

## Sayfa 14

**31-37 Şirket Hakkında Sorulacak Sorular**

Nerede çalışabileceğinizi biraz öğrenmeye ne dersiniz? Çünkü iş, günlük yapılacaklar listenizden ibaret değildir.

31. Şirketin kuruluşunu okudum, ama bana daha fazla bilgi verebilir misiniz ...?
32. Bölümünüzde kıyafet yönetmeliği var mı?
33. Önümüzdeki birkaç yıl içinde bu şirketi nerede görüyorsunuz?
34. Şirketin uzaktan çalışmaya ilişkin politikaları nelerdir?
35. Yeni ürünleriniz veya büyüme planlarınız hakkında bana ne söyleyebilirsiniz?
36. Şirketin odaklandığı mevcut hedefler nelerdir ve bu ekip bu hedeflere ulaşmayı desteklemek için nasıl çalışıyor?
37. Şirketin geleceği konusunda sizi en çok heyecanlandıran şey nedir?

**Takım Hakkında Sorulacak 38-44 Soru**

Her gün birlikte çalıştığınız insanlar iş hayatınızı gerçekten yapabilir veya bozabilir. Ortaya çıkarmak için bazı sorular sorun sizin için doğru ekip olup olmadığı.

38. Bana birlikte çalışacağım takımdan bahseder misiniz?
39. En yakın kiminle çalışacağım?
40. Doğrudan kime rapor edeceğim?
41. Bana doğrudan raporlarımdan bahseder misiniz? Güçlü yönleri ve takımın en büyük zorlukları nelerdir?
42. Önümüzdeki altı ay içinde bu departmanda daha fazla kişiyi işe almayı bekliyor musunuz?
43. Başka hangi departmanlar bununla en yakın şekilde çalışıyor?
44. Bu departmandaki ortak kariyer yolları nelerdir?

**Kültür Hakkında Sorulacak 45-54 Soru**

Ofis düğmeli muhafazakar mı yoksa pantolonunun yanından uçup giden bir yer mi? İnce olanı öğrenin, ama oh-so-şirket kültürünün önemli yönleri.

45. Şirket ve ekip kültürü nasıldır?
46. Buradaki çalışma ortamını nasıl tanımlarsınız - çalışma tipik olarak işbirlikçi mi yoksa daha bağımsız mı?
47. Bana birlikte yaptığınız son takım etkinliğinden bahseder misiniz?
48. Resmi bir misyon beyanı veya şirket değerleri var mı? (Not: Bunun Google'a uygun olmadığından emin olun!)
49. En sevdiğiniz ofis geleneğiniz nedir?
50. Siz ve ekip öğle yemeğinde genellikle ne yaparsınız?
51. Ekipte ofis dışında takılan var mı?
52. Hiç başka şirket veya departmanlarla ortak etkinlikler yapıyor musunuz?
53. Burada çalışmanın, çalıştığınız herhangi bir yerden farklı olan nedir?
54. Siz katıldığınızdan beri şirket nasıl değişti?

**Sonraki Adımlar Hakkında Sorulacak 55-58 Soru**

Ayrılmadan önce, görüşmecinin ihtiyaç duyduğu tüm bilgilere sahip olduğundan ve sonraki adımlarda net olduğunuzdan emin olun. bu soruları sormak.

55. Geçmişimin bu role uygun olduğu konusunda sizi endişelendiren herhangi bir şey var mı?
56. Görüşme sürecinde sonraki adımlar nelerdir?
57. Size sağlayabileceğim, yardımcı olabilecek başka bir şey var mı?
58. Sizin için son soruları cevaplayabilir miyim?

## Sayfa 15

**SDLC ve AGILE****1. Yazılım Testi nedir?**

- İşlevsel ve otomasyon araçlarını kullanarak yazılım hatalarını bulma amacıyla bir program veya uygulamayı yürütme süreci
- Bir yazılım programını / uygulamasını doğrulama / doğrulama süreci
- Test uzmanları, geçmek için test değil, yaklaşımı kırmak için test etmelidir.

**2. Yazılım Gereksinimleri Belirtimi nedir?**

- Yazılım gereksinimleri belirtimi, müşteri ile tedarikçi arasında bir sözleşme görevi gören bir belgedir.
- Bu SRS, son kullanıcının o uygulamayla ilgili tüm gereksinimlerini içerir. SRS iletişim olarak kullanılabilir müşteri ve tedarikçi arasındaki ortam.
- Geliştirici ve test uzmanı, uygulamayı SRS'de yazılı gereksinimlere göre hazırlar ve inceler.
- Belgelenen SRS, müşteri için tüm gereksinimler dikkate alınarak İş Analisti tarafından hazırlanır.

### 3. Yazılım Geliştirme Yaşam Döngüsü (SDLC) - SDLC nedir?

- SDLC, yazılım veya uygulama **oluşturma** aşamalarını tanımlar .
  - o Proje Planlama
  - o Gereksinim Toplama (Projeyi planlamak için kullanılan bilgilerin toplanması, Risklerin belirlenmesi)
  - o Tasarım (Uygulama nasıl inşa edilecek)
  - o Kodlama (geliştirme) (Gereksinimlere göre geliştiriciler uygulamayı yazacaktır)
  - o **Test**
  - o Üretim (dağıtım) (Ürünün piyasaya sürülmesi)
  - o Bakım (Ürünün stabil olduğundan emin olmak, hatalarla ilgili müşteri raporlarına bakmak ve onu düzeltmek)

### 4. Yazılım Testi Yaşam Döngüsü (STLC) - STLC nedir?

- STLC, yazılım veya uygulamanın **test** edilmesindeki aşamaları tanımlar . STLC sürecinde farklı faaliyetler gerçekleştirilmektedir.
  - ürünün kalitesini artırmak.
  - o Gereksinim analizi
  - o Test Planlama
  - o Test Tasarımı
  - o Test Ortamı Kurulumu
  - o Test Yürütme
  - o Test Raporlama

### 5. STLC ve SDLC arasındaki fark nedir?

- STLC, SDLC'nin bir parçasıdır. STLC'nin, SDLC setinin bir alt kümesi olduğu söylenebilir.
- STLC, yazılım veya ürün kalitesinin sağladığı test aşamasıyla sınırlıdır. SDLC'nin eksiksiz ve hayati bir rolü vardır. bir yazılımın veya ürünün geliştirilmesi.
- Bununla birlikte, STLC, SDLC'nin çok önemli bir aşamasıdır ve nihai ürün veya yazılım, STLC sürecinden geçmek.
- STLC aynı zamanda yayın sonrası / güncelleme döngüsünün bir parçasıdır, bilinen kusurların giderildiği SDLC'nin bakım safhası veya yazılıma yeni işlevler eklendi.

### 6. Gereklik nedir?

- Gereksinimler, kullanıcıların yazılım veya ürünle ilgili beklentilerini iletir.
  - Müşteriden gereksinimleri toplama, analiz etme ve belgeleme süreci, gereksinim mühendisliği olarak bilinir.
  - Gereksinim mühendisliğinin amacı, sofistike ve açıklayıcı SRS Sistem Gereksinimlerini geliştirmek ve sürdürmektir.
- Şartname 'Dokümanı

- 15 -

## Sayfa 16

### 7. Gereksinim nereden geliyor?

- Müşteriler uygulama için şartlar verir
- Son kullanıcılarla konuşun → bu uygulamayı en çok kullanacak kişiyle
- Ortaklarla konuşun -
- Etki Alanı Uzmanlarıyla Konuşun - bu uygulamayı daha önce benzer şekilde geliştirmiş olan kodlayıcılar ve geliştiriciler veya bir uzman inşa edilmekte olan ürünün türü
- Sektör Analistleri ve rakipler hakkında bilgiler

### 8. Test ne zaman başlar?

- Test, gereksinimleri test etmekle başlar (en olası cevap gibi görünen kodlama aşamasından sonra değil.)
- İlk etapta gereksinimin doğru olduğundan emin olmalıyız. Yanlış gereksinimle hata oluşturmak imkansızdır ücretsiz uygulama.

### 9. Gereksinimin iyi mi kötü mü olduğu nasıl anlaşılır?

- Gereksinim (SMART) olmalıdır
  - o **S**pecific → Kullanıcı giriş gerekir. Geçerli kullanıcı adı ve şifreye sahip yetkili kullanıcı giriş yapabilmelidir
  - o **M** kolay → Kullanıcı çok hızlı giriş yapabilmelidir (giriş düğmesine tıkladıktan 2 saniye sonra).
  - o **bir** tainable
  - o **R** ealistik
  - o **T** 2 saniyede (çok hızlı makbuz indirmek için estabale → Kullanıcı gerektiği mümkün

### 10. Neden test ediyoruz?

- Hatasız uygulama oluşturmak için.
- Son kullanıcıyı ve müşteriyi memnun etmek için.
- Daha fazla gelir elde etmek için harika ürünler oluşturmak.
- Test etmeyi seviyorum ve test etmek benim tutkum.

**11. Test uzmanının ana sorumluluğu nedir?**

- Hatayı olabildiğince erken bulmak için. Hatanın çoğunun düzeltildiğinden emin olun.
- Hatasız ve kullanıcı dostu bir uygulama sunarak son kullanıcıyı ve müşteriye satın almak için.

**12. Test görevindeki bir test uzmanının veya Yazılım Geliştirme Mühendisliğinin iş sorumluluğu nedir?**

- Test otomasyonunu yazın ve web veya mobil gibi çeşitli platformlar için aynısını kurun.
- Hata raporunu yönetme ve işleme.
- Geliştirici ve müşteri arasında uygun iletişim kanalını sürdürmek.
- Test senaryolarının hazırlanması ve teslim edilmesi.

**13.% 100 test mümkün müdür?**

- Hayal bile edemeyeceğimiz sınırsız sayıda senaryo olduğu için uygulamayı% 100 test edemiyoruz.
- Yazılım testi, mümkün olduğunca test edebileceğimiz **işlevselliğin önceliğine** dayalı risk temelli bir faaliyetir .
- % 100 test mümkün olmasa da,% 100 müşteri memnuniyetinin kesinlikle mümkün olduğuna inanıyorum.

**14. Pozitif test nedir? Mutlu Yol testi?**

- Uygulamanın geçerli girdilerle test edilmesi. " **Mutlu Yol** " Testi olarak da adlandırılır .
- Örn. Geçerli bir kullanıcı adı ve parola ile oturum açarsanız, bu olumlu bir testtir.

- 16 -

**Sayfa 17****15. Test hiyerarşisi nedir?**

- **Birim testi** → Geliştiriciler, geliştirme sırasında her modülü veya kod bloğunu test eder.
- **Bileşen Testi** → Bileşen, kendi kendine çalışabilen bağımsız bir işlevdir. Örn. Amazon Alıcı İşlevselliği, Satıcı İşlevselliği, Prime Video İşlevselliği.
- **Entegrasyon Testi** → Tüm İşlevleri birleştirin. Bunları entegre ettiğimde, yine de tüm fonksiyonları kullanabilir miyim? Yapmak elbette hepsi hala çalışıyor.
- **Sistem Testi** → Uçtan Uca test. Baştan sona her şeyi test edin.
- **Kabul Testi** → Bir UAT (Kullanıcı Kabul Testi) Ekibi veya İş Analisti de Kabul Testi yapabilir.
- Test tamamlandıktan sonra, QA ekiplerini onaylayabilmeleri için başka bir ekibin kabul testi yapması gerekir.
- test başarılı oldu ve ürünü müşteri için hazır hale getirdi.

**16. 508 Uygunluk testi nedir?**

- (Mülakatta biri 508 testinin ne olduğunu sorarsa, sadece ne olduğunu söyleyin. 508'de 5-10 yıllık tecrübem olduğunu söylemeyin. Uyum testi.)
- **Devlet web siteleri için bir gerekliliktir.**
- Devlet tarafından ve devlet için kullanılan tüm web siteleri. **Engellilerin kullanabileceğinden emin olmalıdır.**
- Örnek: healthcare.gov için Uyumluluk yöneticisi var ve 508 Uyumluluk yapan özel bir QA ekibi var
- web sitesinin engelli kullanıcılar için 508 uyumlu olduğundan emin olmak için test.

