Software Quality and Testing

TURKCELL TECHNOLOGY
19.09.2013

TURKCELL

İSTANBUL TEKNİK ÜNİVERSİTESİ
1773

TURKCELL
AKADEMİ

COURSE CONTENT

1 – Fundamentals of Testing

Definition of Testing, Why It Is Required, Common Testing Principles, Basic Testing Process, Test Psychology

2 – Testing in Software Development

Software Development Models, Sources of Software Defects, Test Levels

3 – Software Test Types

Functional Tests (structural test, regression test, integration test, performance test, web service test)

Non-functional tests (Load Test, Stress Test, Security Test)

COURSE CONTENT

4 – Test Management

Test Team, Testing Roles and Responsibilities

5 – Processes

Software Development Life Cycle (SDLC) Process, Requirement and Business Analysis, Design, Build, Release

6 – Testing in SDLC

Test Planning (Test Case Standards and Criterias), Test Set Criterias, Test Issue Tracking

COURSE CONTENT

7 – Test Design Techniques

Test Cases, Black-Box Testing, White-Box Testing, Experience Based Testing, Test Driven Development, Release Testing

8 – Test Automation & Performance Tools

Automation Tool Selection, Implementation and Risks

9 – Test Deployment & Release Management & Production Tracking Process

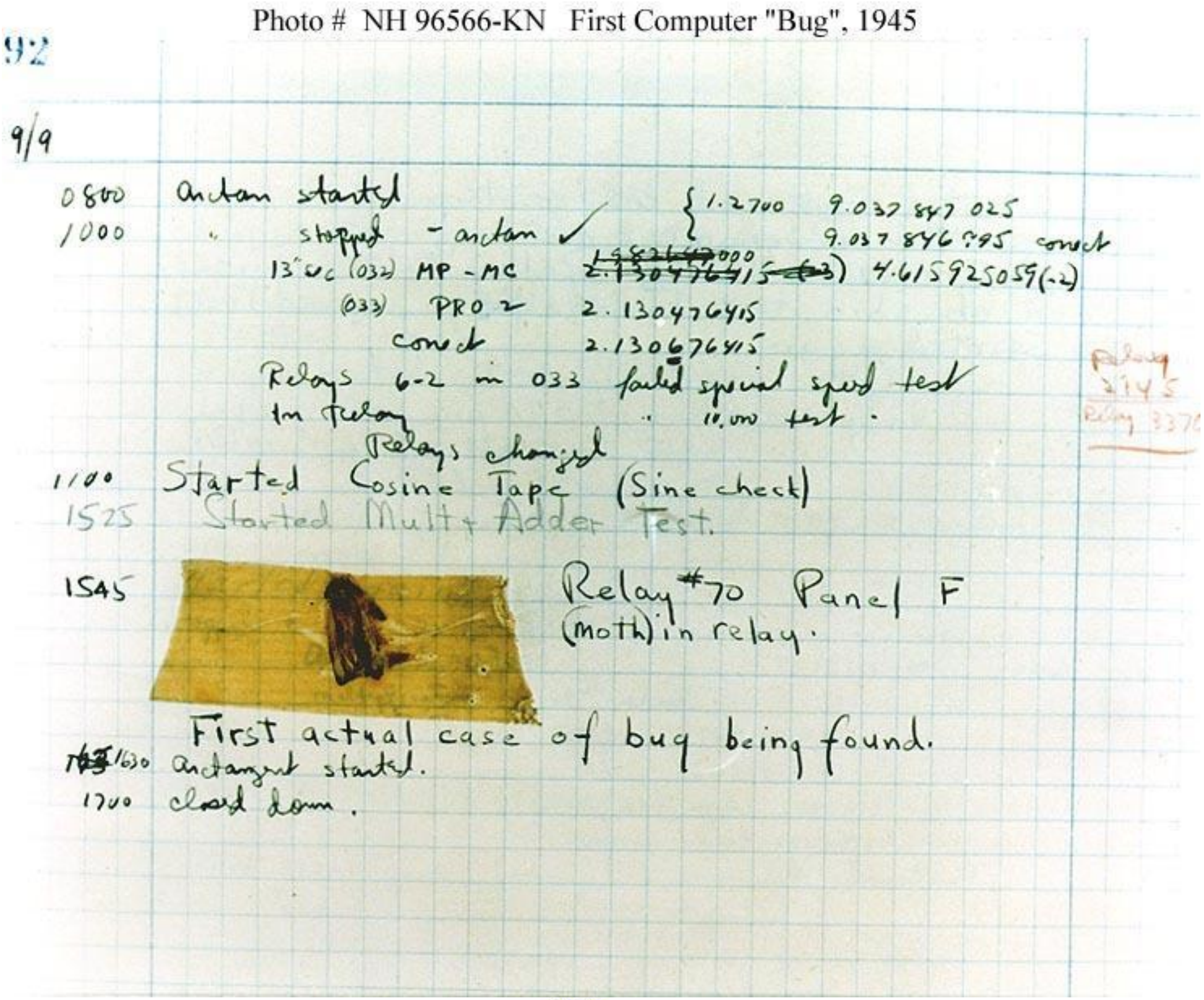


Photo # NH 96566-KN First Computer "Bug", 1945

92

9/9

0800 Arctan started
1000 " stopped - arctan ✓ {1.2700 9.037 847 025
9.037 846 995 conect
13" UC (032) MP - MC ~~1.982147000~~ ~~2.130476415~~ ~~(33)~~ 4.615925059(-2)
(033) PRO 2 2.130476415
conect 2.130676415
Relays 6-2 in 033 failed special speed test
in relay " 11.000 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~143~~1630 Antangent started.
1700 closed down.

Relay 2145
Relay 3370

First Computer Bug

- First computer bug in 1945 Harvard University by Grace Murray Hopper who led to the development of COBOL.
- A butterfly found stuck between the relay contacts while working on the Mark II Aiken Calculator.
- Operators eliminated the butterfly and ‘Debugging’ the word has come from here.

FUNDAMENTALS OF TESTING

- DEFINITION OF TESTING
- WHY IT IS REQUIRED
- COMMON TESTING PRINCIPLES
- BASIC TESTING PROCESS
- TEST PSYCHOLOGY

DEFINITION OF TESTING

General Knowledge About Testing

Testing as a process that takes place throughout the software development life cycle (Test planning,test case preparation, evaluate the results, examining completion (or exit) criteria, reporting on the testing process…)

When a test finds a defect that must be fixed, a programmer must do some work to locate the defect in the code and make the fix. In this process, called debugging, a programmer will examine the code for the immediate cause of the problem, repair the code and check that the code now executes as expected.

