SOFTWARE TESTING

Kerem Hancı

AGENDA

Software Development Life Cycle

Testing Targets

Seven Principles

Testing Process

Test Levels and Test Types

Test Design Techniques

Test Data Management

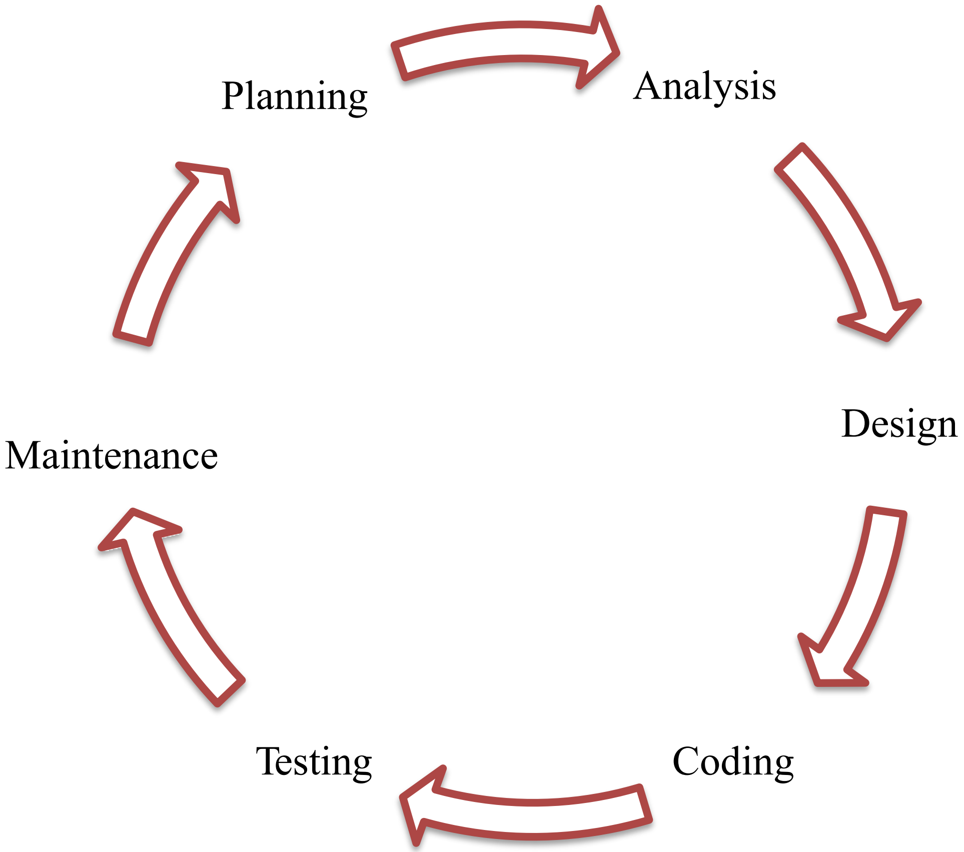
Test Automation

Modern Testing Approaches

Software?

It is a group of machine commands that enable communication of the electronis devices.*

Software Development Life Cycle



Planning

Analysis

Design

Coding

Testing

Maintenance

Software Development Life Cycle

Planing
- Time
- Source
- Requirement

Analysis
- Determining solution
- Optimization
- Feasibility

Design
- System Architecture
- User Interface

Coding
- Development
- Configuration

Testing
- Error Detection
- Quality Standards
- Evaluation
- User Decisions

Maintenance
- Continuity
- Controlling

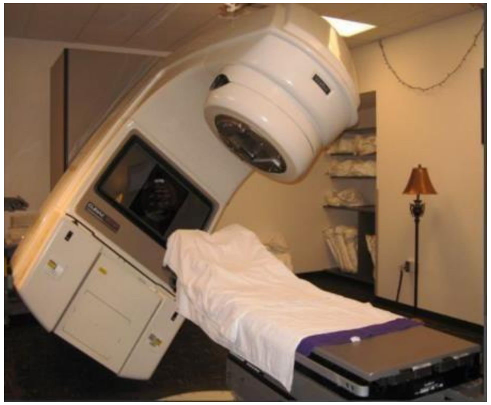
Software Testing?

A research process that includes verifying that the requirements determined for the system of the product are met and determining the differences between the expected and observed situations in case they are not met, in order to provide information to the stakeholders about the quality of the developed product*

*STQR 2016

Why so important?

1994, *China Airlines Airbus A300*
→ 264 dead, 7 seriously injured



1985, Therac 25 Radyasyon Therapy Equipment
→ 3 dead, 3 seriously injured

1996, U.S. Bank
→ 823 people were invested 920 million $ by mistake

1999, Failed Satellite Launch
→ 1.2 billion $ loss

Testing Targets

- ❑ Detecting errors,
- ❑ Confidence about the quality level,
- ❑ Providing information for decision making,
- ❑ Preventing errors

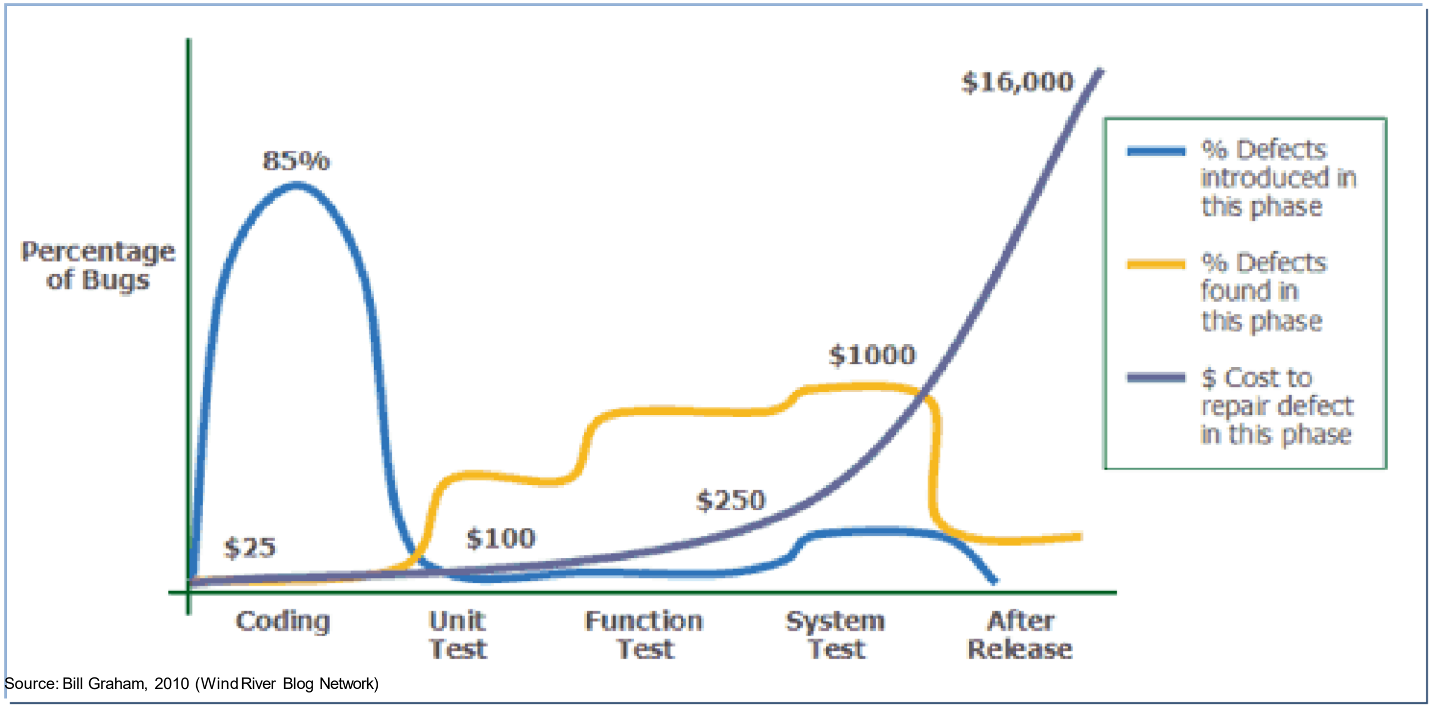
*TTB, 2014, *Test Nedir*, Sertifikalı Test Uzmanı Temel Seviye Ders Programı, Yazılım Test ve Kalite Derneği, Bölüm 1, Turkey, 15.

Testing Principles*

1. Testing shows the presence of defects, not their absence.
2. Exhaustive testing is impossible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence of errors fallacy

* TTB, 2014, *Yedi Test Prensibi*, Sertifikalı Test Uzmanı Temel Seviye Ders Programı, Yazılım Test ve Kalite Derneği, Bölüm 1, Turkey, 16.

