

ISTQB® SERTİFİKALI TEST UZMANI

TEMEL SEVİYE (CTFL) · SYLLABUS v4.0

ÜCRETSİZ ÖRNEK

Sınav Rehberi

Sıfırdan Sınava · Kapsamlı Türkçe Hazırlık

64 Öğrenme Hedefinin Tamamı

Her K3 Hedefi İçin Çözümlü Örnekler

32 Özgün Türkçe Şekil

Tam ve Gerekçeli Çözümlü Deneme Sınavı



Kadir Çamoğlu

ISTQB® ve CTFL®, International Software Testing Qualifications Board'un (ISTQB®) tescilli ticari markalarıdır. Bu kitap bağımsız, gayri resmi bir çalışma kaynağıdır; ISTQB® tarafından onaylanmamış, desteklenmemiş veya sponsorluğu üstlenilmemiştir.

Giriş — Bu Kitap Hakkında

Bu kitap nedir

Bu kitap, CTFL sınavına hazırlık sürecinde size rehberlik edecek kapsamlı bir kaynaktır; müfredatın (syllabus) kuru bir tekrarı değildir. Resmi müfredat, eğitim sağlayıcıları ve sınav soru yazarları için kaleme alınmış yoğun bir başvuru dokümanıdır. Size yalnızca neyin sınanabileceğini belirtir, ancak öğretme amacı taşımaz. Elinizdeki kitap ise tam olarak bu eksikliği gidermeye odaklanır. Müfredatın okuyucuya bıraktığı kavramsal boşlukları pratik örneklerle doldurur. Fikirler arasındaki yapısal bağlantıları da sunarak doğrudan konuyu öğretmeyi hedefler.

Kitabın her sayfasında iki temel ilke gözetilmiştir.

İlki, **resmi kapsama tam bağlılıktır**. Gereksiz ve ileri düzey ayrıntılarla vakit kaybetmezsiniz. Sınav; altı bölüme yayılmış **64 öğrenme hedefi (LO)** ve **anahtar kelimeler** üzerine kuruludur. Kitapta her bölüm bir öğrenme hedefiyle birebir eşleşir. Tüm anahtar kelimeler ise ilk geçtikleri yerde tanımlanır. Kısacası, v4.0 kapsamı dışındaki hiçbir içerik karşınıza çıkmaz.

İkincisi, **v4.0 terminolojisine sadakattir**. 2023 müfredatı; eski 2018 sürümüne kıyasla adları, listeleri ve vurguları değiştirdi. Bu kitap güncel terimleri birebir esas alır. Örneğin beşinci test prensibinin güncel resmi adı “testler yıpranır (tests wear out)”dır; eskiden “pestisit paradoksu (pesticide paradox)” olarak bilinirdi (bazı eğitimciler bunu “antibiyotik direnci”ne de benzetir). Böylece kitapta çalıştığınız terim, sınavda tam olarak karşılaştığınız terim olur. Ayrıca kitap boyunca kritik kavramların İngilizce orijinalleri ilk geçişlerinde ve özet tablolarında parantez içinde verilir (örneğin *komut kapsamı (statement coverage)*, *giriş kriteri (entry criteria)*); böylece Türkçe çalışırken İngilizce kaynaklarla da rahatça köprü kurabilirsiniz.

Bu kitap, tek başına **kendi kendine yeten bir hazırlık kaynağıdır**. Bir bölümü okuyup ardındaki soruları çözebilir, hata yaptığınızda nedenini başka bir dokümana bakmadan doğrudan öğrenebilirsiniz. Sınavda ölçülen tüm temel konular bu kitapta yapılandırılmış şekilde anlatılmıştır. Üstelik içerideki tüm sorular bu esere özel olarak yazılmıştır; resmi örnek sınavlar dahil hiçbir kaynaktaki soruları tekrar etmez.

Bu kitap kimin için

CTFL sınavının bir ön koşulu yoktur; bu kitabın da öyle. Bu çalışma, doğrudan şu kitleler için kaleme alınmıştır:

- **Yazılım testine yeni başlayanlar:** Temel kavramları en baştan öğrenmek isteyen mezunlar ve kariyerini bu alana taşımak isteyenler.
- **Geliştiriciler, iş analistleri ve proje yöneticileri:** Uluslararası geçerliliği olan bir sertifika hedefleyen ve sınav kapsamını net bir şekilde görmek isteyen profesyoneller.
- **Sektörde çalışan test uzmanları:** Pratik bilgisini teorik bir çerçeveye oturtmak ve terminolojisini v4.0 ile güncellemek isteyenler.
- **Sınava yeniden girecek adaylar:** Öğrenme hedefleri (LO) haritasını ve soru ağırlık tablolarını kullanarak spesifik eksiklerini hızla kapatmak isteyenler.

Bu kılavuz, yazılım testine sıfırdan başlayanlar düşünülerek, temel kavramları en yalın haliyle aktaracak şekilde tasarlanmıştır. Terimler ilk kullanımda açıklanır ve her kavram

gerçekçi bir işyeri örneğine bağlanır. Buna karşın, içerik yüzeysel anlatımlardan uzak tutulmuştur: Tüm tanım, kavram ve ayrımlar, doğrudan sınavda karşılaşıcağınız resmi standartlarla tam uyumludur.

Kitabın tamamı boyunca tek bir fikir öne çıkar ve bunu baştan söylemekte yarar var: CTFL, **pratik bir çekirdeği olan, kelime hazinesi ve ayrımlar sınavıdır**. Kırk sorunun otuz ikisi, resmi kavramları hatırlayıp *ayırt edebilme* yetinizi ölçer (K1 ve K2); kalan sekizi ise sekiz adlandırılmış tekniği küçük problemlere *uygulayabilme* yetinizi ölçer (K3). Bu sınavda yanlış şıkların çoğu, kıl payı ıskalayan seçeneklerdir. Başka bir deyişle, tipik yanlış şık rastgele değil, komşu kavramdır. Bu yüzden ayrımları bilinçli olarak öğrenmek, bu kitabın üzerine kurulduğu hazırlık stratejisidir.

Bu kitap nasıl düzenlenmiştir

Kitap; konuya giriş ve yönlendirmeye başlayan, ardından teorik öğretime, pratik uygulamaya ve son olarak başvuru kaynaklarına uzanan doğrusal bir yapıda ilerler:

- **Sınav Tanıtımı**. Sizi doğrudan sınava hazırlar. Formatı, bölüm ağırlıklarını, üç **K-seviyesinin** beklentilerini, soruların kurgulanma mantığını ve iki adımlı yanıtlama stratejisini bu bölümde bulabilirsiniz.
- **Bölüm 1-6**: Müfredattaki altı bölümü birebir yansıtır. Her bölüm; soru sayılarını, K-seviyesi dağılımlarını ve ilgili anahtar kelimeleri içeren bir **Bölüm Özeti** ile açılır. Ardından her bir **öğrenme hedefi (LO)** sade bir dille ve gerçek dünya örnekleriyle anlatılır. Her hedef için bir **Sınav Odağı** notu ve hemen ardından o konuya odaklı birkaç **Hemen Pekiştir** sorusu yer alır; bu soruların cevabı ve kısa gerekçesi hemen altındadır, böylece öğrendiğinizi anında sınarsınız. Her **K3 hedefi** içinse adım adım çözümlü bir örnek ile kısa bir alıştırmaya bulunur. Bölümler; bir sayfalık özet tablo, **Sık Karıştırılanlar** kutusu, **Sınav İpuçları ve Tuzakları** alanı ve o bölüme ait karışık **Alıştırma Soruları** ile sona erer.
- **Gerekçeli Yanıtlar**. Kitabın sonunda yer alan bu bölüm, alıştırmaların detaylı açıklamalarını içerir. Doğru seçeneğin gerekçesini ve en güçlü çeldiricinin neden yanlış olduğunu açıklarken, gerektiğinde ilgili konuya geri dönebilmeniz için sizi doğrudan doğru sayfaya yönlendirir.
- **Sözlük**, altı bölümün tüm resmi anahtar kelimelerini, derli toplu ve ISTQB Sözlüğüne sadık biçimde tanımlar.
- Kitabın sonundaki **Sınav Hazırlık Rehberi ve Yol Haritası** bölümü; haftalara bölünmüş iki farklı çalışma planı (2 haftalık ve 4 haftalık), sınavda en çok karıştırılan kavram çiftlerinin ana listesini ve zengin kaynakçayı içerir.
- **Deneme Sınavı 1**. Kitabın en sonunda, gerçek sınavın resmi yapısını birebir izleyen tam bir deneme sınavı yer alır: 40 soru, bölüm dağılımı 8-6-4-11-9-2 ve K1/K2/K3 bilişsel dengesi. Sınavı süreli bir şekilde tamamladıktan sonra; cevap anahtarı, başarı bantları ve her sorunun ayrıntılı gerekçeli açıklaması (en güçlü çeldirici + anahtar çıkarım) sayesinde kendi performansınızı değerlendirebilirsiniz. Bu deneme, kitabı tek başına yeterli bir hazırlık kaynağı hâline getiren son halkadır.

Bölüm numaraları üzerine bir not. Bu kitap konulara syllabus bölüm numarasıyla değil, *adıyla* atıfta bulunur (örneğin, “sınır değer analizi”, “gözden geçirme süreci”). Sınav hiçbir zaman bir bölüm numarasını hatırlamanızı istemez; kavramları sorar, dolayısıyla numaralandırmayı ezberlemeniz hiç gerekmez.

Nasıl okunmalı ve çalışılmalı

İlk okumayı mutlaka sırayla yapın: Bölümler tamamen birbirinin üzerine inşa edilmiştir. Örneğin; ilk bölüm sonraki kısımlarda sürekli kullanacağınız temel terminolojiyi (insan hatası, hata, arıza, test süreci, test çalışma ürünleri gibi) tanımlar. Benzer şekilde, ikinci bölümdeki test seviyeleri; dördüncü bölümdeki tekniklerin, beşinci bölümdeki test piramidinin ve çeyreklerinin temelini oluşturur.

“Oku-Çöz-Gözden Geçir” döngüsünü uygulayın: Bir konuyu okurken, Sınav Odağı’nın hemen ardındaki **Hemen Pekiştir** sorularını oracıkta çözün; bunlar tüyoyu tazeyken pekiştirmek içindir ve cevapları hemen altındadır. Bölümü bitirdiğinizde ise, cevap anahtarına bakmadan önce bölüm sonundaki karışık **Alıştırma Sorularını** çözün. Ardından, özellikle şans eseri doğru yaptığınız sorular dahil, her sorunun gerekçeli açıklamasını mutlaka okuyun. Bu açıklamalar sadece doğru şıkkı göstermek için değil, sorunun temelindeki o kritik nüansı öğretmek için yazılmıştır.

K3 hedeflerine iki kat ağırlık verin: Sınavdaki 40 sorunun 8’i, öğrendiğiniz teorik bilgiyi pratik bir probleme uygulamanızı (K3 seviyesi) bekler. Birçok adayın en çok zorlandığı yer tam olarak bu beşte birlik kısımdır. Sınavda sizden denklik payları, sınır değer analizi, karar tabloları, durum geçiş testi veya ATDD ile test senaryosu türetmeniz; test eforu hesaplamanız ya da bir hata raporu hazırlamanız istenebilir. Bunların her biri için kitapta adım adım çözümlü örnekler bulacaksınız; bu kısa alıştırma mutlakla kalem kağıt kullanarak, elle yapın.

Ekleri dinamik bir referans olarak kullanın: Bölüm okumalarınız sırasında takıldığınız ya da kararsız kaldığınız anlarda sözlüğe ve kavram çiftlerine dönmekten çekinmeyin. Bununla birlikte, sık karıştırılan kavramlar listesi ve sözlük, hazırlığınızın son haftasındaki o son pekiştirme süreci için de harika birer tekrar aracına dönüşecektir.

Sınav Tanıtımı — CTFL Sınavı Nasıl İşler

Formata bir bakış

Parametre	Değer
Sorular	40 çoktan seçmeli soru, her biri 1 puan
Şıklar	4 şık, tam olarak biri doğru ("Yalnızca BİR seçenek işaretleyin.")
Süre	60 dakika; sınav ana dilinizde değilse +%25 (75 dakika)
Geçme notu	40 puanın 26'sı (%65)
Uygulama	ISTQB Üye Kurulları ve akredite sınav sağlayıcıları aracılığıyla; kurulunuza göre kâğıt veya elektronik, yüz yüze veya çevrim içi gözetimli

Sınavda yanlış yanıtlar puan kaybettirmez. Şansınızı artırmak için emin olmasanız bile **tüm soruları mutlaka işaretleyin.**

Sorular nereden gelir

Sınav yapısı, resmi ve sabit bir genel şablona dayanır. Soru kombinasyonlarında ufak tefek öğrenme hedefi (LO) değişiklikleri olabilse de, her sınavda ana konu başlıklarının soru dağılımı ve K-seviyelerinin genel ağırlığı standart kalır:

Bölüm	Konu	Soru
1	Testin Temelleri	8
2	Yaşam Döngüsü Boyunca Test	6
3	Statik Test	4
4	Test Analizi ve Tasarımı	11
5	Test Faaliyetlerinin Yönetimi	9
6	Test Araçları	2

K-seviyesi	Ne ister	Soru	Süre kılavuzu
K1 — Hatırlama	Resmi bir terimi ya da liste öğesini tanımlamak, hatırlamak, tanımak	8	her biri ~1 dakika
K2 — Anlama	İki kavramı karşılaştırmak, sınıflandırmak, açıklamak, örneklendirmek, ayırt etmek	24	her biri ~1 dakika
K3 — Uygulama	Bir prosedürü yürütmek: test senaryosu türetmek, hesaplamak, önceliklendirmek, rapor hazırlamak	8	her biri ~3 dakika

Bu tablodan şimdiden çıkarabileceğimiz önemli noktalar var:

- **Sınavın kalbi 4. ve 5. bölümlerde atar:** Toplam 40 sorunun 20'si bu iki bölümden gelir ve sınavdaki 8 pratik (K3) sorusunun tamamı buradadır. Dolayısıyla en çok çalışma zamanını bu iki alan hak eder; elinizdeki kitap da en kapsamlı ve detaylı anlatımı bu bölümlere ayırmıştır.
- **Zaman yönetimi dengeli değildir:** 60 dakikada 40 soru çözmek ilk bakışta baskı yaratabilir. Ancak teorik (K1 ve K2) seviyesindeki 32 sorunun her biri, yaklaşık birer dakikada yanıtlanacak şekilde tasarlanmıştır. Bu soruları hızla geçip kazandığınız süreyi hesaplama gerektiren 8 pratik (K3) soruya saklamalısınız.
- **Adayları genellikle şaşırtan bir diğer kural da şudur:** Müfredat bölümlerinin başındaki anahtar kelimeler, öğrenme hedefleri içinde doğrudan geçmese bile sınavda soru olarak karşınıza çıkabilir. Müfredat sizden bu kelimeleri sadece ezberlemenizi (K1) değil, tanımlarını anlamanızı ve benzer terimlerden ayırt edebilmenizi (K2) de bekler. Pratikte, bir kelimeyi sadece duymuş olmak yetmez; onu en yakın çeldiricisinden ayırmanız gerekir. İşte bu yüzden kitapta her anahtar kelime ilk geçtiği yerde net bir şekilde tanımlanmış ve Sözlük kısmında eksiksiz olarak bir araya getirilmiştir.

CTFL soruları nasıl kurulur

Sınavdaki soruların büyük bir kısmı şu üç temel formattan birine sahiptir:

- **Teorik Hatırlama (Genellikle K1):** Doğrudan müfredattaki listeleri sorgular. Örneğin; "Aşağıdakilerden hangisi tipik bir test hedefidir?" sorusunda, seçeneklerin biri doğrudan resmi listeden alınırken, kalan üçü kapsam dışıdır.
- **Senaryo Analizi (K2'lerin Çoğu):** Soru kökünde birkaç cümlelik bir iş simülasyonu verilir ve sizden net bir ayırım yapmanız istenir: Bu bir doğrulama testi mi, yoksa regresyon testi mi? Proje riski mi, ürün riski mi? Giriş kriteri mi, çıkış kriteri mi?
- **Pratik Uygulama ve Hesaplama (Tüm K3'ler):** Soru; size küçük bir spesifikasyon, tablo veya diyagram vererek bir görev yükler: %100 kapsam için kaç test senaryosu gerekir? Üç nokta kestiriminin sonucu nedir? Hangi test senaryosu önceliklidir?

Bu sınavda yanlış şıklar hiçbir zaman rastgele seçilmez. CTFL çeldiricileri genellikle **doğruyu teğet geçen** seçeneklerden oluşur. Yanıt kurgulanırken; hedef kavramın kardeşi (onaylama yerine regresyon), farklı bir adıma ait doğru bir liste, mantıklı görünen ama bir adım kaydırılmış bir sınır değeri ya da v4.0 ile adı değişmiş eski bir 2018 terimi çeldirici olarak karşınıza çıkar.

İki adımlı yanıtlama stratejisi

Bu kitaptaki çözümlü örneklerin tamamı aynı iki adımlı yöntemi temel alır. Sınavda yüksek başarı için bu yaklaşımı bir alışkanlık haline getirin:

1. **Belirgin şekilde yanlış olan seçenekleri eleyin:** Genellikle iki şıkkı hızla eleyebilirsiniz. Çünkü bu seçenekler ya tamamen başka bir bölümün kavramına aittir ya temel bir tanımla çelişir ya da senaryoda verilen mantığa uymaz.
2. **Kalan iki şıkkı tek bir kritere indirgeyerek karşılaştırın:** Kendinize şu soruyu sorun: "Bu iki seçenek arasındaki temel fark nedir?" Ardından soru köküne geri dönün ve doğruyu belirleyecek o anahtar kelimeyi arayın. Örneğin; "düzeltmeden sonra"

ifadesi doğrulama testine, “değişikliğin sistem genelinde yan etki yaratmadığı” vurgusu regresyon testine, “başlamadan önce” şartı ise giriş kriterine işaret eder. Soru kökü, o ipucunu neredeyse her zaman içinde barındırır.

Pratik uygulama (K3) sorularında ise 2. adımı şu stratejiyle değiştirin: **Çözümü önce kağıt üzerinde kendiniz yapın, seçeneklere daha sonra bakın.** Doğru yanıt şıklardan bağımsız olarak önceden türetmek, her seçeneği tek tek senaryoda deneyerek vakit kaybetmekten çok daha güvenli bir yöntemdir.

Bölüm 1 — Testin Temelleri

Bölüm Özeti

Sınavdaki soru sayısı	40 üzerinden 8 (K1 = 2, K2 = 6)
Öğrenme hedefi	14
İşin özü	Testin tanımı ve kapsamı, testin neden gerekli olduğu, yedi temel prensip, yedi test aktivitesi ve çıktıları; ayrıca insan boyutu (roller, beceriler, bütün ekip yaklaşımı ve bağımsızlık).

Bu bölümde tanıtılan anahtar kelimeler: kapsam (coverage) · hata ayıklama (debugging) · hata (defect) · insan hatası (error) · arıza (failure) · kalite (quality) · kalite güvence (quality assurance) · kök neden (root cause) · test analizi (test analysis) · test esası (test basis) · test senaryosu (test case) · test tamamlama (test completion) · test koşulu (test condition) · test kontrol (test control) · test verisi (test data) · test tasarımı (test design) · test yürütme (test execution) · test uyarılma (test implementation) · test gözetimi (test monitoring) · test nesnesi (test object) · test hedefi (test objective) · test planlama (test planning) · test prosedürü (test procedure) · test süreci (test process) · test sonucu (test result) · test (testing) · test çalışma ürünleri (testware) · izlenebilirlik (traceability) · sağlama (validation) · doğrulama (verification)

Bu bölüm, kitabın ana omurgasını oluşturur. Sonraki tüm bölümler burada kuracağımız terminoloji üzerine inşa edilecektir. Sınavda bu bölümden tam 8 soru sorumluluğunuz bulunuyor. Sorular genellikle tanımlara, resmi listelere (yedi prensip ve yedi aktivite) ve birbirine yakın kavramlar arasındaki nüanslara odaklanır; örneğin insan hatası, hata ve arıza arasındaki farklar sıkça sorgulanır.

Nasıl Çalışmalı? Bu bölümde herhangi bir hesaplama veya pratik uygulama (K3) hedefi yer almaz. Dolayısıyla buradaki amacınız problem çözmek değil, güçlü bir terminoloji altyapısı kurmak ve resmi listelerin ardındaki mantığı kavramaktır. Terminoloji zihninizde netleştğinde, bu bölümün sorularını çok daha rahat ve hızlı çözebildiğinizi fark edeceksiniz.

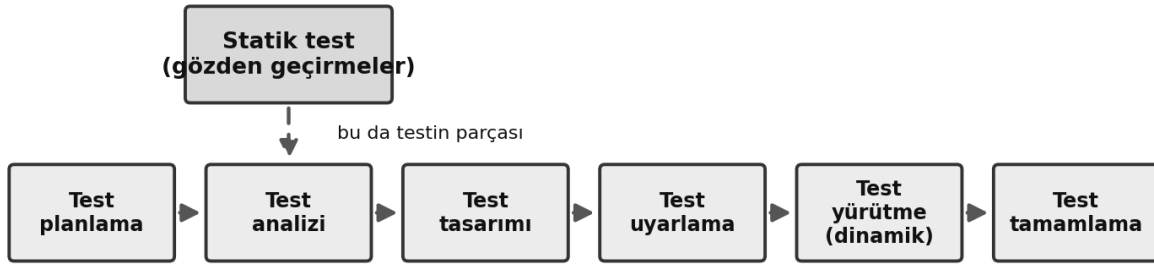
1.1 Test Nedir?