DEFINITION OF TESTING

Software Process Testing

In development testing (which includes component, integration and system testing), the main objective may be to cause as many failures as possible so that defects in the software are identified and can be fixed.

The users of the software may carry out acceptance testing to confirm that the system works as expected and to gain confidence that it has met the requirements.

We may continue to test the system once it is in operational use. In this case, the main objective may be to assess system characteristics such as reliability or availability.

WHY IT IS REQUIRED

Software Systems Context

These days, almost everyone is aware of software systems. We encounter them in our homes, at work, while shopping, and because of mass-communication systems. More and more, they are part of our lives. We use software in day-today business applications such as banking and in consumer products such as cars and washing machines.

Most people have had an experience with software that did not work as expected: an error on a bill, a delay when waiting for a credit card to process and a website that did not load correctly.

WHY IT IS REQUIRED

Why Software Fails

We know that people make mistakes - we are fallible.

We agree that any human being, programmers and testers included, can make an error. These errors may produce defects in the software code or system, or in a document.

- Requirement based defects
- Design based defects
- New and complicated technology
- Unsufficient test process
- On time delivery KPI
- Infrastructure problems etc.

WHY IT IS REQUIRED

The Role of Test

Rigorous testing is necessary during development and maintenance to identify defects, in order to reduce failures in the operational environment and increase the quality of the operational system.

Executing tests helps us move towards improved quality of product and service, but that is just one of the verification and validation methods applied to products.

We may also be required to carry out software testing to meet contractual or legal requirements, or industry-specific standards.

WHY IT IS REQUIRED

Testing and Quality

Testing helps us to measure the quality of software in terms of the number of defects found, the tests run, and the system covered by the tests.. When testing does find defects, the quality of the software system increases when those defects are fixed, provided the fixes are carried out properly.

A well-designed test will uncover defects if they are present and so, if such a test passes, we will rightly be more confident in the software and be able to assert that the overall level of risk of using the system has been reduced.

