ISTQB Foundation Level Essential

Dr.Ahmet Unudulmaz

SIEMENS

İÇERİK

ISTQB Sertifikasyon Süreci

Yazılım Testinin Temelleri
- Yazılım Testi Nedir
- Yazılım Testi Neden Gereklidir
- Yedi Test Prensibi
- Test Süreci
- Test Etme Psikolojisi
- Temel Tanımlar
- Risk Bazlı Testler

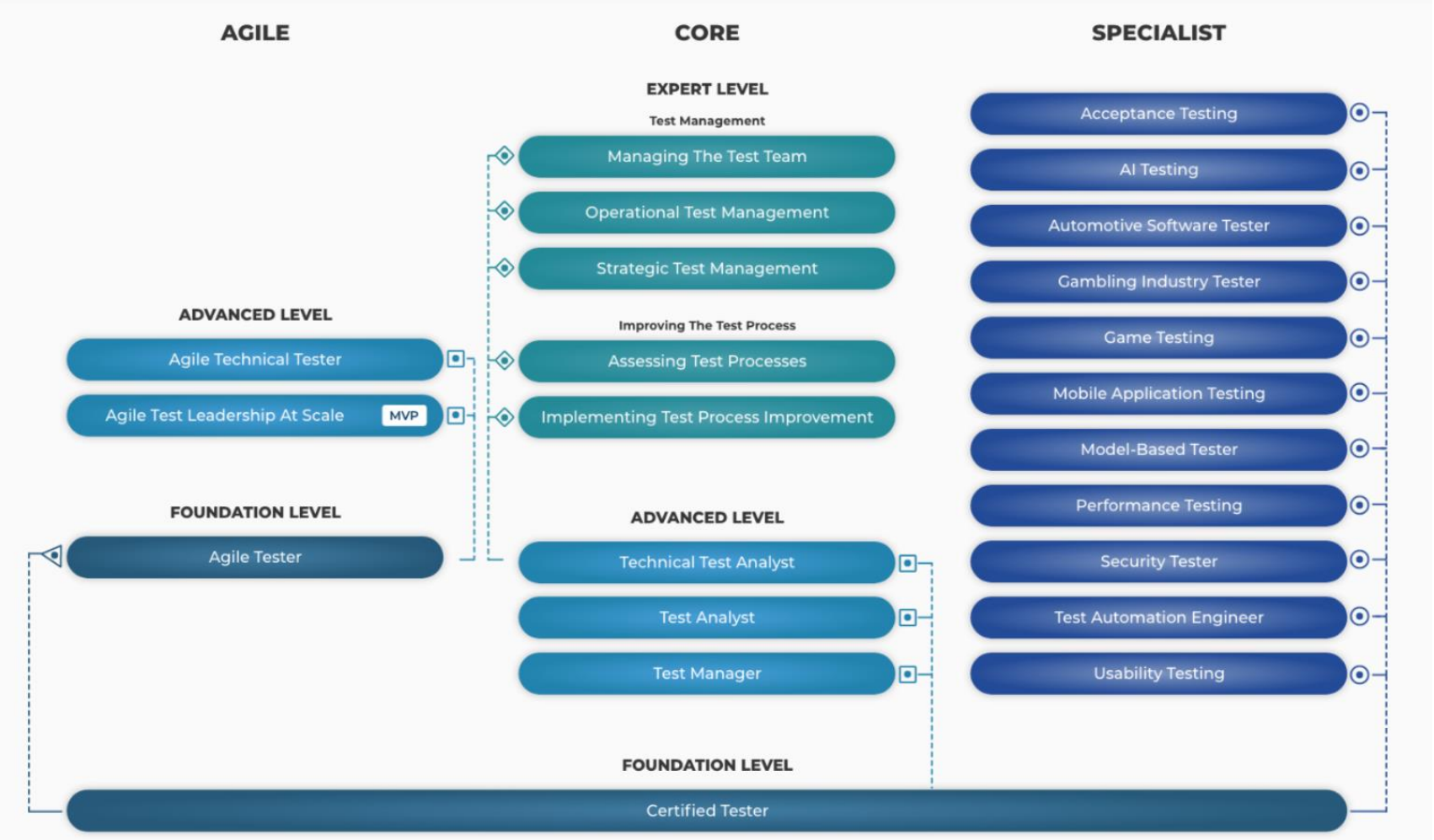
Yazılım Geliştirme Yaşam Döngüsü Boyunca Test
- Yazılım Geliştirme ve Yaşam Döngüsü Modelleri
- Test Seviyeleri
- Test Çeşitleri

Test Teknikleri

Test Otomasyon

SIEMENS

ISTQB Sertifikasyon Süreci



SIEMENS

Yazılım Testinin Hedefleri

- Gereksinimler, kullanıcı hikâyeleri, tasarım ve kod gibi çalışma ürünlerini değerlendirmek
- Belirtilen tüm gereksinimlerin yerine getirilip getirilmediğini doğrulamak
- Test nesnesinin eksiksiz olup olmadığının ve kullanıcıların ve diğer paydaşların beklediği şekilde çalıştığının sağlamasını yapmak
- Test nesnesinin kalite seviyesi hakkında güven oluşturmak
- Hataları önlemek
- Arızaları ve hataları tespit etmek
- Özellikle test nesnesinin kalite seviyesiyle ilgili olarak sağlıklı kararlar almalarını sağlamak için paydaşlara yeterli bilgiyi sunmak
- Yazılımın kalitesiz olma riskini düşürmek (örneğin, canlıda meydana gelebilecek arızaların önceden tespiti)
- Sözleşmeden kaynaklanan, yasal veya düzenleyici gereksinimlere veya standartlara uymak ve/veya test nesnesinin bu gereksinimlere veya standartlara uyumluluğunu doğrulamak

“Bug” Nedir?**

Error : İnsanların aksiyonu sonucunda hatalı sonuç üretilmesi

Fault : Yazılım içindeki hatalar

- Defect veya Bug olarakta adlandırılabilir
- Çalışıtırılarsa system hatasına sebebiyet verebilir

Failure : Yazılımdaki sapmalar (Sistemin Hatası)

A phone ringing in an adjac
ent cubicle momentarily distracts a programmer, causing the programmer to improperly program the logic that checks the upper boundary of an input variable. Later, during system testing, a tester notices that this input field accepts invalid input values. The improperly coded logic for the upper boundary check is:

a) The root cause
b) The failure
c) The error
d) The defect

Select ONE option.

Yazılım Niye Hatalı Olur ?

- Kompleks yazılım
- Zaman baskısı
- İnsani yanılma payı
- Deneyimsizlik veya yetersiz yetkinlik
- Gereksinimler ve tasarım ile ilgili yanlış iletişim de dâhil olmak üzere proje ekip üyeleri arasındaki yanlış iletişim
- Kodun, tasarımın, mimarinin, çözülecek temel problemin ve/veya kullanılan teknolojilerin karmaşıklığı
- Özellikle sistem içi ve sistemler arası etkileşimlerin sayısının fazla olduğu durumlarda, sistem içi ve sistemler arası arayüzler hakkındaki yanlış anlaşılmalar
- Yeni, henüz tecrübe edilmemiş teknolojiler

Teste Ne Zaman Başlanmalı ?