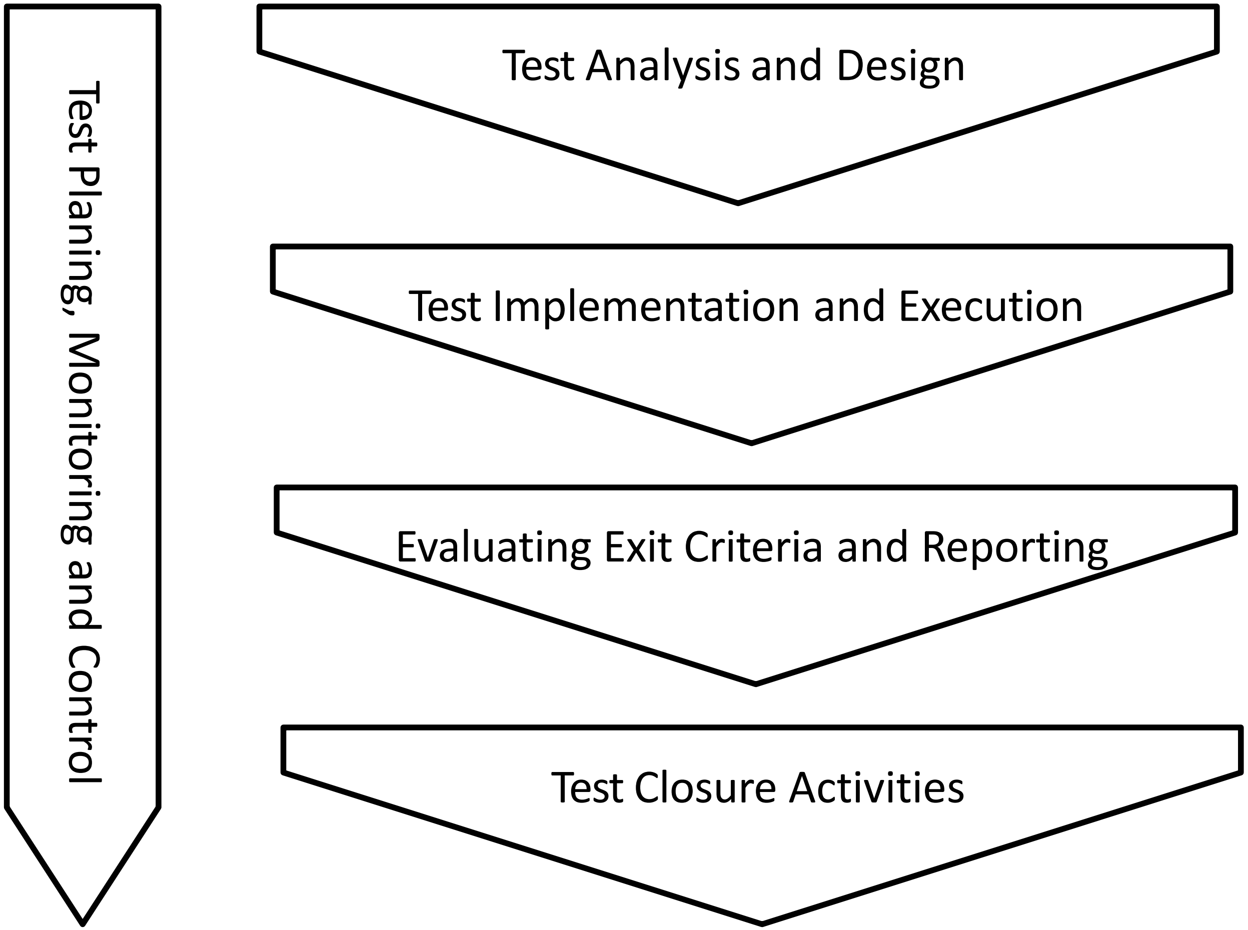
Early Detection



Source: Bill Graham, 2010 (Wind River Blog Network)

The sooner an error is detected, the less it costs.

Testing Process



Test Planing, Monitoring and Control
Test Analysis and Design
Test Implementation and Execution
Evaluating Exit Criteria and Reporting
Test Closure Activities

STLC Example

Requirement

- HES CODE must include 10 characters.

Test Analysis - Test Conditions

- The field has a 10-string limit
- Only letters and numbers should be allowed.
- Two hyphens shouldn’t be included.

(T4P7-7823-10)

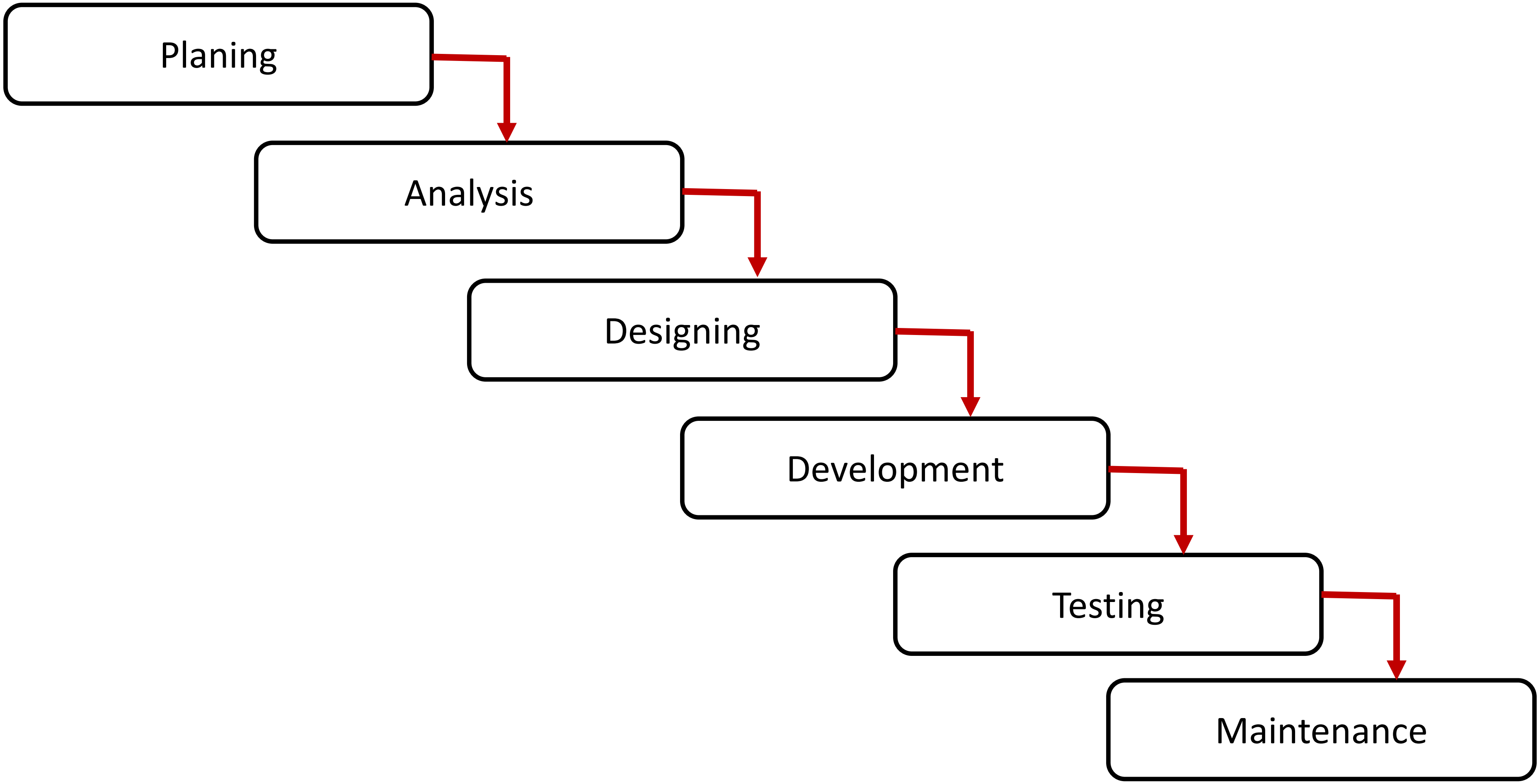
STLC Example

Test Design - Test Cases

- 10 characters should be accepted
- Greater or less than 10 characters should not be accepted
- Letters and numbers should be accepted
- Punctuation or special characters should not be accepted
- Valid – Invalid HES Code
- Hypens should be assigned automatically

HES CODE ..... - ..... - .....