WHY IT IS REQUIRED

Testing and Quality

When we start a new project, it is worth learning from the problems encountered in previous projects or in the production software. As testers, we are also interested in looking at defects found in other projects, so that we can improve our processes. Process improvements should prevent those defects recurring and, as a consequence, improve the quality of future systems.

Organizations should consider testing as part of a larger quality assurance strategy, which includes other activities (e.g., development standards, training and root cause analysis).

WHY IT IS REQUIRED

Cost of Quality

Cost of Quality = Cost of improving the quality of application + Cost of finding and fixing defects

Test Team

Test Automation Tools

Fixing the defects

Testing the corrections

Solving the customer problems

Damaging the image of the company

Missing opportunities for new or cross-selling

Result = Increasing customer satisfaction and decreasing costs

WHY IT IS REQUIRED

Cost of Quality - Case Study

Is it reduces the cost of quality to invest in test?

A company that offers services only through information technology, 1000-error correction within every three months and we assume that the costs of these errors in the example below

The cost of fixing defects by developers=10$

The cost of fixing defects by test engineers=100$

The cost of fixing defects by customers=1000$

WHY IT IS REQUIRED

Cost of Quality - Case Study

None defined a Test Process

Total defects found in three months=1000

Defects found by developers=250

Defects found by customers=750

Cost of quality:	
Test Team	0$
Automation Tools	0$
Total Investment	0$
Cost of fixing the defects:	
Application Development Team	
Fixed defect number	250
Cost	2.500$
Test Team	
Fixed defect number	0
Cost	0
Customer Support	
Fixed defect number	750
Cost	750.000$
Cost of quality	**0$**
Cost of fixing the defects	**752.500$**
TOTAL=	**752.500$**

WHY IT IS REQUIRED

Cost of Quality - Case Study

Defined Test Process

Cost of quality:

Test Team	100.000$
Automation Tools	20.000$
Total Investment	120.000$

Total defects found in three months=1000

Defects found by developers=250

Defects found by testers=550

Defects found by customers=200

Cost of fixing the defects:

Application Development Team

Fixed defect number	250
Cost	2.500$

Test Team

Fixed defect number	550
Cost	55.000$

Customer Support

Fixed defect number	200
Cost	200.000$

Cost of quality	**120.000$**
Cost of fixing the defects	**257.500$**
TOTAL=	**377.500$**

Software Quality and Testing

TURKCELL TECHNOLOGY
06.10.2011

GENERAL TESTING PRINCIPLES

Fundamental Principles of The Test

7 Fundamental Principles Common For All Testing :

Principle 1 – Testing shows presence of defects. Testing cannot prove that there are no defects.

Principle 2 – Exhaustive testing is impossible. Testing everything (all combinations of inputs and preconditions) is not feasible except for trivial cases. Instead of exhaustive testing, we use risks and priorities to focus testing efforts.

Principle 3 – Early testing. Testing activities should start as early as possible in the software or system development life cycle.

GENERAL TESTING PRINCIPLES

Fundamental Principles of The Test

Principle 4 – Defect clustering. A small number of modules contain most of the defects discovered during prerelease testing or show the most operational failures.

Principle 5 – Pesticide Paradox. If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new bugs. The test cases need to be regularly reviewed and revised.

Principle 6 – Testing is context dependent. Testing is done differently in different contexts.

Principle 7 – Absence-of-errors fallacy. Finding and fixing defects does not help if the system built is unusable and does not fulfill the users' needs and expectations.

TURKCELL

TURKCELL AKADEMİ

WHY IT IS REQUIRED

Testing Duration

100% testing coverage is not possible, how much should the software/system be tested?

- Test closure point will vary according to the requirements of the projects.

- Duration of the test phases varies depending on the risk of the application within the scope of the project.

- The risks that may arise during the test phases determines the types and duration of the tests.

TURKCELL

İSTANBUL TEKNİK ÜNİVERSİTESİ 1773

TURKCELL AKADEMİ

.

GENERAL TESTING PRINCIPLES

Risk Analysis

There are two kinds of risks.

1. Product Risks

Extensibility

Environment

Software characteristics (Security, Usability, Reliability)

Technical complexity

Performance

Data integrity

2.Project Risks

Deadlines for the projects phases

Budget

Non-efficient usage of the resources

Team competencies

GENERAL TESTING PRINCIPLES

Risk Assessment

The purpose of the test is to understand and minimize the risks.

For a good risk management strategy, all the risks identified at the analysis phase should be considered.