Test, Sadece Yazılımı Çalıştırmaktan İbaret Değildir

Genelde “test” denildiğinde akla ilk gelen şey yazılımı çalıştırmak ve sonuçları izlemektir. Oysa gerçek anlamda test, çok daha geniş bir faaliyetler bütünü kapsar. Süreci planlamak, neyin test edileceğini analiz etmek, testleri tasarlamak, hazırlamak (uyarlamak), sonuçları raporlamak ve test nesnesinin kalitesini değerlendirmek bu bütünün birer parçasıdır.

Kısacası test süreci, henüz ortada tek bir satır kod bile yokken başlar ve yazılım yürütme aşaması tamamlandıktan sonra da devam eder. Yazılımı çalıştırarak yürüttüğünüz sınav süreci, **dinamik test (dynamic testing)** olarak adlandırılır. Yazılımı çalıştırmadan, örneğin bir gereksinim belgesini gözden geçirerek yapılan çalışmalar ise **statik test (static testing)** kapsamındadır (Statik test konusunu Bölüm 3’te detaylıca ele alacağız).

Şekil 1.1, bu faaliyetlerin birbiriyle nasıl ilişkilendiğini ve statik testin sürece nerede dahil olduğunu göstermektedir.



Şekil 1.1 — Test, yürütmeden ibaret değildir. Aktiviteler sırayla ilerler ve statik test (gözden geçirmeler) de testin bir parçasıdır.

Şekilde	Anlamı
Soldan sağa zincir	Test aktiviteleri olağan sıralarıyla: planlama, analiz, tasarım, uyarlama, yürütme, tamamlama
“Test yürütme (dinamik)”	Yürütme yazılımı çalıştırır; bu yüzden <i>dinamik test</i> denen kısımdır
Test analizine kesik bir okla bağlanan “Statik test (gözden geçirmeler)”	Bir belgeyi gözden geçirmek gibi statik test de testtir; hiçbir yazılım çalıştırılmasa bile

Çalışmanın en başında netleştirilmesi gereken, kitabın ilerleyen kısımlarında da sıkça referans vereceğimiz çok önemli bir terim çifti bulunmaktadır:

Doğrulama (Verification): “Ürünü doğru mu geliştiriyoruz?” sorusuna yanıt arar. Yani yazılımın, kendisi için belirlenen teknik gereksinimleri karşılayıp karşılamadığını denetler.

Sağlama (Validation): “Doğru ürünü mü geliştiriyoruz?” sorusunun peşindedir. Yani yazılımın, kullanıcıların ve paydaşların gerçek iş ihtiyaçlarını tam olarak karşılayıp karşılamadığına bakar.

Test süreci bu iki kavramı da kapsar. Bir ürün, doğrulama aşamalarını eksiksiz geçmesine rağmen sağlama aşamasında başarısız olabilir. Örneğin bir bordro yazılımı, kendisine verilen tüm yazılı formülleri kurallarına uygun biçimde işleyebilir. Ancak bu formüller güncel mevzuatla örtüşmüyorsa, tüm testler teknik olarak “başarılı” görünse bile sistem işlevsel olarak hatalı sonuçlar üretecektir.

Tipik test hedefleri (LO 1.1.1 — K1)

Müfredat, test süreçlerinin arkasındaki dokuz temel nedeni sıralar. Sınavda karşınıza çıktığında bu maddelerin her birini geçerli birer test hedefi olarak tanıyabilmeniz gerekir.:

1. Gereksinimler, kullanıcı hikayeleri, tasarımlar ve kod gibi iş ürünlerini değerlendirmek,
2. Arızaları tetiklemek ve hataları ortaya çıkarmak,
3. Test nesnesinin hedeflenen kapsamda test edilmesini sağlamak,
4. Düşük yazılım kalitesinden kaynaklanan risk seviyesini azaltmak,
5. Belirlenen gereksinimlerin karşılandığını doğrulamak,
6. Test nesnesinin sözleşmesel, yasal ve düzenleyici (mevzuata ilişkin) gereksinimlere uygunluğunu doğrulamak,

7. Paydaşların bilinçli kararlar alabilmesi için onlara net bilgi sağlamak,
8. Test nesnesinin kalitesine yönelik güven oluşturmak,
9. Test nesnesinin eksiksiz olduğunu ve paydaş beklentilerine uygun çalıştığını doğrulamak (sağlamak).

Bu listede daha önce değindiğimiz iki kritik kavrama dikkat edin: 5. ve 6. maddeler doğrudan **doğrulama (verification)** hedefleriyken, 9. madde net bir **sağlama (validation)** hedefidir.

Dokuz maddelik bu uzun listeyi zihninizde daha kolay yapılandırmak için hedefleri dört temel grupta toplayabiliriz:

Hedef Grubu	İlgili Maddeler
Değerlendirme ve Keşif	İş ürünlerini değerlendirmek (1) · Arızaları tetikleyip hataları ortaya çıkarmak (2)
Gereksinim ve Uyumluluk Denetimi	Belirlenen gereksinimleri doğrulamak (5) · Sözleşmesel, yasal ve mevzuat uyumluluğunu doğrulamak (6) · Paydaş beklentilerini karşılamak (9)
Kapsam ve Risk Yönetimi	Hedeflenen kapsama ulaşılmasını sağlamak (3) · Risk seviyesini azaltmak (4)
Bilgilendirme ve Güven Tesis Etme	Karar süreçleri için bilgi sağlamak (7) · Kaliteye yönelik güven oluşturmak (8)

Pratik Hayatta: Bir canlıya çıkış (sürüm) kararından hemen önce, ürün sahibi (Product Owner) test ekibine şu kritik soruyu sorar: “Üründen ne kadar eminiz?” Ekip bu soruya; mevcut test kapsamı oranlarıyla ve kalan risk analizleriyle yanıt verir. İşte o anda, gerçekleştirilen test faaliyetleriyle hiçbir yeni hata bulunmamış olsa bile, test süreci 7. ve 8. hedefleri (bilgi sağlama ve güven oluşturma) başarıyla yerine getiriyor demektir. Yani testin değeri, sadece hata bulmaya bağlı değildir.

Sınav Odağı: Sınavda sıkça karşılaşılan soru kalıplarından biri, birbirine çok benzer şıklar arasından “gerçek bir test hedefini” ayırt etmenizi hedefler. Karşınıza çıkacak yanıltıcı seçenekler genellikle test dışındaki farklı disiplinlerin faaliyetleridir. Örneğin; hataları onarmak hata ayıklamanın (bir geliştirme aktivitesi), süreçleri iyileştirerek hataları önlemek kalite güvencenin (QA) işidir. Yazılımda “hiç hata kalmadığını kanıtlamak” ise (1. prensipte göreceğimiz üzere) zaten mantıken imkansızdır. Bir seçeneği işaretlemeyen önce, o şıkkın bir geliştirme veya QA aktivitesi olmadığından emin olun.