Waterfall Model



Planing

Analysis

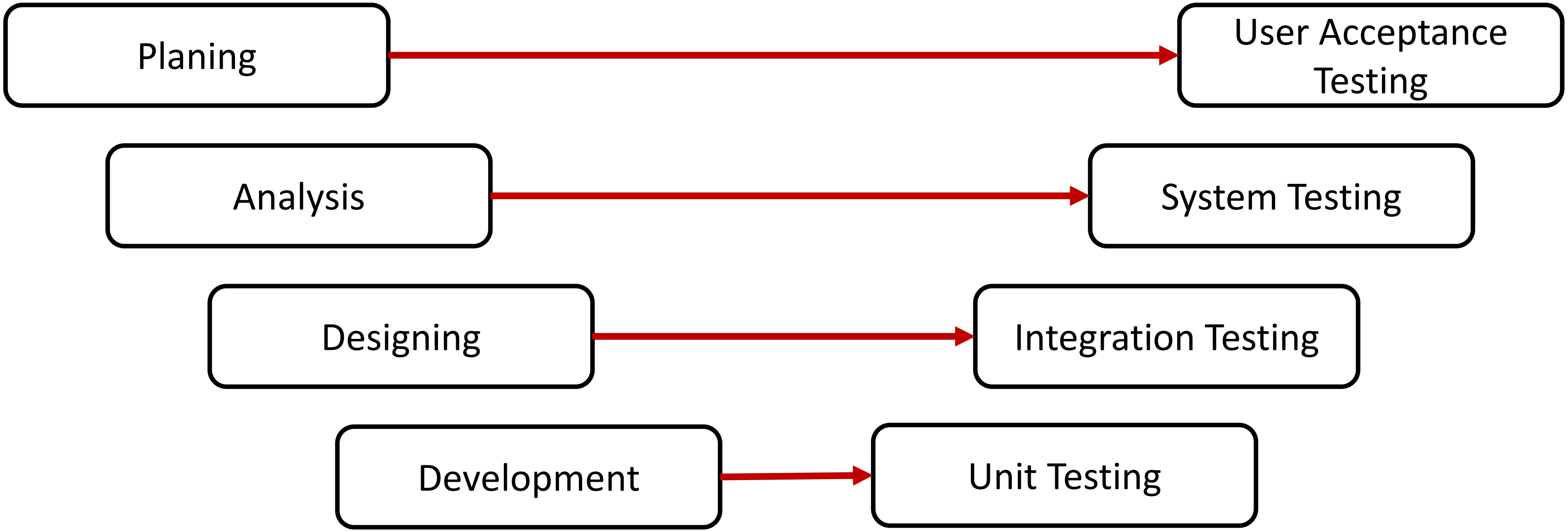
Designing

Development

Testing

Maintenance

V Model



Planing

User Acceptance Testing

Analysis

System Testing

Designing

Integration Testing

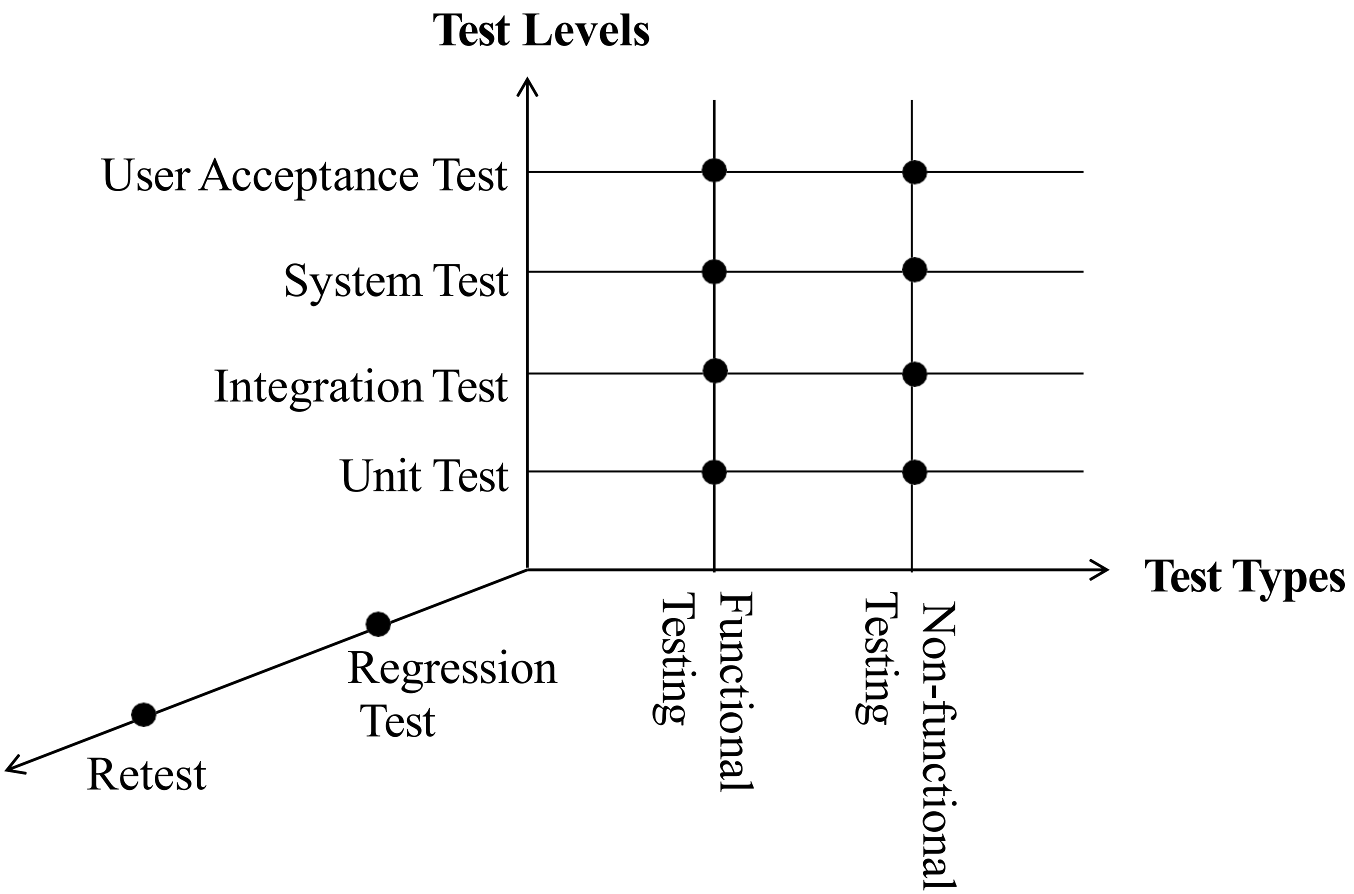
Development

Unit Testing

Agile Model

Sprint 1	Sprint 2	Sprint 3		Sprint n
Planing	Planing	Planing		Planing
Analysis	Analysis	Analysis		Analysis
Designing	Designing	Designing	....	Designing
Development	Development	Development		Development
Testing	Testing	Testing		Testing

Test Levels & Test Types



Functional Testing

- ❑ It is the test group in which the behaviors that the software should perform are tested.
- ❑ In addition to the functions that the software should do, it is very important to include the functions that the software should not do in terms of error detection.

Non-Functional Testing

❑ It is the test group in which the software is tested how it performs the behaviors it should perform.

❑ They are carried out taking into account ISO 9126 quality standards.

❑ Includes tests for usability, reliability, efficiency, maintainability and portability

Testing for Effects

- ❑ **Retest:** It is the type of test performed to confirm the corrected error.

- ❑ **Regression Test:** A type of test performed to confirm that the corrected error does not cause any other effects.

Risk

The impact and probability of situations which may cause adverse consequences if it occurs.

Risk in the software process; represents possible errors which occur when the product is taken into the live environment.

Risk Identification

❑ Expert Comments

❑ Risk Drafts

❑ Checklists

❑ Experiences

Risk = Impact X Probability

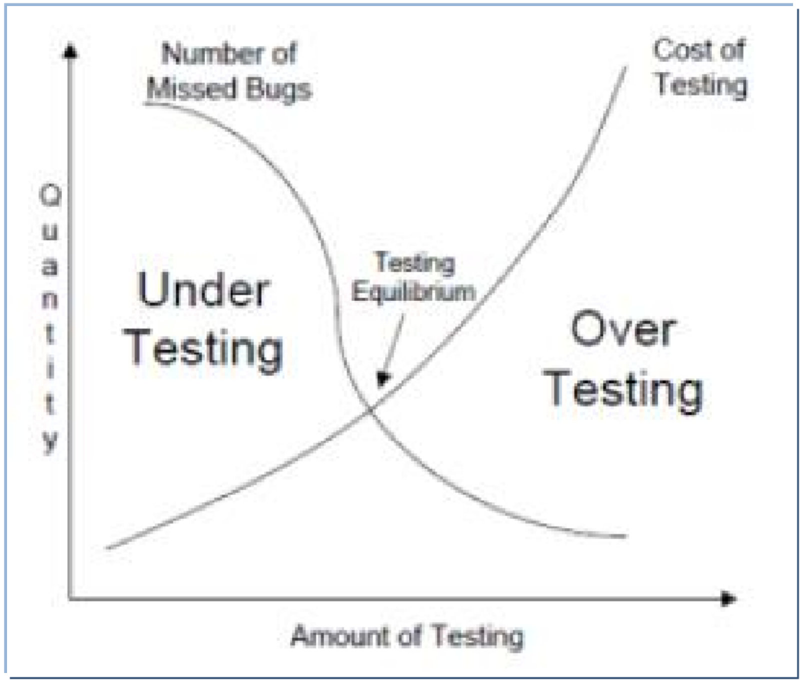
Impacts; are the impressions that the error has left on the customer, user or other stakeholders. For example; loss of reputation, loss of business, loss of trust etc.

Probability; indicates the probability of an error occurring. For example: Olasılığa code complexity, team conflicts etc.

Risk Types

- **Project Risks:** They are not related to the software itself, but are problems that may occur in the project where the software is implemented. For example; personnel issues, contrat issues etc.
- **Product Risks:** They are the problems with the software itself. ISO 9126 quality characteristics are taken into account when determining these risks. For example; functionality, usability, efficiency etc.

Software Testing and Risk Management



Overtesting increases the cost of development and hurts the budget, while undertesting increases the likelihood of encountering risks.

Defect Management

If there is difference between the expected results mentioned in the test steps and the actual results while the test scenarious are being run, it is called *anomaly*.

Anomalies are recorded and reported, and if they do not reflect the truth in other work products, they are considered as errors.

