

# i2i Academy

## Training Document

**Topic**

S  
o  
f  
t  
w  
a  
r  
e

Copyright of  i2i Systems Turkey 2016

The copyright in this work is vested in i2i Systems Turkey and the information contained herein is confidential. This work (either in whole or in part) must not be modified, reproduced, disclosed or disseminated to others or used for purposes other than that for which it is supplied, without the prior written permission of i2i Systems Turkey. If this work or any part hereof is furnished to a third party by virtue of a contract with that party, use of this work by such party shall be governed by the express contractual terms between the i2i Systems Turkey which is a party to that contract and the said party.

Document Name	Qualitative Assurance
	SQL – EX – 02

Document Difficulty Level			
Beginner	Junior	Senior	Expert
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



---

## Document History

Date	Author	Ver	Comments
<b>01.06.2016</b>	Kadriye Dönmez	1.0	Initial Draft
<b>01.06.2016</b>	Mennan Tekbir	1.1	Org.

## Software Quality Assurance

### Exercise SQA\_EX\_02:

**Definiton :** Please answer following questions:

1. Unit test ile integration test arasındaki farklar nelerdir?
2. Statik tekniklerden walkthroughs un ne olduğunu nasıl yapıldığını açıklayınız.
3. Defect priority ve defect severity nedir? Priority Urgent, severity low; priority low, severity urgent olan iki örnek veriniz.
4. İyi bir test uzmanı nasıl olmalıdır?
5. İyi bir test case nasıl olmalıdır?
6. Yapay zekanın gelişimi, test tekniklerine pozitif bir etki etmiş midir?
7. Bir fabrikanın kalite sorumlusu olarak çalışığınızı düşünün. Kalite kontrol bandına gelen bir tükenmez kalem nasıl test edersiniz?

**Objectives :** To understand software testing concepts and evaluate SQA Training

### Solution of SQA\_EX\_02:

1. Unit test ile integration test arasındaki farklar nelerdir?

Unit test, fonksiyon veya sınıf gibi küçük parçaların doğru çalışıp çalışmadığını kontrol eder. Genellikle dış sistemlerle bağlantı kullanılmaz. Örneğin, toplama yapan bir fonksiyonun sonucu doğru verip vermediğini test etmek unit testtir.

Integration test ise birden fazla bileşenin birlikte çalışmasını test eder. Örneğin, bir kullanıcının sipariş verip ödeme yapması süreci buna örnektir. Bu testler, sistemin genel işleyişini test eder ve daha geniş kapsamlıdır.

Sonuç olarak, unit test küçük parçaları, integration test ise bu parçaların birlikte çalışmasını kontrol eder. Unit test daha hızlı ve sık kullanılırken, integration test sistemin genelini değerlendirir.

2. Statik tekniklerden walkthroughs un ne olduğunu nasıl yapıldığını açıklayınız.

Statik test teknikleri yazılımın ürününe değil yazılımın ortaya çıkması için gerekli kod, gereksinim, doküman gibi çıktıların test edilmesiyle ilgilidir. Yani yazılım çalıştırılmaz uygun tekniklerle çıktıların üzerinden geçilir. Bu yüzden statik test teknikleri olarak anılır.

**Walkthrough** (inceleme yürütmesi), yazılım geliştiricisinin ya da belge sahibi kişinin, oluşturduğu doküman veya kodu diğer ekip üyelerine sunduğu ve birlikte inceledikleri yapısal bir toplantıdır. Walkthrough sırasında katılımcılar belgede anlam bütünlüğü, gereksinimlere uygunluk ve olası hatalar açısından görüş bildirir. Katılımcılar genellikle geliştirici, testçi, iş analisti ve proje yöneticisi gibi farklı rollerdendir.

