

# ISTQB Chapter 1

## Yazılım Testi Nedir?

- Test süreci aynı zamanda test planlama, test analizi, test tasarımı, testin uyarlanması, test ilerlemesini ve sonuçlarını raporlama ve bir test nesnesinin kalitesini değerlendirme gibi faaliyetleri içerir
- Bazı testler, test edilen birimin veya sistemin çalıştırılmasını içerir; bu testlere dinamik testler denir. Diğer testler, test edilen birimin veya sistemin çalıştırılmasını gerektirmez; bu testlere ise statik testler denir.
- Test etme ve hata ayıklama farklı aktivitelerdir. Bazı durumlarda, **test uzmanları başlangıçtaki testlerden ve son onaylama testinden sorumluyken, yazılımcılar hata ayıklama ve ilgili birim testlerini gerçekleştirir.** Bununla birlikte, çevik yazılım geliştirmede ve diğer bazı yaşam döngülerinde test uzmanları hata ayıklama ve birim testlerine dâhil olabilirler.

## Yedi Test Prensipleri

1. Testin amacı, yazılımda hataların olduğunu göstermektir; yazılımda hata kalmadığını ispatlamak değildir.
2. Yazılımı %100 test etmek imkansızdır.
3. Erken test, zaman ve para tasarrufu sağlar.
4. Hatalar yazılımın belli alanlarında yoğunlaşır.
5. Antibiyotik direnci
6. Yazılım testi, projenin bağlamına, koşullarına göre değişiklik gösterir.
7. Yeni hata bulamıyoruz başarılı bir yazılım elde ettik yanılgısı.

1. Testing shows the presence of defects, not their absence
2. Exhaustive testing is impossible

3. Early testing saves time and money
4. Defects cluster together
5. Beware of the pesticide paradox
6. Testing is context dependent
7. Absence-of-errors is a fallacy

Bir test süreci aşağıdaki ana aktivite gruplarından oluşur:

□ Test planning □ Test monitoring and control □ Test analysis □ Test design □ Test implementation □ Test execution □ Test completion

## **Test planlama - Test Planning**

Defining the overall approach to testing //Genel yaklaşimin tanımlanması

Test planning is performed continuously in all life cycle processes and activities

Formulating the test schedule. //Takvim

“Determining the exit criteria” , DETERMINING the testing criteria

## **• Test gözetimi ve kontrolü - Test monitoring and control**

değerlendirme

## **• Test analizi - Test analysis**

Writing test conditions.

“During test analysis, the test basis is analyzed to identify testable features and define associated test conditions. Syllabus s.19”

Identification of the features which should be tested // Test edilmesi gereken özelliklerin belirlenmesi

## • Test tasarımı - Test Design

Developing test cases.

“Designing the test environment and identifying any required infrastructure and tools.  
(Test ortamının tasarlanması ve gerekli altyapı ve araçların belirlenmesi.) Syllabus s.20”

Test tasarımı sırasında test koşulları; üst seviye test senaryoları, üst seviye test senaryo grupları ve test yazılımları gibi ayrıntılar belirlenir. Bu nedenle, test analizi “ne test edilecek?” sorusuna cevap verirken, test tasarımı “nasıl test edilecek?” sorusuna cevap verir.

## Test uyarlama -Test implementation

Test implemantationda

1.Test prosedürleri yazılır.

2. Creating the test suite from the test cases for efficient test execution.(Etkin test yürütme için test senaryolarından test paketi oluşturma.)

3.*Preparing test data and **ensuring it is properly loaded in the test environment.***  
(**Test verilerinin hazırlanması ve test ortamında düzgün şekilde yüklenmesinin sağlanması.**)

Building the test environment(cevre) and Creation of a test suite

test tasarımı “nasıl test edilecek?” sorusuna cevap verirken, test uyarlama “şu anda testleri koşturmak için gerekli şeylere sahip miyiz?” sorusuna cevap arar.

Test tasarımı ve test uyarlama görevleri genellikle birlikte hayata geçirilir.

“Creating test suites from the test procedures and (if any) automated test scripts”

- “*In exploratory testing and other types of experience-based testing, test design and implementation may occur, and may be documented, as part of test execution.* Syllabus s.21” Experience-based (tecrübeye dayalı testlerde) ise test implementation ile test execution birlikte ele alınır.

## • Test koşumu - Test execution

Defect'lerin raporlanması test execution faaliyetleri sırasında yapılır.