Hemen Pekiştir

HP-C1-1.1.1-a. Aşağıdaki test hedeflerinden hangisi bir **sağlama (validation)** hedefidir?

- a. Belirlenen gereksinimlerin karşılandığını doğrulamak
- b. Test nesnesinin sözleşmesel ve yasal uygunluğunu doğrulamak
- c. Test nesnesinin, paydaşların beklediği gibi eksiksiz çalıştığını doğrulamak (sağlamak)
- d. Test nesnesinin gereken kapsamda test edilmesini sağlamak

► **Yanıt: c.** “Doğru ürünü mü inşa ediyoruz?” sorusu sağlamadır; (a) ve (b) gereksinimlere uyumu denetleyen doğrulama hedefleri, (d) ise kapsam hedefidir.

HP-C1-1.1.1-b. Aşağıdakilerden hangisi meşru bir test hedefi **değildir**?

- a. Test nesnesinin kalitesine güven oluşturmak
- b. Paydaşlara karar verebilmeleri için bilgi sağlamak

- c. Hataları üreten geliştirme sürecini iyileştirmek
- d. Yetersiz yazılım kalitesinin risk seviyesini azaltmak

► **Yanıt: c.** Süreci iyileştirerek hataları önlemek kalite güvencenin işidir; testin bir hedefi değildir. Diğer üçü resmi listede yer alır.

HP-C1-1.1.1-c. “Belirtilen gereksinimlerin karşılandığını doğrulamak” hangi tür test hedefidir?

- a. Doğrulama (verification)
- b. Sağlama (validation)
- c. Kapsam hedefi
- d. Risk azaltma hedefi

► **Yanıt: a.** Belirtilen (yazılı) gereksinimlere uyumu denetlemek doğrulamadır; paydaş ihtiyacına uyum ise sağlama olurdu.

Test ve hata ayıklama (LO 1.1.2 – K2)

Test ve hata ayıklama, aralarındaki iş bölümü son derece net olan iki farklı faaliyettir.

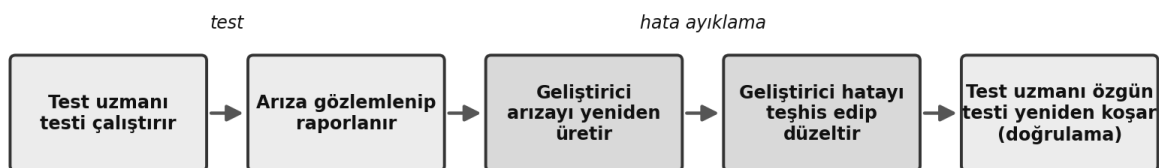
Test, yazılımı çalıştırarak arızaları tetikler (dinamik test) veya bir iş ürünü üzerindeki hataları doğrudan ortaya çıkarır (statik test). **Hata ayıklama (debugging)** ise tamamen bir yazılım geliştirme faaliyetidir; amacı, ortaya çıkan bir arızanın kaynağını bulmak, analiz etmek ve ortadan kaldırmaktır.

Karşılaştırma Alanı	Test	Hata Ayıklama
Ana Amaç	Arızaların oluştuğunu göstermek veya hataları doğrudan bulmak	Arızanın nedenini bulmak, analiz etmek ve gidermek
Sorumlu Rol	Test Uzmanları (ve bütün ekip)	Yazılım Geliştiriciler
Tipik İş Akışı	Testleri yürütür, arızayı gözlemler ve raporlar	Arızayı yeniden üretir, hatayı teşhis eder ve düzeltir
Sonraki Adım	—	Doğrulama testi: Düzeltmenin işe yarayıp yaramadığını görmek için özgün test yeniden koşular

Dinamik test sırasında bir arıza tespit edildiğinde, hata ayıklama süreci genellikle üç adımdan oluşur: 1. arızayı yeniden üretmek, 2. kök nedeni teşhis etmek 3. ve hatayı gidermek.

Statik testlerde ise süreç çok daha basittir. Ortada çalışan bir yazılım olmadığı için yeniden üretilecek bir arıza da yoktur; hata zaten belgede veya kodun içinde doğrudan görünür durumdadır. Bu yüzden buradaki hata ayıklama, doğrudan ilgili kusuru düzeltmekten ibarettir.

Şekil 1.2, dinamik test sırasında gözlemlenen bir arızanın ardından işleyen standart döngüyü göstermektedir.



Şekil 1.2 — Test, hata ayıklama ve doğrulama testi. Test arızayı görünür kılar, hata ayıklama nedenini giderir, doğrulama testi düzeltmeyi denetler.

Zincirdeki adım	Hangi aktivite
Test uzmanı bir testi çalıştırır, bir arıza gözlemlenip raporlanır	Test
Geliştirici arızayı yeniden üretir, sonra hatayı teşhis edip düzeltir	Hata ayıklama
Test uzmanı özgün testi yeniden çalıştırır	Doğrulama testi

Pratik Hayatta: Akşam eve geldiğinizde banyo tavanından yere su damladığını fark ettiniz. Bu sızıntıyı çıplak gözle görmek ve durumu apartman yönetimine bildirmek **testtir**; görevi sadece sorunu görünür kılmak ve mevcut durumu belgelemektir.

Üst kat komşunuzun tesisatçıyı çağırması, tesisatçının tavanı inceleyip “*Bu su nereden geliyor? Küvet gideri mi tıkanı, çamaşır makinesinin borusu mu patladı, yoksa duvarın içindeki ana temiz su borusu mu sızdırıyor?*” diye kaçak testi yapması, sorunun patlak bir borudan kaynaklandığını **teşhis etmesi** ve o boruyu yenisiyle değiştirmesi **hata ayıklamadır**.

Tamirat bittikten sonra üst kattaki tüm musluklar açılarak boruya kasıtlı olarak tazyikli su verilmesi ve sizin banyonuzun tavanından hâlâ su damlayıp damlamadığının kontrol edilmesi ise **doğrulama testidir**.

Sınav Odağı: Senaryo soruları genellikle bir olay dizisi verir ve sizden bu faaliyetleri sırasıyla adlandırmanızı ister. Standart doğru akış **test** → **hata ayıklama** → **doğrulama testi** şeklindedir. Sınavdaki en popüler yanıltıcı şık, doğrulama testi ile regresyon testinin yerini değiştirmektir. Bu iki kavramın farkına Bölüm 2’de detaylıca değineceğiz; ancak şimdilik şu ipucunu aklınızda tutun: Doğrulama testi sadece *başarısız olan özgün testi* yeniden çalıştırır; regresyon testi ise yapılan değişikliğin *sistemin başka bir yerini* bozup bozmadığını kontrol eder.

Hemen Pekiştir

HP-C1-1.1.2-a. Bir geliştirici, raporlanan bir arızayı kendi ortamında yeniden üretir, koddaki kusuru teşhis eder ve düzeltir. Bu faaliyet aşağıdakilerden hangisidir?

- Test
- Statik test
- Hata ayıklama
- Doğrulama testi

► **Yanıt: c.** Bir arızanın nedenini bulup gidermek hata ayıklamadır (geliştirme faaliyeti). Test yalnızca arızayı görünür kılar; düzeltmeyi denetlemek ise doğrulama testidir.