**17. Risk temelli test nedir?**

- Risk Temelli test, bir ürünün işlevlerinin, çıktıların önceliğine göre test edilmesi olarak tanımlanır.
- Risk Temelli test, bir ürünün ticari etkisi ve olasılığı olan önemli özelliklerinin test edilmesini içerir.
- bu özelliklerin başarısızlığı çok yüksektir.
- Bir ürünün tüm işlevlerinin önceliği, iş gereksinimine ve ardından yüksek önceliğe göre belirlenir.
- önce işlevsellikler sonra orta ve sonra düşük öncelikli işlevler test edilecektir.
- Risk Bazları testi, bir ürünün tüm işlevlerini test etmek için yeterli zaman olmadığında yapılacaktır.
- % 100 test yapılamadığından risk analizi yapmalıyız. Analize dayanarak testimize öncelik vermeliyiz
- önce yüksek riskli alanı test edin. Örneğin:
  - o En kritik işlevler
  - o En sık kullanılan işlevler
  - o En karmaşık işlevler vb ...

**18. Bu regresyon paketini inşa etmek ne kadar sürdü?**

- 3 yıl sürdü; 2 test cihazı 1 manuel test cihazı + 1 otomasyon test cihazı
- koştığımızda:
  - o yayınlanmadan önce
  - o büyük hata düzeltilmesinden sonra
  - o büyük yeni işlevlerden sonra
- Test senaryolarını nerede tuttuğumuz ve takım olarak tek bir sprintte birden fazla uygulanacak kararı nerede aldığımız
- bazı senaryoları test edersiniz.

**19. Regresyon paketini çalıştırırken bize bir zorluk söyler misiniz?**

- Arızalar. Çünkü regresyon paketi çok uzun zaman önce geliştirildi ve neyin değiştiğini bilmiyorsunuz. Bir düğmesi değişmiş olabilir.

- 17 -

## Sayfa 18

### 20. Kaç çevreniz var?

- Geliştirme Ortamı
  - o Birim testi
  - o Test ortamından daha az kararlı
- Test ortamı
  - o Manuel test burada gerçekleşir
  - o Üretim ortamını aynen kopyalar
  - o Değişiklikler aralıklarla dağıtılır
  - o Otomatik **duman testleri** burada yapılır
    - Uygulamanın diğer önemli testleri gerçekleştirmek için yeterince kararlı olup olmadığından emin olmak için test ortamına karşı çalışır faaliyetler.
    - Değişiklikler Test ortamına her dağıtıldığında çalışır
    - Geliştirici ortamında çalıştırılabilir
  - o Otomasyon testleri burada yapılır
  - o Otomatik Entegrasyon testleri burada çalıştırılır
- Üretim Öncesi Ortam
  - o UAT ortamı
  - o Demo burada gerçekleşir
  - o yük / performans testi burada gerçekleşir
  - o Değişiklikler büyük aralıklarla dağıtılır
  - o Otomatik ana **regresyon testleri** burada (yayınlanmadan önce)
    - UAT ortamına karşı çalışır
    - Yeni değişikliklerin herhangi bir kusurla sonuçlanıp sonuçlanmadığını öğrenmek için
    - Büyük hata düzeltmelerinden ve her sürümden sonra çalışır
    - Bu test, test planında kararlaştırılır
  - o Çok kararlı
- Üretim ortamı

### 21. Fonksiyonel test nedir?

- Fonksiyonel test ekibi manuel test cihazı olarak da adlandırılabilir, otomasyon ekibi tarafından da yapılabilir (otomasyon fonksiyonel test yapmak). Kara kutu testi veya manuel test cihazlarına benzer. Sadece uygulamanın belirli işlevselliğini test etmek. Örn. Yapabilmek Kullanıcı Girişi? Kullanıcı çıkış yapabilir mi? Uygulama görünümünü ve hissini test etmemek.

### 22. Fonksiyonel olmayan test nedir?

- Performans testi, Güvenlik testi, Örn. 2000 kullanıcı aynı anda uygulamaya giriş yapabilir mi? Kullanıcı şuraya taşınabilir mi sonraki sayfa 1 saniye içinde mi?

### 23. Birim testi nedir? Hiç birim testi yaptınız mı?

- Beyaz kutu testinin bir parçasıdır. Geliştirme ortamından kodu dağıtmadan önce geliştiriciler tarafından yapılır QA ortamına.
- Geliştiriciler tarafından yapıldığı için henüz birim testi yapmadım. Ama bence öğrenebilirim ve gerekirse yapabilirim.

### 24. Bileşen testi nedir?

- Uygulamanın her bir bileşenini ayrı ayrı test etmek. Uygulamada tek bileşenli olabilir. Bir bileşende bağımsız işlevsellik. Örn. amazon.com'da Satıcı işlevi tek bileşen olabilir. Alıcı başka biri olabilir bileşen. Ayrıca Amazon ana videoları başka bir bileşen olabilir.

- 18 -

## Sayfa 19

### 25. Duman Testi →

- **TEST YAPMA sırası:** Kod → Birim Testi → Entegrasyon Testi → Sağlık Testi → **Duman Testi** → Fonksiyonel Test
- Projemizde; oturum açma, kullanıcıyı görüntüleme, kullanıcı ayrıntıları sayfası, yeni kullanıcı oluşturma ve görev oluşturma
- Bu beş modülde, geliştirici ilk olarak duman testini gerçekleştirecektir.
  - gibi modüller; Kullanıcı, geçerli oturum açma kimlik bilgileriyle oturum açabilir veya kapatabilir, oturum açtıktan sonra yeni kullanıcı oluşturulabilir veya oluşturulamaz, kullanıcı

oluşturuldu, görüntülendi ya da değil vb.

#### 26. Regresyon testinin hangi bölümü otomatikleştirilmelidir?

- Kararlı olan testler
- Sık sık tekrarlanıyor
- Basittir ve test cihazı girişi gerektirmez, otomasyon için iyi adaylardır

#### 27. Regresyon testlerinizin etkili olmasını nasıl sağlıyorsunuz?

- Regresyon testleri kusurların yakalanmasına izin verecek kadar geniş ve detaylı olmalıdır. Yinelenen testi de ortadan kaldırırsınız durumlarda, test senaryolarını ve otomatik testleri mümkün olduğu kadar birleştirin.

#### 28. Yazılımda bir dizi kritik hata düzeltilmiştir. Tüm hatalar, raporlarla ilgili tek bir modüldedir. Test yöneticisi

- sadece rapor modülünde regresyon testi yapmaya karar verir.
- Regresyon testi diğer modüllerde de yapılmalıdır çünkü bir modülü sabitlemek diğer modülleri etkileyebilir.

#### 29. Girdi ve çıktı kapsamına ulaşmak için hangi teknik kullanılabilir?

- İnsan girdisine, bir sisteme arabirimler yoluyla girdiye veya entegrasyon testinde arabirim parametrelerine uygulanabilir.

#### 30. Regresyonunuzu nasıl yürütürsünüz? Ne sıklıkla, kaç sanal makine, kaç gün, kaç test?

- Gerileme her sürümden önce planlanmıştır ve yılda 4 kez yayınliyoruz (2 İlkbahar sürümü ve 2 Sonbahar sürümü).
- Gerileme, büyük bir hata düzeltilmesi olduğunda da gerçekleşir.
- Yaklaşık 500 özellik dosyası ve 1300 senaryo.
- Regresyon testleri jenkins tarafından başlatılır. Testler jenkins sunucusunda (VM) yürütülür. Linux sunucum RedHat.
- En son çalıştırma 12 saatten fazla görünüyor.
- **Diğer bir cevap ise ;** Bir dizi regresyon testi oluşturdum. Regresyon etiketli özellik dosyalarıdır. Ve bir isim var regresyon testlerini başlatan jenkins. Testi tetiklemek için maven komutunu kullanır. Maven komutu şunları içerir: bu etiket adı: mvn test -D cucumber.options = "- etiketler @Regression".
- Yürütmenin sonunda, jenkins ayrıntılı test adımları ve ekran görüntüleriyle HTML raporu oluşturur.

#### 31. Kara kutu testi nedir? Farklı kara kutu test teknikleri nelerdir?

- Kara kutu testi, yazılımı iç yapısını bilmeden test etmek için kullanılan yazılım test yöntemidir. kodun veya programın.
- Bu test genellikle bir uygulamanın işlevselliğini kontrol etmek için yapılır. Farklı kara kutu test teknikleri;
  - o Eşdeğerlik Bölümleme
  - o Sınır değer analizi
  - o Neden efekt grafiği oluşturma

#### 32. Eşdeğerlik bölümleme testi nedir?

- Eşdeğerlik bölümleme testi, uygulama girdi test verilerini her birine bölen bir yazılım test tekniğidir. Test senaryolarının türetilebileceği eşdeğer verilerin en az bir kez bölünmesi. Bu test yöntemi ile zamanı azaltır yazılım testi için gereklidir.
- Örnek: Bir not hesaplama sistemini test ederken, bir test zamanı 90'dan 100'e kadar olan tüm puanların bir not vereceğini belirler. A, ancak 90'ın altındaki puanlar olmayacak.
- Girdi ve çıktı kapsamına ulaşmak için hangi teknik kullanılabilir? İnsan girişine, arayüzler üzerinden giriş uygulanabilir bir sisteme veya entegrasyon testindeki arayüz parametrelerine.

- 19 -

## Sayfa 20

#### 33. Sınır değeri testi nedir?

- Girdi ve çıktı denklik sınıflarının kenarlarında, altında ve üstünde sınır koşullarını test edin.
- Örneğin, maksimum 1000 \$ ve minimum 100 \$ çekebileceğiniz bir banka başvurusu, yani sınırda değer testi, ortada vurmak yerine yalnızca kesin sınırları test ediyoruz. Bu, maksimumun üzerinde test ettiğimiz anlamına gelir limit ve minimum limitin altında.
- Örneğin, kredi kartının: Etkinleştirilme tarihi alt sınırdır. 10/2019 son kullanma tarihi üst sınırdır. 0 \$ daha düşük harcama sınırı sınırı. 25.000 \$ harcama limiti için üst sınırdır.

#### 34. Sınır değer analizi neden iyi test senaryoları sağlar?

- Çünkü değerler aralığının 'kenarlarına' yakın farklı durumların programlanması sırasında sıklıkla hatalar yapılır.

#### 35. Neden karar tabloları kullanıyoruz?

- Eşdeğerlik bölümleme ve sınır değer analizi teknikleri genellikle belirli durumlara veya girdilere uygulanır. Ancak, farklı girdi kombinasyonları farklı eylemlerin gerçekleştirilmesine neden oluyorsa, bunu kullanarak göstermek daha zor olabilir. eşdeğerlik bölümleme ve sınır değeri analizi, daha çok kullanıcı arayüzüne odaklanma eğilimindedir.
- Diğer iki spesifikasyona dayalı teknik, karar tabloları ve durum geçiş testi daha çok iş üzerine odaklanmıştır mantık veya iş kuralları. Karar tablosu, şeylerin kombinasyonlarıyla (örneğin girdiler) başa çıkmanın iyi bir yoludur.
- Bu teknik bazen '**neden-sonuç**' tablosu olarak da anılır . Bunun nedeni, ilişkili bir mantık olmasıdır. Bazen karar tablosunun türetilmesine yardımcı olmak için kullanılan '**neden-sonuç grafiği**' olarak adlandırılan diyagram oluşturma tekniği .

#### 36. Beyaz kutu testi nedir ve beyaz kutu testi türlerini listeler?

- Beyaz kutu test tekniği, iç yapının analizine dayalı olarak test senaryolarının seçimini içerir (Kod kapsamı,

- bir bileşenin veya sistemin şube kapsamı, yol kapsamı, koşul kapsamı vb.).
- Kod Tabanlı test veya Yapısal test olarak da bilinir. Farklı beyaz kutu testi türleri
  - o Bildirim Kapsamı o Karar Kapsamı

### 37. Beyaz kutu testinde neyi doğrularsınız?

- Koddaki güvenlik açıklarını doğrulayın
- Koddaki eksik veya bozuk yolları doğrulayın
- Belge özelliklerine göre yapının akışını doğrulayın
- Beklenen çıktıları doğrulayın
- Uygulamanın tam işlevselliğini kontrol etmek için koddaki tüm koşullu döngüleri doğrulayın
- Satır kodlamasını doğrulayın ve % 100 testi kapsayın

### 38. Gri Kutu Testi nedir?

- Gri kutu testi, kara kutu ve beyaz kutu testinin karmasıdır.
- Gri kutu testinde, test mühendisi bileşenin kodlama bölümü bilgisine sahiptir ve test senaryoları veya testi tasarlar sistem bilgisine dayalı veriler.
- Bu test cihazında kod bilgisi vardır, ancak bu, beyaz kutu testi bilgisinden daha azdır. Bu bilgiye dayanarak test senaryoları tasarlanır ve test edilen yazılım uygulaması bir kara kutu gibi davranır ve test cihazı uygulamayı test eder. dışarıda.