- Projenin ilk günü bile test vardır (Tasarım , tool seçimleri vb.)
- İlk safha gereksinim analizleriyle başlar
- Projenin amaç, test stratejisinin belirlenmesi gerekli
- Yazılım geliştirme süreçleriyle entegre olmalı

Kalite Maliyeti

Uygulama Kalitesinin arttırılması için yapılan harcama

+

Çıkan hataların düzeltilmesi için yapılan harcama

TEST PRENSİPLERİ

- Test Hatasızlığı Değil Hatalılığı Gösterir
- Bir Ürün %100 test edilemez
- Erken Test
- Hataların Kümelenmesi
- Böcek ilacı paradoksu – Testcase update
- Test içerik bağımlıdır (Hız / Performans / Güvenlik)
- Hata kalmadı yanılgısı

As a result of risk analysis, more testing is being directed to those areas of the system under test where initial testing found more defects than average.

Which of the following testing principles is being applied?

a) Beware of the pesticide paradox.
b) Testing is context dependent.
c) Absence-of-errors is a fallacy.
d) Defects cluster together.

Select ONE option.

A product owner says that your role as a tester on an Agile team is to catch all the bugs before the end of each iteration. Which of the following is a testing principle that could be used to respond to this statement?

a) Defect clustering
b) Testing shows the presence of defects
c) Absence of error fallacy
d) Root cause analysis

Select ONE option.

TEST SÜREÇLERİ

Test Planı

- Master Test Plan
- Rol / Sorumluluk
- Risk / Kaynak / Efor

Test Analiz

- Doküman İnceleme
- Test Dizayn
- Neyin Test Edileceğinin Tespiti
- Limitler / Koşullar

Test Tasarımı

- Test Case Özellikleri
- Test Case İsimleri

Test Uyarlama

- Test Prosedür
- Test Case İçerik
- Test Ortamı

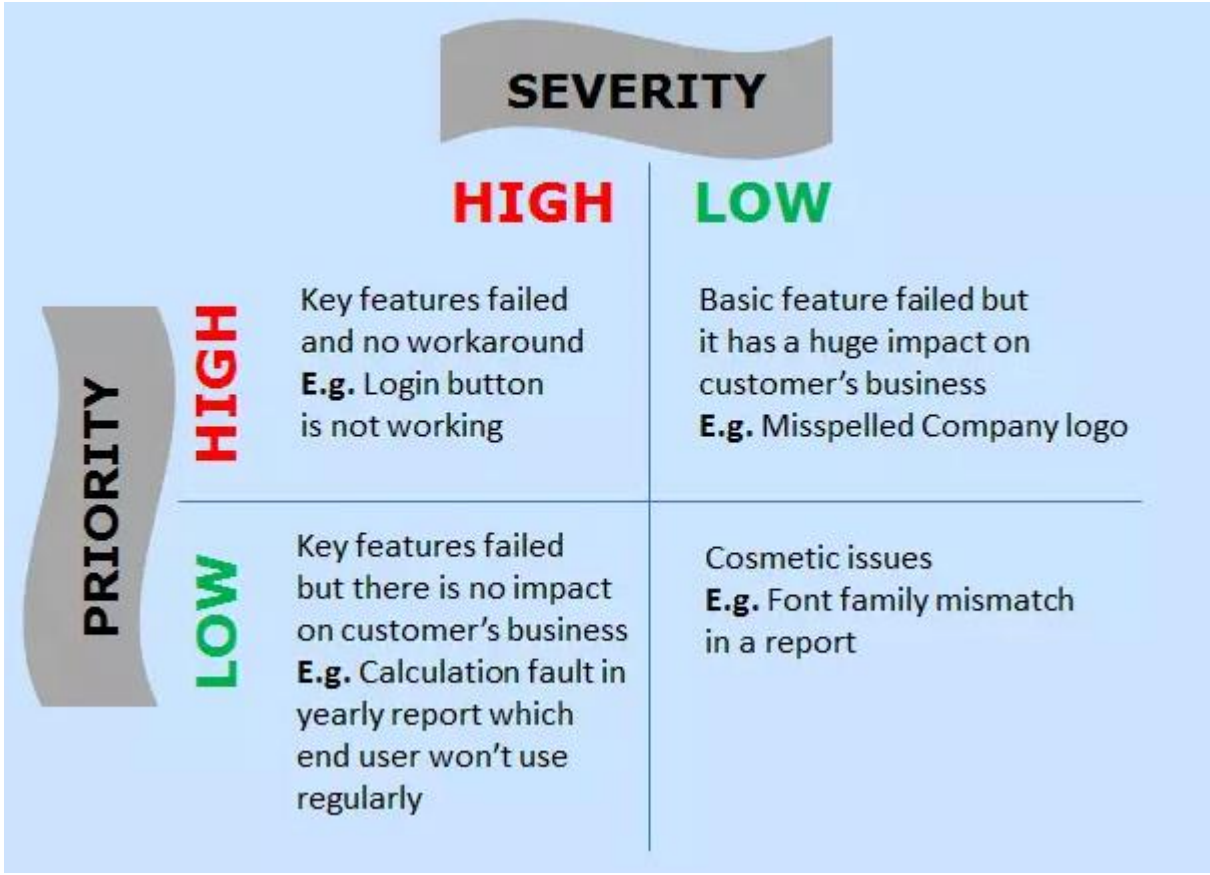
Test Uygulama

- Çalıştırma

Priority & Severity

Priority (Öncelik) : Hatanın müşteriye etkisi

Severity (Önem) : Hatanın sisteme etkisi, çözülmesi için gerekli zaman & kaynak maliyeti



Önemli Tanımlar

Test Basis : Teste girdi olan dokümanların tamamına verilen isim (dayanak)

Test Object : Test edilen ürün

Test Objectives : Testin hedefi & amacı , projeye göre değişir

Test Approach : Test strajesinde belirlenen test yaklaşımı

Önemli Tanımlar (2)

Test Policy : Firmanın testten ne anladığını, standartların, organizasyon yapısının içinde bulunduğu doküman (3-5 sf olmalı) – Roller, ünvanlar, test standartları gibidir

Test Strategy : Firmada yapılması gereken test aktivite ve tiplerinin içeren ve bunların nasıl yapılacağından bahsedilen doküman (Test seviyeleri, girdi & çıktı kriterleri, test teknikleri vs)

Fedex

Taşımacılık yapması (Policy) | Uçak ve araba ile taşıması (Strategy)

Paketi 3 saatte göndermesi (Hedef) | Uçakla göndermesi (Yaklaşım)

Önemli Tanımlar (3)

Verification (Doğrulama) : Sistem bileşenlerinin yazılım geliştirme sürecinde her safhada konan hedefleri karşılayıp karşılamadığının doğrulanması

- Yazılım doğru üretildi mi?

Validation (Sağlama) : Sistemin gereksinimlere Uygun çalışıp çalışmadığın kontrolü

- Doğru yazılım üretildi mi?