HP-C1-1.1.2-b. Bir gereksinim belgesindeki veya kaynak koddaki hata, statik test (gözden geçirme) sırasında tespit edilmiştir. Bu hatanın 'hata ayıklama' (debugging) süreci için aşağıdakilerden hangisi doğrudur?

- Dinamik teste kıyasla daha kolaydır; çünkü henüz bir arıza (failure) tetiklenmemiştir, kusur doğrudan belgede/kodda görünür ve giderilir.
- Dinamik testle bulunan bir hataya kıyasla daha zordur; çünkü kod çalıştırılmadığı için hatanın kök nedeni (root cause) izole edilemez.
- Dinamik test süreçleriyle tamamen aynı zorluktadır; çünkü her iki süreçte de hatanın düzeltilmesi için önce doğrulama testi (confirmation testing) yapılması şarttır.

d. Statik testlerde kod koşturulmadığı için hata ayıklama ve doğrulama faaliyetlerine hiçbir şekilde ihtiyaç duyulmaz.

► **Yanıt: a.** Statik testlerde kod çalıştırılmadığı için bir arıza (failure) tetiklenmez; dolayısıyla dinamik testlerdeki gibi arızayı yeniden üretme ve kök neden analizi yapma ihtiyacı doğmaz. Kusur (defect) doğrudan belgede veya kaynak kodda görünür durumdadır ve doğrudan giderilir.

HP-C1-1.1.2-c. Test ile hata ayıklama arasındaki temel ayrım aşağıdakilerden hangisidir?

- Test nedeni bulur; hata ayıklama arızayı gösterir
- Test arızayı tetikler/gösterir; hata ayıklama nedeni bulup giderir
- İkisi de yalnızca geliştiricinin sorumluluğundadır
- İkisi de yalnızca dinamik testte gerçekleşir

► **Yanıt: b.** Test arızaları açığa çıkarır (ya da hatayı doğrudan bulur); hata ayıklama ise nedeni bulup gideren geliştirme faaliyetidir.

1.2 Test Neden Gereklidir?

Testin başarıya katkıları (LO 1.2.1 — K2)

Test süreçleri, projenin başarısına çok yönlü katkılar sağlar. Bu katkıları genel ve soyut faydalar olarak görmek yerine, her birini somut ve ölçülebilir birer değer olarak ele almak gerekir:

- **Hataları Erken Tespit Etmek:** Hata düzeltme maliyetlerinin en düşük olduğu erken aşamalarda kusurları yakalamak, proje bütçesini korumanın ve verimliliği artırmanın en etkili yoludur. Henüz kod yazılmadan önce, gözden geçirilen bir gereksinim belgesindeki hatayı düzeltmek sadece birkaç dakikalık bir revizyon gerektirirken; aynı yanlış anlaşılmanın canlı ortamda fark edilmesi büyük sistem kesintilerine, acil kriz yönetimlerine ve kullanıcı memnuniyetsizliğine yol açar.
- **Kaliteyi Doğrudan Değerlendirmek:** Yazılım Geliştirme Yaşam Döngüsü (SDLC) boyunca üretilen tüm iş ürünlerinin kalitesine dair somut veriler sunar. Bu veriler, yönetimin canlıya çıkış (sürüm) kararları gibi kritik proje adımlarını güvenle atmasını destekler.
- **Son Kullanıcının Bakış Açısını Temsil Etmek:** Geliştirme süreci boyunca son kullanıcıların ve paydaşların beklentilerini masada tutar. Böylece teslim edilen ürünün, hedef kitlenin gerçek ihtiyaçlarını karşılama olasılığını ciddi oranda artırır.
- **Yasal ve Sözleşmesel Uyumluluk Sağlamak:** Test kanıtlarının ve raporlarının zorunlu olduğu durumlarda, kuruluşun sözleşmesel, yasal veya mevzuattan kaynaklanan standartları eksiksiz karşılamasına yardımcı olur.

Pratik Hayatta: Bir hastane randevu yazılımı geliştirilirken, henüz tek bir satır kod bile yazılmadan gereksinim belgeleri gözden geçirilir. Bir test uzmanı şu soruyu yöneltir: *“Bir hasta, aynı saat dilimine iki farklı doktordan randevu almak isterse sistem ne yapacak?”* Bu senaryonun kuralı analiz dokümanında tamamen unutulmuştur. Eksikliği kağıt üzerinde fark edip çözmek en fazla on dakikalık bir toplantıya mal olur; fakat bu durumun canlı ortamda binlerce hasta tarafından deneyimlenmesi tam bir operasyonel kriz yaratacaktır.

Sınav Odağı: Bu hedef genellikle senaryo-eşleştirme sorularıyla sınılanır. Size bir olay verilir ve bunun testin meşru bir katkısı olup olmadığını belirlemeniz istenir. Temel mantık nettir: Test; hataları erken bularak, kaliteyi değerlendirerek, son kullanıcıyı temsil ederek ve yasal/sözleşmesel kanıtlar sunarak başarıya ortak olur. Sınavdaki en kritik tuzak, testi yazılıma kalite “**enjekte eden**” veya hataları “**onaran**” bir faaliyet olarak gösteren şıklardır. Kaliteyi inşa etmek bütün ekibin ortak sorumluluğudur; hatayı onarmak ise hata ayıklamanın (geliştiricinin) işidir. İşaretleme yapmadan önce, şıkkın bir kodlama veya onarım işini değil, testin sağladığı **bilgi, analiz ve değerlendirme** rolünü betimlediğinden emin olun.

Hemen Pekiştir

HP-C1-1.2.1-a. Bir gereksinim belgesindeki belirsiz bir kural, henüz tek satır kod yazılmadan yapılan gözden geçirmede fark edilir ve düzeltilir. Bu durum testin hangi katkısına örnektir?

- Hataları erken tespit etmek
- Yasal ve sözleşmesel uyumluluğu sağlamak
- Son kullanıcının bakış açısını temsil etmek
- Ürüne doğrudan kalite eklemek

► **Yanıt: a.** Hata, en ucuz aşamada (kod yazılmadan önce) yakalanmıştır. (d) yanlıştır: test kaliteyi inşa etmez; kaliteyi bütün ekip inşa eder.

HP-C1-1.2.1-b. Aşağıdakilerden hangisi testin gerçek bir katkısı **değildir**?

- Hataları erken ve ucuz aşamada bulmak
- İş ürünlerinin kalitesini doğrudan değerlendirmek
- Bulunan hataları onarmak
- Düzenleyici uyumluluk için kanıt sağlamak

► **Yanıt: c.** Hataları onarmak hata ayıklamadır (geliştirme faaliyeti); testin sağladığı şey bilgi ve değerlendirmedir.

HP-C1-1.2.1-c. Bir dış denetim sürecinde, ürünün canlıya çıkışı öncesinde resmi test kanıtları talep edilmiş; proje ekibi de detaylı test raporlarını sunarak bu yasal yükümlülüğü yerine getirmiştir. Bu durum, test faaliyetlerinin projenin başarısına sağladığı hangi temel katkıyı doğrudan örneklendirir?

- Yasal/sözleşmesel uyumluluğu desteklemek
- Hataları erken bulmak
- Son kullanıcı bakış açısını temsil etmek
- Kaliteyi inşa etmek

► **Yanıt: a.** Test kanıtının istendiği durumlarda raporlar, düzenleyici/sözleşmesel uyumluluğu destekler.