### 39. Statik ve dinamik test arasındaki fark nedir?

- **Statik test** : **Statik test** sırasında kod yürütülmez ve yazılım dokümantasyonu kullanılarak gerçekleştirilir.
- **Dinamik test** : Bu testi gerçekleştirmek için kodun çalıştırılabilir bir biçimde olması gerekir.

### 40. Bakım testi nedir?

- Mevcut yazılımın değiştirilmesi, taşınması veya kullanımdan kaldırılmasıyla tetiklenir.

- 20 -

## Sayfa 21

### 41. Entegrasyon Testi nedir?

- Entegrasyon testi kara kutu testidir. Entegrasyon testi, birimlerin belirli bir görevi tamamlamak için birlikte çalışın.
- Entegrasyon testinin amacı, uygulamanın farklı bileşenlerinin birbiriyle etkileşime girdiğini doğrulamaktır. Test senaryoları, bileşenler arasındaki arayüzlerin kullanılması amacıyla geliştirilmiştir.
- Gerçek sonuçlar ve beklenen sonuçlar aynı olduğunda entegrasyon testi tamamlanmış kabul edilir. Entegrasyon testi yapılır birim testinden sonra. Entegrasyon testi yapmak için başlıca üç yaklaşım vardır:
  - o **Yukarıdan Aşağıya Yaklaşım** → bileşenleri yukarıdan aşağıya entegre ederek test eder.
  - o **Aşağıdan yukarıya yaklaşım** → Kontrol akışının altından üst düzey bileşenlere doğru gerçekleşir
  - o **Büyük patlama yaklaşımı** → **Bunda** , tam bir sistem oluşturmak için farklı modül bir araya getirilir ve ardından test yapılır. üzerinde gerçekleştirildi.

### 42. Ölçeklenebilirlik Testi nedir?

- Ölçeklenebilirlik testi, işlevsellik ve performans yeteneklerini geliştirmek ve iyileştirmek için yapılan testtir. uygulama. Böylelikle uygulama, son kullanıcıların gereksinimlerini karşılayabilir.
- Ölçeklenebilirlik ölçümleri, uygulama performansının yük ve stres koşullarında değerlendirilmesi yapılarak yapılır. Şimdi bu değerlendirmeye bağlı olarak, uygulamanın yeteneklerini iyileştiriyor ve geliştiriyoruz.

### 43. Depolama Testi nedir?

- Depolama Testinde, uygulamanın, verileri DB'ye depolamaktan sorumlu olan işlevlerini test ederiz.
- GUI'de veya ön uçta son kullanıcı tarafından girilen veriler, veritabanında depolanan verilerle aynıdır.
- Depolama testi, uygulamanın ön ucundan alınan verilerin doğru yerde ve veri tabanında doğru şekilde.

### 44. Stres Testi nedir?

- Stres testi, yazılımı, üzerindeki baskıyı arttırsak uygulamanın çökmediğini kontrol etmek amacıyla test eder. uygulama üzerinde çalışan kullanıcı sayısını artırarak uygulama.
- Uygulama tarafından halledilemeyen birçok işlemi ateşleyen uygulama üzerine de stres uygulayabiliriz.
- Uygulama yeteneklerini değerlendiren uygulama üzerinde stres testi yapıyoruz, uygulama yeteneklerini belirtilen sınırlar dahilinde veya ötesinde değerlendiriyoruz. belirlemek için gereksinimler.
- Genel olarak bu, çok yüksek düzeyde yük ve stres koşullarında gerçekleştirilen bir tür performans testidir.

### 45. Test Donanımı nedir?

- Test koşturma takımı, uygulamayı farklı testlerde çalıştırarak test etmek için gerekli olan bir yazılım ve test verileri koleksiyonudur. stres, yük, veri odaklı ve davranışını ve çıktılarını izleme gibi koşullar. Test Demeti iki ana bölümden oluşur:
  - o Test yürütme motoru
  - o Test komut dosyası deposunu test edin
- Otomasyon testi, testlerin yürütülmesini kontrol etmek ve gerçek sonuçları beklenen sonuçlarla karşılaştırmak için bir aracın kullanılmasıdır. Sonuçlar. Aynı zamanda test ön koşullarının oluşturulmasını da içerir.

**46. Test kapsamı nedir?**

- Test kapsamı, sahip olduğumuz test senaryosu ve bu test senaryolarının hangi fonksiyonel alanı kapsadığı anlamına gelir.

**47. V-Modeli nedir?**

- Test faaliyetlerinin yazılım geliştirme aşamalarıyla nasıl bütünleştiğini gösteren bir yazılım geliştirme modeli.

- 21 -

---

**Sayfa 22****48. Aşağıdakilerden hangisi, test yakalama ve tekrar oynatma olanakları sağlayan test araçlarının kullanımından en çok yararlanacaktır?**

- Gerileme testi
- Entegrasyon testi
- Sistem testi
- Kullanıcı Kabul Testi

**49. Kabul testi nedir?**

- Kabul testi QA testinden sonra yapılacaktır. Mevcut projemde UAT ekibi tarafından yapılıyor. UAT ekibinden sonra kabul testini gerçekleştirerek kod üretime gidecektir.
  - o Geliştirme ortamı (geliştiricilerin kod yazdığı ve birim testi gerçekleştirdiği)
  - o QA ortamı (uygulamayı test ettiğimiz yer.)
  - o UAT ortamı (kod QA ortamı test edildikten sonra UAT ortamına konuşturılacaktır. UAT testi ekibi, iş gereksinimlerine uyduğundan emin olmak için testler gerçekleştirecektir. Aynı zamanda evreleme ortamı olarak da adlandırılır.
  - o Üretim ortamı (son kullanıcının gerçek uygulamayı görebildiği zamandır)

**50. UAT (Kullanıcı Kabul Testi) ile Sistem testi arasındaki fark nedir?**

- **Sistem Testi** : Sistem testi, sistem bir bütün olarak teste girdiğinde kusurları bulmaktır, aynı zamanda **son** olarak da bilinir.
  - testi bitirmek için** . Bu tür testlerde uygulama baştan sona kadar devam eder.
- **UAT** : Kullanıcı Kabul Testi (UAT), bir ürünü belirleyen bir dizi özel testten geçirmeyi içerir.
  - ürünün kullanıcılarının ihtiyaçlarını karşılayıp karşılamayacağı.

**51. Sürekli entegrasyon nedir?**

- Geliştiriciler, uygulamada yeni kod değişiklikleri yaparken kodları sisteme teslim edebilir ve teslim alabilir.
- Bir geliştirici sisteme yeni bir kod eklediğinde, **Sürekli tümleştirme (CI) sunucusu olarak** adlandırılan bir sunucu vardır .
- **CI sunucusu** sürekli olarak yeni kod arıyor. Yeni kod uygulamaya eklendiğinde, CI sunucusu,
  - kodun teslim edildiğini hemen anlıyor . *(bununla entegre bir araç var, belki Jenkins veya başka bir şey).*
- Bu araç, uygulamanın temel işlevselliğini kontrol etmek için otomatik duman testini başlatacaktır. Sonra hava diyecek
  - bu kod, uygulamayı olumsuz yönde etkiledi veya etkilemedi.

**52. Kod üretim ortamına nasıl dağıtılır?**

- Yerelden
  - o pull ve push kullanarak kodu Git'e kontrol edin (benim şirketimde bu SVN'dir)
  - o Birim testleri çalıştırın
  - o Değişiklikleri sunucuya dağıtın
    - Geliştirici kodu her kontrol ettiğinde jenkins tarafından otomatik olarak yapılır.
    - Geliştirme ortamına dağıtım değişikliklerini geçtikten sonra
- Geliştirme ortamından
  - o Jenkins tarafından yapılan Test ortamındaki değişiklikleri dağıtın.
  - o Planlanabilir veya manuel olarak tetiklenebilir
- Testten → Jenkins tarafından yapılan değişiklikleri dağıtın
- Ön prodüksiyondan

**53. Çevik Çerçeve?**

- **Rol** : PO, SM, Takım
- **Törenler** : - Baskı Planlama, Günlük Scrum, Sprint İncelemesi, Sprint Retro, Bakım Oturumu
- **Eserleri** : Ürün birikim, - Sprint birikim, -Burnout grafik

- 22 -

---

**Sayfa 23****54. Çevik nedir?**

- Çevik, şelale metodolojisine alternatif olan **yinelemeli** ürün **geliştirme** metodolojisidir.
- Scrum: Takım, bir sonraki sprint için yapılacak iş miktarını planlar
- Kanban: Sprint planlaması yok, hikayeler olduğu gibi toplanıyor, ancak diğer her şeye hala sahipsiniz

##### 55. Neden Agile'a ihtiyacımız var? Şelale ve Çevik mi?

- Şelale metodolojilerinin aşağıdaki dezavantajları olduğundan;
  - o Gereksinim, belge imzalandıktan sonra değiştirilemez veya değiştirilmesi zor olamaz.
  - o Şelalede bir aşamayı tamamlamadan bir sonraki aşamaya geçemezsiniz. Örneğin, kodlama aşamasından önce tamamlanan test başlatılamaz.
  - o Müşteri, geliştirme yaşam döngüsünün çok ileri aşamalarına kadar ne elde edeceğini göremez.
  - o Üretime gitmek daha uzun sürüyor. Ürün pazara girdiğinde çoktan modası geçmiş olabilir.
- Agile'ın aşağıdaki avantajları vardır:
  - o Değişiklik memnuniyetle karşılanmaktadır. Örneğin, sprint demosundan sonra müşteri bir şeyi beğenmezse, geri bildirimde bulunun ve ürünü geliştirin. Gereksinim değişikliği tamam.
  - o Yinelemeli geliştirme süreci olduğundan, geliştirme ekibi işlevsellik geliştirebilir, geri bildirim alabilir ve bir sonraki yinelemeyi iyileştirir. Böylece ürün sürekli olarak gelişecektir.
  - o Scrum master yardımıyla atıklar çevik olarak ortadan kaldırılır. Örneğin, bloke olursam, beklemek ve zamanımı boşa harcamak zorunda kalmam.
  - o Ekip üyeleri birbirleriyle verimli bir şekilde iletişim kurduğundan, yinelenen çabaları önleyerek daha üretken olabiliriz.
  - o Waterfall, C # .NET gibi araçları ve platformu vurgular, ancak Agile insanları vurgular. En iyi araca sahip olabilirsiniz ama son insanlar bu araçları kullanıyor. İlham alan insanların daha az ürüne sahip olsalar bile harika ürünler yapabileceklerine inanırım. para veya daha az kaynak.

##### 56. Önceki projelerinizde ne tür Agile metodolojisi kullandınız?

- Aşırı programlama (XP), Kanban ve Scrum'ı duydum. Ama sadece scrum ile çalıştım.

##### 57. Scrum, Çevik bir çerçevedir, değil mi? Diğer birkaç Agile çerçeveyi adlandırın.

- Evet, Scrum bir Çevik çerçevedir. Çok az Çevik çerçeve –Özellik Odaklı Geliştirme Test Güdümüdür
- Geliştirme, Kanban

##### 58. Scrum'daki farklı roller nelerdir? Scrum rolleri?

- **Ürün sahibi** , aslında projenin paydaşıdır.
  - o Ekip önünde proje gereksinimlerini temsil eder.
  - o Neyin inşa edileceğine dair bir vizyona sahip olmaktan ve detaylı vizyonunu takıma aktarmaktan sorumludur.
  - o Çevik bir scrum yazılım geliştirme projesinin başlangıç noktasıdır.
- **Scrum takımı** , belirli bir takımın başarılması için performans gösteren bireylerin kolektif katkısıyla oluşur.
  - o proje.
  - o Ekip, talep edilen ürünün zamanında teslimi için çalışmak zorundadır.
- **Scrum ustası** - Scrum ustası, scrum takımının olup olmadığını kontrol eden scrum takımının lideri ve koçudur.
  - o taahhüt edilen görevleri düzgün bir şekilde yürütmek.
  - o Ayrıca sprint hedefine ulaşabilmeleri için takımın verimliliğini ve üretkenliğini artırmaktan sorumludur.
  - o etkili bir şekilde.

