risk Rr is revised as (4).

$$Rr = \Sigma((Ro \times Tr) + \Sigma((Rm \times (Tt - Tr)) \cdots (4)$$

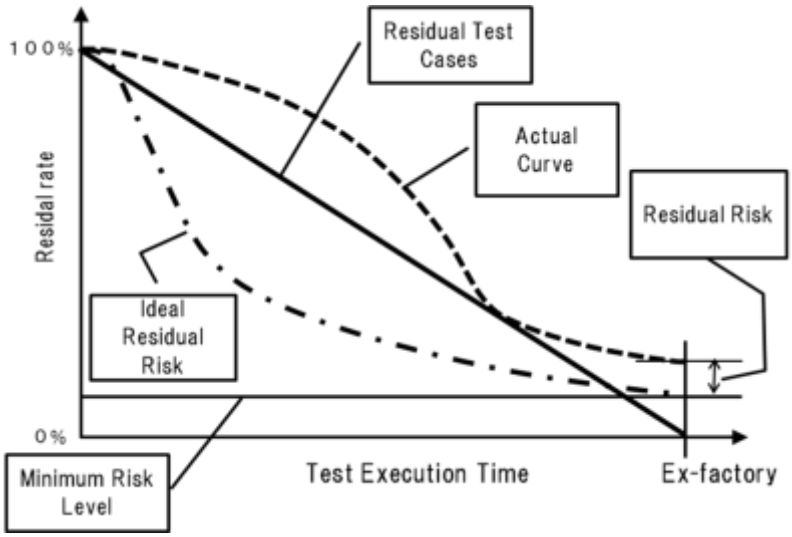

Table 6. Revised Residual Risk Chart

4. Case Study

Figure 7 shows the applied case. The graph has four lines, planned residual test cases, actual residual test cases, planned residual risk and actual residual risk. In this case, the planned residual risk would have reduced by the same ratio as the execution of test cases. But actually the code implementation of functions was delayed and the progress of test execution was slow because some tests were blocked and we needed to issue additional confirmation tests. When problems occurred (found defects), at first the residual risk was slightly increased. The order of the test execution were also changed by the delay of the code implementation and fixing the defects. We could only issue tests for lower level of risk due to the change in the execution order of the test cases so that the reduction of the level of risk was slower than planned. At the planned release day the residual test cases were about 40% and the residual risk was more than 50%. We had to suspend the release schedule and continue testing all test cases. At last the residual risk reached the minimum risk level. The point is we draw the planned and actual lines. And we analyzed and took some action for the differences to minimize the residual risk.

