


**Vietnam and Japan
Joint ICT HRD Program**

ITSS Software Development
**Chapter 14. Quality management
in Testing Phase**

Nguyen Thi Thu Trang
trangntt-fit@mail.hut.edu.vn

1

Content

- 
1. Quality Management and Quality Assurance
 2. Change Control of the software product

2

1.1. What is Objective of the Test?

Until when should the testing continue?

In what condition can we finish the testing?

When the ending time scheduled for testing has come?

When the prepared all test cases is carried out?

When enough testing is carried out to ensure the required quality?

Who and how to make the judgments?

Enough testing? And Enough quality?

1.1 What is Objective of the Test? (2)

As a result of the discussion, we can say;

Test has the following two objectives;

- To detect as many errors as possible!

- It is necessary to carry out effective and enough testing.

- To evaluate quality of the software, and then ensure the software has the required quality.

- How to evaluate and ensure the software has the required quality?

1.2. How to evaluate quality of the software - Reliability

Basic Procedure of the quality evaluation;

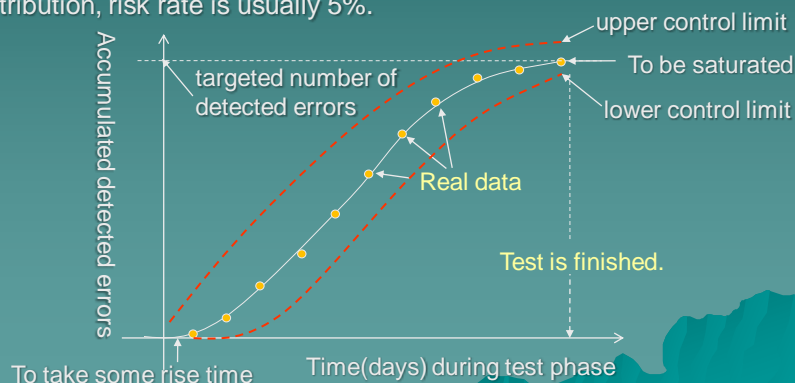
1. To forecast the number of errors existed in the software.
 - : Based on the **the actual performance of experienced software development project**, considering developed size, function type, running platform, development environment, and development team.
2. To do Test the enough test cases covering the entire software.
 - : To manage comparing the growth curve model* with targeted number of detected errors.
3. To analyze and evaluate based on the forecast data.
 - : Finally also to evaluate comparing the growth curve model with targeted number of detected errors.

*: To be explained in next slide

The 3rd party department should manage the quality evaluation procedure. And then, they should do final evaluation and make judgment deliver or not.

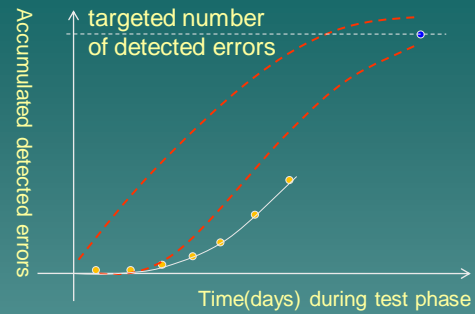
1.3. Growth curve model of Reliability

- Can be modeled “the accumulated detected errors vs. Test progress” as “S” character graph such as Gompertz curve.
- If plotted real data is saturated in the last stage of the testing, it can be understood as errors are converged, and reliability is stable.
- Upper/lower control limit are set up by statistical process of binomial distribution, risk rate is usually 5%.



1.3 Quality management using Growth curve model in upper test phase (1)

Accumulated detected errors are lower than lower control limit.
How to understand this situation?
What actions are needed?



Case1: Quality of the software is better than the supposed quality.
Case2: Detection ability of the testing carried out is lower.

→At first, please evaluate the test cases and test environments!!
Judgment of "Quality of the software is better" will be last!!

1.3 Quality management using Growth curve model in upper test phase (2)

Accumulated detected errors are upper than upper control limit.
How to understand this situation?
What actions are needed?



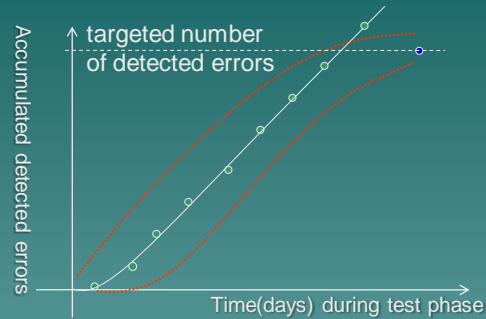
Case1: Quality of the software is worse than the supposed quality.
Case2: Detection ability of the testing carried out is very good.

→If forecast of final accumulated detected errors is more than targeted number of detected errors, the case1 is most probable.
Actions for quality improvement are needed.

1.4 Quality management using Growth curve model in last test phase (1)

Accumulated detected errors are exceeded targeted number of detected errors and are not saturated.

How to understand this situation?
What actions are needed?