##### 59. Bir scrum takımını nasıl tanımlarsınız?

- 5 rock yıldızını bir araya getirirseniz, bu onların bir ekip oldukları anlamına gelmez veya harika ürünler geliştirebilecekleri anlamına gelmez.
- Benim için ekip, aynı hedefi paylaşan, aynı yöne hareket eden, birbirine güvenen bir grup insandır.
  - o ve harika bir ürün oluşturmak için birbirleriyle etkili bir şekilde iletişim kuracak ve işbirliği yapacak. Yıldız olmamalı bireysel ama bir yıldız takımı.

## Sayfa 24

##### 60. Scrum Master'ın sorumlulukları nelerdir?

- İzleme ve izleme
- Gereksinimleri doğru anlamak
- Proje hedefine ulaşmak için çalışın
- Süreç kontrol ustası ve kaliteli usta
- Ekibi müfrezelerden koruyun
- Takımın performansını iyileştirmek
- Toplantıları yönetin ve sorunları çözün
- Çatışmaların ve engellerin çözümü
- İletişim ve raporlama

##### 61. Negatif test senaryosu nedir?

- Negatif test senaryoları, yıkıcı bir şekilde test etme fikrine dayanılarak oluşturulur. Örneğin, ne olacağını test etmek
- Uygulamaya uygun olmayan girişler girilirse. Yanlış giriş bilgisi

##### 62. "Scrum of Scrums" teriminden ne anlıyorsunuz?

- Halihazırda yedi ekibin üzerinde çalıştığı aktif bir proje varsayalım. Her takım kendi başına liderlik etmekten sorumludur.
- saldırı toplantısı. Ancak, farklı ekiplerle koordinasyon sağlamak ve iletişim kurmak için ayrı bir



saldırı toplantısı. Büyükkelçi olarak bilinen her ekipten kendisini temsil etmekten sorumlu bir ekip lideri vardır.

Scrums ekibi.

- Scrum takımları arasında koordinasyonu sağlamak için düzenlenen scrum toplantısı, scrums olarak bilinir.

### 63. Sevk edilebilir ürün / artım?

- Ürünün parçası yapıldı ve her sprintten ek işlevler almaya devam ediyor
- Artış, geliştirme ekibinin *Bitti Tanımına uygun olmalıdır*
  - o Ürün artımı teslim edildiğinde, "Bitti Tanımı" nı karşılaması gerekir
  - o Kabul kriterleri yerine getirildi
  - o Ürün sahibi kullanıcı hikayelerini kabul eder
- Artış PO tarafından kabul edilebilir olmalıdır

### 64. BurnDown Grafiği nedir?

- İşin tamamlanma oranının ve yapılması gereken işin ne kadar kaldığının grafik temsili

### 65. Doğrulama ve Doğrulama nedir?

- Doğrulama, testçiler ve geliştiriciler tarafından geliştirme sırasında gerçekleşir; yazılımı geliştirme aşamasında değerlendirme sürecidir aşama ve belirli bir uygulamanın ürününün belirtilen gereksinimleri karşılayıp karşılamadığına karar vermek.
- Test uzmanları tarafından doğrulama; geliştirme sürecinin sonunda yazılımın değerlendirilmesi ve olup olmadığının kontrol edilmesi sürecidir. müşteri gereksinimlerini karşılar.

### 66. Hazırın Tanımı Nedir?

- Kabul Kriterleri temizlendi / gözden geçirildi ve Puan / saat verildi

### 67. Otopark nedir?

- Çevik'te bu şu anlama gelir: Diğer insanlarla gerçekten alakalı olmayan bir sorununuz olduğunda karşılaştığınızda, devam etmemeliyiz. diğer insanların zamanını boşa harcadığımız için bu konuyu toplantıda tartışmak. <Hadi **park yeri** ögesi yapalım > anlamına gelir bu konuyla ilgilenen herkes görüşmeden sonra konuşabilir.

- 24 -

## Sayfa 25

### 68. Sprint iş akışı nedir?

- Bir hikaye nasıl yapılır ve yaşam döngüleri nasıl ilerler - bir şey engellendiğinde ne olur, vb.

### 69. Kullanıcı Hikayesi nedir?

- (Not: temelde, bir kullanıcı hikayesi sadece bir gerekliliktir) Kullanıcı hikayesi, kısa ve basit bir tanım minimum **sevk edilebilir** üründür.
- Normalde şu şekilde görünür: <**son kullanıcı**> olarak <**eylem**> yapmak istiyorum, böylece < **Fayda**> yapabilirim .
  - o Amazon kullanıcısi olarak giriş yapabilmeliyim, böylece çevrimiçi malzeme satın alabilirim

### 70. "Sevk edilebilir" dediniz, bununla ne demek istiyorsunuz?

- Bir kullanıcı olarak kullanıcı adımı kullanıcı adı alanına koymak istediğimi gerçekten söyleyemezsin.
- Böylece kullanıcı adımı oraya yazabilirim. Tam işlevsellik olmalı. Kullanıcı adı girmek, gönderilebilir bir işlev değildir. Ancak giriş yapabilmek tam bir işlevselliktir. Sevk edilebilir derken bunu kastediyorum.

### 71. Destan nedir?

- Epic, tek bir sprintte tamamlayamayacağımız büyük bir kullanıcı hikyesidir.
- Örneğin, bir kullanıcı olarak yerel mağazayı ziyaret etmek zorunda kalmamak için çevrimiçi satın almak istiyorum. Bu hikaye çok büyük ve olamaz bir sprintte tamamlandı. Yani, kullanıcı hikyesini yerine Epic diyebiliriz. Aşağıdakiler gibi birden çok kullanıcı hikyesine bölünmelidir:
  - o Müşteri olarak hesabımı görebilmek için giriş yapabilmek istiyorum.
  - o Müşteri olarak satın alabilmek için bir ürünü arayabilmek istiyorum.
  - o Müşteri olarak satın alacağım ürünün parasını ödeyebilmek için ödeme aşamasına geçmek istiyorum.
  - o Müşteri olarak hesabımı koruyabilmek için çıkış yapabilmek istiyorum.
  - o Gördüğünüz gibi <Bir müşteri olarak satın alabilmek istiyorum ...> birden çok kullanıcı hikyesine bölünebilir. Takım seçebilir her sprintte bir veya daha fazla kullanıcı hikyesini.

### 72. En son projenizde çevik deneyim?

- Sprintimiz 4 haftadır ve her 3 sprint'i yayın döngüsü olarak yayınlıyoruz
- Ekibimde 7 kişi var. 3 geliştirici (Shwan, Simon, Sinan), 1 otomasyon (Me) ve 1 fonksiyonel test edici (Usman), ayrıca 1 SM (Yasin) ve 1 PO (Simon B.).
- Sprint Planlama Toplantısı ile bir sprint başlatıyoruz ve
  - o ekibin öncelikli özellikleri ve ürün iş yığını öğeleri hakkında tartışırız ve
  - o uygulamanın geliştireceğimiz kısmını öğreniyoruz.
  - o dayalı hikaye seçimi *hız ve kapasite*
    - **Hız** : Bir sprintte iletilen hikaye puanı / demo sayısı. Örneğin: ekip 30 hikaye puanı planladıysa (İş değeri); bir sprintte değerinde kullanıcı hikayesi var ve planlandığı gibi sunabiliyorsa, takımın hızı 30'dur
    - **Kapasite** : Bir sprint için toplam kullanılabilir saat sayısı Takımın kapasitesidir. Tatil ve PTO saatlerini hesaplar

o Bu toplantı her hafta yapılır ve yaklaşık 1 saat sürer. Sprint Bakım toplantısından daha genel bir fikir ediniz

görevler için bazı tahminler vermek için.

- Ekip, SM ve PO, çalışma öğelerinin ilgili ve yararlı olmasını sağlamak için bir araya gelir
- Kullanıcı öykülerinin PO'suna sorular sorun
- Kabul kriterlerini yeniden tanımlayın
- Yeni hikayeler yazmak
- Destanları kullanıcı hikayelerine bölme
- Doğru tahminde bulunmak / gereğinden az / fazla tahminde bulunmak için hikayeyi anlayın

#### Nasıl tahmin edersiniz?

Deneyimlerime ve hikayenin karmaşıklığına dayanarak ve daha önce üzerinde çalıştığım bir şey.

- Sprint başladıktan sonra Günlük Standup Toplantısı yaparız
  - o her gün sabah ve dün ne yaptık, bugün ne yapacağız ve herhangi bir engelleyici var mı tartışıyoruz.
  - o Sadece sprint hakkındaki bilgileri senkronize ediyoruz.

- 25 -

## Sayfa 26

- Sprint sonu, genellikle Sprint Demo / Gözden Geçirme Toplantısı yaparız.
  - o Sadece müşteriye sprint oluşturduğumuzu göstermek içindir (PO geri bildirimde bulunabilir)
  - o Ekibimde bir SDET olarak, bazen sunum yaptım ve konferans odası.
  - o Müşteri, paydaşlar veya iş adamları bilmedikleri sorular sorarlar.
- Sprint Demosundan sonra Sprint Retrospektif Toplantısı yapıyoruz.
  - o Sprint Retro'da, son sprintte neyin iyi olduğunu, ne tür hatalar yaptığımızı konuşuyoruz.
  - o Bunların üzerinden geçer ve aynı hataları bir daha yapmayacağımızdan emin oluruz.
  - o İyi bir şey ve iyileştirmeler yapsaydık, yapmaya devam ederdik.
  - o Sprint gözden geçirme toplantısında veya sprint sonunda yapılan bu toplantı; 2-3 saat sürer.

### 73. Kabul kriterleri nedir?

- Kabul kriterleri, kullanıcı hikayesinin başarılı bir şekilde geliştirilip geliştirilmediğini bilme şeklimizdir.
- Bir hikayenin ne zaman "tamamlandığını" belirlemek için kullanıcının bakış açısından açıklanan gereksinim beyanları ve beklendiği gibi çalışmak
- 3 bölümden örnekler
  - o Giriş → geçerli e-posta adresi
  - o İşlem → mesajlaşmayı işletme
  - o Sonuç → pazarlama mesajı tasarımı, pazarlama tarafından sağlanan özelliklerle eşleşiyor

### 74. Fare deliği nedir?

- Agile takımında çok fazla iletişim olduğu için, takımın birçok konuyu tartışması gerekir. Ama bazen tartışma bir konu için çok uzun sürecek ve gerçekten verimli değil. Bunun <fare deliği> olduğunu söyleyeceğiz, bu, yapmamamız gerektiği anlamına gelir bu konuyu çok uzun süre ele alın ve ilerleyin.

### 75. Ne tür Test senaryoları?

- Farklı senaryoları ele alıyorum
  - o Olumlu
  - o Negatif
  - o Sınır Değer Analizi

### 76. Test Vakası?

- Test durumu, Test Edilmekte Olan Uygulama ile karşılaştırılacak özel bir durumdur. Test adımları, ön koşullar hakkında bilgi içerir, test ortamı ve çıktılar.
- Test senaryosu, işlevselliği ve test adımlarını açıklar.
  - o Test Vakası Kimliği
  - o Adım numarası
  - o İşlevselliğin açıklaması
  - o Beklenen sonuç
  - o Gerçek Sonuç

### 77. Genellikle bir haftada kaç Test vakası (regresyon süitinizde) tamamlıyorsunuz?

- 10 küçük test durumu, 7-8 orta, 2-3 büyük
- VEYA Projeye bağlıdır. COOLSIS'te 2000 test vakamız var. 4Stay'de yaklaşık 700 test vakamız var.

### 78. Regresyon süitinizi çalıştırmak ne kadar sürer?

- Projeye bağlıdır. Şu anki projemde, regresyon paketindeki 2000 test vakasından 1500 civarında zaten otomatik. Paralel yürütme gerçekleştirmek için 10 sanal makine kullanırsak, otomatik testi yürütmek 2 ila 3 gün sürer durumlarda. Ayrıca, manuel test ediciler bazı manuel test durumlarını yürütecekler, ancak ne kadarını yürüttüklerinden emin değilim. inanıyorum yalnızca bazı önemli test durumlarını yalnızca önceliklendirmeden sonra yürütürler.

