

# **Test Management**

Whitepaper

Andreas Erlacher

[andreas.erlacher@microsoft.com](mailto:andreas.erlacher@microsoft.com)

# Content

1	Introduction .....	3
2	Testing Process .....	4
3	Test Types .....	6
3.1	Development Tests .....	6
3.1.1	Unit tests .....	6
3.1.2	Integration tests .....	6
3.1.3	Function tests .....	6
3.1.4	Security Tests .....	6
3.2	Technical Tests .....	6
3.2.1	System Tests .....	6
3.2.2	Security Tests .....	7
3.2.3	Performance Tests.....	7
3.2.4	Load Tests .....	7
3.2.5	Stress Test.....	7
3.2.6	Configuration Test.....	7
3.2.7	Monitoring Test .....	7
3.2.8	Rollback Test .....	7
3.2.9	Relaunch Test.....	7
3.2.10	Installation Test.....	8
3.3	Domain Specific Tests.....	8
3.3.1	Business Cycle Test .....	8
3.3.2	User Interface Test .....	8
3.3.3	Usability Test .....	8
3.3.4	User Acceptance Test .....	8
3.4	Regression Test .....	8
3.5	Acceptance Test.....	9
4	Test Management .....	10
4.1	Test Case Definition .....	13
4.2	Automation .....	14
4.3	Tools.....	14
4.4	Roles .....	14
4.5	Environments .....	15
4.6	Results.....	15
5	Appendix .....	16
5.1	Acronyms.....	16
5.2	References .....	16

# 1 Introduction

“We did it right”

After completion of a software development project, a suitable verification of the results is necessary in order to anchor this conviction in the consciousness of all participants.

In general the verification of software results— beside reviews – is done by testing. Testing is a tool of the Quality Management and serves the quality assurance. The goal is to guarantee the product maturity to meet the business requirements. Therefore the quality measurement metrics defined in ISO/IEC 9126 are taken as criteria (functionality, reliability, efficiency, usability, maintainability, portability and its subgroups; see Reference [3]).

Testing should help to answer following questions:

- |          |  |
|----------|--|
| 1. Why?  | To ensure the required quality.                          |
| 2. When? | During the development until shipment.                   |
| 3. What? | The entire product from different viewpoints.            |
| 4. How?  | With appropriate methods for the different viewpoints.   |
| 5. Who?  | Members of the interest groups to any defined viewpoint. |

As part of the Software Development Process Testing starts at the beginning and accompanies the Development until the product is shipped. Testing is defined through:

1. Process Steps
2. Tools
3. Roles

The different perspectives per interest group are summarized in Test Types. These are essential to accomplish the desired quality and depth with testing.

Test Management is the instance of the Test Process in a specific project. It animates theory to real life.

This document gives a general idea on the Test process, Test Management and some already proven Test Types. It is based on experiences obtained in several projects.

## 2 Testing Process

The Testing Process is the base for Test Management and an essential part of the entire Software Development Process.

Following image shows the imbedding of Testing into the Management Processes of the Software Life Cycle (source: Reference [2]):

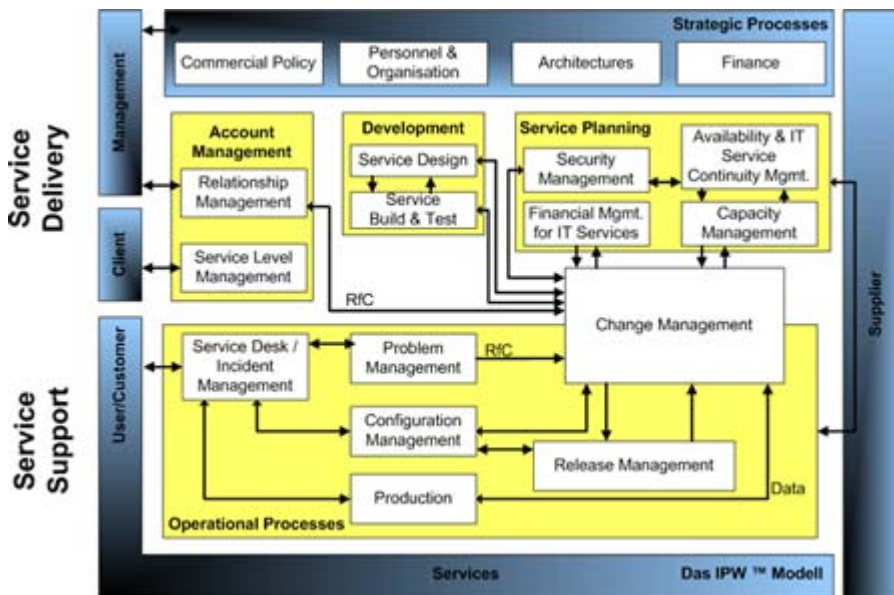


FIGURE 2-1 SOFTWARE MANAGEMENT PROCESSES (SOURCE: ITIL, REFERENCE [2])

The process sketched is based on the Process Definition by Vivtek (Reference [1]).