There are important steps in the risk management :

- Identification and evaluation of risks
- Prioritization of risks
- Preparation the risk management strategy for preventing against the risk

GENERAL TESTING PRINCIPLES

Risk Assessment

Risk based testing approach :

- Selects the most appropriate testing techniques.
- Provides critical input for the test duration.
- Allows the identification of the priority of the tests.
- Ensures the detection of the significant errors as soon as possible.
- Focuses on high risk areas.
- Identifies the training that will be organized for the project team within the help of the assesment of the risks.

GENERAL TESTING PRINCIPLES

FMEA

FAILURE MODE & EFFECT ANALYSIS TECHNIQUE

A procedure in product development and operations management for analysis of potential failure modes within a system for classification by the severity and likelihood of the failures.

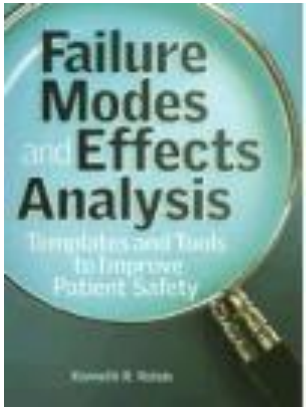
Step 1 – Specify The Quality Risk Categories

Step 2 – Determine The Possible Risk Areas

Step 3 – Ranking The Possible Risk Areas

Step 4 – Determine Risk Priority Number (RPN)

Step 5 – Determine Testing Density Using RPN



GENERAL TESTING PRINCIPLES

FMEA

Step 1 – Specify The Quality Risk Categories

Functionality

Performance

Load Capacity

Operations and sustainability

Data Quality

Documentation

Integration…..

GENERAL TESTING PRINCIPLES

FMEA

Step 2 – Determine The Possible Risk Areas

Entering the possible risks under the quality risk categories found in Step 1.

Example : TTECH-Billing Application

Category=Functionality 1.000

Case 1.001:Reproduction of two invoices for the same period

Case 1.002:Not taken Access Fees

Category=Performance 2.000

Case 2.001:Long lasting Billing phase ….

GENERAL TESTING PRINCIPLES

FMEA

Step 3 – Ranking The Possible Risk Areas

Grading the risks entered in Step 2.

1. According to the importance of the system (severity)
2. According to the priority of the customer (priority)
3. According to the priority of the realization (likelihood)

GENERAL TESTING PRINCIPLES

FMEA

Ranking The Risk – Severity

According to the importance of the system

1. Data loss
2. Loss of functionality
3. Repairable loss of functionality
4. Partial loss of functionality
5. Cosmetic Risk

GENERAL TESTING PRINCIPLES

FMEA

Ranking The Risk – Priority

According to the priority of the customer

1. Urgent
2. Mandatory
3. Major
4. Low
5. Optional

GENERAL TESTING PRINCIPLES

FMEA

Ranking The Risk – Likelihood

According to the priority of the realization

1. Probable
2. Possible
3. Improbable

GENERAL TESTING PRINCIPLES

FMEA

Example : TTECH-Billing Application

Category=Functionality 1.000

Case 1.001:Reproduction of two invoices for the same period

Severity =1 – Data loss

Priority = 1 – Urgent

Likelihood = 2 – Possible

Case 1.002:Long lasting Billing phase

Severity =2 – Loss of functionality

Priority = 1 – Urgent

Likelihood = 2 – Possible

GENERAL TESTING PRINCIPLES

FMEA

Step 4 – Determine Risk Priority Number (RPN)

After ranking the severity, priority and likelihood, the RPN - Risk Priority Number can be easily calculated by multiplying these three numbers.

The failure modes that have the highest RPN should be given the highest priority for corrective action.

GENERAL TESTING PRINCIPLES

FMEA

Example : TTECH-Billing Application

Category=Functionality 1.000

Case 1.001:Reproduction of two invoices for the same period

Severity =1 – Data loss

Priority = 1 – Urgent

Likelihood = 2 – Possible

RPN value for the risk = 1x1x2 =2

Case 1.002:Long lasting Billing phase

Severity =2 – Loss of functionality

Priority = 1 – Urgent

Likelihood = 2 – Possible

RPN value for the risk = 2x1x2 =4

GENERAL TESTING PRINCIPLES

FMEA

Step 5 – Determine Testing Density Using RPN

Testing density of the unit or the module of the software is determined according to the RPN value.