Test Stratejisi

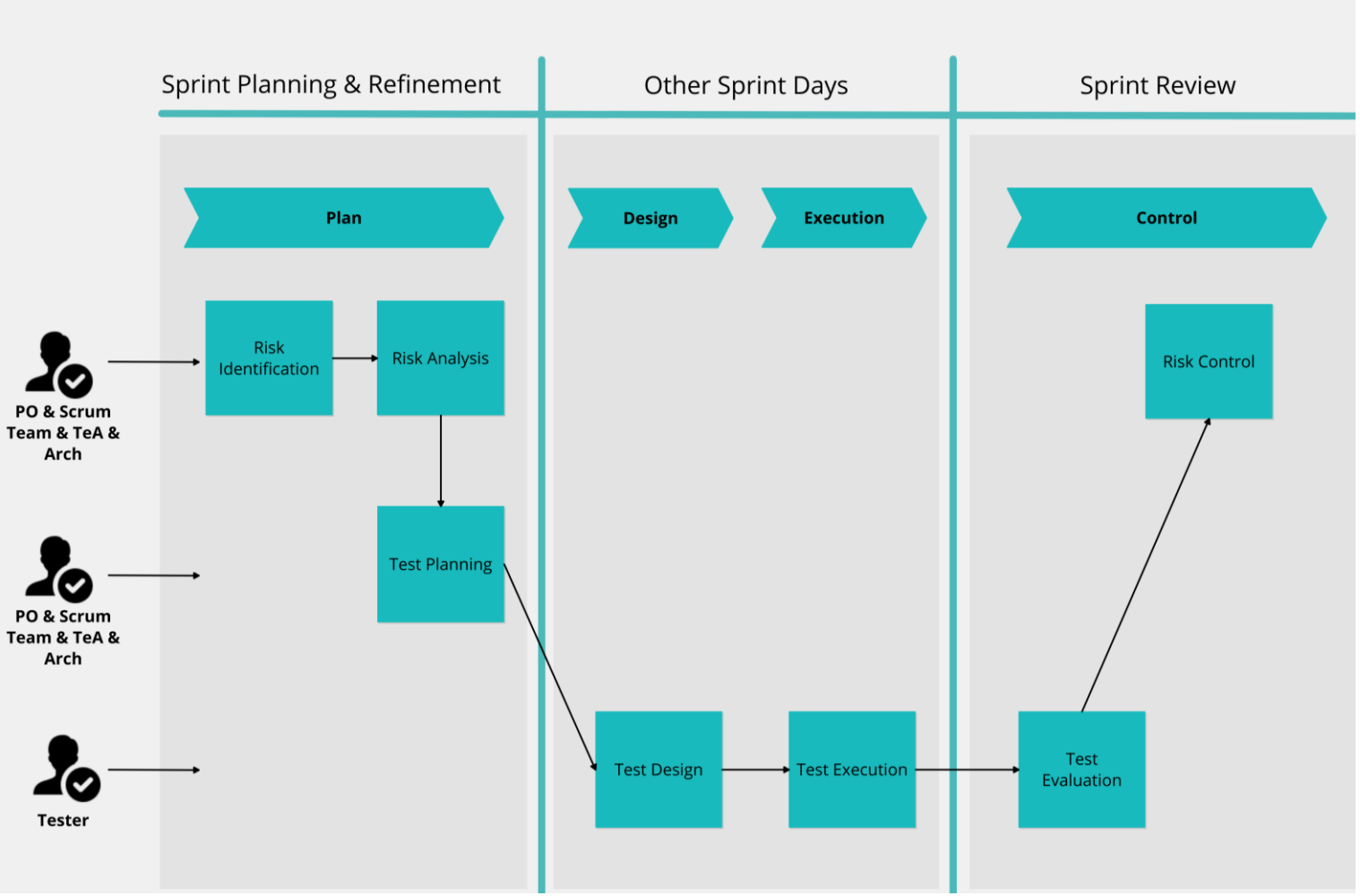
SIEMENS

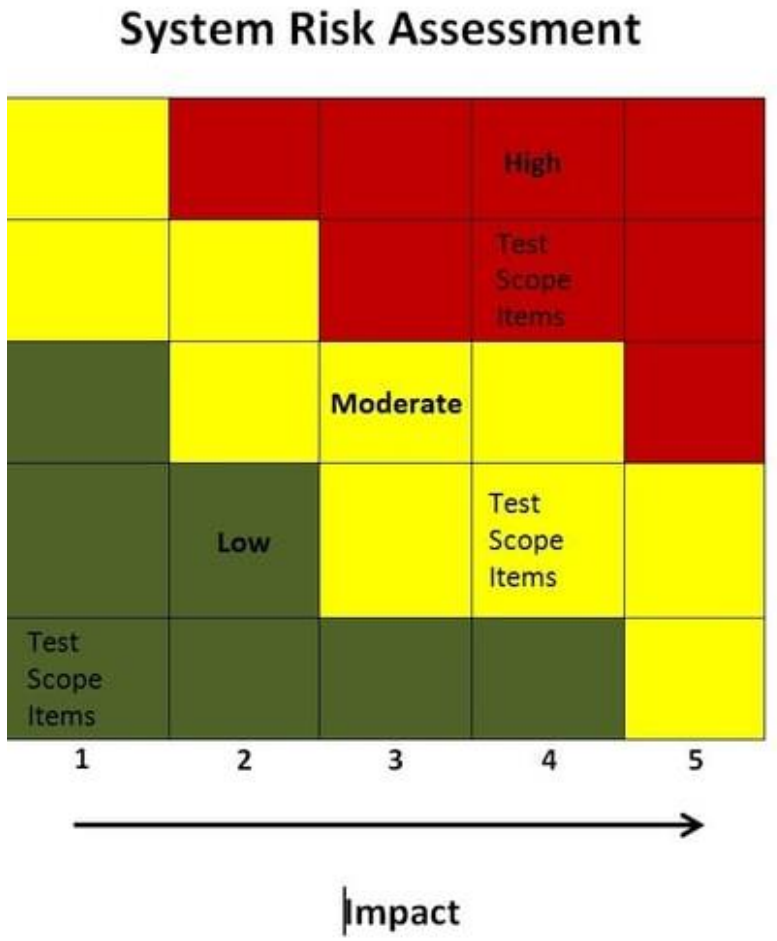
Riskler

ISTQB'ye göre RİSKLER

- Ürün Riski
  - Fonksiyonalite
  - Security
  - Güvenilirlik
  - Kullanılabilirlik
  - Performans
  - Bakım
- Proje Riski
  - Scope
  - Timeline
  - Resource

Risk Bazlı Test Yaklaşımı





SIEMENS

Which of the following is a project risk?