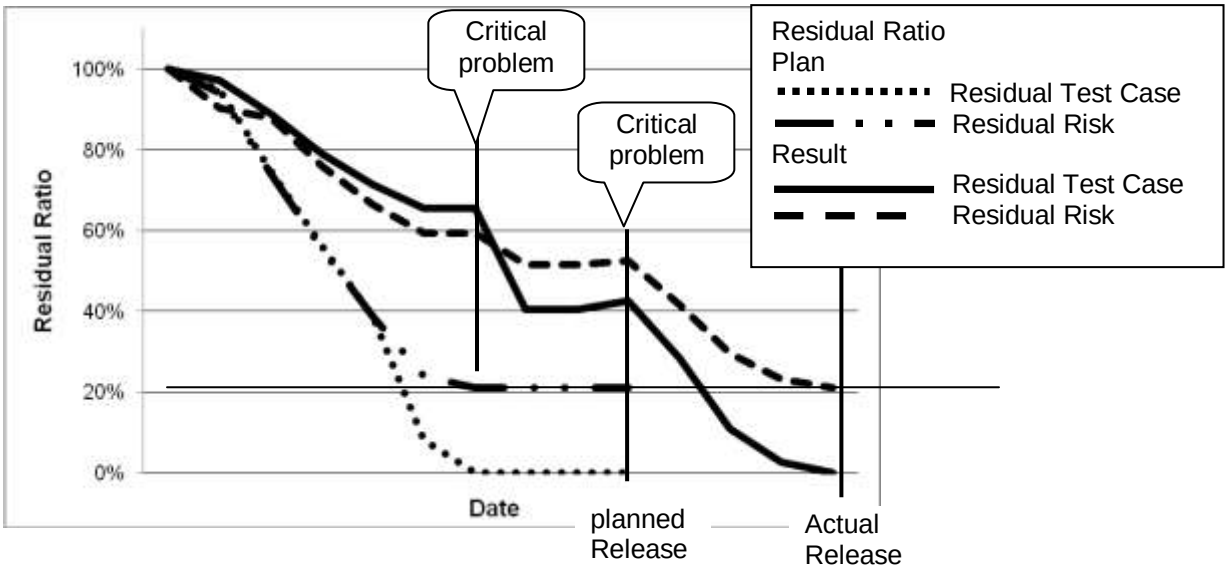

Table 6. Case Study

5. Consideration

5.1 Visualization of Risk

We can visualize the mitigation of risk by the residual risk graph. As shown in Figure 7, if we analyze the variation of risks and differences between the planned and actual risk and testing progress, we can issue some appropriate countermeasures dynamically in order to minimize the quality and delivery risk. In this case study, the test execution was delayed. The main cause was the delay of code implementation. But the test team in the design section did not use the risk-based testing approach. The priorities of system tests were different from QA's therefore some tests were blocked and lost their efficiency. And also we had some serious problems and increased rework such as bug fixing and re-test. At the planned release date, the test progress was 60% and the residual risk was 50% so that we determined that we could not release the code. So we understand the usefulness of expression of residual risks for the judgment of quality. In this case study, the level of risk was calculated simply as expression (1), and we confirmed this approach was practical. On the other hand, in the case that the risk likelihood is minimum, we cannot change the value of likelihood by testing. This is the problem in the next step.

5.2 The motivation of agreement with the result.

To monitor the test progress using the residual risk graph means to evaluate the level of risk mitigated in accord with the testing progress. While analyzing and considering the level of risk and priorities, we would have the motivation as follows; since we want to reduce the actual level of risk, the higher risks should be mitigated faster.

5.2.1 The test environment for the higher risk should be prepared faster

Usually the test team plans the order of test execution so that the higher the risk, the earlier the test coverage. For that purpose the design team should implement and deliver the higher risk functions earlier for testing. And we need the appropriate resources and testing equipment and environment in the appropriate timing. If we test embedded products, we also need the actual working products as a test item.

5.2.2 We need the confidence of the result of risk analysis

The result of risk analysis is the basis of risk-based testing. So, the test team would attempt to improve the accuracy of analysis and prevent missing the risk items. But it would be difficult for them to understand the software architecture or the code complexity to estimate the likelihood, or to get the customer information to estimate the impact of risk to the customer. They need to communicate with their stakeholders and make

agreement on the result of risk analysis. By improving this, the communication between the test team and stakeholders is better activated.

5.2.3 We want to keep risks lower from the beginning

An ounce of prevention is worth a pound of cure. Using the risk-based test approach, we will notice that risk prevention is the better choice. The testing will become the confirmation activity for risk prevention.

5.2.4 Defect database

Considering the risk means considering the defects. We notice that we need to understand the defects that should be analyzed and classified. This information will be used in every process of risk-based testing, risk identification, risk analysis, test analysis, test design and test execution. Therefore, we will have a motivation to make defect database. In fact, the QA test teams feel this motivation and they have had meeting periodically

5.3 The traceability of risk

The risk-based testing needs traceability from the risk items through test case and test specification. If we use the residual risk graph, the traceability is much more important. Especially the test design is the most important process, because we should design the test by the result of risk analysis.
The higher risks tend to find more defects, or tend to test more important areas of the system or both.[6] So we allocate our testing effort into higher risk areas in the test design process. This means for higher risk areas, we should consider more kinds of test types, using more testing points of view with appropriate testing methods, more variations of the test conditions, and more test cases. Or for lower risk areas, we should select more simple tests, because the resources are limited.
On the other hand, sometimes we should consider the other factor of priority. For example, the characteristics of the domain, or policy of the project, even lower risk are, we should design to allocate more testing effort appropriately. It is so important to check whether the test cases are appropriate or not, that we need to review them. But the test cases are too detailed and too much to review effectively. Therefore the test design document is very important. We can understand the testing strategy and intention of testing effort allocation and its balance. So we can acquire the testing confidence. Using the risk-based testing approach we will be motivated to improve test design process, and I found the phenomena in the field. In this case, still we need to improve the test design. It is the subject of the next step.

5.4 The merits for the stakeholders

*If the residual risk on the residual risk graph is reduced and test progress goes according to plan, the stakeholders have a confidence in the quality of product and delivery. On the other hand if it does not go well, they should share the sense of crisis and consider the risk treatment immediately. This means the stakeholders and test team have the same value of risk and take actions collaboratively using this graph. At the release judgment, we understand the residual risk directly by the residual risk graph. And we can find what kind of risk by the traceability document. On the other hand, all stakeholders realize the risk by the agreement of the risk analysis, therefore we can share the same sense of risk intuitively and immediately. We take these activities contiguously the quality of test planning and test design will be improved by these motivations. This test process improvement will be very helpful for the stakeholders, especially the project manager, becaus*e they can understand the test strategy, test plan (when and what), and test design (how).