Testing Density	RPN Interval
Extensive Testing	1-5
Balanced Testing	6-20
Opportunity Testing	21-50
Reporting Observed Bugs	51-

GENERAL TESTING PRINCIPLES

FMEA

Example : TTECH-Billing Application

Category=Functionality 1.000

Case 1.001:Reproduction of two invoices for the same period

Severity =1 – Data loss

Priority = 1 – Urgent

Likelihood = 2 – Possible

RPN value for the risk = 1x1x2 =2 **(Extensive Testing)**

Case 1.002:Long lasting Billing phase

Severity =2 – Loss of functionality

Priority = 1 – Urgent

Likelihood = 2 – Possible

RPN value for the risk = 2x1x2 =4 **(Extensive Testing)**

FUNDAMENTAL TEST PROCESS

Planning and Control

Test Planning:

During test planning, we make sure we understand the goals and objectives of the customers, stakeholders, and the project, and the risks which testing is intended to address. This will give us what is sometimes called the mission of testing or the test assignment.

- Determine the scope and risks and identify the objectives of testing,
- Determine the test approach (techniques, test items, coverage, identifying and interfacing with the teams involved in testing, testware),

FUNDAMENTAL TEST PROCESS

Planning and Control

Test Planning:

- Implement the test policy and/or the test strategy and set the goals,
- Determine the required test resources (e.g. people, test environment, PCs),
- Schedule test analysis and design tasks, test implementation, execution and evaluation,
- Determine the exit criteria

FUNDAMENTAL TEST PROCESS

Planning and Control

Test Control:

Test control is an ongoing activity. We need to compare actual progress against the planned progress, and report to the project manager and customer on the current status of testing, including any changes or deviations from the plan.

- Measure and analyze the results of reviews and testing,
- Monitor and document progress, test coverage and exit criteria,
- Provide information on testing,
- Initiate corrective actions,
- Make decisions

FUNDAMENTAL TEST PROCESS

Analysis and Design

Test analysis and design is the activity where general testing objectives are transformed into tangible test conditions and test designs.

- Review the test basis (such as the product risk analysis, requirements, architecture, design specifications, and interfaces), examining the specifications for the software we are testing.
- Identify test conditions based on analysis of test items, their specifications, and what we know about their behavior and structure.

FUNDAMENTAL TEST PROCESS

Analysis and Design

- Design the tests, using techniques to help select representative tests that relate to particular aspects of the software which carry risks or which are of particular interest, based on the test conditions and going into more detail.
- Evaluate testability of the requirements and system.
- Design the test environment set-up and identify any required infrastructure and tools.

FUNDAMENTAL TEST PROCESS

Implementation and Execution

During test implementation and execution, we take the test conditions and make them into test cases and testware and set up the test environment.

Implementation:

- Develop and prioritize your test cases and create test data for those tests.
- Create test suites from the test cases for efficient test execution.
- Implement and verify the environment.

FUNDAMENTAL TEST PROCESS

Implementation and Execution

Execution:

- Execute the test suites and individual test cases, following our test procedures.
- Log the outcome of test execution and record the identities and versions of the software under test, test tools and testware.
- Compare actual results with expected results.
- Where there are differences between actual and expected results, report discrepancies as incidents.
- Repeat test activities as a result of action taken for each discrepancy.

FUNDAMENTAL TEST PROCESS

Evaluating Exit Criteria and Reporting

Evaluating exit criteria is the activity where test execution is assessed against the defined objectives.

- Check test logs against the exit criteria specified in test planning
- Assess if more tests are needed or if the exit criteria specified should be changed
- Write a test summary report for stakeholders

FUNDAMENTAL TEST PROCESS

Test Closure Activities

Evaluating exit criteria is the activity where test execution is assessed against the defined objectives.