Error reports should be clear, objective and contain all the details.

Defect Report

❑ Environment

❑ Scenario

❑ Actual and expected result

❑ Data

❑ Detection time

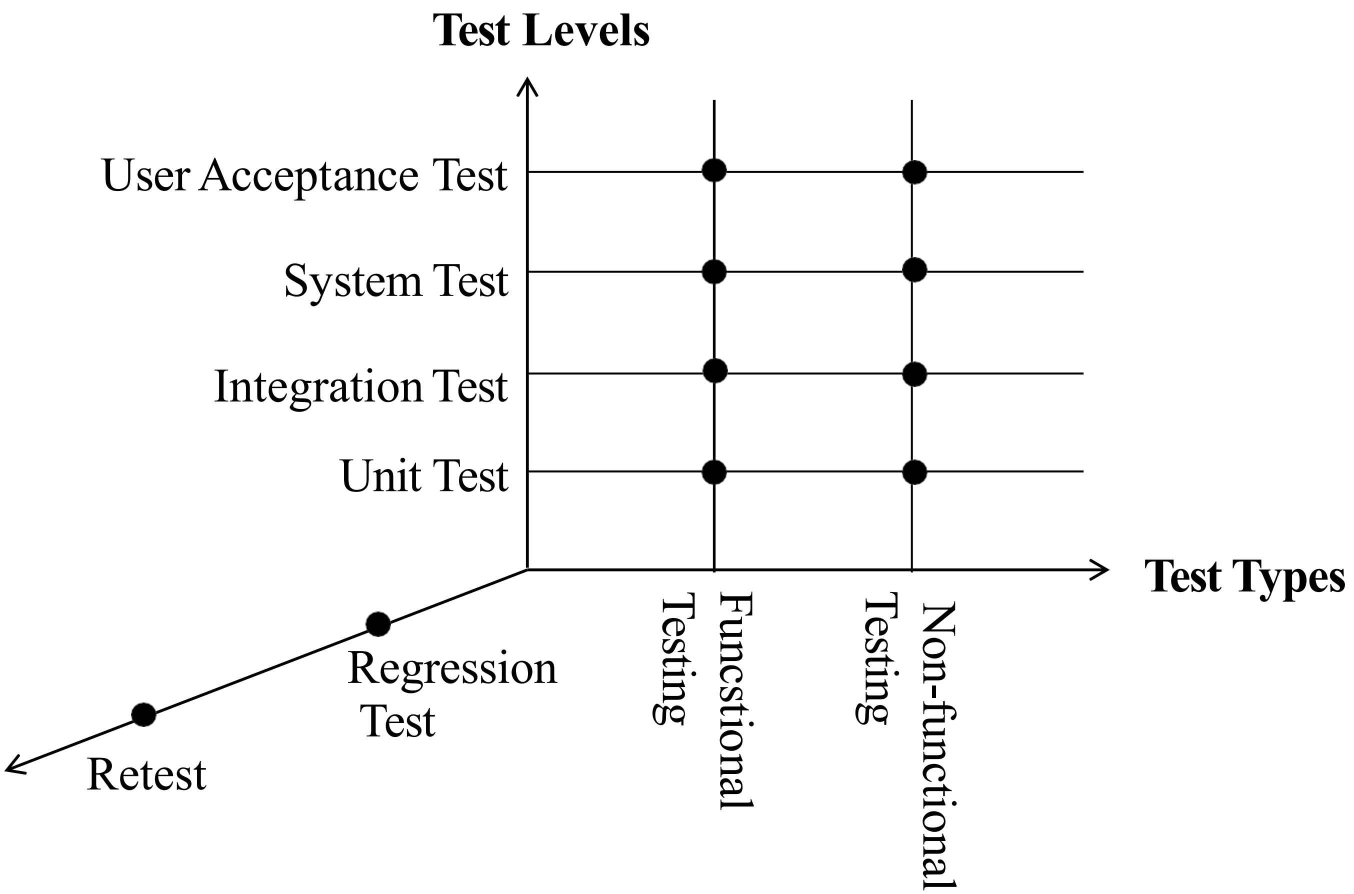
❑ Priority

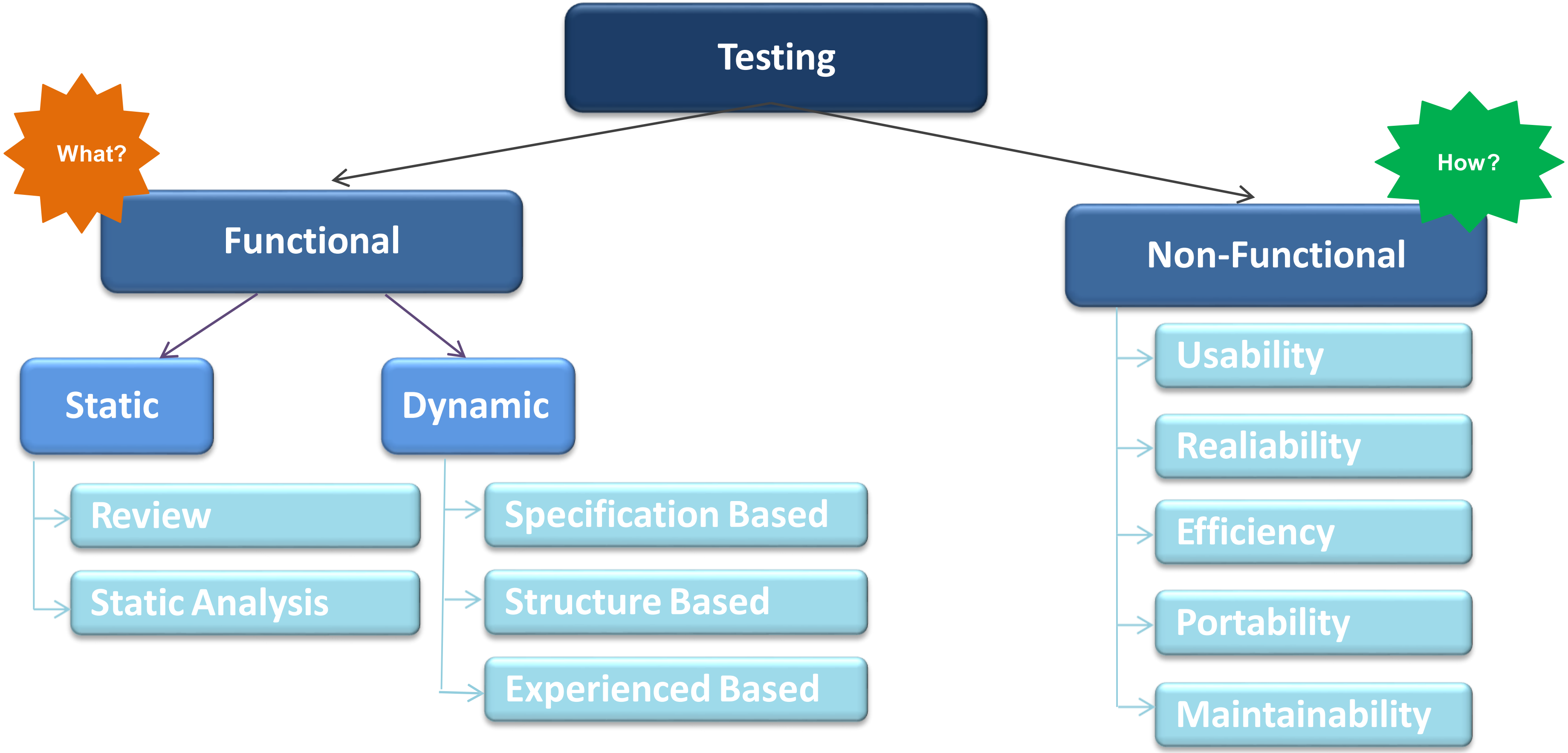
❑ Severity

❑ Assigned developer

❑ Tester

Test Levels & Test Types





Testing

What?

Functional

Static

Review

Static Analysis

Dynamic

Specification Based

Structure Based

Experienced Based

How?