The result should consider and support following phases:

1. Planning of the necessary activities
2. Preparation of the necessary resources
3. Execution of the tests
4. Analysis of the results
5. Product acceptance

These requirements lead to the following generic process:



FIGURE 2-2 TESTING PROCESS

According the specified process definition in each process step the following points have to be determined:

- |                    |   |
|--------------------|---|
| 1. Tasks:          | Describes the necessary activities                  |
| 2. Goals:          | List of the goals of reaching that                  |
| 3. Metrics:        | Measurement of the goals                            |
| 4. Input:          | Necessary resources to the execution                |
| 5. Output:         | Result of the execution                             |
| 6. Responsibility: | Owner and his team                                  |
| 7. Methods:        | The procedure to reach around the goals             |
| 8. Tools:          | The remedies in order to obtain the desired results |

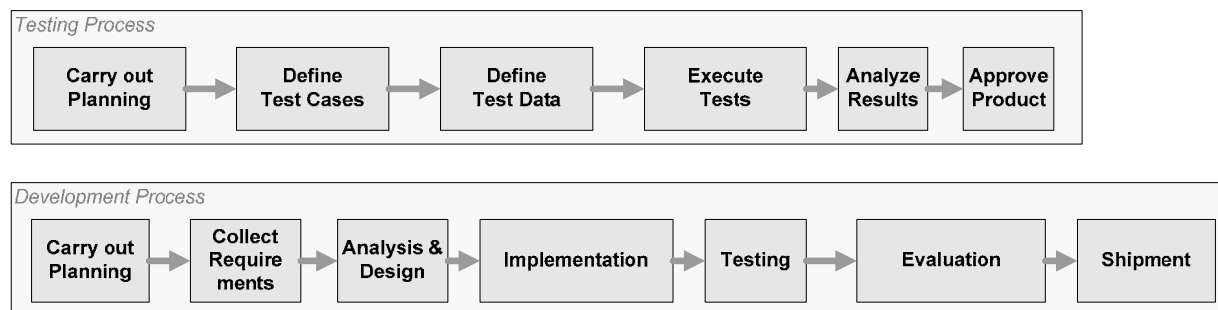
The concrete definition and implementation of the list mentioned above strongly depends on the respective business and its infrastructure.

Proven roles in the Testing Process are:

- |                    |  |
|--------------------|--|
| 1. Test Manager:   | Responsible for the entire Test Management.                        |
| 2. Domain experts: | Participates in test cases creation and provision of the test data |
| 3. Tester:         | Executes the tests. Collects and documents the results.            |

The imbedding of Tests into the whole Software Development Process is an important success factor for a project. The Testing Process starts with the planning of the project and ends with the deployment of the generated software.

The following image shows schematically the dependency of the test process on a simplified software development process. These dependencies –including possible iterations- depend on the selected Software Development method (VModell, XP, RUP, Scrum,...).



**FIGURE 2-3 TESTING PROCESS AND DEVELOPMENT PROCESS**

The Business Cases (Use Cases) are the base for the Test Cases. A Test Case can be seen as a Use Case for a Business Case.

Definition and Collecting test cases should start in the Analysis phase of the Software development process. Many test cases are domain-specific and therefore should be agreed with the ones who define the business requirements (customer). Furthermore the test cases can be used to review the requirements.

## 3 Test Types

Test types serve the consideration of the different views per interest groups on the product. The test type influences also the selection of the appropriate tools and method.

Special attention on the error behavior of the tested units is important with all tests.

### 3.1 Development Tests

The development tests cited here describe certain tests of different granularity which are executed by the development team in addition to other QS-tasks (e.g. Code-Reviews).

These tests can be automated depending on the chosen technology.

#### 3.1.1 Unit tests

The behavior of the smallest units of the software is tested. With Unit tests, all self-sufficiently executable software components are tested against the detailed specification. The target is to test all code produced during implementation.