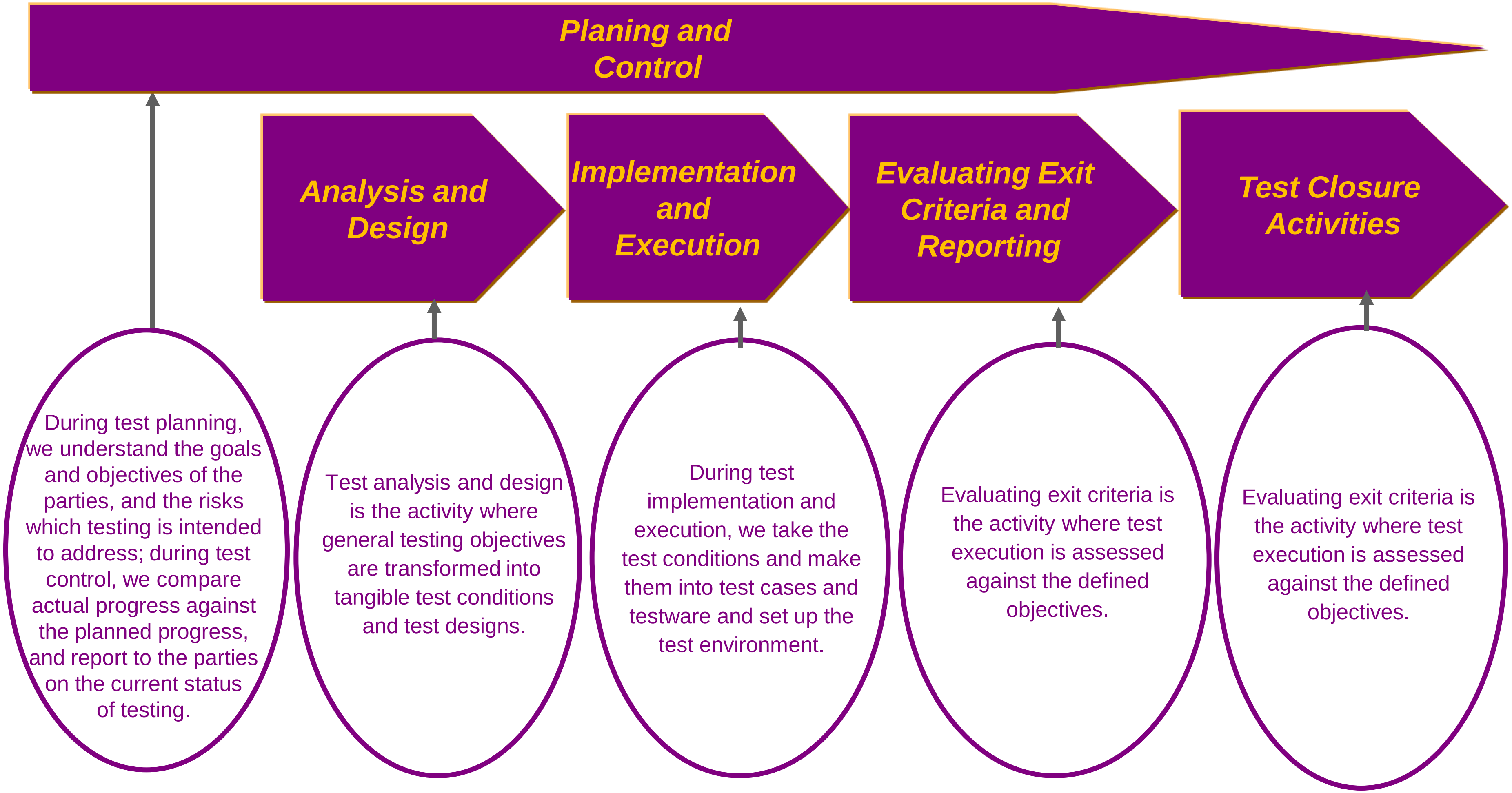
- Check which planned deliverables we actually delivered and ensure all incident reports have been resolved through defect repair or deferral.
- Finalize and archive testware, such as scripts, the test environment, and any other test infrastructure, for later reuse.

FUNDAMENTAL TEST PROCESS

Test Closure Activities

- Hand over testware to the maintenance organization who will support the software and make any bug fixes or maintenance changes, for use in confirmation testing and regression testing.
- Evaluate how the testing went and analyze lessons learned for future releases and projects.

FUNDAMENTAL TEST PROCESS



Planing and Control

Analysis and Design

Implementation and Execution

Evaluating Exit Criteria and Reporting

Test Closure Activities

During test planning, we understand the goals and objectives of the parties, and the risks which testing is intended to address; during test control, we compare actual progress against the planned progress, and report to the parties on the current status of testing.

Test analysis and design is the activity where general testing objectives are transformed into tangible test conditions and test designs.

During test implementation and execution, we take the test conditions and make them into test cases and testware and set up the test environment.