Non-Functional

Usability

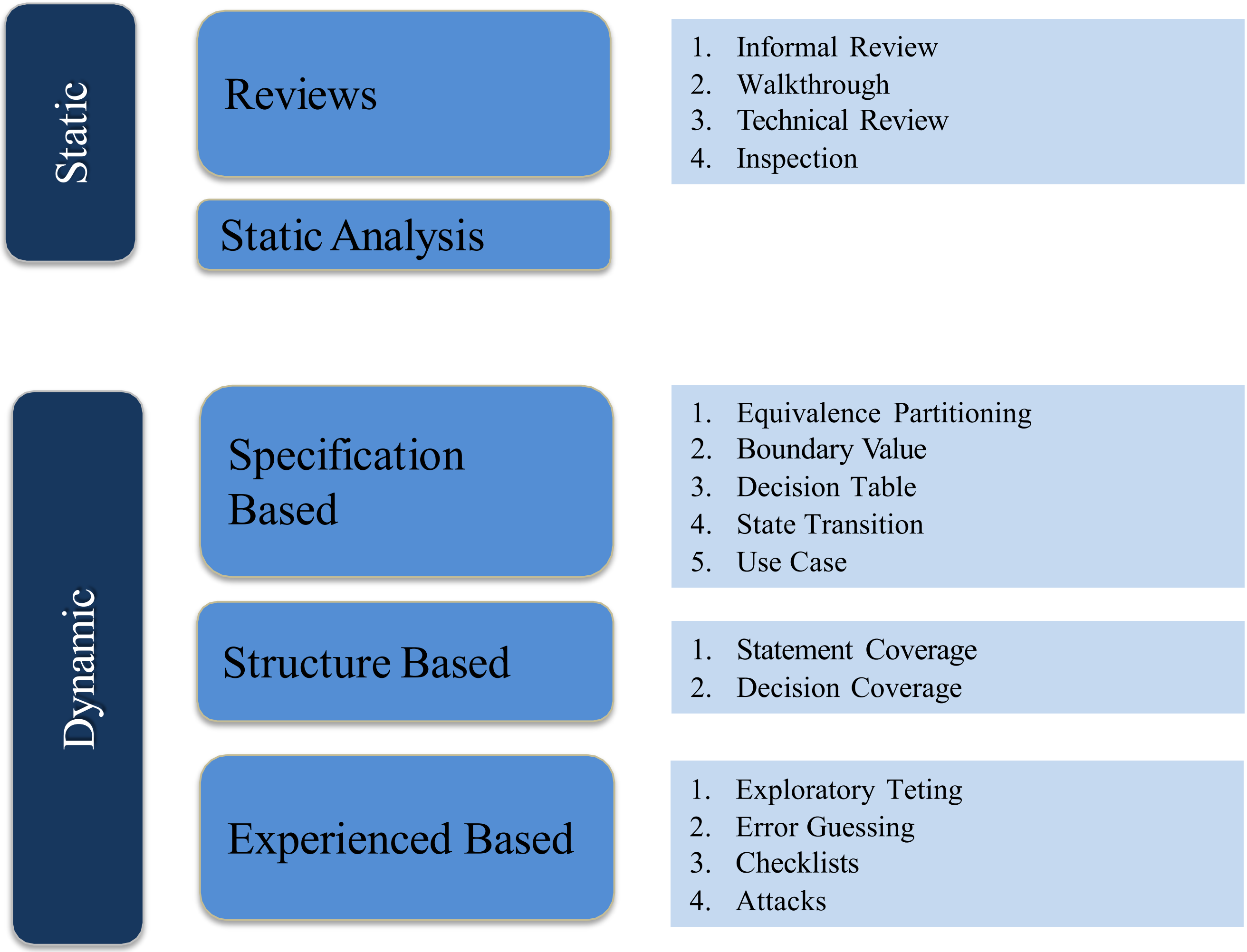
Realiability

Efficiency

Portability

Maintainability

Test Tasarım Teknikleri



Static

Reviews

1. Informal Review
2. Walkthrough
3. Technical Review
4. Inspection

Static Analysis

Dynamic

Specification Based

1. Equivalence Partitioning
2. Boundary Value
3. Decision Table
4. State Transition
5. Use Case

Structure Based

1. Statement Coverage
2. Decision Coverage

Experienced Based

1. Exploratory Teting
2. Error Guessing
3. Checklists
4. Attacks

Static Testing

Before the code starts to run, the tests performed by reviewing the requirements, analysis and design documents or to find the errors of the code by static analysis are called static testing.

Static testing is of great importance in the software development life cycle, as correcting the errors detected after the code is run will be much more costly than correcting the errors detected during static testing.

Static Testing Types

❑ Reviews

❑ Static Analysis

Reviews

- □ Informal

- □ Walkthrough

- □ Technical Review

- □ Inspection

Static Analysis

It is the process of reviewing the code to find software errors before the code is run. It helps to find implementation errors that are hard to find before the code is run.

Dynamic Testing

All of the tests performed after the code is compiled and run are called dynamic tests. Depending on the type of software to be tested, test design techniques to be applied dynamically and the application methods of these techniques may differ.

Dynamic Testing Types

❑ Specification Based

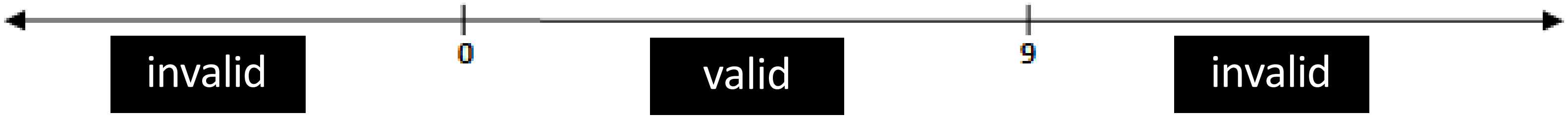
❑ Structure Based

❑ Experience Based

Equivalence Partitioning

It is a test design technique in which inputs or outputs showing the same behavior are grouped and tested. By selecting and testing representative data from each of the created groups, the number of test data is reduced. Inputs in the same group are expected to give the same expected result.

Equivalence Partitioning



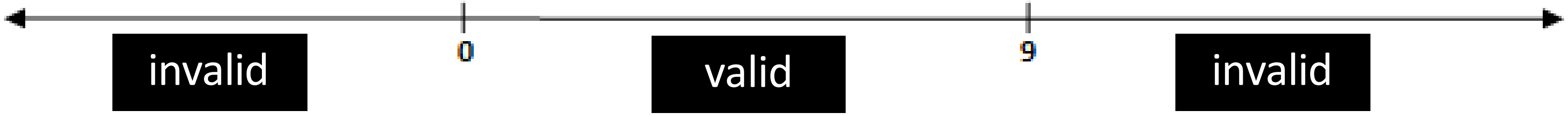
Valids : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Invalids : (-∞, -1] U [10, ∞)

Boundary Value Analysis

It is a test design technique used to test limits of eauivalence partitioning. With this technique, the validation of the breakpoints is ensured, it is confirmed that the correct breakpoint is in the right denominator.

Boundary Value Analysis



Two valued : -1, 0, 9, 10

Three valued : -1, 0, 1, 9, 10, 11

Decision Table

It is a test design technique that is used to script and run all combinations of conditions specified in analysis and requirement documents. It is one of the most effective methods used for comprehensive testing of the product as it provides confirmation of all combinations of conditions.

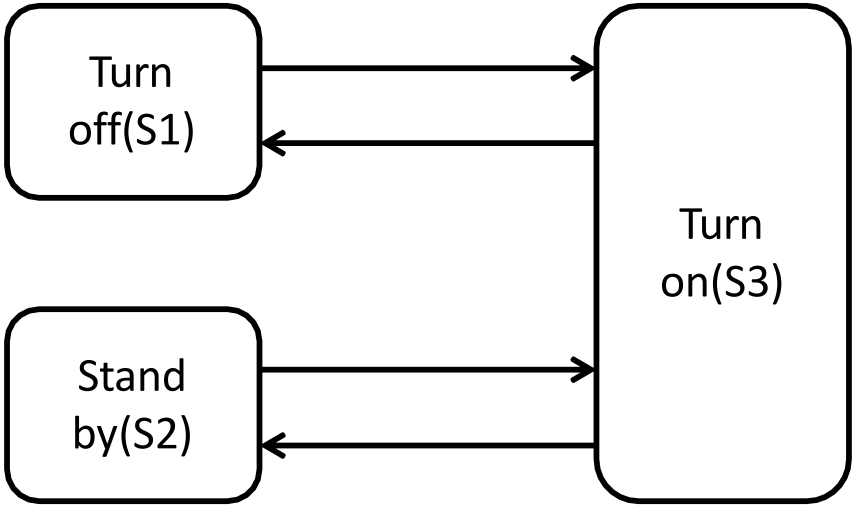
Decision Table

Let's say a vacation site offers a discount to customers under 25 or married. 10% discount for customers under the age of 25 and 20% discount for married customers. In this case, let's examine the holiday expense calculations with the decision table technique.