Unit tests are testing the base of the product. Because of that they are of fundamental importance (e.g.: for maintenance and extensibility).

#### 3.1.2 Integration tests

The integration test the correct orchestration of the components tested in the Unit test. At this point it is useful to test the inter-component communication paths – especially in case of errors.

#### 3.1.3 Function tests

With function tests the required In/Out-behaviour is checked. As far as it is possible without knowledge of the source-code all commands (executable functions) are tested. Feasibility studies are also made within function tests.

#### 3.1.4 Security Tests

The developer has to enable the components to ensure the fulfillment of the required security levels. According the Threat Model and the chosen implementation technology the code has to be tested for security (see also Reference [4], p. 47, 567ff).

### 3.2 Technical Tests

With technical tests the demanded product characteristics are verified without knowing the internals of the software. They are pure black-box tests.

The results of each Test Type have to be compared with the requirements listed in the Technical Requirements Specification.

#### 3.2.1 System Tests

With the general System Tests the correct and full implementation of all functional requirements are proved (e.g. Interfaces) as stated in the Technical Requirements Specification such as the ability to be operated in the target environment (Hardware and Software), interaction with other systems, correct implementation of the required Business Processes.

### 3.2.2 Security Tests

Additional to the security tests done during development, the whole applications has to be tested for vulnerabilities. The Levels and Threat Model defined in the Technical Requirements Specification build the base for that verification.

- Application Level Security: Ensures e.g. the proper rights for users to navigate within the application.
- System Level Security: Prevents unwanted and prohibited access from outside.

### 3.2.3 Performance Tests

The following application's capabilities are tested:

- Response speed - in which time does the user receive a reaction of the system?
- Response time – in which time does the user receive an answer on data input?
- Transaction times - in which time are transactions completed?
- Side construction at the screen (dependent on the connection – subjectively)
- Execution time for processes or database-scripts

### 3.2.4 Load Tests

It is supposed to be shown how the system reacts under the determined maximal load (e. g. concurrent users).

### 3.2.5 Stress Test

The stress test gathers the behavior of the system in the case of overload over longer periods or under extreme load above the demanded allowable maximal load.

### 3.2.6 Configuration Test

It reviews whether the application functions corresponding to the specification in correlation with different HW/SW-constellations and parameter attitudes.

### 3.2.7 Monitoring Test

The behavior and messages of the monitoring are checked against the requirements and the Operations Instructions. The results are essential for SLA-Management.

### 3.2.8 Rollback Test

This test can be exercised either independently or as part of the Relaunch Test. It should verify data consistency and that in case of an error no data garbage is produced.

### 3.2.9 Relaunch Test

Special attention is paid on the consistency of the system including used resources after a partly or total unexpected shutdown followed by a restart.

It is recommended to perform a data backup test with a full-, differential and incremental backup and a restore test on a clean machine.

### 3.2.10 Installation Test

The purpose of this test is on the one hand to review the Installation Guidelines with respect to consistency, correctness and completeness and on the other hand to install the product in the target environment as described in the Installation Guidelines.

The test was successful if the software was installed correctly and all necessary information is covered in the guidelines.

## 3.3 Domain Specific Tests

The professional tests investigate the product from the domain expert's point of view. Especially the correct implementation of the business cases is tested.

### 3.3.1 Business Cycle Test

In this test general recurring and ad-hoc activities and workflows are tested as well as data life cycles (CRUD) and if necessary the according access rights.

### 3.3.2 User Interface Test

The implementation of the specified requirements and further accompanying documents is verified with respect to layout and user navigation. Special attention is paid on the correct compliance of business given (CI) and recommendations (e.g. WAI).

### 3.3.3 Usability Test

With Usability-tests, the access rights of single roles as well as its navigation possibilities are reviewed through the entire system including its consistency (z. B. logical disruptions in the navigation tree).

### 3.3.4 User Acceptance Test

The appropriateness of the workflows, the error tolerance of the system, the ease of learning for the user are tested as well as the intelligibility of the help functions. At the same time peculiarities of the system's user are to be considered especially.

## 3.4 Regression Test

The main purpose of this test type is to repeat already exercised tests to guarantee the proper function of tested (unchanged) software parts after changes were applied to the system. Additionally possible side effects caused by e.g. bug fixing can be detected. This test type has to be exercised after each bug fix cycle and is an excellent candidate to be supported by automation.