- 26 -

**Sayfa 27****79. Otomatikleştirilmiş komut dosyanızı çalıştırdığınızda ne yaparsınız ya da regresyon çalıştırdığınızda ne yaparsınız?**

- İlk olarak, betiğini çalıştırmalıyım. Komut dosyası yürütme tamamlandığında, olup olmadığını görmek için çalıştırma sonucunu analiz etmeliyim.  
başarısız test durumları. Başarısız test senaryoları varsa, meşru uygulama sorunu nedeniyle başarısız olup olmadığını belirlemem gerekir veya bazı komut dosyası sorunlarından kaynaklanıyor. (komut dosyası da otomasyon kodu sorunu nedeniyle başarısız olabilir) uygulama sorunundan kaynaklanıyorsa, El ile yeniden üretmeye ve yeniden üretebilsem bir kusur kaydetmeye çalışacağım. Senaryomdan kaynaklanıyorsa, düzeltmem gerekiyor. Ama bu değil çoğu zaman durum.

**80. Otomatikleştirmek için attığınız adımlar nelerdir?**

- İşlevselliği öğrenin
  - o Okuma gereksinimleri
  - o BA ile bilgi aktarımı oturumu
  - o Takım arkadaşlarına sor
- Manuel olarak test edin
  - o Her adımı doğru anladığımdan emin olmak
  - o Beklenen sonuçları anlayın
- Otomatikleştirin
  - o POM sayfaları oluşturun
    - Kullanacağım gerekli öğeleri / yöntemleri ekleyin ve PageFactory tasarım desenini ekleyin
    - Singleton desenli bir sürücü sınıfı oluşturun
- TestNG Assertions'ı kullanarak testleri doğrulayın

**81. Otomasyon ile manuel arasında konunun yüzde kaç var?**

- % 80-85 otomasyon % 15-20 manuel

**82. Manuel test yerine otomatik testi ne zaman seçersiniz?**

- Test senaryoları yüksek öncelikli test durumlarıysa.
- İşlevsellik kritik işlev ise.
- Sallama veya duman testi test durumları.
- Test senaryoları çok uzun ve yürütmek çok zorsa. Önceliğe dayalı regresyon testi durumları.
- Mümkün olduğunca otomatikleştirmeliyiz.

**83. Sprintinizde otomasyonu ne zaman yaparsınız?**

- Geliştiriciler kendi paylarına düşeni yaptıklarında
- Kod, KG / test ortamına dağıtıldığında
- Test çerçevesi kurulduğunda
- Tüm manuel testler yapıldığında
- Duman testleri geçiyor

**84. Test Planı nedir?**

- Test planı, test kapsamını açıklayan bir kelime belgesidir
  - o Yüksek seviye test döngüsü
  - o Kusur yaşam döngüsü
  - o Giriş Kriterleri (teste başlamak için gerekenleri tanımlar)
  - o Çıkış Kriterleri (testin ne bittiğini tanımlar)

- 27 -

**Sayfa 28****85. Test planlarındaki tablolar nelerdir?**

- Test tasarımı, kapsam, test stratejileri, yaklaşım Test planı belgesinin içerdiği çeşitli detaylardır.
  - o Test senaryosu tanımlayıcısı
  - o Kapsam
  - o Test edilecek özellikler
  - o Test edilmeyen özellikler
  - o Test stratejisi ve Test yaklaşımı
  - o Teslimatların test edilmesi
  - o Sorumluluklar
  - o Risk ve Beklenmedik Durumlar

**86. Test planı ile QA planı arasındaki fark nedir?**

- Bir test planı, ürünün test etmek için ne yapılması gerektiğini ortaya koyar ve kalite kontrolün hataları belirlemek için nasıl çalışacağını ve kusurlar.

- Öte yandan, bir kalite güvence planı, hataları ve kusurları test etmek ve düzeltmek yerine önlenmesiyle ilgilidir.

#### 87. Akran değerlendirmesi nedir?

- Akran incelemeleri, aynı ekipte çalışan kişiler arasında yapılan incelemelerdir. Örneğin, yazılmış bir test senaryosu bir QA mühendisi tarafından bir geliştirici ve / veya başka bir QA mühendisi tarafından incelenebilir.

#### 88. Bir sistemi veya modülü yeterince test etmek için yeterli test senaryosunun oluşturulduğunu nasıl anlarsınız?

- Her gereksinimi karşılayacak en az bir test senaryosu olduğunda yeterli test senaryosunun oluşturulduğunu söyleyebilirsiniz. Bu, uygulamanın tüm tasarlanmış özelliklerinin test edilmesini sağlar.
- A2-İhtiyaç izlenebilirlik matrisine sahip olmamızın nedeni budur. Kaç gereksinimin olduğunu söyleyebiliriz test senaryoları kapsamına girer ve kaçının hala RTM'den kaldığı.

#### 89. Test durumlarını kim onaylar?

- Test senaryolarının onaylayıcısı bir kuruluştan diğerine değişir. Bazı kuruluşlarda, QA sorumlusu, vakaları test ederken bir başkası onları akran incelemelerinin bir parçası olarak onaylar.

#### 90. Test planlarını ve test senaryolarını kim yazıyor?

- Test planları tipik olarak kalite güvence sorumlusu tarafından yazılırken, test uzmanları genellikle test senaryoları yazarlar.

#### 91. Test tasarım tekniğinin amacı nedir?

- Test koşullarının belirlenmesi ve test senaryolarının belirlenmesi.

#### 92. Test senaryosu ile Test komut dosyası arasındaki fark nedir?

- Test senaryosu terminolojisi çoğunlukla Manuel Test için kullanılırken, Test Komut Dosyası çoğunlukla Otomasyon Testi için kullanılır
- **Test senaryosu, girdi değerlerini, beklenen çıktıyı ve yürütme için ön koşulları belirten bir dokümantasyondur.**  
**test.** Aynı zamanda senaryonun nasıl test edileceğine dair alt düzey ayrıntıların bir düzenidir.
- Yazılım testinde bir test komut dosyası, test edilen sistemde test edilen sistem beklendiği gibi çalışır.

#### 93. Bir test stratejisine neler dahil edilmelidir?

- Test stratejisi, uygulamanın nasıl test edileceğine ve tam olarak neyin test edileceğine dair bir plan içerir (*kullanıcı arayüzü, modüller, süreçler, vb.*). Test için sınırlar belirler ve manuel veya otomatik testin kullanılıp kullanılmayacağını belirtir.

- 28 -

## Sayfa 29

#### 94. Komut dosyası başarısız olduğunda ne yapacaksınız?

- Deneyimlerime göre, başarısızlığı tespit edeceğim,  
o uygulama hatası, senkronizasyon hatası, komut dosyası sorunu veya ortamın çalışmamasından kaynaklanıyorsa, öncelikle sonucu şu şekilde analiz ederim:  
Jenkins aracılığıyla yeniden üretin, yalnızca başarısız olanı çalıştırın,  
o senkronizasyon sorunundan kaynaklanıyorsa, örtük, açık ve bazı özel beklenenleri kullanarak ekstra süre ekleyeceğim  
koşullar,  
o Komut dosyası sorunu varsa, komut dosyamda hata ayıklayacağım (tanımlayacağım) ve düzelteceğim, istisnaları analiz edeceğim,  
o eğer gerçek bir kusursa, o zaman kusuru kaydedirim.

#### 95. Test Senaryosu?

- Test edilen uygulamanın uçtan uca işlevselliğinin beklendiği gibi çalıştığından emin olun
- Test edenin, eylemi nasıl kullandıklarını kontrol etmek ve gerçekleştirmek için ayağını son kullanıcıların ayakakabalarına koymasına gerekir.  
test edilen uygulama
- TS kendisiyle ilişkili birçok test senaryosuna sahip olabilir, TS'yi yürütmeden önce senaryo için test senaryoları düşünmemiz gerekir.
- Test Senaryosu: Giriş sayfasını doğrulayın  
o Test Senaryosu 1: Geçerli bir kullanıcı adı ve şifre girin  
o Test Senaryosu 2: Şifrenizi sıfırlayın  
o Test Senaryosu 3: Geçersiz kimlik bilgilerini girin
- Her test senaryosunda, yürütme için ayrıntılı adımlar ve koşullar bulunur

#### 96. Gereksinim İzlenebilirlik Matrisi (RTM)

- RTM, tüm test senaryolarının gereksinimi karşılayıp karşılamadığından emin olmak için kullanılır. Excel sayfası gibidir.

#### 97. İşlevsel bir özellik yoksa veya herhangi bir sistem ve geliştirme yoksa bir sistem için bir test geliştirmek için neler yapılabilir?

- **belgeler?**
- İşlevsel özellikler veya sistem geliştirme belgeleri olmadığında, test uzmanı kendini alıştırmalıdır  
ürün ve kod ile. Piyasadaki benzer ürünleri bulmak için araştırma yapmak da faydalı olabilir.

#### 98. Fonksiyonel test türleri nelerdir?

- Birim Testi
- Duman testi

- Sağlık testi
- Entegrasyon Testi
- Sistem Testi
- Gerileme testi
- UAT (kullanıcı kabul testi)

**99. Akıl sağlığı testi ile duman testi arasındaki fark nedir?**

- Akıl sağlığı testi yapıldığında, ürün, sırayla test grubu ile bir ön test turundan geçirilir.  
düşme işlevi gibi temel işlevleri kontrol etmek için. Duman testi ise geliştiriciler tarafından yapılmaktadır.  
müşterinin gereksinimlerine göre.

**100. Akıl sağlığı testinde hangi adımlar yer alır?**

- Sağlık testi, duman testine çok benzer. Yapmak için yapılan bir bileşenin veya uygulamanın ilk testidir.  
en temel düzeyde çalıştığından ve daha ayrıntılı testlere devam etmek için yeterince kararlı olduğundan emin olun.

**101. WinRunner ile Rational Robot arasındaki fark nedir?**

- WinRunner işlevsel bir test aracıdır, ancak Rational Robot hem işlevsel hem de performans testi yapabilir. Ayrıca,  
WinRunner 4 doğrulama noktasına ve Rational Robot 13 doğrulama noktasına sahiptir.

- 29 -

## Sayfa 30

**102. Kalite Güvencesi ile test arasındaki fark nedir?**

- Kalite Güvencesi'nin hedefleri, test etme hedeflerinden çok farklıdır.
- QA'nın amacı hataları önlemektir, testin amacı ise hataları bulmaktır.

**103. Rastgele testi açıklayın.**

- Rastgele test, uygulamanın rastgele oluşturulan girdi verilerini nasıl işlediğini kontrol etmeyi içerir. Veri türleri  
tipik olarak yok sayılır ve rasgele bir harf, sayı ve diğer karakterler dizisi veri alanına girilir.

**104. Kalite Kontrol ve Kalite Güvencesi arasındaki fark nedir?**

- Kalite kontrolü (QC) ve kalite güvencesi (QA) birbiriyle yakından bağlantılıdır, ancak çok farklı kavramlardır. QC, bir  
Geliştirilen ürün, kalite güvencesinin amacı, geliştirme sürecinin,  
sistem veya uygulama gereksinimleri karşılayacaktır.

**105. Proje geliştirmede QA'nın rolü nedir?**

- QA ekibi, geliştirme için yürütülecek sürecin izlenmesinden sorumludur.
- QA ekibinin sorumlulukları, test yürütme sürecini planlamaktır.
- QA Lideri, zaman tablolarını oluşturur ve ürün için bir Kalite Güvence planı üzerinde anlaşır.
- QA ekibi, QA sürecini ekip üyelerine iletir. QA ekibi, test senaryolarının gereksinimlere göre izlenebilirliğini sağlar.