Walkthrough yapılışı:

- Belge sahibi, katılımcıları belirler ve bir toplantı düzenler.
- Toplantıda belge ya da kod parça parça okunur ve herkes görüşlerini belirtir.
- Amaç hata bulmaktan çok, belgeyi herkesin anlamasını ve iyileştirme önerileri sunmasını sağlamaktır.
- Elde edilen notlar belge sahibi tarafından göz önünde bulundurularak düzenleme yapılır.

3. Defect priority ve defect severity nedir? Priority Urgent, severity low; priority low, severity urgent olan iki örnek veriniz.

Defect priority ve defect severity, yazılım testlerinde bulunan hataların (bug) sınıflandırılması için kullanılan iki önemli kavramdır.

**Defect Severity (Hata Şiddeti):** Bir hatanın teknik olarak yazılım üzerindeki etkisini, hatanın sistemin işleyişini bozup bozmadığını belirtir.

High severity: Sistem çöküyor veya ana işlev çalışmıyor.

Low severity: Görsel bir hata, yazım yanlışı gibi işlevsel olmayan sorunlar.

**Defect Priority (Hata Önceliği):** Bir hatanın ne kadar acil düzeltilmesi gerektiğini ifade eder. Genellikle proje yöneticisi ya da müşteri ihtiyacına göre belirlenir.

High priority: Hemen düzeltilmeli.

Low priority: Sonraya bırakılabilir.

Örnekler:

Priority: High , Severity: Low: Mobil uygulama ekranında “Şifremi unuttum” butonunun rengi arka planla aynı. (sistem işleyişi doğru ama kullanıcıyı etkilediği için acilen düzeltilmeli)

Priority: Low, Severity: High: Mobil uygulamada cihaz dili Türkçe ayarlandığında bazı ekranlarda içerikler kayıyor. (arayüzün bozulması ciddi bir sorun ama hata sadece bu dil seçeneğinde oluşuyor bu nedenle önceliği düşük)

4. İyi bir test uzmanı nasıl olmalıdır?

İyi bir test uzmanı, detaylara dikkat eden, analitik düşünebilen ve sabırlı biri olmalıdır. Hataları bulmak için sorgulayıcı yaklaşmalı, yazılımın işleyişini anlamalıdır. Ayrıca test araçları, otomasyon ve temel yazılım bilgilerine hâkim olmalıdır.

5. İyi bir test case nasıl olmalıdır?

İyi bir test case, açık ve net olmalıdır, belirli bir işlevi test etmelidir, test sonucunda alınması beklenen çıktı belirtilmelidir, tekrarlanabilir olmalı ve kullanıcı senaryolarına uygun hazırlanmalıdır.

6. Yapay zekanın gelişimi, test tekniklerine pozitif bir etki etmiş midir?

Evet, pozitif etki etmiştir. Yapay zekâ otomatik test oluşturma, hata tahmini ve test veri üretimi gibi alanlarda test süreçlerini daha hızlı ve verimli hâle getirmiştir. AI tabanlı araçlar, test senaryolarını optimize ederken insan hatalarını azaltır. Bu sayede yazılım testleri daha kısa sürede, daha yüksek doğrulukla gerçekleştirilebilmektedir.

7. Bir fabrikanın kalite sorumlusu olarak çalıştığınızı düşünün. Kalite kontrol bandına gelen bir tükenmez kalem nasıl test edersiniz?

Bir tükenmez kalem kontrol ederken öncelikle görsel kontrolleri yapardım. Mürekkep seviyesini inceler, kalemin üzerindeki yazıların düzgün ve silinmemiş olup olmadığını kontrol ederdim. Ardından işlevsellik kontrolüne geçerdim. Kalemin açılıp kapanma mekanizmasının sorunsuz çalışıp çalışmadığını, kesintisiz yazıp yazmadığını ve yazarken fazla mürekkep akıtıp akıtmadığını gözlemlerdim. Son olarak, dayanıklılık kontrolü yapardım. Kalem yere düştüğünde fiziksel bir hasar oluşup oluşmadığını kontrol eder, ardından kontrollerimi tamamlardım.