### 3.5 Acceptance Test

It serves as the final compliance verification with the requirements and is a mix of chosen test cases out of test types mentioned above. Possible candidates are:

1. Installation
2. Configuration
3. User Acceptance
4. Relaunch
5. Rollback

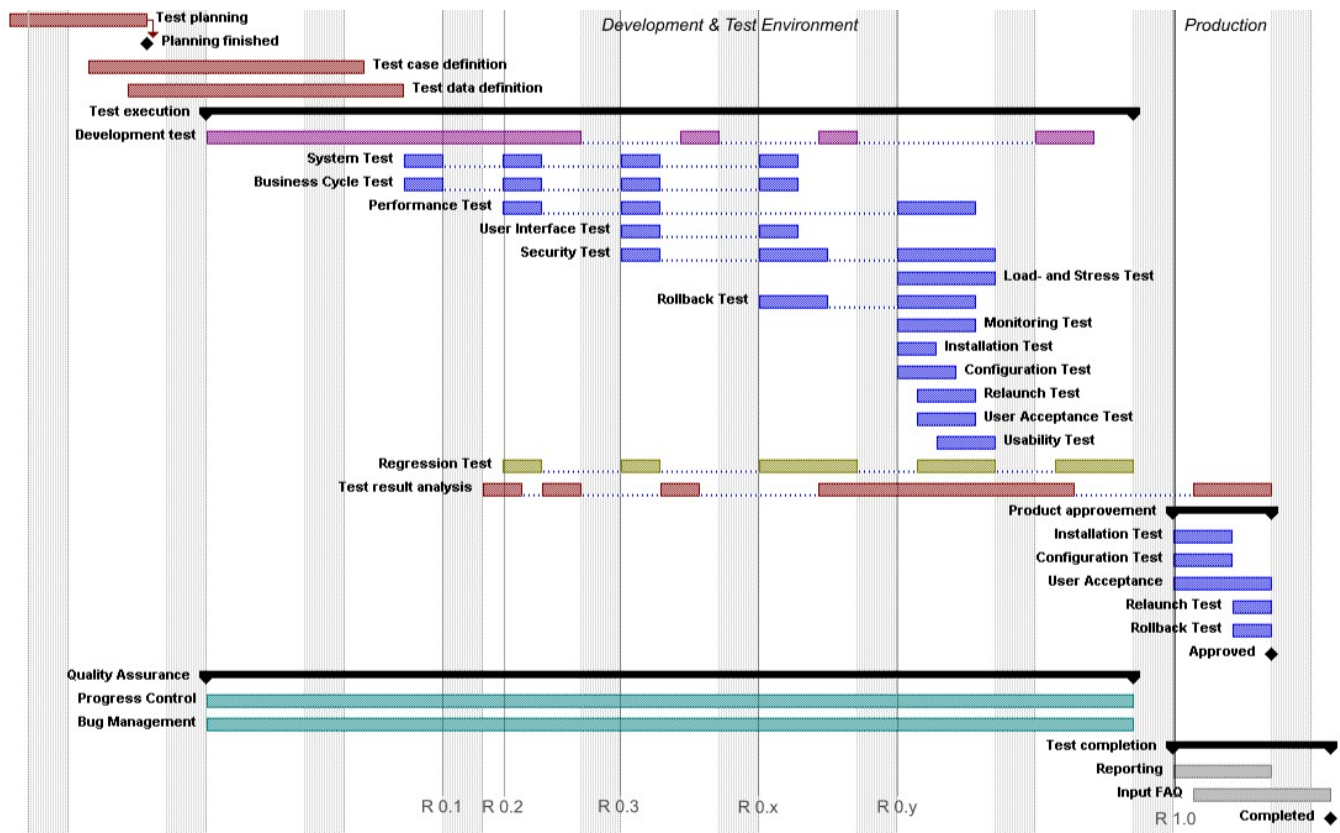
Also all resources as defined in the project result plan have to be approved such as:

1. Handbooks (Installation, Administration, User)
2. Software Documentation
3. Quality Assurance Documents
4. Education materials
5. SLA fulfillment

## 4 Test Management

Testing is time intensive and expensive. An important factor of success is to already consider the testing efforts during the planning phase of a project.

The following picture shows an example of possible Test Type allocation during a Project Life Cycle.