**106. İyi bir QA veya Test yöneticisi yapan nedir?**

- Yazılım geliştirme süreci hakkında bilgi
- Üretkenliği artırmak için ekip çalışmasını iyileştirir
- Yazılım, test ve kalite kontrol mühendisleri arasındaki işbirliğini geliştirir
- QA süreçlerini iyileştirmek için.
- İletişim yetenekleri.
- Toplantılar düzenleyebilir ve odaklanmış halde tutabilir

**107. Regresyon testi ile yeniden test arasındaki fark nedir?**

- Regresyon testi, bir modül veya sistemde yapılan değişikliklerin olumsuz bir etkiye sahip olmadığından emin olmak için testler gerçekleştiriyor  
önceki sürümlerde. Yeniden test sadece aynı testi tekrar çalıştırmaktır. Regresyon testi yaygın olarak sorulan kılavuzdur  
mülakat sorularını test etmek ve dolayısıyla bu konuyu anlamak için daha fazla araştırmaya ihtiyaç vardır.

**108. Hata önem derecesi ve hata önceliği arasındaki farkı açıklayın.**

- Hata şiddeti, hatanın uygulama veya sistem üzerindeki etki düzeyini ifade ederken, hata önceliği seviyeyi ifade eder  
bir düzeltmeye ihtiyaç duyan aciliyet.
- Genellikle ciddiyet, maddi kayıp, çevreye verilen zarar, şirketin itibarı ve can kaybı olarak tanımlanır.  
Bir kusurun önceliği, bir hatanın ne kadar hızlı düzeltilmesi ve canlı sunuculara dağıtılması gerektiğiyle ilgilidir.

**109. Sistem testi ile entegrasyon testi arasındaki fark nedir?**

- **Sistem testi** için tüm sistem bir bütün olarak kontrol edilir,
- **entegrasyon testi** için ise , ayrı modüller arasındaki etkileşim test edilir.

**110. Fonksiyonel ve yapısal testler arasındaki farkı açıklayın.**

- İşlevsel test, test edenin sistemin veya  
uygulama spesifikasyona göre çalışır. Yapısal testler ise koda veya algoritmalara dayanmaktadır.  
ve beyaz kutu testi olarak kabul edilir.

## Sayfa 31

## 111. Pilot ve Beta testi arasındaki fark nedir?

- Bu ikisi arasındaki farklar aşağıda listelenmiştir:
  - o Ürün son kullanıcıya sunulmak üzereyken bir beta testi, daha önceki aşamada pilot test yapılır. geliştirme döngüsünün.
  - o Beta testinde uygulama, uygulamanın kullanıcı gereksinimlerini karşıladığından emin olmak için birkaç kullanıcıya verilir ve herhangi bir gösterici içermez, oysa pilot test ekibi üyesi olması durumunda kaliteyi iyileştirmek için geri bildirimde bulunur. uygulamanın.

## 112. Alfa testi nedir?

- Geliştiricinin sitesinde son kullanıcı temsilcileri tarafından ön sürüm testi.

## 113. Başarısızlık nedir?

- Başarısızlık, belirtilen davranıştan uzaklaşmadır.

## 114. Test karşılaştırmaları nelerdir?

- Bazı yazılımlara bazı girdiler koyarsanız bu gerçekten bir test midir, ancak yazılımın bunları üretip üretmediğine asla bakmazsanız doğru sonuç?
- Testin özü, yazılımın doğru sonucu verip vermediğini kontrol etmektir ve bunu yapmak için karşılaştırmalıyız. Yazılımın üretmesi gereken şey için ne ürettiği.
- Bir test karşılaştırmacı, bu karşılaştırmaların yönlerini otomatikleştirmeye yardımcı olur.

## 115. Yazılım testi sırasında Risk analizinin nasıl gerçekleştirileceğini açıklayın?

- Risk analizi, uygulamada riski belirleme ve test etmek için önceliklendirme sürecidir. Aşağıdakilerden bazıları riskler:
 

1. Yeni Donanım.	3. Yeni Otomasyon Aracı.	5. Uygulamanın kullanılabilirliği
2. Yeni Teknoloji.	4. Kod tesliminin sırası.	kaynakları test edin.
- Bunları üç kategoriye ayırıyoruz:
  - o Yüksek büyüklük: Hatanın uygulamanın diğer işlevselliği üzerindeki etkisi.
  - o Orta: Uygulamada tolere edilebilir ancak istenmez.
  - o Düşük: tolere edilebilir. Bu tür bir riskin şirket işi üzerinde hiçbir etkisi yoktur.

## 116. İpek Testi nedir?

- İpek Testi, uygulamanın regresyon ve işlevsellik testini gerçekleştirmek için geliştirilmiş bir araçtır. İpek Testi bir araç Windows, Java, web veya geleneksel istemci / sunucu tabanlı uygulamaları test ederken kullanılır.
- Silk Test, test planının hazırlanmasında ve bu test planlarının yönetiminde, testin doğrudan erişimini sağlamak için yardımcı olur. veritabanı ve alanın doğrulanması.

## 117. Ana Test Planı ile Test Planı arasındaki fark nedir?

- Ana Test Planı, uygulamanın tüm test ve riskle ilgili alanlarını içerirken, Test senaryosu dokümanı şunları içerir: test durumları.
- Ana Test planı, genel geliştirme sırasında çalıştırılacak her bir testin tüm ayrıntılarını içerir. uygulama, test planı ise test gerçekleştirmenin kapsamını, yaklaşımını, kaynaklarını ve programını açıklar.
- Ana Test planı, uygulama üzerinde gerçekleştirilecek her testin açıklamasını içerirken, test planı yalnızca birkaç test senaryosunun açıklamasını içerir. Birim testi, Sistem testi, beta testi vb. gibi test döngüsü sırasında
- Tüm büyük projeler için Master Test Planı oluşturulur, ancak küçük proje için oluşturulduğunda bunu test planı olarak adlandırdık.

## Sayfa 32

## 118. Bir test ne zaman başarılı kabul edilir?

- Testin amacı, uygulamanın gereksinimlere göre çalışmasını sağlamak ve olabildiğince çok mümkün olduğunca hatalar ve hatalar. Bu, daha fazla işlevselliği kapsayan ve daha fazla hata ortaya çıkaran testlerin, en başarılı ol.

## 119. Kusur nedir?

- Beklenen sonuç gerçek sonuçla eşleşmediğinde, bu kusurdur.

## 120. Kusur yoğunluğu tanımlansın mı?

- Hata yoğunluğu, kod satırı başına toplam kusur sayısıdır.

**121. Kusur Yaşam Döngüsü (DLC) nedir?**

- Yeni → Atanmış → Açık → Sabit → Yeniden Test Edildi → Kapat

**122. Kusur kategorileri nelerdir?**

- **Yanlış** : Gereksinimler uygulamada yanlış uygulanmıştır.
- **Eksik** : Müşteri tarafından verilen gereksinim ve uygulama bu uygulamayı karşılamadığında.
- **Ekstra** : Son müşteri tarafından verilmeyen ürüne dahil edilen bir gereksinim. Bu her zaman bir farktır şartname ancak ürünün kullanıcıları tarafından istenen bir özellik olabilir.

**123. Bir kusur bulduğunuzda ne yapmalısınız?**

- Bir kusur bulursam, bunu bildirmeden önce, geçerli bir kusur olduğundan emin olmam gereken hatayı yeniden oluşturuyorum.
- Küçük bir sorunsal, geliştirici masasına gideceğim ve hemen düzeltebilir.
- Büyük bir sorunsal, JIRA'ı açar ve kusuru günlüğe kaydedirim.
- Hata olup olmadığından emin değilsem KOBİ ile konuşacağım (konu uzmanı, uygulamayı bilen kişi demektir hiç yoktan iyidir).

**124. Test nasıl yapılmalıdır?**

- Test, uygulamanın teknik gereksinimlerine göre yapılmalıdır.

**125. Geliştirici kusur olmadığını söylüyorsa ne yapmalı?**

- Her zaman gerçek bir kusur olduğundan emin oluyorum, bu yüzden onu yeniden üretiyorum.
- Ekran görüntülerini alıyorum ve kusuru yeniden oluşturmak için tüm adımları atıyorum.
- Aslında mevcut projemde karşılaştığım en büyük zorluklardan biri de bu.

**126. Bir programı test edip hataların% 100'ünü bulabilir misiniz?**

- Bir uygulamadaki tüm hataları bulmak imkansızdır çünkü çoğunlukla kaç tane hata olduğunu hesaplamamın bir yolu yoktur. Orada Programın karmaşıklığı, programcının deneyimi gibi böyle bir hesaplamayla ilgili birçok faktör var. ve benzeri. Bu Manuel test mülakat soruları, test uzmanları tarafından dikkate alınan en zor sorulardır.

**127. Hata ayıklama ve test etme arasındaki fark nedir?**

- Hata ayıklama ile test etme arasındaki temel fark, hata ayıklamanın genellikle aynı zamanda sorunu çözen bir geliştirici tarafından yürütülmesidir. hata ayıklama aşamasında hatalar. Öte yandan test, hataları düzeltmek yerine bulur. Bir testçi bulduğunda hata, genellikle bir geliştiricinin düzeltebilmesi için rapor ederler.

**128. İyi bir test olarak kabul edilen nedir?**

- Bir nesnenin veya sistemin işlevselliğinin çoğunu kapsayan test, iyi bir test olarak kabul edilir.

- 32 -

**Sayfa 33****129. Test ne zaman durdurulmalıdır?**

- Test edilen sistemin risklerine bağlıdır. Testi durdurabileceğiniz bazı kriterler vardır.
  - o Son Tarihler (Test, Sürüm)
  - o Test bütçesi tükendi
  - o Hata oranı belirli bir düzeyin altına düşüyor
  - o Belirli bir yüzde ile tamamlanan test senaryoları
  - o Test bitimleri için alfa veya beta dönemleri
  - o Kod, işlevsellik veya gereksinimlerin kapsamı belirli bir noktaya kadar karşılanır

**130. Yukarıdan aşağıya ve aşağıdan yukarıya test arasındaki fark nedir?**

- Yukarıdan **Aşağıya** test, sistemle başlar ve ünite seviyesine kadar ilerler.
- **Aşağıdan yukarıya** testi, ters yönde, birim seviyesinden genel sisteme arayüzle kontrol eder. İkisinin de değeri var ama Aşağıdan yukarıya test, genellikle hataları düzeltme maliyetinin daha düşük olduğu geliştirme döngüsünün başlarında kusurları keşfetmeye yardımcı olur.

**131. Oluşturduğunuz çalıştırılabilir dosyaların ortalama boyutu nedir?**

- Bu, yürütülebilir dosyalar ile deneyimimiz hakkında basit bir röportaj sorusudur. Oluşturduğunuz herhangi birinin boyutunu biliyorsanız, sadece bu bilgiyi sağlayın.

**132. Ön uç ve arka uçta testler yaptınız mı?**

- Ön Ucu test ettiğimde, uygulamayı açarak ve kullanıcı arayüzünde test gerçekleştirerek aslında kullanıcı arayüzünü test ediyorum. Eğer yaptıysam Kullanıcı arayüzündeki herhangi bir şey varsa, değişikliğin veritabanında da yapıp yapılmadığını görmek için arka uç testi yapmam gerekiyor. İçin Örneğin, bir ebeveyn iletişimi bilgilerini güncellediğimde veya yeni bir uygulama oluşturduğumda, veritabanına bağlanıyorum ve verilere değişiklikler uygulanır veya yeni uygulama oluşturulur veya oluşturulmaz.

**133. Ön Uç Testi ile Arka Uç testi arasındaki fark nedir?**

- Ön Uç Testi, Grafik Kullanıcı Arayüzünde gerçekleştirilirken Arka Uç Testi, veri tabanlarının testini içerir.

- Ön uç, kullanıcının etkileşime girebileceği web sitesi görünümünden oluşurken, arka uç durumunda gerekli olan veritabanıdır. verileri saklamak için.
  - Son kullanıcı, ön uç uygulamanın GUI'sine veri girdiğinde, bu girilen veriler veritabanında saklanır. Kaydetmek
- Bu verileri veritabanına SQL sorguları yazıyoruz.

#### 134. Test sırasında bulduğunuz en zor sorun nedir?

- *(Bu, örnek vermeniz gereken basit bir mülakat sorusudur)* . Bu, en zor manuel testlerden biridir.

Mülakat soruları, cevabınız işinizi belirleyecektir. Problem çözme becerilerinizin ve işin. Öyle bir cevap vermelisiniz ki problem çözme becerileriniz ve yeni şeyler öğrenme hevesiniz, ve işe olan bağlılığınızla cevaplarınızla belirtilecektir.