5.5 The motivation for the master test plan

In this risk-based testing practice, the communication between testing team and design team was improved. As a result, at the risk analysis process the design team recognizes the risk that QA found in test last time. The design team feeds it back into their design and takes the preventive measures. As the next step, at the risk analysis process, the design team and the test team had a motivation to consider the distribute the risk to the appropriate test level, e.g. component and integration test level (by the design team) or system test level (by the test team) as in Figure 8. As a result, they had the motivation to express the distribution of the risks in the master test plan to each level test plan. It will make clear the responsibilities of scope of testing and prevent missing the risks at the boundary of the test levels.

5.6 Process risk

The higher the risk, the earlier the test coverage, as in Figure 4 is ideal. This means the design and implementation should be earlier so that the higher risk area can be tested earlier. But usually it would take more time to develop and/or implement the higher risk potion and the delivery would be delayed. Also if the product includes the hardware factor, its schedule would impact to the delivery We should consider these problems as the project risks.

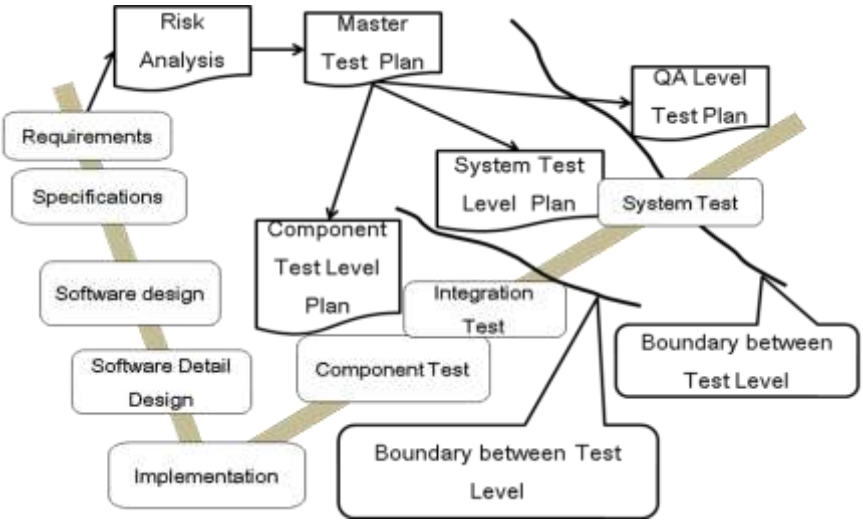

Table 6. Master Test Plan ： Risk Distribution

6. The next step

We need the traceability from the risk analysis through test execution. The key of the traceability is the test design process. In the test design process we consider how to reflect the result of risk analysis into the test cases. The risk-base testing approach has such so clear objective to improve the test design methods and process. Rex Black calls this kind of risk-based testing approach an analytical risk-based test strategy. He also says that it is not perfect. Because we will not have all of the information we need for a perfect risk assessment at the beginning of the project. Even with periodic reassessment of risk we will miss some important risks. Therefore, and analytical risk-based testing strategy should blend reactive strategies during test.[4]

7. Conclusion

Using the Residual Risk graph, we can review the status of the residual risk, the product risks and the project risks expression in real time. This can improve communication in the project, especially the test team and design team so that the test team can be motivated to improve the process,

References

[1] Risk management - Principles and guide lines on implementation, ISO 31000：2009, International Organization for Standardization, 2009
[2] Rex Black, *Critical Testing Process*, Addison-Wesley Professional, 2003.
[3] Rex Black, “Five Fallacies of Risk Based”, Testinghttp://www.rbcs-us.com/blog/2010/09/23/five-fallacies-of-risk-based-testing/, (accessed on 2011/6/12)
[4] Certified Tester Advanced Level Syllabus version 2007, International Software Testing Qualifications Board , 2007.
[5] Certified Tester Foundation Level Syllabus version 2011, International Software Testing Qualifications Board , 2011.
[6] Rex Black, “Advanced Software Testing vol.2”, Rock Nook Inc., California, U.S.A.,2009
[7] Paul Gerard, Neil Thompson, “Risk Based E-Business Testing”, Artech House Publishers, MA, USA,