There are two conditions:

- ❑ Being under the age of 25
- ❑ Being married

State Transition



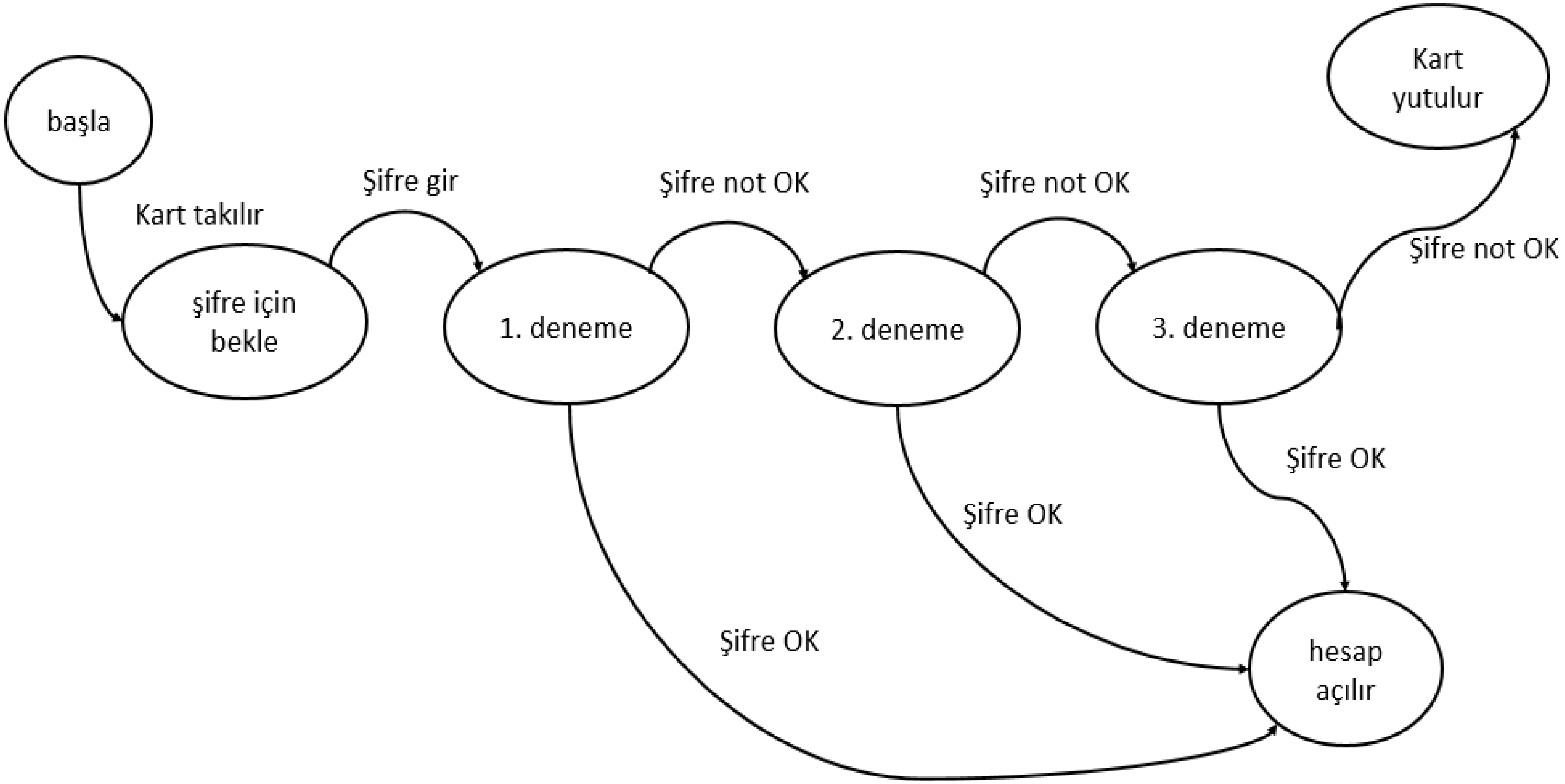
Zero switch

- ❑ $S_1 \to S_3$
- ❑ $S_3 \to S_1$
- ❑ $S_2 \to S_3$
- ❑ $S_3 \to S_2$

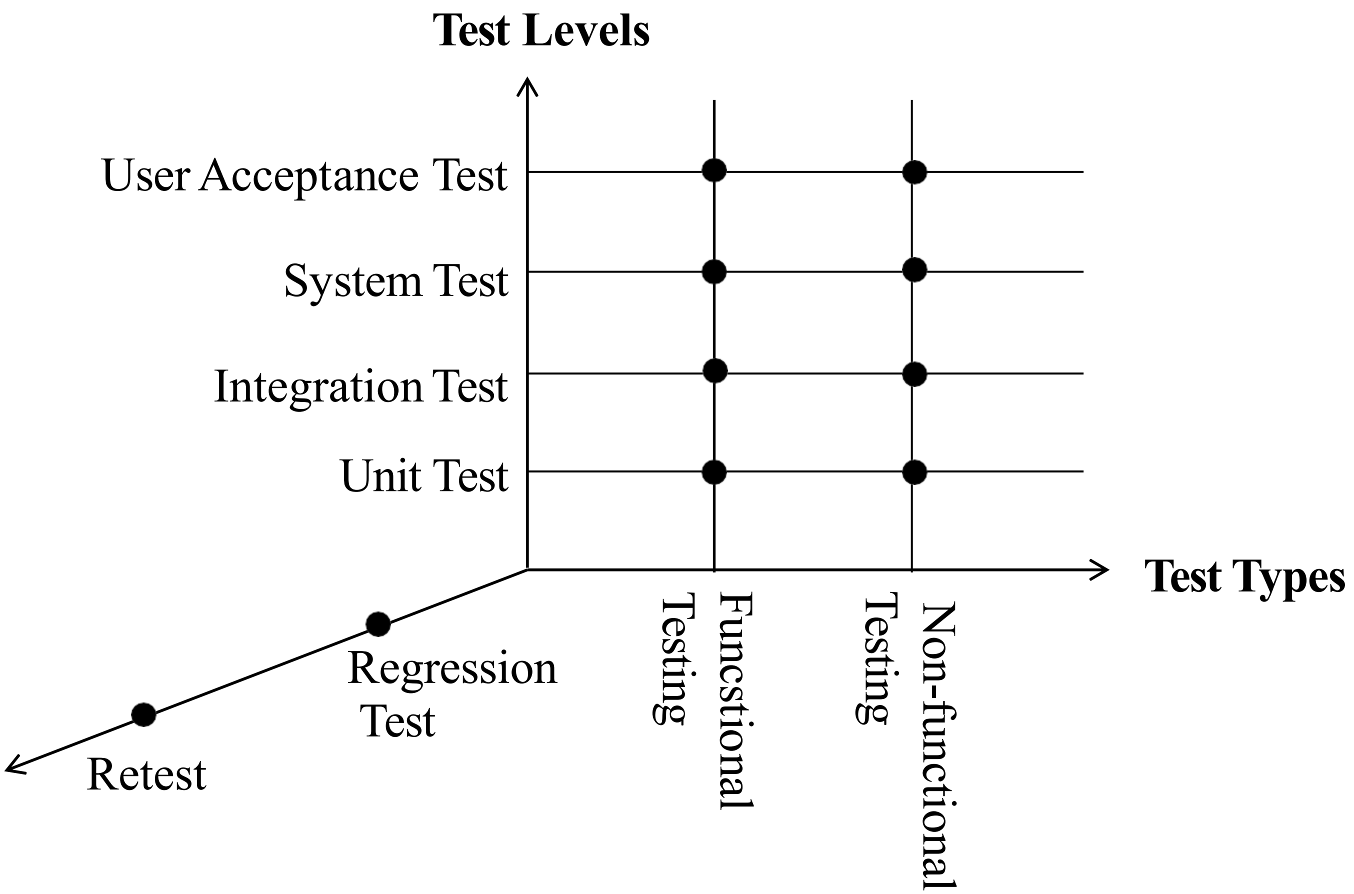
One switch

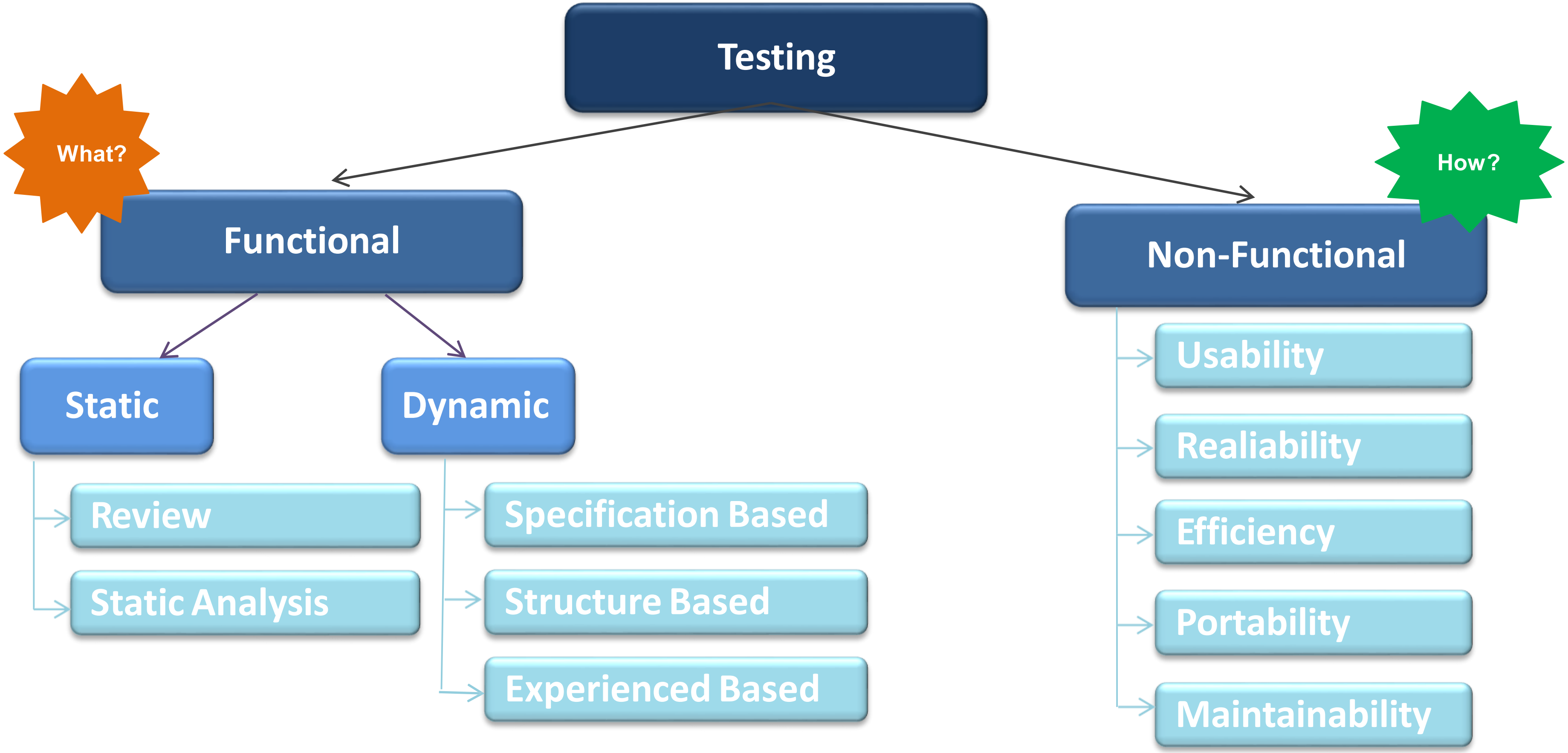
- ❑$S_1 \to S_3 \to S_1$
- ❑$S_1 \to S_3 \to S_2$
- ❑$S_2 \to S_3 \to S_2$
- ❑$S_2 \to S_3 \to S_1$
- ❑$S_3 \to S_1 \to S_3$
- ❑$S_3 \to S_2 \to S_3$