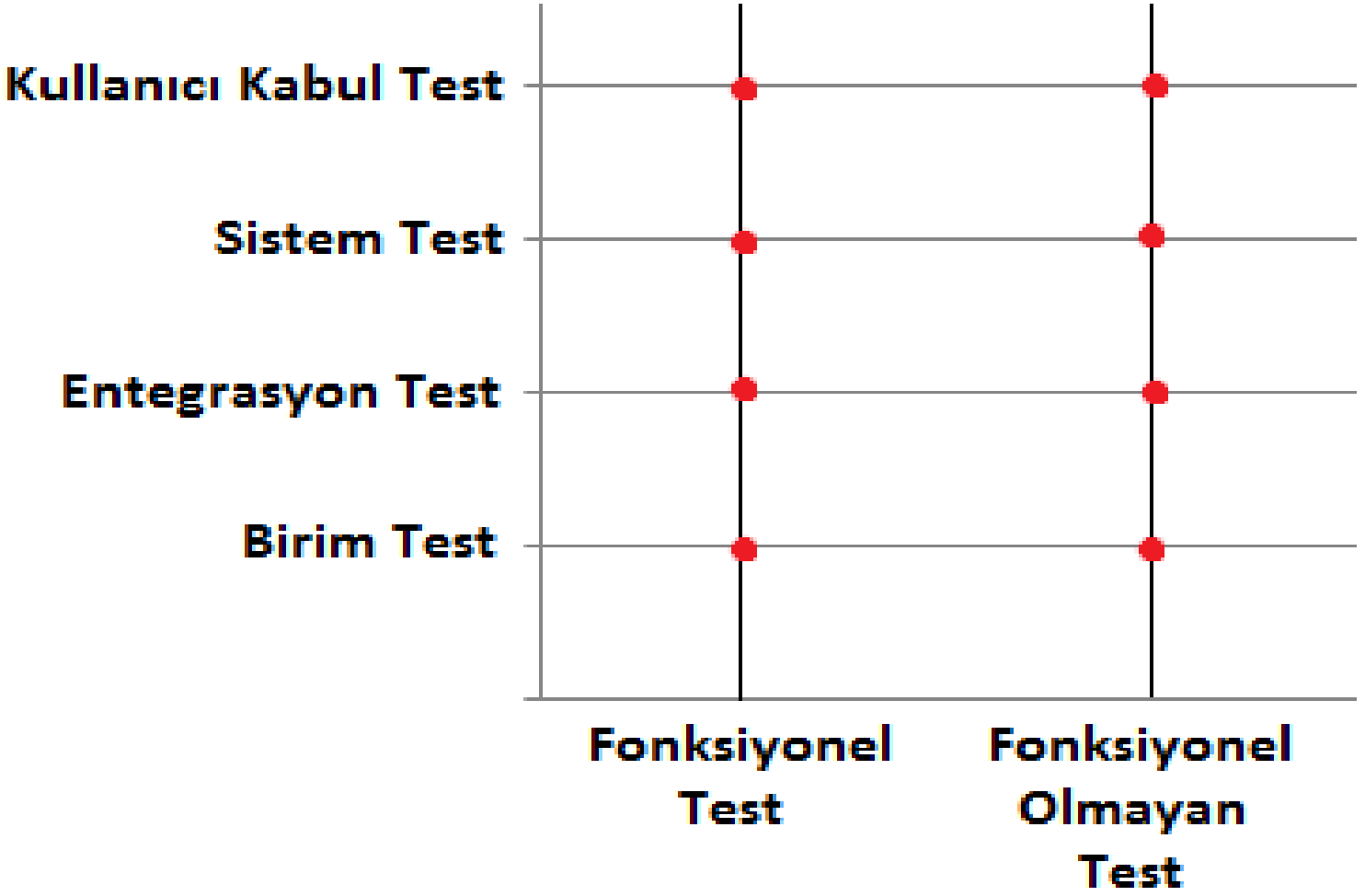
a. A defect that is causing a performance issue
b. A duplicate requirement
c. An issue with a data conversion procedure
d. A schedule that requires work during Christmas shutdown

Question #36 (1 Point)

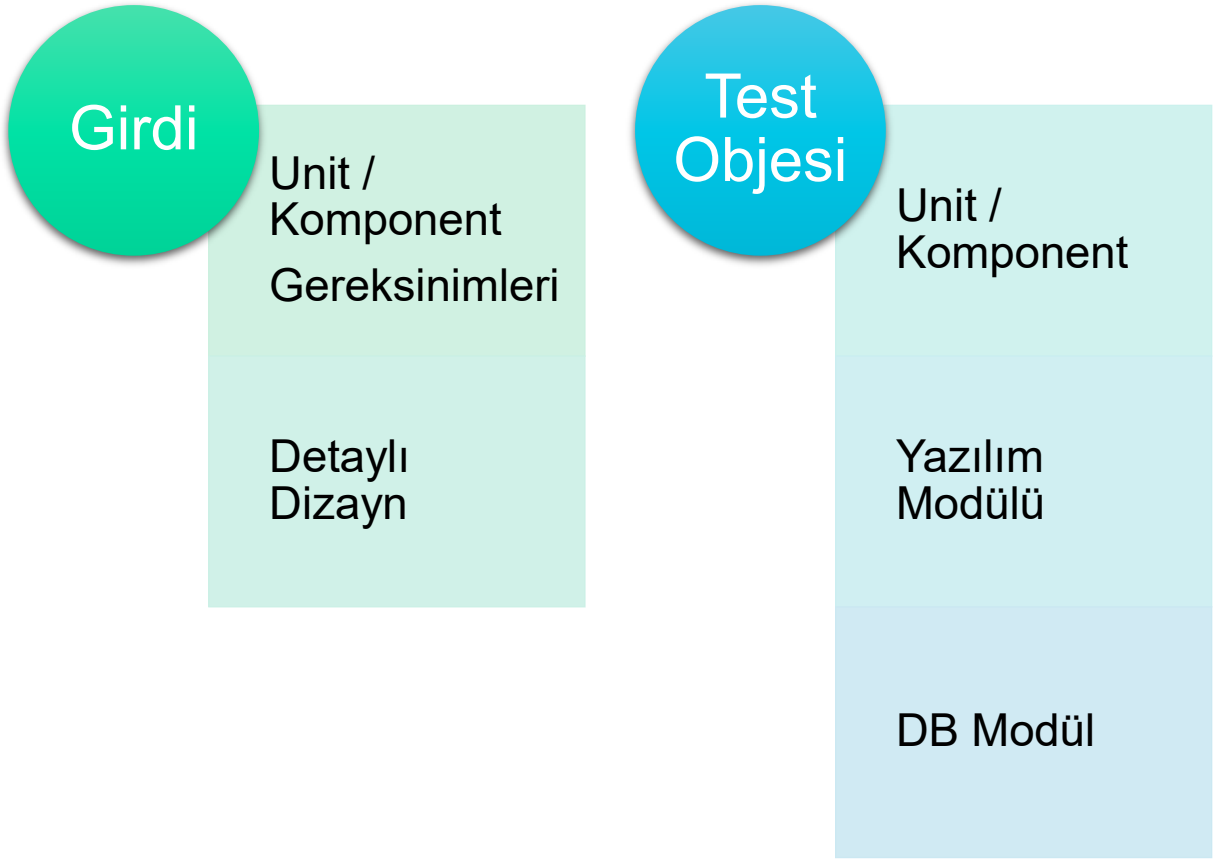
Which of the following is MOST likely to be an example of a PRODUCT risk?

a) The expected security features may not be supported by the system architecture.
b) The developers may not have time to fix all the defects found by the test team.
c) The test cases may not provide full coverage of the specified requirements.
d) The performance test environment may not be ready before the system is due for delivery.