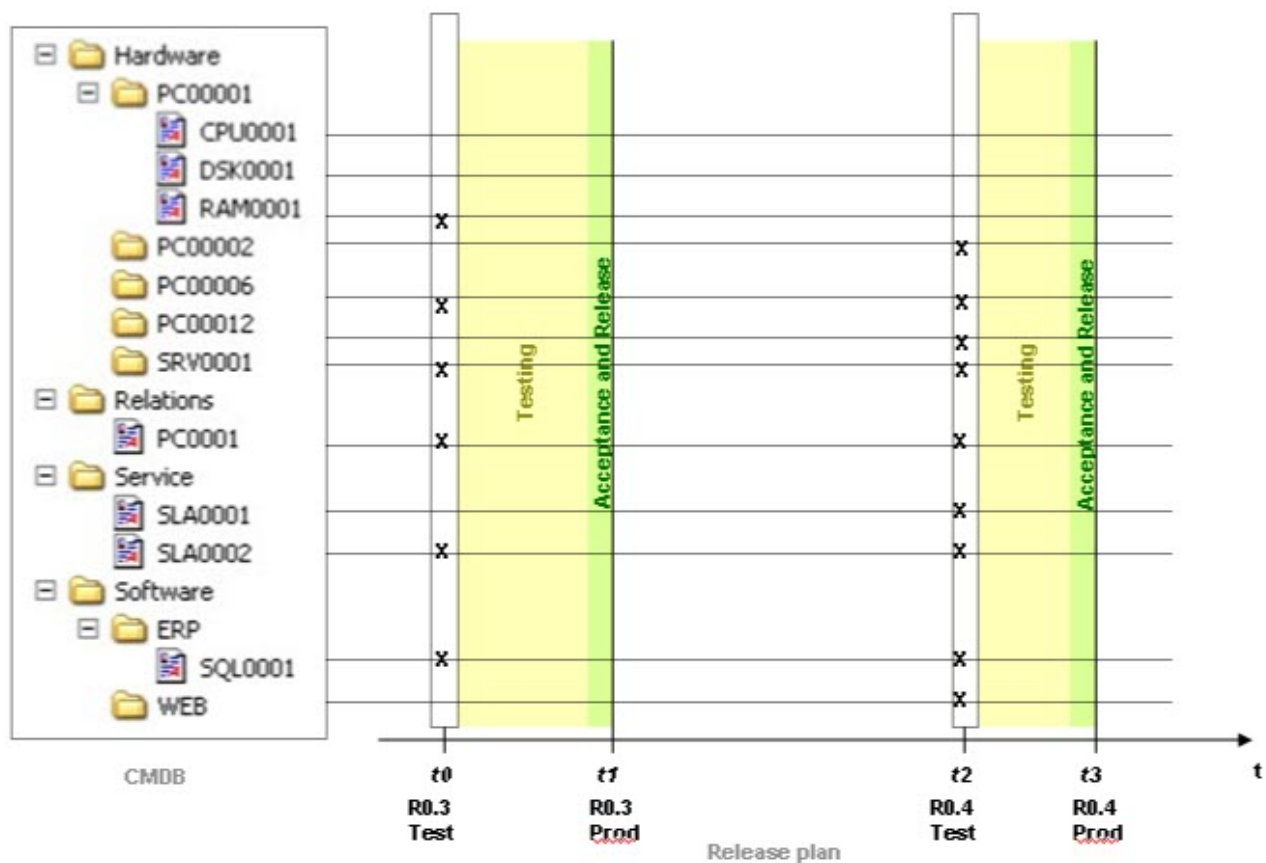
**FIGURE 4-1 TESTING WITHIN THE PROJECT CONTEXT**

The number of Releases and which test types are executed at what Release stage depend on various factors like chosen Software development methodology, availability of human and system resources, special requirements as specified in the Business and Technical Specification (e.g. Performance, Security) and many others. It is essential to start with Test management at Project start up.

Testing has to be in synch at least with following processes to ensure effectiveness and efficiency (see also Reference [2]):

- Change Management: Ensure holistic Error/Bug fixing for the Product Life Cycle
- Release Management: Correct transfer of the proper components
- Configuration Management: Store the companies assets and build the base for Incident Management

The orchestration of the different Management Processes is shown in the following picture.