State Transition



Test Levels & Test Types





Testing

What?

Functional

Static

Review

Static Analysis

Dynamic

Specification Based

Structure Based

Experienced Based

How?

Non-Functional

Usability

Realiability

Efficiency

Portability

Maintainability

Performance Testing

It is to measure the performance of the system under a certain load and to ensure that it reaches the desired performance.

It aims to solve the bottlenecks of the system under heavy load with systems such as code and database.

«**Only conducting Performance Testing at the conclusion of system or functional testing is like ordering a diagnostic blood test after the patient is dead.**»

Scott Barber

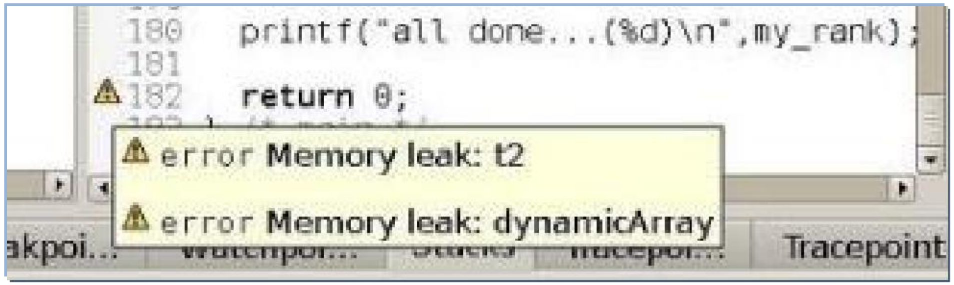
Performance Testing

Load Testing: It is a test type that gives information about how much load the system is working with maximum performance. It is done to understand how much load the system can tolerate.

Performance Testing

Stress Testing: It is uploading to the system periodically with the maximum number of users. It is to measure the system's response to such situations in a chaos environment and to determine the recovery level of the system when the malfunction is resolved.

Reliability Testing



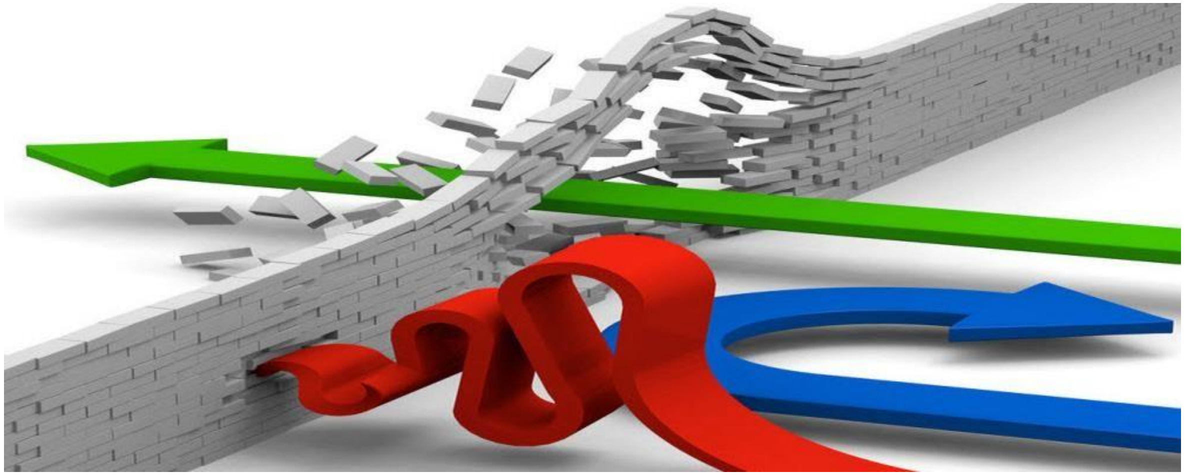
```
printf("all done...(%d)\n",my_rank);

return 0;
```

error Memory leak: t2

error Memory leak: dynamicArray

Static Code Analysis: Data flow anomalies and memory leaks are detected without running the written code. It is based on compliance with coding standards and quality metrics.

Reliability Testing



Penetration Testing: It is the process of detecting security vulnerabilities with cyber attacks and interventions using methods that predict malicious attacks, simulating attempts to infiltrate the system with these vulnerabilities and reporting all these transactions.

Usability Testing

These are tests that measure the learnability and suitability of the product for the end user.

Usability Testing

User tests consist of 5 components :

Learnability: Can users easily complete simple operations on the interface they use for the first time?

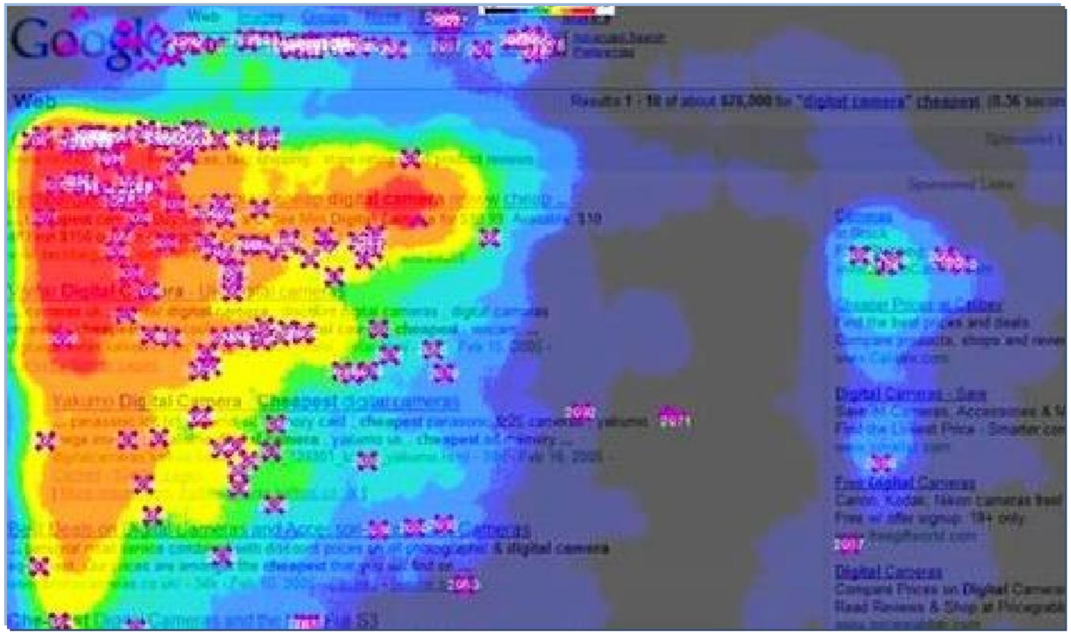
Efficiency: Can users complete assigned tasks quickly?

Memorability: Can users still use the software efficiently after the time spent without using the software after the first use?

Error Tolerance: How many mistakes does the user make, how serious are the mistakes, or can the user easily correct the mistakes?