TEST SEVİYELERİ ve ÇEŞİTLERİ

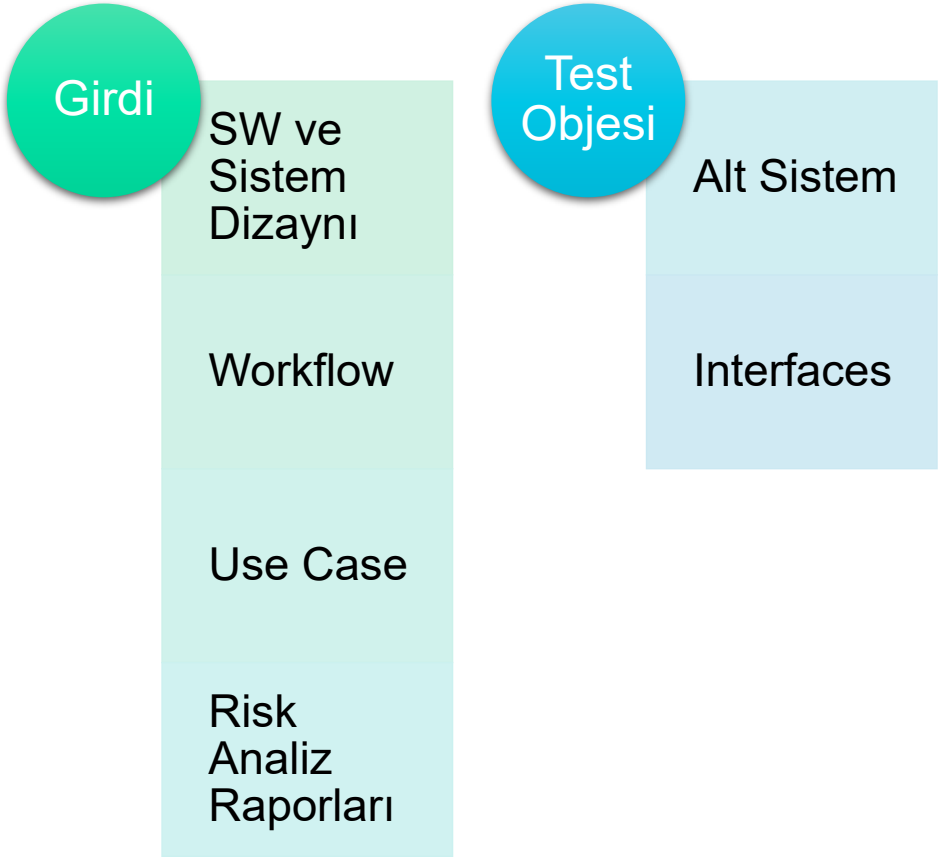


BİRİM TESTİ

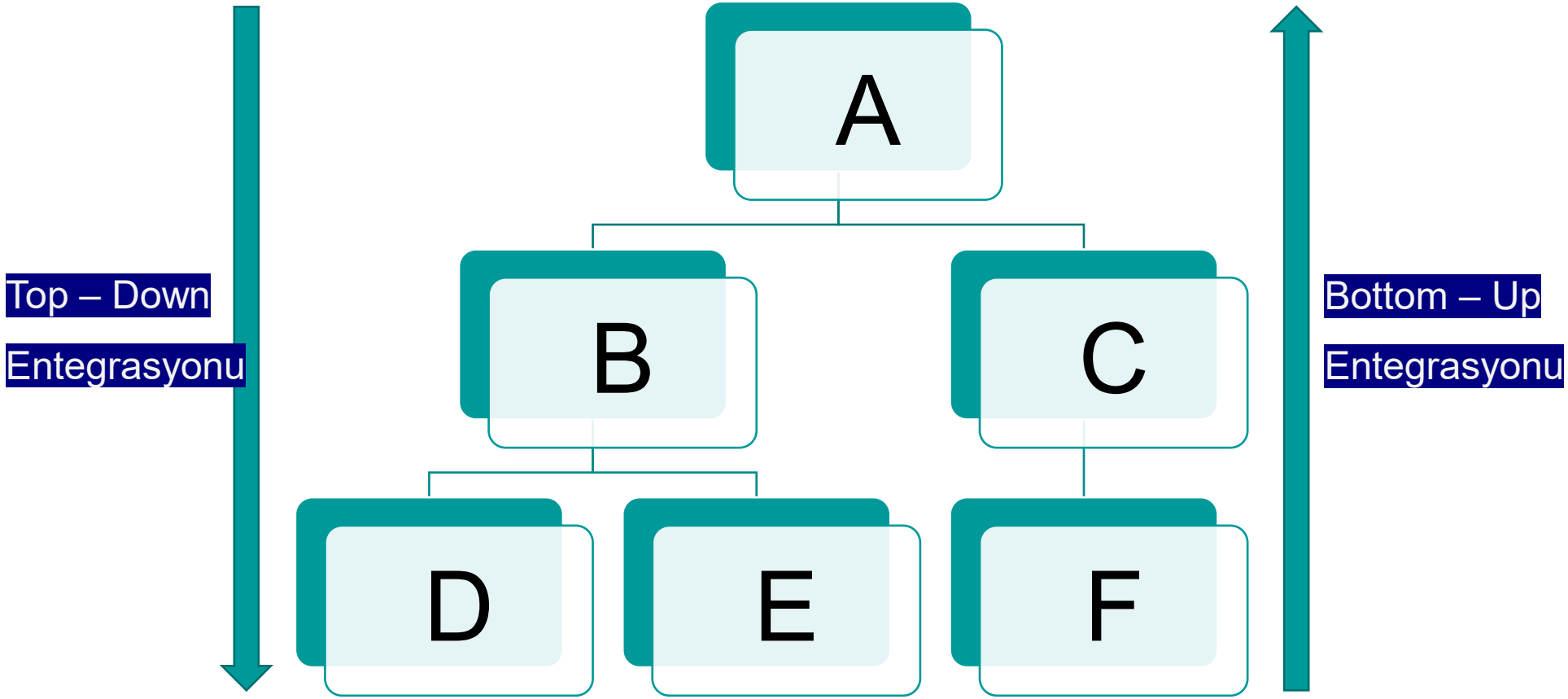

Girdi
Unit / Komponent Gereksinimleri
Detaylı Dizayn
Test Objesi
Unit / Komponent
Yazılım Modülü
DB Modül

BİRİM TESTİ

ENTEGRASYON TESTİ



ENTEGRASYON TESTİ – 2



SİSTEM TESTİ

Girdi

- SW ve Sistem Gereksinimleri
- Risk analiz raporları
- Use Case

Test Objesi

- Sistem
- Tam entegre edilmiş uygulamalar

KULLANICI KABUL TESTİ

Girdi

Kullanıcı Test Senaryoları

Test Objesi

Nihai Ürün

Alfa + Beta Testleri

TEST ÇEŞİTLERİ

Fonksiyonel

- Doğruluk (Accuracy)
- Uygunluk (Suitability)
- Uyumluluk (Interoperability)

Fonksiyonel Olmayan

- Güvenilirlik (Reliability)
- Kullanılabilirlik (Usability)
- Taşınabilirlik (Portability)
- Sürdürülebilirlik (Maintainability)
- Performans