#### 135. Scrumda karşılaştığınız zorluk nedir?

- Scrum, çapraz fonksiyonlu takımı vurguladığından (bu, geliştiricinin test edebilmesi ve test uzmanlarının geliştirebilmesi gerektiği anlamına gelir) geleneksel bir QA test cihazı olarak geliştirme ekibinin bir parçası olmak zordur. Çünkü genellikle KG'ler nasıl kod yazılacağını bilmiyor. Bu yüzden kendimi çok rekabetçi tutmalıyım. Ne zaman vaktim olursa, Java gibi daha çok kodlama öğreniyorum.
- Zaman değişikliği sorunu → Veritabanına girilen tarihi bir tarih kaydettiğinde, bunlar daha erken.

#### 136. Otomasyon Testi nedir?

- İnsan müdahalesini azaltan testin otomatik olarak gerçekleştirilmesi süreci, bu otomasyon testidir.
- Otomasyon testi QTP, Selenium, WinRunner vb. Bazı otomasyon araçları yardımıyla yapılır.
- Otomasyon testinde, uygulamayı test etmek için test komut dosyasını çalıştıran bir araç kullanıyoruz; bu test komut dosyası oluşturulabilir manuel veya otomatik olarak. Test tamamlandığında araçlar otomatik olarak test raporu ve rapor oluşturur

- 33 -

## Sayfa 34

#### 137. Ne zaman otomatikleştireceksin?

- Çok fazla manuel çaba gerektiriyorsa. En az bir kez manuel çalıştırıyorum ve ardından otomatikleştiriyorum.
- Otomasyon, tekrar eden çoğu işlevsellik için iyidir

#### 138. Hangi testler otomatikleştirilebilir?

- Regresyon testleri
- Fonksiyonel testler
- Veri tabanı
- Duman testleri
- API

#### 139. Ne zaman otomatikleştirmeyeceksiniz?

- İşlevsellik değişmeye devam ederse
- İşlevsellik tüm proje boyunca yalnızca bir kez kullanılırsa
- **Ad-hoc testi** otomatikleştirilemez.

#### 140. Bir scrum sprintinin süresi nedir? Sprint ne kadar sürüyor?

- Mevcut projemde senaryo döngümüz 4 haftadır. Sprintiniz burada ne kadar sürer? 2 hafta mı 4 hafta mı? (bazen iyidir soru sormak için. ATM gibi davranmamanız gerektiğini unutmayın. Genellikle sadece soruya cevap veren kişileri unuturlar. Orada bir denge olmalıdır.)
- Ekip sayımız 7 kişidir. 1 SM, 1 PO, 3 geliştirici, 1 MT, 1 AT

#### 141. Hız nedir?

- Hız, takımın baskıya göre ilerleme hızıdır.
- İki farklı scrum takımıyla karşılaştıramayacağımı da söyleyebilirim.

#### 142. "Yapı kırıcı" nedir?

- Yapı kırıcı, yazılımda bir hata olduğunda ortaya çıkan bir durumdur.
- Bu ani beklenmeyen hata nedeniyle, derleme işlemi durur veya yürütme başarısız olur veya bir uyarı oluşturulur.
- Daha sonra test edenin sorumluluğu, hatayı ortadan kaldırarak yazılımı normal çalışma aşamasına geri döndürmektir.

#### 143. Scrum'daki engeller hakkında ne biliyorsunuz ? Bazı engel örnekleri verin.

- Engeller, scrum ekibinin karşılaştığı, çalışma hızlarını yavaşlatan engeller veya sorunlardır.
- Scrum ekibinin işlerini "Bitti" yapmasını engellemeye çalışan bir şey varsa, bu bir engeldir.
- Engeller herhangi bir biçimde olabilir. Bazı engeller şu şekilde verilmiştir:
  - o Kaynak eksik veya hasta ekip üyesi
  - o Teknik, operasyonel, organizasyonel sorunlar
  - o Yönetim destek sistemi eksikliği
  - o İş sorunları
  - o Hava durumu, savaş vb. dış sorunlar
  - o Beceri veya bilgi eksikliği
- Çözüm: Ekip çalışması, sıkı çalışın, iyi iletişim kurun, çevrimiçi bağlantı, rehberlik ve eğitim

#### 144. Çevik ve Scrum arasındaki fark ve benzerlik nedir?

- Çevik, geniş bir spektrumdur, proje yönetimi için kullanılan bir metodolojidir, Scrum ise sadece Çevik



- Süreci ve adımlarını daha kısaca açıkla.
- Çevik bir uygulamadır, oysa scrum bu uygulamayı sürdürmek için bir prosedürdür.
- Çevik → projeleri adım adım veya aşamalı olarak tamamlamayı içeren benzerlik. Çevik metodoloji dikkate alınır doğada yinelenmeli olmak. Bir Agile formu olan Scrum, Agile ile aynıdır. Aynı zamanda artımlı ve yinelenmelidir.

- 34 -

## Sayfa 35

### 145. Artış nedir? Açıklamak.

- Artış, bir sprint sırasında tamamlanan tüm ürün biriktirme kalemlerinin toplamıdır .
- Her artış, kümülatif olduğu için önceki tüm sprint artış değerlerini içerir.
- Hedefimize ulaşmak için bir adım olduğu için sonraki sürümde mevcut modda olmalıdır.

### 146. Daily stand-up'tan ne anlıyorsunuz?

- Günlük stand-up, hemen hemen tüm ekibin bir araya geldiği günlük bir toplantıdır (en çok tercihen sabahları yapılır).
  - Aşağıdaki üç soruya cevap bulmak için 15 dakika -
  - Dün ne yaptın? Bugün için planın nedir?
  - Görevinizi tamamlamanıza engel olan herhangi bir engel veya engel var mı?
- Günlük stand-up, ekibi motive etmenin ve gün için bir hedef belirlemesini sağlamanın etkili bir yoludur.

### 147. Scrum hakkında ne biliyorsunuz?

- Scrum, yazılım geliştirme için Scrum ve Kanban tabanlı bir modeldir.
- Bu model, özellikle sürekli bakıma ihtiyaç duyan, çeşitli programlama hataları olan veya bazı ani değişiklikler var.
- Bu model, bir programlama hatası veya kullanıcı hikayesi için minimum sürede bir projenin tamamlanmasını teşvik eder.

### 148. Agile kalite stratejilerinden bazılarını belirtiniz mi?

- Yineleme
- Yeniden düzenleme
- Dinamik kod analizi
- Kısa geribildirim döngüleri
- İncelemeler ve inceleme
- Standartlar ve Talimatlar
- Dönüm noktası incelemeleri

### 149. Çevik Manifesto ve Prensiplerini biliyor musunuz? Kısaca açıklayın.

- Bu, çevik / scrum rollerinin çoğunun ipuçlarında olması gereken teoridir.
- Bu sorunun bir parçası olarak dört manifesto değeri ve 12 ilke olabildiğince açıklanmalıdır.
- % 100 doğru bir şekilde anlatılmasa bile iyi olmalı, ancak değerlerin ve ilkelerin niyetleri gelmeli dışarı örneğin
- Bildirir
  - Süreçler ve araçlardan ziyade bireyler ve etkileşimler
  - Kapsamlı dokümantasyon yerine çalışan yazılım
  - Sözleşme müzakeresi yerine müşteri işbirliği
  - Bir planı takip etmek yerine değişime yanıt vermek
- Rehber ilkeler
  - Müşteri Memnuniyeti
  - Hoş Geldiniz Değişen Gereksinimler
  - Çalışan Yazılım Sıkça Verilir (Aylar yerine Haftalar)
  - İş İnsanları ve Geliştiriciler Arasında Yakın, Günlük İşbirliği
  - Proje, güvenilmesi gereken motive olmuş bireyler etrafında inşa edilmiştir.
  - Yüz Yüze Görüşme, iletişimin en iyi şeklidir
  - Çalışan yazılım, ilerlemenin birincil ölçüsüdür
  - Sürdürülebilir kalkınma, sabit bir tempoyu koruyabilen
  - Teknik mükemmelliğe ve iyi tasarıma sürekli dikkat
  - Basitlik - Yapılmayan iş miktarını en üst düzeye çıkarma sanatı - esastır

- 35 -

## Sayfa 36

- En iyi mimariler, gereksinimler ve tasarımlar kendi kendini organize eden ekiplerden ortaya çıkar
- Ekip düzenli olarak nasıl daha etkili olabileceği üzerine düşünür ve buna göre ayarlanır

### 150. Yakma ve yanma çizelgelerinin faydası nedir?

- Yakma çizelgesi bir projede tamamlanan iş miktarını gösterirken, yakma çizelgesi miktarı gösterir

- bir projeyi tamamlamak için geriye kalan iş..
- Bu nedenle, yanma ve yanma çizelgeleri bir projenin ilerlemesini izlemek için kullanılır.

#### 151. Çevik modelin herhangi bir dezavantajı var mı? Öyleyse, açıkla.

- Evet, Agile modelinin bazı dezavantajları var, bazıları şöyle;
  - o Bir görevi tamamlamak için gereken çaba hakkında bir tahminde bulunmak kolay değildir. Durumda daha sorunlu hale geliyor gereken toplam çaba hakkında bir fikir edinmek zorlaştığından büyük projeler için.
  - o Bazen, projenin tasarımına ve dokümantasyonuna düzgün şekilde odaklanmak mümkün değildir
  - o Müşterinin gereksinimlerinin doğru anlaşılması durumunda nihai proje müşteriyi karşılamayacaktır. Gereksinimler. Böylece müşteri memnuniyetsizliğine yol açacaktır.
  - o Yalnızca Çevik metodolojiler konusunda hatırı sayılır deneyime sahip lider önemli kararlar alabilir. The Deneyimi çok az olan veya hiç olmayan ekip üyeleri karar verme sürecine dahil olmazlar, bu nedenle ilerleme şansları olmaz. onların bilgisi.

#### 152. Agile'da Zero Sprint ve Spike'ı tanımlayın.

- Sıfır Sprint, Çevik'teki ilk sprint'in hazırlık adımı olarak tanımlanabilir.
  - o Projeye fiilen başlamadan önce yapılması gereken bazı faaliyetler vardır.
  - o Bu aktiviteler Sıfır sprint olarak kabul edilir; bu tür faaliyetlerin örnekleri şunlardır: geliştirme, biriktirme listelerinin hazırlanması vb.
- Spike, sprintler arasında alınabilecek hikaye türüdür.
  - o Çiviler, tasarım veya araştırma, tasarım gibi teknik konularla ilgili faaliyetler için yaygın olarak kullanılmaktadır. prototip oluşturma ve keşif.
  - o İki tür sivri uç vardır - işlevsel sivri uçlar ve teknik sivri uçlar.

#### 153. Scrum Master'ın rolü nedir?

- Scrum ustası, Scrum takımının lideri ve koçudur.
- SM, performansı etkileyebilecek her türlü engelden takıma hizmet etmekten ve korumaktan sorumludur.
- SM'nin ana rolü, takımını sprint hedefine ulaşması için motive etmektir.
- Kendini organize eden ve motive olmuş bir ekip oluşturmaya odaklanmıştır ve her bir üye, aşağıdakilerin uygulanmasına aşinadır. Çevik ve Scrum ilkeleri ve uygulamaları.
- SM, adanmış görevleri düzgün bir şekilde yerine getirip getirmediklerini scrum ekibini uygun bir şekilde kontrol eder.
- Sprint hedefine ulaşabilmeleri için takımın verimliliğini ve üretkenliğini artırmaktan da sorumludur. etkili bir şekilde.

#### 154. Scrum'daki bir hikaye noktası hakkında ne biliyorsunuz?

- Scrum'daki bir hikaye noktası, belirli bir işlemi gerçekleştirmek veya tamamlamak için gereken toplam çabanın tahmin edilmesine yönelik birimdir. görev.

#### 155. Sashimi'nin Scrum metodolojisindeki rolü nedir?

- Sashimi, Scrum metodolojisinde önemli bir rol oynar.
- Sashimi, geliştiriciler tarafından oluşturulan tüm işlevlerin tamamlandığını kontrol etmek için Scrum tarafından kullanılan bir tekniktir.
- Bu tekniği kullanarak, kullanılan analiz, tasarım, kodlama, test ve dokümantasyon gibi tüm gereksinimler bir ürünün yapısı kontrol edilir ve ancak bundan sonra ürün görüntülenir.