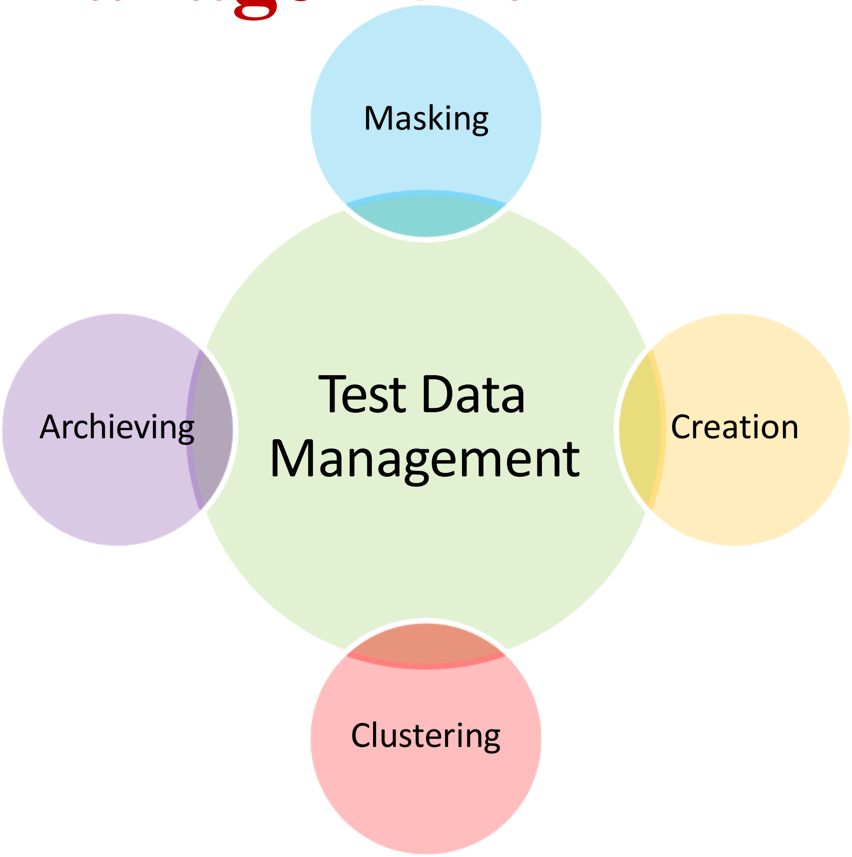
Satisfaction: Are users satisfied with the software?

Usability Testing



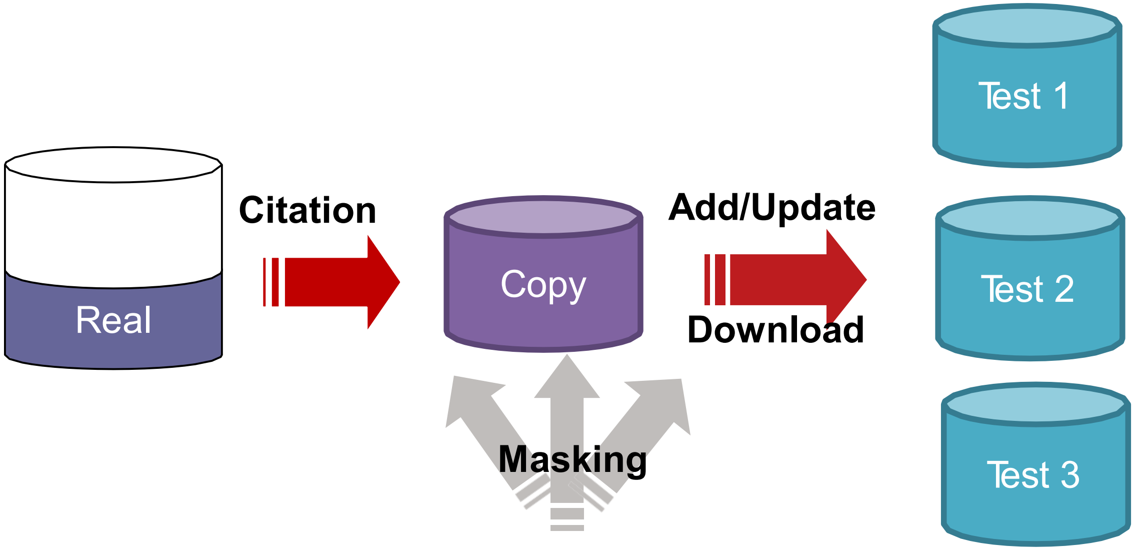
By using the eye tracking device, the regions where the users concentrate on the tested application can be determined with the heat mapping technique.

Test Data Management



It is the process of analyzing test data requirements, designing test data structures, creating and maintaining test data.

Test Data Management



With test data management, reliable and real data is masked and provided in a live environment, preventing errors caused by false data and saving time.

Test Automation

It is the comparison of predicted results with actual results during software testing and the use of specific software to control the running of tests.

Test Automation

Questions used when making test automation decision :

1.Period?

2. Maturity and Maintenance?

3. Infrastructure?

4. Modularity?

5. The Difference Between Manual and Automation?

Test Automation

Automated test types:

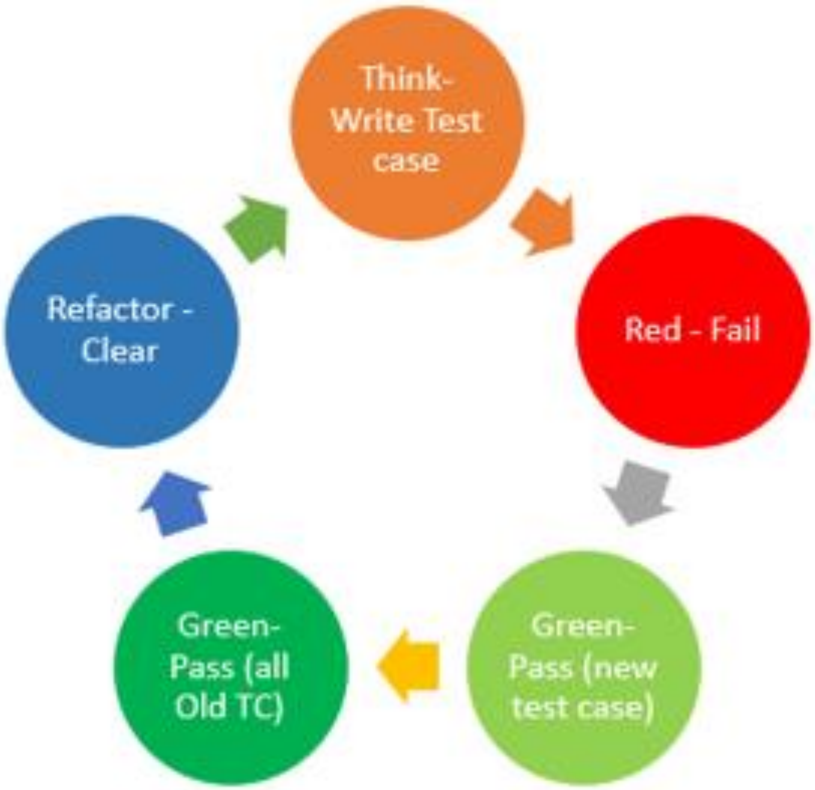
1. Unit tests

2. Repetitive tests

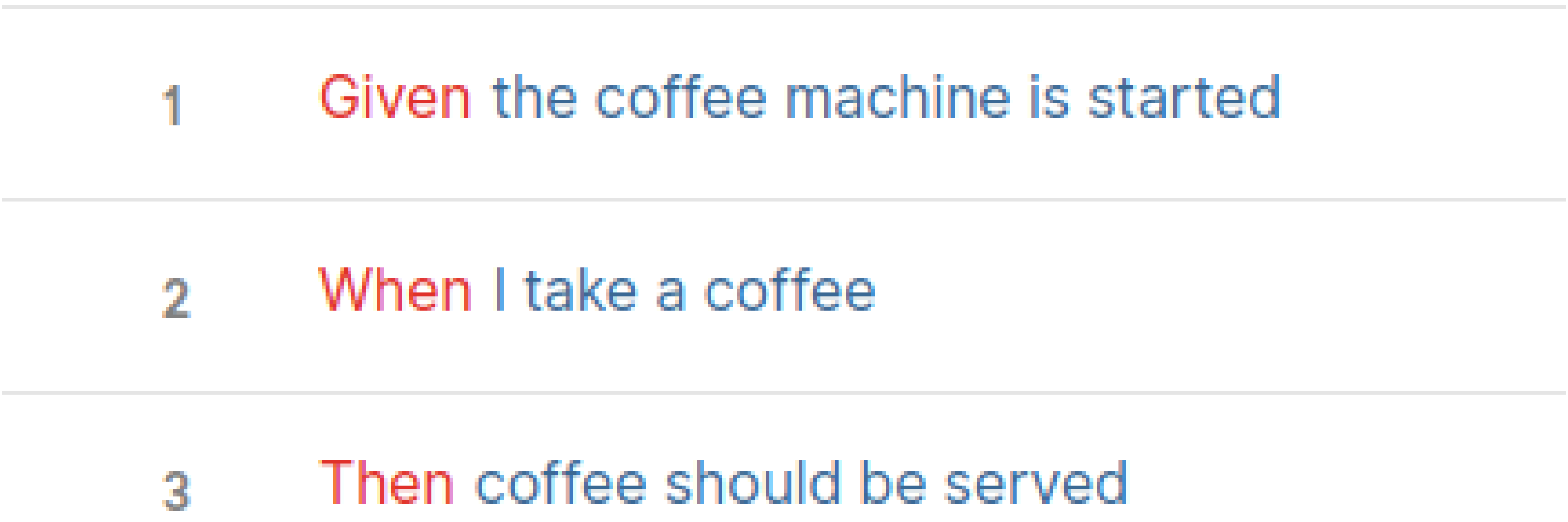
3. Regression Tests

4. Multi Platform Tests

Test Driven Development



Behaviour Driven Development



```
Given the coffee machine is started
When I take a coffee
Then coffee should be served
```

Thank You

kerem_hanci@icloud.com