Evaluating exit criteria is the activity where test execution is assessed against the defined objectives.

Evaluating exit criteria is the activity where test execution is assessed against the defined objectives.

FUNDAMENTAL TEST PROCESS

CRITICAL PERFORMANCE INDICATORS (KPI)

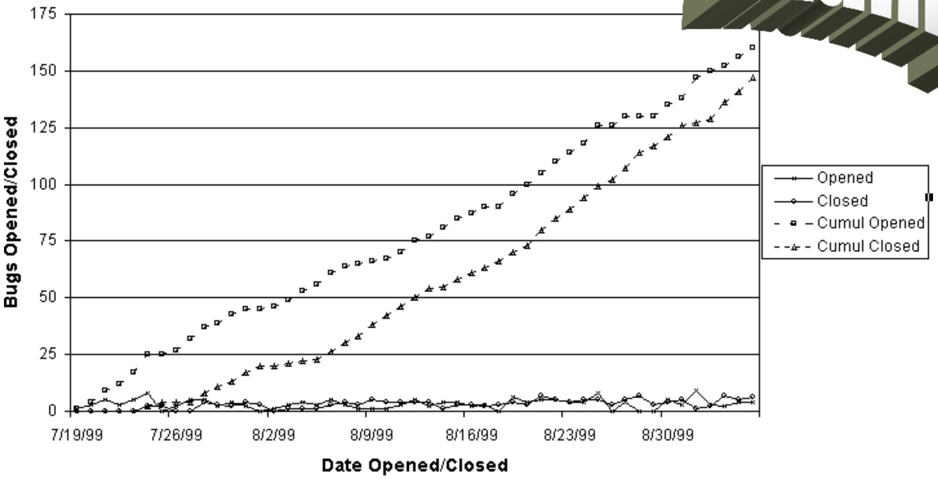
KEY PERFORMANCE INDICATORS	MEASURE	TARGET	ACTUAL	DEVIATION
Percent of test requirements turned into tests.	%	90%	80%	10%
Percent of tests executed that have passed.	%	50%	45%	5%
Average time for finding a bug.	hr	1	2	100%
Average time for fixing a bug.	hr	10	15	15%
Average age of open bugs.	hr	20	24	20%
Number of times a bug is reopened (if high).	unit	200	232	16%
Number of irreproducible bugs (perhaps as a percentage of total bugs found).	unit	100	178	78%
Total bugs rejected.	unit	50	78	56%
....				
....				

Example

FUNDAMENTAL TEST PROCESS

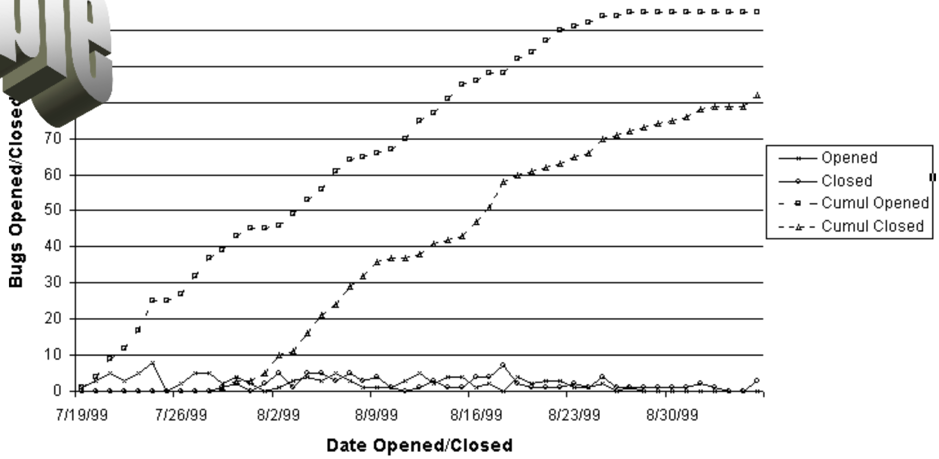
PERIODIC REPORTS

Example



Endless Bug Discovery

(System never stabilizes)



Ignored Bug Reports

(Persistent quality gap)