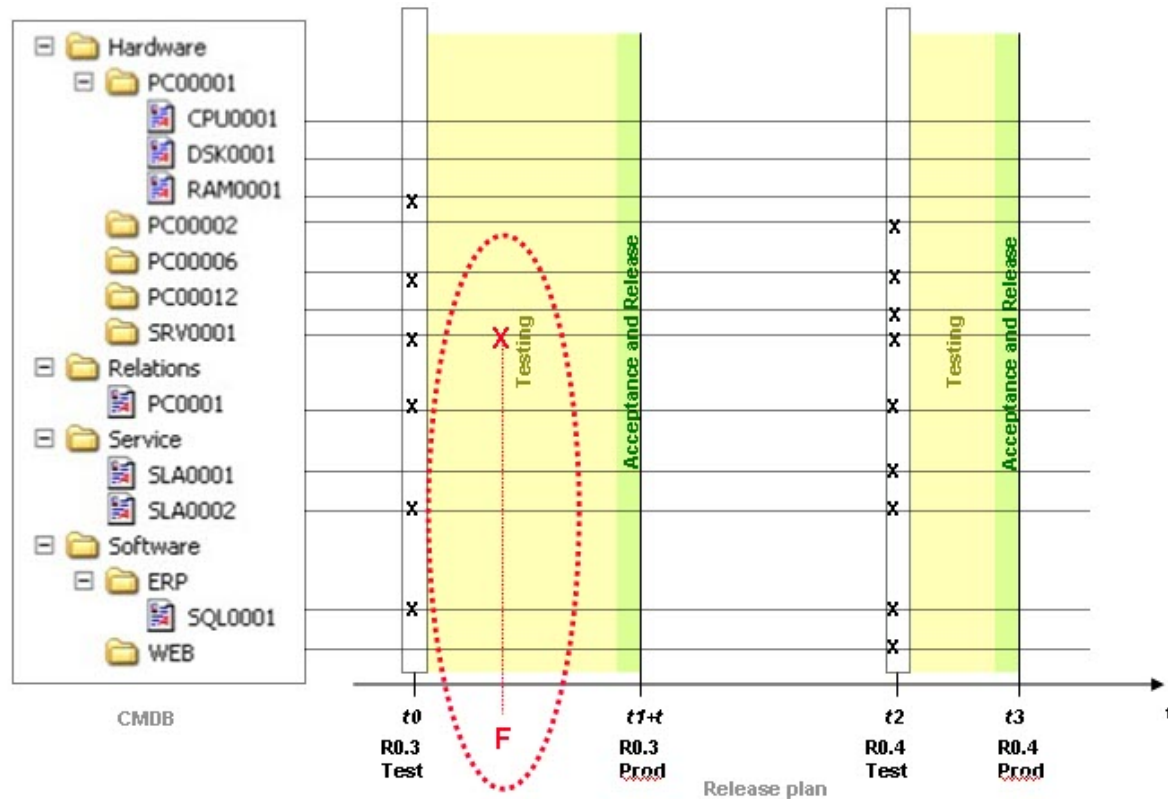
**FIGURE 4-2 COLLABORATION OF TEST-, RELEASE- AND CONFIGURATION MANAGEMENT**

The Configuration Management Database (CMDB) contains already approved Software and Hardware. The system to be built is not part of the CMDB unless it is released and put into production. In the production environment it can impact existing assets of the CMDB. These impacts are marked as 'X' in the picture above.

Whether the Test or Release Management is primary involved in error/bug resolving depends on the timely occurrence of a bug/error. Basically there exist two scenarios when an error can occur:

1. While the system is tested
2. During operation (after acceptance and release)

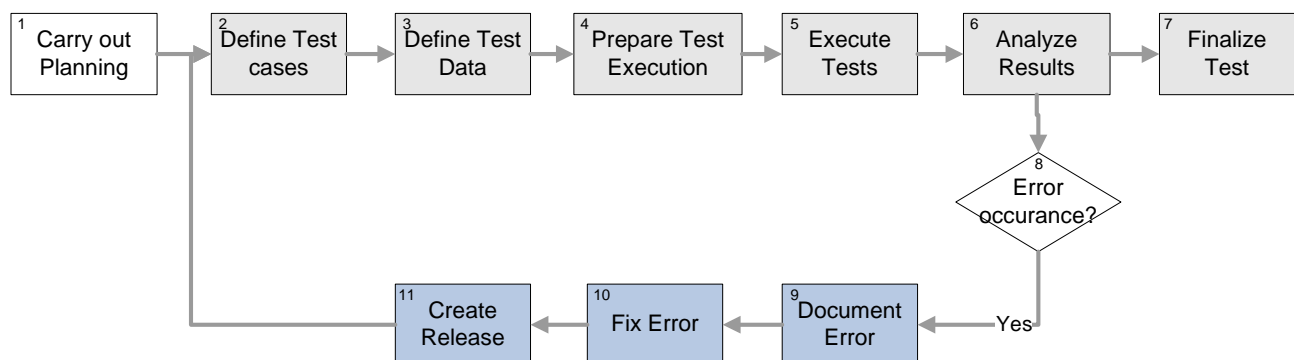
Errors occurring during Testing can have impacts on the Release Plan:



**FIGURE 4-3 IMPACT OF ERRORS OCCURING DURING TEST PHASE ON RELEASE PLAN**

The scheduled acceptance and release must be postponed by time  $t$  because the error has to be fixed.