Test ve kalite güvence (LO 1.2.2 — K1)

Sektörde “test” ile “kalite güvence (Quality Assurance, QA)” terimleri çoğu zaman aynı anlamda kullanılır. Bu iki kavram birbiriyle son derece ilişkili olsa da kesinlikle aynı şey değildir.

Karşılaştırma Alanı	Test (Bir Kalite Kontrol Biçimi)	Kalite Güvence (QA)
Ana Odak	Ürün odaklıdır.	Süreç odaklıdır.

Karşılaştırma Alanı	Test (Bir Kalite Kontrol Biçimi)	Kalite Güvence (QA)
Niteliği	Düzeltilicidir: Mevcut iş ürünlerindeki veya koddaki hataları ortaya çıkarmaya çalışır.	Önleyicidir: Gelecekte daha az hata oluşsun diye standartlar, süreçler ve iyi uygulamalar kurar.
Temel Soru	“Geliştirilen bu üründe bir hata var mı?”	“Kalite üreten, doğru bir yöntem ve süreçle mi çalışıyoruz?”
Çıktıların Kullanımı	Düzeltililebilmeleri için spesifik hataları ve arızaları listeler.	Geliştirme ve test süreçlerinin ne kadar iyi işlediğine dair geri bildirim sağlayarak sistemi iyileştirir.

Kısacası; test inşa edilen ürünü incelerken, kalite güvence (QA) bu ürünün inşa edilme yolunu ve yöntemlerini iyileştirir. Her iki disiplin de kalitenin artırılması için birlikte çalışır ve birbirini besler: Örneğin test ekibinin bulduğu kararlı hata örüntüleri, QA ekibinin geliştirme süreçlerindeki tıkanıklıkları veya eksiklikleri tespit edip iyileştirmesi için en değerli girdiyi sağlar.

Sınav Odağı: Bu hedefe yönelik sorular tamamen “**Ürün ve Düzeltici**” ile “**Süreç ve Önleyici**” arasındaki ayrım üzerine kurulur. Karşınıza çıkacak yanıltıcı şıklar genellikle süreç düzeyindeki, hata önlemeye yönelik bir faaliyeti betimleyip buna “test” der; ya da test faaliyetinin kendisinin gidip hataları doğrudan düzelttiğini iddia eder. Sınıflandırmanızı her zaman odağın ürün mü yoksa süreç mi olduğu kriterine göre yapın.

Hemen Pekiştir

HP-C1-1.2.2-a. “Tanımlı geliştirme sürecinin tüm projelerde kurallara uygun işleyip işlemediğini denetlemek” aşağıdakilerden hangisine örnektir?

- Test (bir kalite kontrol biçimi)
- Kalite güvence (QA)
- Doğrulama testi
- Hata ayıklama

► **Yanıt: b.** Süreç odaklı ve önleyici bir faaliyet → kalite güvence. Test, ürün odaklı ve düzelticidir.

HP-C1-1.2.2-b. Aşağıdakilerden hangisi testin (bir kalite kontrol biçiminin) ayırt edici özelliğidir?

- Süreç odaklı ve önleyicidir
- Ürün odaklı ve düzelticidir
- Hataların oluşmasını baştan engeller
- Yalnızca süreçleri denetler, ürünü değil

► **Yanıt: b.** Test mevcut ürünü inceler ve var olan hataları bulur (ürün odaklı, düzeltici); önleme ve süreç odağı QA'ya aittir.

HP-C1-1.2.2-c. Bir ekip, gelecekte daha az hata oluşması için kod inceleme standartları ve süreç kontrol listeleri tanımlıyor. Bu faaliyet hangisidir?

- Test (kalite kontrol)
- Kalite güvence (QA)
- Doğrulama testi
- Regresyon testi

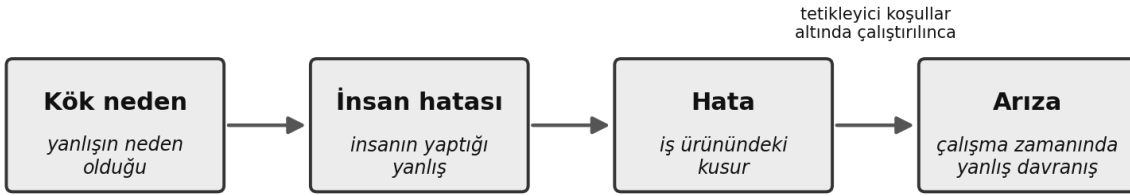
► **Yanıt: b.** Önleyici ve süreç düzeyinde bir çalışma olduğundan kalite güvencedir.

İnsan Hataları, Hatalar, Arızalar ve Kök Nedenler (LO 1.2.3 — K2)

Bu konu, müfredattaki (syllabus) en kritik kavramsal zincirlerden biridir ve sınavda hata payı bırakmayacak kadar net anlaşılmalıdır. Süreç doğrusal bir neden-sonuç ilişkisi izler:

- **İnsan Hatası (Error):** Bir kişinin; zaman baskısı, karmaşıklık, yorgunluk veya eğitim eksikliği gibi nedenlerle yaptığı yanlış eylemdir.
- **Hata / Kusur (Defect / Bug):** İnsan hatasının bir iş ürününe yansıyan somut sonucudur. Bir gereksinim dokümanında, tasarım belgesinde, kod satırında veya konfigürasyon dosyasında bırakılan kusurdur.
- **Arıza (Failure):** Koddaki bir hata **çalıştırıldığında** yazılımın gerçekleştirdiği yanlış davranıştır. Sistemin yapması gerekeni yapmaması veya yapmaması gereken bir şeyi yapması durumudur; yani kusurun dışarıdan görünür hale gelmesidir.

Şekil 1.3, bu doğrusal akışı ve hatanın görünür bir arızaya dönüşmesi için gereken tetikleyici mekanizmayı göstermektedir.



Şekil 1.3 — Kök neden → insan hatası → hata → arıza. Bir hata, ancak tetikleyici koşullar altında çalıştırıldığında arızaya dönüşür.

Öge	Anlamı
Kök neden	Yanlışın neden olduğu
İnsan hatası	İnsanın yaptığı yanlış
Hata	İş ürününde bırakılan kusur
Arıza	Çalışma zamanındaki yanlış davranış; yalnızca hata tetikleyici koşullar altında çalıştırıldığında ortaya çıkar

Ancak bu zincir her zaman kendiliğinden ve kusursuzca işlemez. Sınav tam da bu doğrusal akışın dışına çıkan şu iki önemli istisnayı ölçer:

1. **Her hata bir arızaya yol açmaz:** Bazı hatalar, sistemin nadiren çalışan veya hiç uğranmayan kod bloklarında saklanır (buna bazen *uyuyan/dormant hata* denir). Örneğin, bir rezervasyon sisteminde sadece 29 Şubat tarihlerinde çalışan hatalı bir hesaplama kodu olsun. Bu hata üç yıldır kodun içinde durmasına rağmen, aradaki yıllarda canlı ortamda tetikleyici bir artık gün yaşanmadığı için hiçbir arızaya neden olmamıştır. Hata koda mevcuttur; sadece henüz arıza üretmemiştir.
2. **Her arıza, test nesnesindeki bir hatadan kaynaklanmaz:** Sistem bazen yazılımın kendisinde hiçbir kusur olmamasına rağmen arıza verebilir. Müfredat buna çarpıcı bir fiziksel örnek verir: Dışarıdan gelen bir radyasyon veya güçlü elektromanyetik alanlar, hatasız yazılmış bir donanım yazılımı (firmware) belleğini anlık bozarak arızaya yol açabilir. Daha gündelik bir test senaryosu düşünürsek: Koştuğunuz bir test, test ortamına geçen ayın güncel olmayan döviz kuru tablosu yüklendiği için başarısız olabilir. Yazılım kendisine verilen girdi için tamamen doğru davranmıştır; ortadaki

arıza yazılımın kodundaki bir hatadan değil, tamamen **çevresel/ortamsal faktörlerden** kaynaklanmaktadır.