*Test Execution* —> *Recording the IDs and versions of the test item(s) or test object, test tool(s), and testware , Logging the outcome of test execution*

Measuring and analysing results

Comparing expected und actual results

## • Test tamamlama -Test completion

Handover of Testware

## Sınav Notlari

- Debuggingde tester görev alabiliyor ama sadece identifying defects kısmında yer alabilir.
- Testing kalite değerlendirmede kullanılabilir.
- “To find failures and defects”, testing’in temel/ana hedefleri arasındadır.
- Determine whether enough component testing was executed. —> Bu testTeamın değil test leadin görevidir.
- Debugging is Identifying the cause of a defect, repairing the code and checking the fix is correct.
- TEst ve Debug arasındaki fark —>Testing shows failures caused by defects. Debugging finds, analyzes, and removes the causes of failures in the software.
- the MAIN benefit of designing tests early in the life cycle —> It helps prevent(engel olmak) defects from being introduced into the code.//Hataların koda girmesini önlemeye yardımcı olur.
- The effect of testing is to: Give an indication (gösterge) of the software quality;

- A bug or defect is: The result of an error or mistake;
- When is visible to end-users is a deviation from the specific or expected behavior, this is called..? a Failure
- Failure is Incorrect program behavior due to a fault in the program.
- Which of the following could be a reason for a failure?—>Software fault
- Exhaustive testing ise tüm olasılık ve ön koşulların düşünülerek test nesnesinin test edilmesidir.
- It is normally impossible to test all input / output combinations for a software system.
- It is not practically possible to test them all
- Testing can demonstrate that there are defects.
- When in the lifecycle should testing activities start?—> As early as possible
- “Defects cluster together” prensibine göre sistemdeki hatalar sistemin küçük bir bölümünde toplanır.
- Testing can show the presence of defects.
- Testing reduces the probability of uncovered defects.
- Testing can show that a previously present defect has been removed.
- Testing cannot prove that software is correct.
- Testers should begin to review documents as soon as drafts are available.//Test uzmanları, taslaklar hazır olur olmaz belgeleri incelemeye başlamalıdır.
- Defect clustering —> Hata kümeleme
- During Test **Planning** could faults be found **most cost effectively**
- Entry criteria; hangi şartlarda test sürecine başlanacağını ifade etmektedir. Örneğin gerekli veri tabanı veya test nesnelerinin hazır olmaması durumunda test faaliyetlerine başlanamaz. Exit criteria; hangi şartlarda test faaliyetlerinin durdurulacağını ifade etmektedir. Örneğin, planlanan testler uygulandıysa (executed), tanımlanan test kapsamına ulaşıldıysa, çözümlenmeyen defect'ler önceden belirlenen ve Kabul edilebilir limit içindeyse, kalan tahmini defect sayısı çok düşükse ve yazılımın fonksiyonel-fonksiyonel olmayan özellikleri tatmin ediciyse

test faaliyetlerine son verilebilir. Bu durumlarda exit criteria sağlanmış demektir. Exit kriteri Test Planning aşamasında kararlaştırılır.

- “ to Plan when to stop testing “ is the purpose of test completion criteria in a test plan: to Plan when to stop testing

- correct order

1. Define the objectives of testing.

2. Create bi-directional traceability between test basis, test conditions and test cases.

//Çift yönlü izlenebilirlik oluşturmak

3. Building the test environment.

4. Comparing actual results with expected results.

5. Check test logs against exit criteria.

- Specification of the test cases required to test a feature(Bir özelliği test etmek için gereken test senaryolarının belirtilmesi).

Test Design denilince aklımıza “How to test?” gelmeli.

Designing and prioritizing test cases and sets of test cases

- **Test Scenarios:** What are the areas to be tested? Sorusunun cevabıdır. Yani hangi alanlar test edilecektir sorusunun cevabı test scenarios’tur. ÖRNEK: User login, User can view/edit personal details.
- **Test Cases:** How to test scenarios? Sorusunun cevabıdır. Yani, test edilecek alanları nasıl test edeceğimizi test case’ler ile belirleriz. Test case’de ilgili senaryonun nasıl gerçekleştirileceği adım adım belirtilir. Test case’lerde test data, expected result ve preconditions gibi veriler de yer almalıdır.

ÖRNEK: 1) Launch browser, 2) Open URL, 3) Provide valid username and password, 4) Click on Login button.

- **Test Conditions:** Test scenarios ile hemen hemen aynı anlamda kullanılır. “What to test” sorusunun cevabıdır. “An item or event that could be verified by one or more test cases”
- Test condition, requirements, specifications veya user story’lerden çıkarılabilir.

- **Test Procedures:** Is a collection of test cases arranged sequentially in order to execute a test.
- **Test Scripts:** Commonly used to refer to a test procedure specification, especially an automated one.??.
- Sonuç olarak test case bir test senaryosu içindeki bir adımı test etmek için yazılır. Test case'de bu adımı test etmek için gerekli veriler (test data), ön gereklilikler (preconditions) ve o test adımlarını yapınca karşılaşmamız öngörülen beklenen sonuç (expected result) bulunmaktadır.
- Test prosedürleri ise bir testi uygulamak (veya run-koşturmak) için belirli bir sıraya göre dizilmiş test case'lerini ifade etmektedir.