The necessary bug fixing process while testing changes the Test process as shown below:

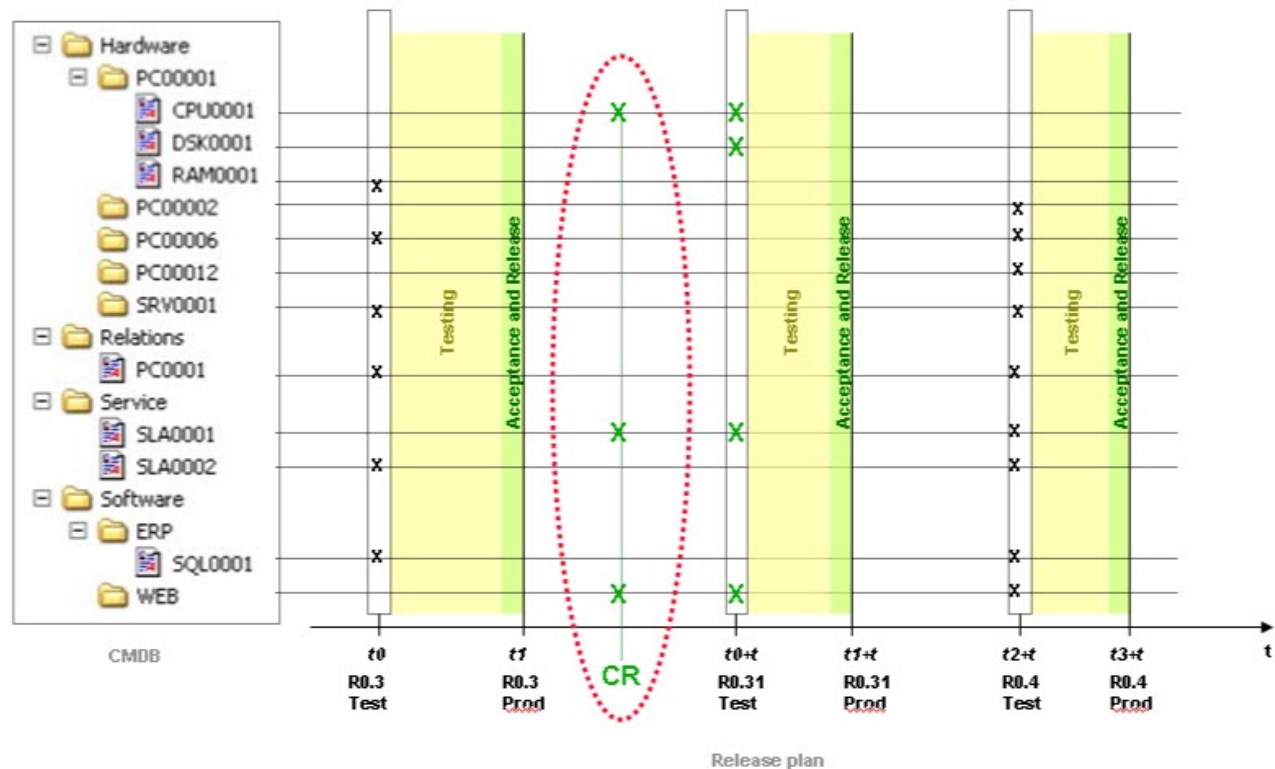


**FIGURE 4-4 ACTIVITIES OF A TEST CYCLE**

There are several enhancements in the process shown above compared to the generic process shown in Figure 2-3:

1. An additional step for activities prior to Test Execution
2. The Test finalization finishes each test cycle whereas the Test Process is finalized by the Product approval

Errors occurring during operation can have tremendous impacts on the Project Plan:



**FIGURE 4-5 BUG OCCURENCE DURING OPERATION**

Errors during operation can cause a Change Request (CR) which (together with other CRs) start a new development cycle producing a new release of type delta, minor or major.