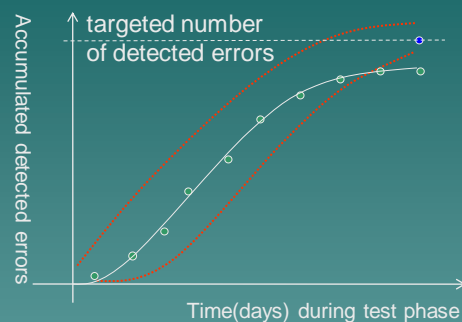
Suppose that quality of the software is worse than the supposed quality and also detection ability of the testing carried out is lower.
What actions are needed?

→Quality analysis and Testing evaluation should be done!!

1.4 Quality management using Growth curve model in last test phase (2)

Accumulated detected errors are lower than targeted number of detected errors and are saturated.

How to understand this situation?
What actions are needed?



Case1: Quality of the software is better than the supposed quality.

Especially, there are few errors caused by upper stream such as Requirements analysis/definition or Architectural design.

Case2: Detection ability of the integrated/system testing is lower.

→At first, please evaluate and improve the test cases and test environments for the integrated/system testing.

1.5. Evaluation of the software quality using Growth curve model

The basic conditions for judgment to meet the required quality and stable are below;

1. The number of detected defects falls within the upper limit to the lower limit of the target number of detected errors.
- and,
2. The number of accumulated detected errors became saturated at last stage of the testing.

If the both of two conditions are not satisfied, the reasons why the conditions are not satisfied are analyzed.

1.5. Evaluation of the software quality – Analysis of Growth curve model

How to analyze the reason why the software quality is unstable.

1. In which software component, how many errors are detected?
→ Identify the component whose quality is not so good.
2. What type of errors are detected?
→ In which process/stage is each error built-in?
→ To evaluate the requirement, or design processes!!
3. In which test process is each error detected? In unit test? In program test? Or....?
→ To evaluate testing ability!!

→ How to improve the quality based on the analysis result.

1.5. Evaluation of the software quality - focusing on individual components

In order to probe the cause of unstable quality, it is very effective way to analyze focusing on individual software components/function such as user interface module, control logic module, and DB access routine?

-In a lot of cases, a few software components/functions are major causes of unstable quality of whole software.

-Why do only a few components/functions are so bad quality?

Various causes can be possible! However, if we narrow down the cause of the problem, improvement is easy way.

→ If DB access routines are poor quality, use a high quality library routines or well-established generation tool.

1.6. Final Judgment of Delivery

The 3rd party department for quality assurance should manage the quality evaluation procedure.

And then, they should do final evaluation and make judgment deliver or not.

-They should do the sampling test for evaluation the final quality including non- functional quality. Sometimes it is necessary to assist the evaluation by developing team.

-They also should conduct the **Software Qualification Testing**

They should make the judgement totally!!

Content

1. Quality Management and Quality Assurance



2. Change Control of the software product

15

2.1. Introduction to Change Control

What is the change control?

: For example, the source code of the program should be fixed in a certain stage.

→ After that, when some modifications are made, what procedure is needed for change/modifying the source code?

→ Basic concept of the Change Control is we need some authorized procedure to change/modify the source code!!

Why? Is it needed?

2.2. Why is it need for?

-To make the software quality stable!!

Especially, plurar engineers attend one software development project, the authorized procedures are needed in various scene such as source code modification, requirements change, and documentation for specification change. Typical case is source code!!

“One owner makes the modification plan, and some experts review the plan and be authorized. And then modify the source code. ”

- 1.The process ensure the modification plan is the best answer.
2. The team consensus is established in the process.

→ It is possible to make source code quallity better after the modification.

2.3. Basic Procedure for source code modification

1. Causes to be modified are occured. Ex. Defects are measured.
2. To be identified causative module or processing part.
3. The module owner create the modification plan including such as;
 - Analysis of the cause of the defect.
 - Modification architectture.
 - The coding before the modification
 - The coding after the modification
 - Test Case for assurance test
 - Influential module or function
4. Review to authorize the modification based on the plan.

Then authorize the modification...

2.3. Basic Procedure for source code modification (2)

The owner of the module modifies the module based on the plan.

5. The owner has got the Authorized Source code.

6. Source code modification is done on a trial basis.

7. Test case for assurance test for modification part are carried out. If it is successful.

8. The modification module carry back the authorized source code.

From Step 5 to 8, a configuration management system for source code is assumed.

2.4. Regression test for the modified source code

After some defects are corrected, the regression test is needed!

How many times the regression test is needed?

What timing to do the test is good?

What scope is appropriate and adequate?

Can automated environment be used?

In what scope, can the automated environment be applicable.

The planning for Regressiontest is very important!!

2.5. Basic Procedure for requirements design change

As for documentations for specification, change control/verion management is necessary.

The basic procedure is almost the same as source code modification;

“One owner makes the modification plan, and some experts review the plan and be authorized. And then modify the design can be changed”

Not only in the test phase, in full phase, it is important.

Question?