Performans Testi

- Zaman Bazlı
- Kullanıcı Bazlı
- Kaynak Kullanımı
- Load Test
- Stress Test

ReTest vs Regression Test

Retest	• Bug Fixed
Regresyon Testi	• Yan Etki

Yazılım Geliştirme Modelleri

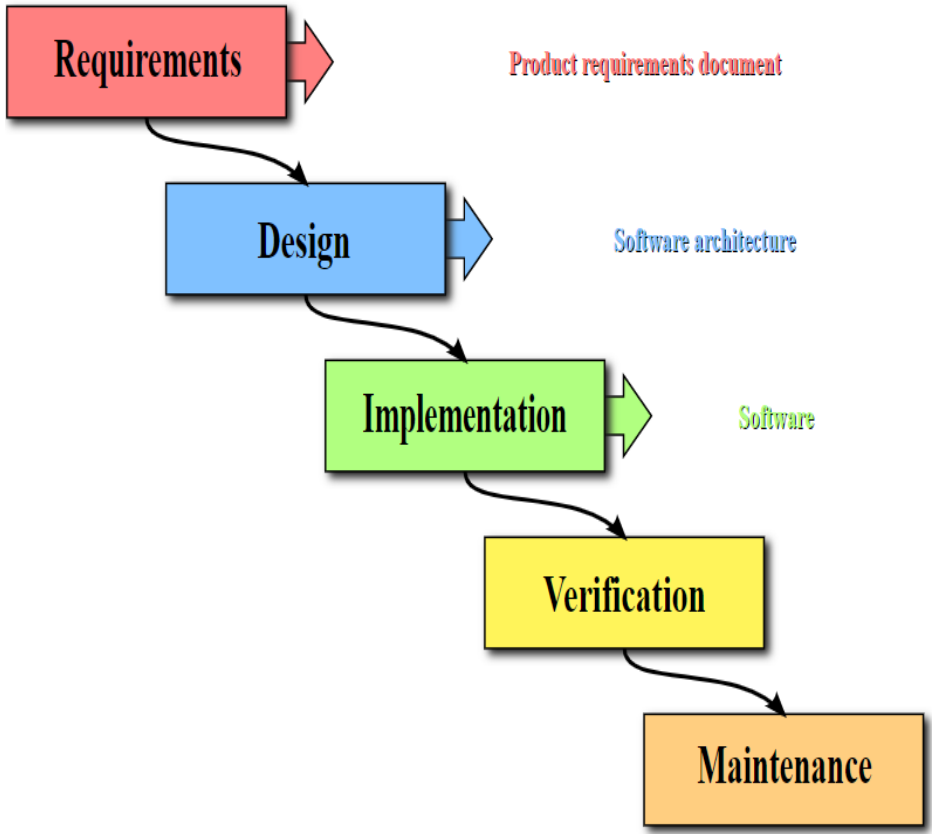
Traditional / Sequential

- Waterfall
- V Model

Iterative / Incremental

- Arttırımlı Geliştirme
- Spiral
- Agile (Scrum & XP)

Waterfall



- Şelalenin her basamağında yer alan aktiviteler eksiksiz olarak yerine getirilir. Bu bir sonraki basamağa geçmenin şartıdır.
- Her safhanın sonunda bir doküman oluşturulur. Bu yüzden şelale modeli doküman güdümlüdür.
- Yazılım süreci doğrusaldır, yani bir sonraki safhaya geçebilmek için bir önceki safhada yer alan aktivitelerin tamamlanmış olması gerekir.
- Kullanıcı katılımı başlangıç safhasında mümkündür. Kullanıcı gereksinimleri bu safhada tespit edilir ve detaylandırılır. Daha sonra gelen tasarım ve kodlama safhalarında müşteri ve kullanıcılar ile diyaloğa girilmez.

Waterfall (2)

Avantajları

Fazların net bir şekilde sınırlandırılması

Basit planlama ve kontrol olanakları

Basit ve anlaşılabilir bir model

Düşük maliyet

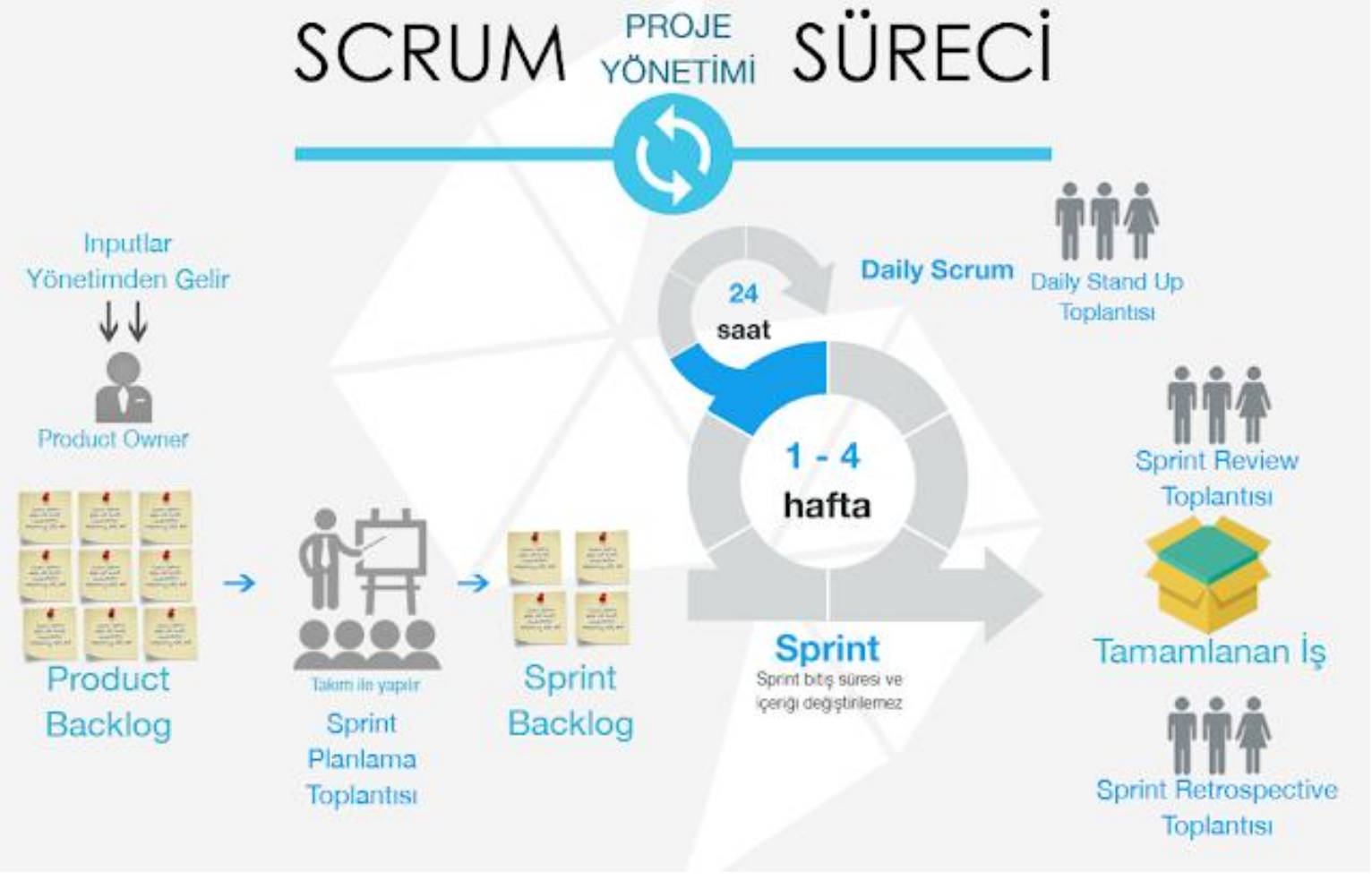
Dezavanzajları

Kullanıcı katılımı sadece tanım aşamasında mümkün

Doküman güdümlü

Sıralama, sınırlandırma ve yeterlilik problemleri

Scrum


SCRUM PROJE YÖNETİMİ SÜRECİ
Inputlar
Yönetimden Gelir
Product Owner
Product
Backlog
Takım ile yapılır
Sprint
Planlama
Toplantısı
Sprint
Backlog
24
saat
Daily Scrum
Daily Stand Up
Toplantısı
1 - 4
hafta
Sprint
Sprint bitiş süresi ve
içeriği değiştirilemez
Sprint Review
Toplantısı
Tamamlanan İş
Sprint Retrospective
Toplantısı

Scrum (2)

Avantajları

- Kısa planlanmış döngüler, değişen gereksinimler ve ani ortaya çıkabilecek değişiklikler için organizasyonunuza esneklik imkânı tanır.
- Müşterileriniz ürünü oluşturmak ve çıkan problemler kolayca çözebilmek için aktif bir şekilde geribildirim sağlar.
- Ekip katılımına öncelik verilerek kurulan iletişim sayesinde sorunların daha hızlı çözülmesini sağlar.
- Proje hızla ilerlerken ortaya çıkan her problem bir sonraki döngüde daha iyi bir çözüm için basamak olarak kullanılır.
- **Dezavantajları**
- Ürün gereksinimleri sürekli değişebileceğinden maliyetler peşinen tahmin edilemez.
- Müşterilerden sürekli geri bildirim almak sizi alaşağı edebilir ya da bazı müşterilerin zamanı ve hiç ilgisi olmayabilir.
- Müşteriler geri bildirim vermek istediklerinde akıllarına gelebilecek yeni istekler ek fazları ortaya çıkarır. Bu durum proje maliyetlerini ve süresini uzatabilir.
- Müşteri ile iletişimi zor olan büyük Kurumsal yapılarda uygulamak zor olacaktır.