## 4.1 Test Case Definition

The complete and consistent test case definition helps to deliver test execution efficiently. A Test case should contain following information:

1. Unique Identifier
2. Description of purpose and intended goals
3. Flag indicating Success / Failure
4. Date of execution
5. Execution's Start and End time
6. Detailed description of each step containing
  - a. Unique Identifier
  - b. What to perform
  - c. What to expect
  - d. Gained Result
  - e. Comment

The description of a test case must be clear and explicit. That ensures that the same Test case executed from different testers will lead to the same result.

Test cases should be reviewed. The necessary time and resources have to be taken into account with Test planning.

## 4.2 Automation

Test task automation is a two edged sword and it is the initiator of a lot of discussions.

On the one hand it seems obvious that automation can spare time and beware testers from performing recurring stupid actions. Testers are humans so the execution of the same tasks over and over again can cause different results – even errors - because of different individual interpretation of test goals (depending on the definition of the expected test results in the test case) as well as the lack of concentration when redelivering the same work for several times.

On the other hand the effort for definition what to automate, the necessary knowledge, possible additional tooling and the maintenance of the automation resources such as scripts have to be considered.

## 4.3 Tools

The tools used for Testing are depending on the enterprise and their software ecosystem e.g. already existing tools.

For proper tool selection the various Process steps, their inputs and outputs have to be considered. Especially following topics should be looked at when choosing Test Tools:

1. Test case definition, documentation and execution
2. Bug/Error capturing, documentation and tracking
3. Test automation

The integration of Error/Bug and Test Management is desired to gain Consistency, Correctness and Completeness (3C) as well as Traceability.

## 4.4 Roles

Roles are an important part of a Process Definition and a key success factor for Process management. Some key roles with Testing:

1. Test Manager: Owner of the process and responsible for the entire test management.
2. Domain experts. Responsible for the full test coverage of the business and technical requirements. Domain experts should also participate in Test execution (act as Testers) as they are also representing the various user groups of the system to be build.
3. Tester: Testers are responsible for the test execution and result capturing and documentation. At least the different Personas (see Reference [5]) identified for the project should be testers. Tester should also participate in Test case definition and reviews.

Testing will fail without the proper role definitions and full and useful casting of the roles with respect to all possible actors interacting with the system.

## 4.5 Environments

The infrastructure is a key factor for successful testing and therefore the availability of the proper environments has to be planned in time. At least three different environments are necessary:

1. Development
2. Test
3. Production

The Development environment should be decoupled from the Test environment to prevent interference from development to test execution. All Development Tests are performed in the Development environment.

The Test environment should be an exact copy of the Production environment (Hardware and Software). Otherwise the Test Results collected in the Test environment are nice to have but they are not very meaningful and cannot even be trusted. It is up to the particular testing setup of a project which tests are executed in the Production environment. At least the Acceptance tests should be done there.

## 4.6 Results

To fulfill the 3Cs, the results of the test execution should be documented and archived in a repository. This has to be considered when the tools are chosen. The tool should also enable an extensible Reporting.

The Test Completion Report is necessary for the final approval.

The Test Result Report is used for the Knowledge Base and serves the continuous improvement process (according CMM).

## 5 Appendix

### 5.1 Acronyms

ACRONYM	DEFINITION
CMDB	Configuration Management Database
CR	Change Request
SLA	Service Level Agreement
WAI	Web Accessibility Initiative ( <a href="http://www.w3.org/wai">www.w3.org/wai</a> )
3C	Consistency, Correctness and Completeness
CMM	Capability Maturity Model

### 5.2 References

NR	FILE / DOCUMENT / REFERENCE	COMMENT
[1]	<a href="http://www.vivtek.com/wftk/process_defn.html">http://www.vivtek.com/wftk/process_defn.html</a>	Process definitions
[2]	<a href="http://www.itil.org/">http://www.itil.org/</a>	IT Infrastructure Library
[3]	<a href="http://www.cse.dcu.ie/essiscope/sm2/9126ref.html">http://www.cse.dcu.ie/essiscope/sm2/9126ref.html</a>	Quality criteria of Software according ISO/IEC 9126
[4]	Writing Secure Code	Microsoft Press, 2003
[5]	<a href="http://www.agilemanagement.net/Articles/MSF/InnovationinMSFv4.0.html">http://www.agilemanagement.net/Articles/MSF/InnovationinMSFv4.0.html</a>	MSF 4.0 Overview and Definitions