Tüm bu döngünün en başında ise **Kök Neden (Root Cause)** yer alır. Kök neden, insan hatasının arkasındaki temel tetikleyicidir. Kök neden analiziyle bu derindeki sebebi çözmek, sadece o anki tek bir hatayı düzeltmekle kalmaz; benzer kaynaklardan beslenen tüm hata sınıflarının gelecekte ortaya çıkmasını kalıcı olarak engeller.

Zincirdeki Adım	Kavramsal Karşılığı	Somut Örnek (Fatura Sistemi)
Kök Neden	Yanlışın en temel sebebi	Vergi kuralı değişikliklerinin, hiçbir gözden geçirme (review) yapılmadan, resmi olmayan e-postalarla ekiplere iletilmesi.
İnsan Hatası	İnsanın yaptığı yanlış	Geliştiricinin, gelen e-postadaki yuvarlama kuralını aceleyle okurken yanlış anlaması.
Hata (Kusur)	Üründeki kusur	Kod bloklarına vergi tutarını yarı-yukarı yerine, her zaman aşağı yuvarlayacak hatalı bir matematiksel kuralın yazılması.
Arıza	Görünür yanlış davranış	Küsuratlı fatura kesilen müşterilerden her ay sistem tarafından eksik ücret tahsil edilmesi (Arızanın dışarıdan fark edilmesi).

Koddaki hatayı düzeltmek sadece bu tek fatura problemini çözer. Ancak kök nedeni ele alıp vergi kuralları için resmi, gözden geçirmeli bir değişiklik yönetim süreci kurmak, geliştiricinin *bir sonraki* kuralı da yanlış okumasını ve benzer tüm hataları baştan engeller.

Sınav Odağı: Senaryo soruları genellikle bir olay akışı betimler ve sizden hangi adımın insan hatası, hata, arıza veya kök neden olduğunu seçmenizi ister. Odak noktalarınız çok nettir: Bir insanın gerçekleştirdiği eylem **insan hatası**, dokümanda veya kodda duran durağan kusur **hata**, ekranda veya çıktıda gözlemediğiniz sistem sapması **arıza**, her şeyin arkasındaki en derin yönetsel veya süreçsel boşluk ise **kök nedendir**. Sınavda bu kavramların yanına, çeldirici olarak sıkça 'yanlış pozitif' (false positive / sahte alarm) veya 'yanlış negatif' (false negative / gözden kaçan kusur) terimleri yerleştirilebilir. Unutmayın; bu kavramlar test süreçlerindeki hatalarla değil, test araçlarının veya testlerin kendi sonuçlarındaki yanlıgılarla ilgilidir ve detaylarına ilerleyen bölümlerde değineceğiz.

Hemen Pekiştir

HP-C1-1.2.3-a. Bir geliştirici, bir e-postadaki kuralı yanlış anlar ve koda yanlış bir indirim oranı yazar. Bu kod canlı ortamda çalışınca müşterilere yanlış tutarlar yansıtılır. Müşterilere yansıyan yanlış tutarlar bu zincirde hangisidir?

- İnsan hatası (error)
- Hata (defect)
- Arıza (failure)
- Kök neden (root cause)

► **Yanıt:** c. Kusurun çalıştırılıp dışarıdan görünür hale gelmesi arızadır. Yanlış anlama insan hatası, koddaki yanlış oran hatadır.

HP-C1-1.2.3-b. Aynı senaryoda (analist mevzuatı yanlış anlar → dokümana yanlış kural yazılır → kod bunu uygular → müşteriye yanlış fatura), zincirdeki **hata (defect)** hangisidir?

- Analistin mevzuatı yanlış anlaması
- Dokümana ve koda yansıyan yanlış kural
- Müşterilere kesilen yanlış faturalar
- Kuralların gözden geçirilmeden e-postayla iletilmesi

► **Yanıt: b.** İş ürününde (doküman/kod) bırakılan kusur hatadır. (a) insan hatası, (c) arıza, (d) kök nedendir.

HP-C1-1.2.3-c. Bir koddaki hatalı hesaplama yalnızca 29 Şubat'ta çalışır ve üç yıldır hiçbir artık gün yaşanmadığı için hiç arıza üretmemiştir. Bu durum hangi gerçeği gösterir?

- Her arıza bir hatadan kaynaklanır
- Her hata mutlaka bir arızaya yol açmaz
- Hatalar belli alanlarda yoğunlaşır
- Test, hataların yokluğunu kanıtlar

► **Yanıt: b.** Çalıştırılmayan (uyuyan) bir hata gerçektir ama tetiklenmediği sürece arıza üretmez.

HP-C1-1.2.3-d. Hatasız yazılmış bir donanım yazılımı (firmware), dışarıdan gelen güçlü bir elektromanyetik etki nedeniyle yanlış davranır. Bu durum hangi gerçeği gösterir?

- Her arıza, test nesnesindeki bir hatadan kaynaklanmaz
- Her hata mutlaka bir arızaya yol açar
- Kök neden her zaman bir insandadır
- Statik test bu arızayı önlerdi

► **Yanıt: a.** Arıza, koddaki bir kusurdan değil çevresel koşullardan da kaynaklanabilir.

Ücretsiz örnek burada sona eriyor

Buraya kadar okuduklarınızı; kitabın giriş bölümleri ile Bölüm 1'in ilk iki alt bölümüydü. Kitabın tamamında sizi şunlar bekliyor:

- Bölüm 1'in kalanı: Yedi test prensibi, test aktiviteleri, roller, beceriler ve test bağımsızlığı
- Bölüm 2-6: Yaşam döngüsü, statik test, test teknikleri, test yönetimi ve test araçları — 64 öğrenme hedefinin tamamı
- Her K3 hedefi için adım adım çözümlü örnek ve kısa alıştırma: denklik payları, sınır değer analizi, karar tabloları, durum geçişi, kestirim, önceliklendirme ve hata raporu
- 40 soruluk tam deneme sınavı: cevap anahtarı, skor bantları ve her soru için gerekçeli açıklama
- 200'den fazla terim içeren sözlük, sık karıştırılan kavram çiftlerinin ana listesi, 2 ve 4 haftalık çalışma planları ve kaynakça

Kitabın tam sürümü: leanpub.com/istqb-ctfl-v4-tr

ISTQB® ve CTFL®, International Software Testing Qualifications Board'un tescilli ticari markalarıdır. Bu kitap bağımsız, gayri resmi bir çalışma kaynağıdır.