Test Tasarım Teknikleri

Dinamik Test Tasarımı

- Specification Based (Black Box)
- Experienced Based
- Structured Based (White Box)

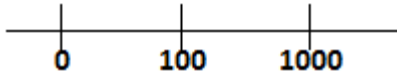
Statik Test Tasarımı

- Walkthrough
- Informal Review
- Technical Review
- Management Review
- Audit
- Inspection

2^n

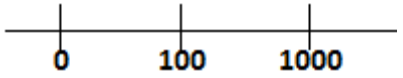
Specification Based (Black Box)

Denklik Sınıfı Test Tekniği



- 0 <
- 0 > ve 100 <
- 100> ve 1000<
- 1000>

Sınır Değer Test Tekniği



- -1 0 1
- 99 100 101
- 999 1000 1001

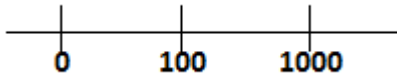
Karar Tablosu Test Tekniği

- Durum
- Aksiyon

İş Senaryo Test Tekniği

- Temel akış
- Alternatif akış
- Hatalı akış

Denklik Sınıfı Test Tekniği (Equivalence Partitioning)



- 0 <
- 0 > ve 100 <
- 100> ve 1000<
- 1000>

Bu teknikle %100 denklik payı kapsamı elde etmek için, test senaryoları, her paydan en az bir değer kullanarak, belirlenmiş tüm payları (geçersiz paylar dâhil) kapsamalıdır. Kapsam, test edilen denklik paylarının sayısının tüm denklik paylarının sayısına bölünmesiyle ölçülür ve normalde yüzde olarak ifade edilir. Denklik paylarına ayırma tüm test seviyelerinde uygulanabilir.

Örnek

Bir vergi sisteminde 4000 TL'ye kadar vergisiz, sonraki 1500 TL için %10, sonraki 28000 TL için %20 ve daha yüksek meblalar için %40 vergilendirme yapılmaktadır. Denklik sınıfı tekniğine göre aşağıdaki very gruplarından hangisi aynı kategoriye düşer

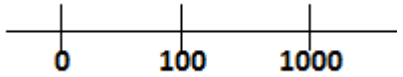
a)5600,28000,7800

b)2800,4200,4800

c)3000,4500

d)28000,50000,60000

Sınır Değer Analizi (Boundary Value)



- -1 0 1
- 99 100 101
- 999 1000 1001

Sınır değer analizi tüm test seviyelerinde uygulanabilir. Bu teknik genellikle bir sayı aralığı (tarihler ve saatler dâhil) gerektiren gereksinimleri test etmek için kullanılır. Kapsam, test edilen sınır eğerlerin sayısının tüm sınır değerlerin sayısına bölünmesiyle ölçülür ve normalde yüzde olarak ifade edilir.

Örnek

Bir vergi sisteminde 4000 TL'ye kadar vergisiz, sonraki 1500 TL için %10, sonraki 28000 TL için %20 ve daha yüksek meblalar için %40 vergilendirme yapılmaktadır. Sınır Değer Analizi için hangi test verisi kullanılmalıdır

a)1500

b)33501

c)4500

d)28000

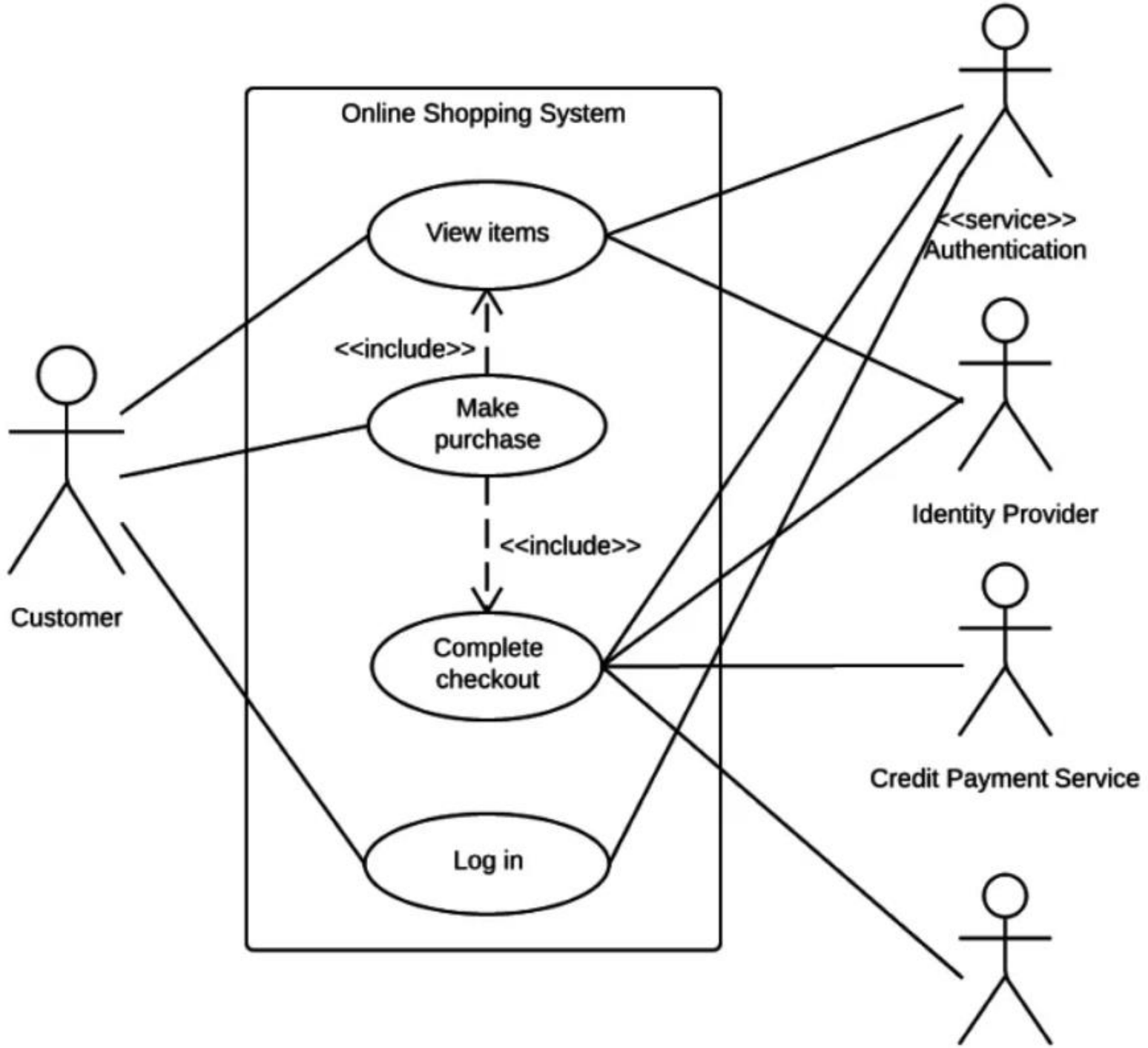
İş Senaryo (Use Case) Test Tekniği

- Her kullanım senaryosu, bir sistemin bir veya daha fazla aktör ile işbirliği içinde gerçekleştirebileceği bazı davranışları belirtir

- Bir kullanım senaryosu, etkileşimler ve aktivitelerin yanı sıra uygun durumlarda önkoşullar, artkoşullar ve doğal bir anlatımla tanımlanabilir.