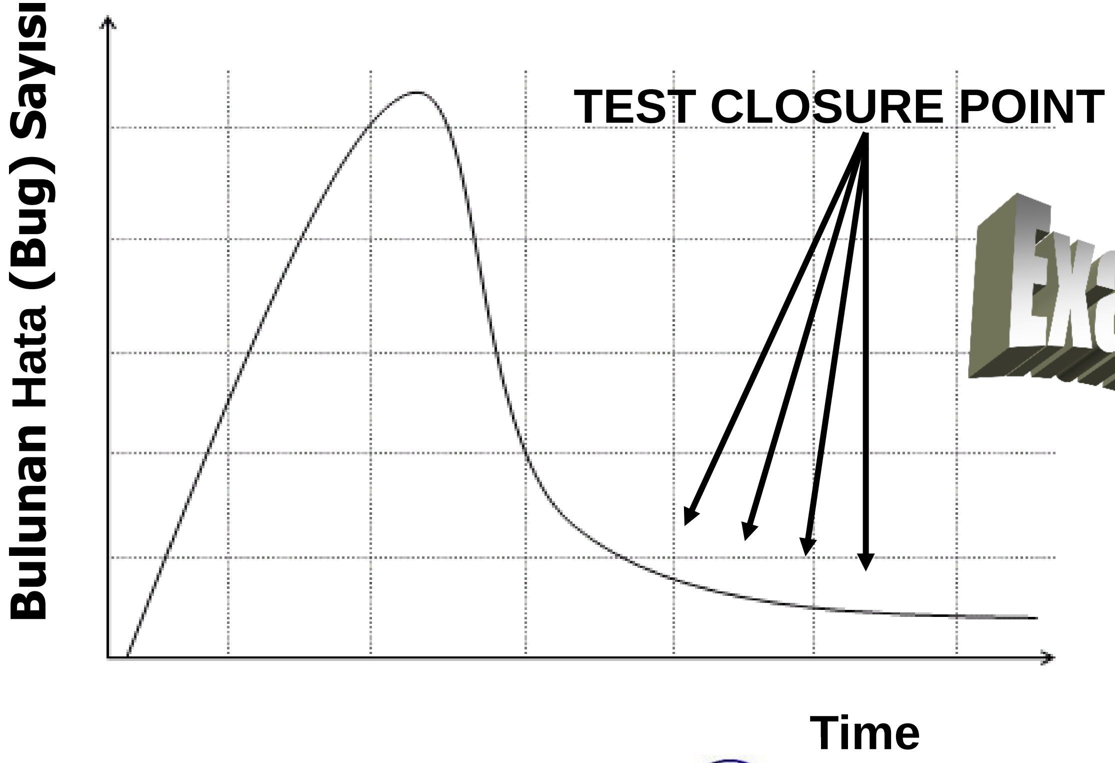
TURKCELL

İSTANBUL TEKNİK ÜNİVERSİTESİ 1773

TURKCELL AKADEMİ

FUNDAMENTAL TEST PROCESS

FINALIZATION OF TESTING



Example

THE PSYCHOLOGY OF TESTING

Independent Testing

Several levels of independence can be identified :

- Tests by the person who wrote the item under test
- Tests by another person within the same team, such as another programmer
- Tests by a person from a different organizational group, such as an independent test team
- Tests designed by a person from a different-organization or company, such as outsourced testing or certification by an external body.

TURKCELL

TURKCELL AKADEMİ

THE PSYCHOLOGY OF TESTING

Software Developer Perspective

Programmers are testers - they test the components which they build, and the integration of the components into the system.

- In general, developers can be seen constructive.
- They want to develop a good product but their

main goal is to implement the software as soon as possible.

- They can miss the big picture because of getting lost in detail.
- They want to make your own instead of the customer's requirement as a purpose of the test.

THE PSYCHOLOGY OF TESTING

Test Engineer Perspective

- In general, test engineers can be seen destructive.
- Test engineers have to be constructive to avoid problems with programmers that may occur during the tests.
- Communication problems between the test engineers, business analysts and programmers is the most common risk factors in the project.
- The biggest risk to make the tests by an independent testing group is the possible communication problems.

TURKCELL

TURKCELL AKADEMİ

THE PSYCHOLOGY OF TESTING

Test Engineer Perspective

Testing can be seen as a destructive activity, we need to take care to report on defects and failures as objectively and politely as possible.

- Communicate findings on the product in a neutral, fact-focused way without criticizing the person who created it.
- Explain that by knowing about this now we can work round it or fix it so the delivered system is better for the customer.
- Start with collaboration rather than battles. Remind everyone of the common goal of better quality systems.