- Aktörler ve sistem arasındaki etkileşimler sistemin durumunda değişikliklere neden olabilir. Etkileşimler; iş akışları, faaliyet şemaları veya iş süreç modelleri ile grafiksel olarak gösterilebilir.

- Bir kullanım senaryosu, istisnai davranış ve hata ele alma dâhil olmak üzere, temel yazılım davranışının olası varyasyonlarını, alternatiflerini içerebilir. Testler, tanımlanmış davranışları (temel, alternatif, istisnai ve hata ele alma) denemek için tasarlanır.



Basic Flow

Alternate Flow

Exception Flow

Structured Based (White Box)

Komut testi, kod içinde yer alan yürütülebilir komutların üzerinden geçilip bu komutların çalıştırılmasıdır. Kapsam, testler tarafından çalıştırılan komutların sayısının test nesnesindeki çalıştırılabilir komutların toplam sayısına bölünmesi ile ölçülür ve normalde yüzde olarak ifade edilir.

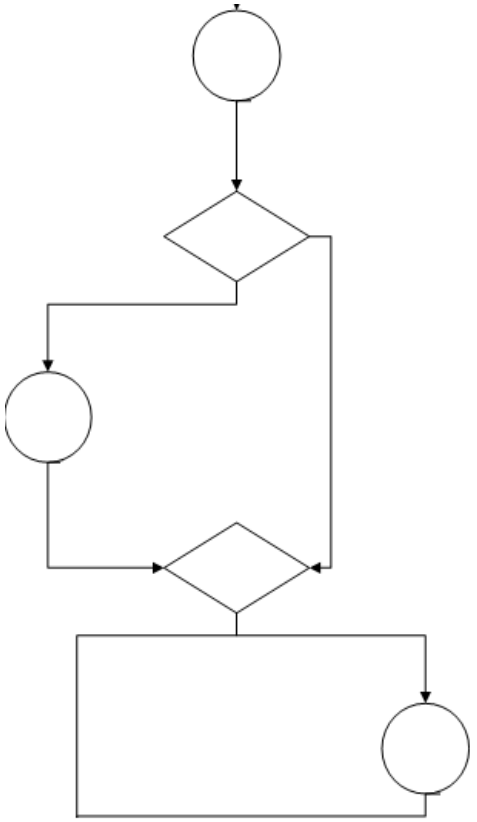
Karar testi, koddaki kararların üzerinden geçilip bu kararların çalıştırılması ve karar çıktılarına dayanarak kodun test edilmesidir. Bunun için test senaryoları karar noktasındaki kontrol akışlarını takip eder (örneğin, bir IF komutu için, biri doğru çıktı ve biri yanlış çıktı; bir CASE komutu için tüm olası çıktıların test senaryoları gerekli olacaktır).

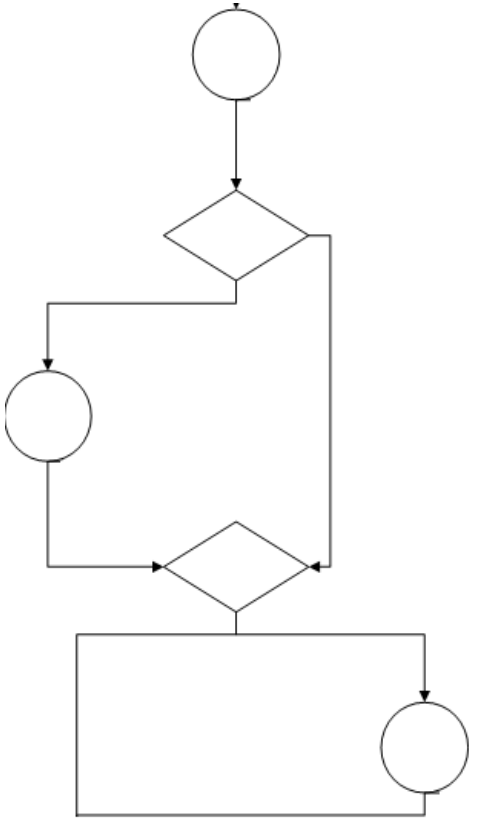
2^n

Structured Based (White Box)

Statement Coverage

Decision Coverage





1

Test Case

2

Test Case

SC <= DC

Question #23 (1 Point)

Which statement about the relationship between statement coverage and decision coverage is true?

a) 100% decision coverage also guarantees 100% statement coverage.
b) 100% statement coverage also guarantees 100% decision coverage.
c) 50% decision coverage also guarantees 50% statement coverage.
d) Decision coverage can never reach 100%.

2^n

Tecrübeye Dayalı Test Teknikleri

Hata Tahmini

- Tecrübeli kişi
- Sözel

Checklist Bazlı

- Deneyime göre hazırlanmıştır
- Yazılıdır

Atak

- Ataklar ile sistem hata tespiti

Keşif Testi

- Önce testi koş
- Sonra testi yaz
- Süre 1 gün

2^n

Statik Test Tasarım Teknikleri

Test Otomasyon

Ne Zaman Otomatize Edilir

- Ne için yapılacak
- Periyot
- Olgunluk & Bakım
- Teknoloji

Ne Şekilde Otomatize Edilir

- Unit /Component / Integration / System Test için kullanılan toollar
- Regresyon testini simüle etmek
- Son Kullanıcıyı Simüle etmek

Questions

SIEMENS

Proje

1)Software Requirement Specification (SRS) Dokümanı hazırlanacak

2)SRS dokümanı ile yapay zeka destekli bir uygulama yazılacak

3)SRS dokümanı ve yazılan uygulamadan test senaryoları çıkarılacak (Prompt kullanmadan)

4)SRS dokümanı ve yazılan uygulamadan test senaryoları çıkarılacak (Prompt kullanarak)

- Ana test senaryolar (Fonksiyonel gereksinimler)
- Alternatif test senaryolar (Fonksiyonel olmayan gereksinimler)
- Risk analizi yapılarak mevcut risk listesi çıkarılacak
- Risk listesine göre alternatif test senaryolar çıkarılacak

5)Bütün gereksinimlerin testlerle eşleştirildiği kontrol edilecek (%100 gereksinim kapsamı - Gerekirse prompt düzenlenip tekrar 4.madde koşulsun)

6)Bütün risklere karşılık gelen test senaryolarının yazıldığı kontrol edilecek (%100 risk indirgeme – Gerekirse prompt düzenlenip tekrar 4.madde koşulsun)

7)Yazılan testlere göre olası hata listesi olşuturulması

References

https://istqb.org/

https://www.turkishtestingboard.org/files/ISTQB_CTFL_Syllabus-v40-TR.pdf

https://www.gasq.org/files/content/gasq/downloads/certification/ISTQB/Foundation%20Level/ISTQB_CTFL_Syllabus-v4.0%20.pdf

https://www.turkishtestingboard.org/files/exams/FL-orneksinav-A-ceviri.pdf