- 36 -

## Sayfa 37

#### 156. Çevik test teriminden ne anlıyorsunuz?

- Çevik test, tamamen yazılım geliştiricinin çevik ilkelerine dayanan bir yazılım test uygulamasıdır. O bir gereksinimlerin ürün sahibi ile ekip arasındaki işbirliğinin sonucu olduğu yinelemeli metodoloji. Çevik ilkeler ve uygulamalar, müşteri gereksinimlerini karşılamak için, başarılı bir şekilde tamamlanır. proje.

#### 157. Scrum yerine şelale kullanılması hiç önerildi mi? Varsa, ne zaman olduğunu açıklayın.

- Evet, bazen Scrum yerine şelale modelinin kullanılması önerilmektedir.
- Müşteri gereksinimleri basit, iyi tanımlanmış, tam olarak anlaşılabilir olduğunda ve tabi olmadığında yapılır. projenin tamamlanmasına kadar değiştirmek.

#### 158. Scrum, projeler için otomatik testlerin kullanılmasını neden teşvik ediyor?

- Scrum, en hızlı yapmak için otomatik (otomatikleştirilmiş performans veya otomatik regresyon) testinin kullanılmasını teşvik eder projenin olası teslimi. *otomatikleştirmek için kullandığınız bazı araçları açıklayabilirsiniz*

#### 159. Çevik için bazı yaygın matrisleri açıklayın.

- **Hız** → Hız, son 3-4 sprintten alınan ortalama puan sayısıdır. Hepsinin toplamı ile ölçülür. hikayelerin onaylanmış tahminleri. Kapasite, ilerleme vb. Hakkında fikir verir.
- **Kümülatif Akış Şeması** → Bunun yardımıyla, tek tip iş akışı üzerinde bir inceleme yapılır. Bu diyagramda / grafikte, x eksenini zamanı temsil ederken y eksenini çaba sayısını temsil eder.
- **İş Kategorisi Tahsisi** → zaman yatırımı hakkında hızlı bilgi veren önemli bir faktördür, örn.

zaman harcanıyor ve zaman faktörü olarak hangi göreve öncelik verilmesi gerekiyor.

- **Zaman Kapsamı** → Test sırasında bir koda verilen zamandır. Yüzde olarak hesaplanır.

test paketi tarafından çağrılan kod satırlarının sayısı ve ilgili kod satırlarının toplam sayısı.

- **Verilen İş Değeri** → Ekibin çalışma verimliliğini ifade eden bir terimdir. İş hedefleri

öncelik, karmaşıklık ve ROI düzeyine göre sayısal değerler 1,2,3 vb. atanır.

- **Kusur Giderme Farkındalığı** → Ekibin kaliteli bir ürün sunmasına yardımcı olan faktördür. Bir kimlik

aktif kusur sayısı, bunların farkındalığı ve giderilmesi, yüksek kaliteli bir ürün sunmada önemli bir rol oynar.

- **Hata Çözüm Süresi** → Ekip üyelerinin hataları (hataları) tespit edip bir öncelik belirlediği bir prosedürdür.

kusur çözümü için. Hataları / hataları düzeltme veya kusur çözme prosedürü, aşağıdakiler gibi birden fazla işlemden oluşur:

Kusur resmini temizleme, kusur tespitini planlama, kusur tespitini tamamlama, çözüm oluşturma ve işleme

bildirir.

- **Sprint Burndown Matrix** → Sprint burndown grafiği, uygulanmayan veya uygulanmayanların sayısını temsil eden bir grafikdir.

Scrum döngüsü sırasında sprintler uyguladı. Bu matrik, sprint ile tamamlanan işin izlenmesine yardımcı olur.

#### 160. Çevik modeli kullandığınız bazı metodolojileri ve gelişmeleri adlandırın.

- Çevik modelin kullanılabileceği metodolojilerden ve geliştirmelerden bazıları şunlardır:

- o Kristal metodolojileri

- o Yalın yazılım geliştirme

- o Dinamik geliştirme ve Özellik odaklı geliştirme

#### 161. Scrum M / Ürün O / Çevik ekip üyesi olarak deneyiminizi paylaşın ve birincil sorumluluklarınız nelerdi?

- Bu sorudaki püf noktayı, açıklarken kendi kendini organize eden ve kendi kendini motive eden bir ekip gösterip göstermediğinizdir.

#### 162. Projenizdeki sprintlerin / yinelemelerin uzunluğu neydi?

- Buradaki fikir, ne tür bir ortamda çalıştığınıza karar vermektir. Kesinlikle gibi takip soruları olacak

bu uzunluk başlangıçta sabitlendi ve hiç değişmedi mi? Bu uzunluktan daha fazlasını mı yoksa daha azını mı denediniz?

- 37 -

## Sayfa 38

#### 163. "Planlama Poker" tekniği hakkında ne biliyorsunuz?

- Scrum Poker olarak da bilinen planlama pokeri, planlama ve tahmin için kullanılan kart tabanlı bir çevik tekniktir. İçin

planlama poker tekniği oturumu başlatmak, çevik kullanıcı hikayesi ürün sahibi tarafından okunur.

- Poker planlama tekniğinde gerçekleştirilen adımlar şunlardır:

- o Her bir tahmincede 0, 1, 2, 3, 5 vb. değerlere sahip bir deste poker kartı vardır ve hikaye puanlarını belirtmek için idealdir.

günler veya takımın tahmin için kullandığı başka bir şey.

- o Her bir tahmincinin ürün sahibi ile bir görüşmesi vardır ve ardından müşterilere göre özel olarak bir kart seçer.

bağımsız tahmin.

- o Aynı değere sahip kartlar tüm tahminciler tarafından seçilirse, bu bir tahmin olarak kabul edilir. Değilse, tahminci

tahminlerinin yüksek ve düşük değerini tartışır.

- o Sonra yine, her bir tahminci özel olarak bir kart seçer ve ortaya çıkarır. Bu poker planlama süreci, bir hedefe ulaşmak için tekrarlanır.

Genel Anlaşma.

#### 164. Projelerinizdeki kullanıcı hikayesi haritalamasını ve hikayelerinin tahminini nasıl yaptınız?

- Planlama poker, tişört, boyutlandırma vb. Gibi herhangi bir tahmin tekniği kullandınız mı? Kullandığınız teknik ne olursa olsun

proje sadece çok açık bir şekilde bahsedin.

#### 165. Çevik test metodolojisi diğer test metodolojilerinden nasıl farklıdır?

- Çevik test metodolojisi, tüm test sürecinin birden çok küçük kod segmentine bölünmesini içerir. İçinde

her adımda bu kod segmentleri teste tabi tutulur.

- Ekip iletişimi gibi çevik test metodolojilerinde yer alan bir dizi ek süreç vardır.

optimum sonuçlar için stratejik değişiklikler ve diğerleri.

#### 166. Scrum takım üyelerini idare ederken projenizde karşılaştığınız en büyük zorluk nedir?

- Dolandırıcılığın ilk aşamalarında genellikle karşılaşılan zorluklar, hızı sabitlemek, ekip üyeleri çatıştır,

zaman boks vb.

- o Uygulama test edilebilecek kadar stabil olmalıdır. o Her zaman zaman kısıtlaması altında test etme

- o Gereksinimleri anlamak.

- o Alan bilgisi ve iş kullanıcı bakış açısı anlayışı.

#### 167. İlk olarak hangi testler yapılmalı?

- Tam Uygulamanın Test Edilmesi. o Regresyon testi.

- Yetenekli test uzmanlarının olmaması. o Değişen gereksinimler.

- Kaynak, araç ve eğitim eksikliği

#### 168. Scrum Master sertifikanız var mı?

- Sertifikalı bir scrum ustasıysanız, sertifika sınavı, alınan puan ve

sertifika sınavını geçtiğiniz yıl. Sertifikanız yoksa, deneyiminizden bahsedin ve vurgulayın belirlen alan. Ayrıca, yakın gelecekte sertifikasyona yatırım yapmayı planlayıp planlamadığınızı görüşmeceye bildirin.

**169. Çevik sertifikanız var mı? Neden bu sertifikayı seçtiniz?**

- Çevik ve Scrum metodolojileri bir projeyi en kısa sürede tamamlamak için kullanılır.
- Çevik ilkelerin uygulanması müşteri memnuniyetiyle sonuçlanırken, scrum, esnek özelliği ile bilinir.

Gereksinimler.

**170. Daha önce offshore ekibiyle çalıştınız mı?**

- Hayır, bilmiyorum. ( *Offshore, temelde ekibin farklı bir ülkede bulunduğu ancak yine de şirketiniz tarafından istihdam edildiği anlamına gelir* )

- 38 -

## Sayfa 39

**171. Yaygın UI test otomasyon araçları nelerdir?**

- Selenium
  - o Salatalık
  - o TestNG
- Appium
- Açılöçer
- Şarap
- UFT / QTP
- Katalon Stüdyo

**172. Test Yazılımı nedir? Test gereçleri?**

- Uygulamanın test edilmesine yardımcı olan yazılımın alt kümesidir.
- Testleri planlamak, tasarlamak ve yürütmek için test yazılımı gerekir. Belgeleri, komut dosyalarını, girdileri, beklenen sonuçları, kurulumu içerir ve testte kullanılan ek yazılım veya yardımcı programlar.
- Test yazılımı, bir yazılım paketini test etmek için gerekli olan tüm yardımcı programların ve uygulama yazılımının birleşimine verilen bir terimdir.
  - Özel olduğu için;
  - o Farklı amaç
  - o Kalite için farklı ölçütler ve
  - o Farklı kullanıcılar

**173. İstemci veya sunucu ortamı testi nasıl etkiler?**

- Veri aktarım veri aktarım hızı, donanım ve veri aktarım hızı gibi testi etkileyen birçok çevresel faktör vardır.
  - sunucu vb. istemci veya sunucu teknolojileri ile çalışırken, testler kapsamlı olacaktır.
- Zaman sınırlarımız olduğunda entegrasyon testini yapıyoruz. Çoğu durumda yükü, stresi ve performansı tercih ederiz.
  - istemci veya sunucu ortamı için uygulamanın yeteneklerini incelemek için test.

**174. Bir test uzmanının tipik deneyimleri veya aşırı yük çalışma günü veya test (SDET) kaynaklarında yazılım geliştirme mühendisi?**

- Test uzmanı için herhangi bir günde üç temel görev her zaman çok fazla zaman alır:
- Projenin gereksinimlerini anlamak.
- Hazırlama ve yürütme, müşterinin beklenen işlevlerine dayalı test senaryoları gerektirir.
- Müşteri için geliştirilen bireysel işlevlerde tanımlanan hataların geliştiriciye raporlanması ve yeniden test edilmesi
  - Beklenen işlevsellüğün herhangi bir yaygın hata olmadan düzgün bir şekilde sunulmasını sağlamak için geliştirici tarafından yeniden teslim edildikten sonra aynı.

**175. Bir test uzmanının, sağlanan ürünün gerçekten taşınmaya hazır olduğuna nasıl karar verebileceğine ilişkin bazı uzman yorumlarını açıklayın. canlı ortamda?**

- Bu kritik kararlardan biridir, bu yüzden asla tek kişi veya genç adamlar tarafından alınmamıştır. Yalnızca geliştirici ve test kullanıcısı
- Bu kararın alınmasına dahil olmadığında, üst yönetim periyodik olarak buna dahil olur. Yönetim testi esas olarak aşağıdakileri sağlar:
- Ürün teslimatının hatasız olduğundan emin olmak için aşağıda doğrulama:
- Test uzmanı tarafından sağlanan hata raporlarını doğrulama. Hatanın nasıl çözüldüğü ve test cihazı tarafından yeniden test edilip edilmediği.
- Test eden tarafından bu belirli işlev, dokümantasyon ve alınan onay için yazılan tüm test senaryolarının doğrulanması
  - aynı test cihazından.
- Yeni işlevlerin mevcut herhangi bir işlevi bozmamasını sağlamak için otomatik test senaryolarını çalıştırın.
- Bazen, gelişen tüm bileşenin test senaryoları tarafından kapsanmasını sağlayan test kapsamı raporunun doğrulanması yazılı.

- 39 -

## Sayfa 40

**176. Sağlanan ürünün gerçekten taşınmaya hazır olduğuna bir test uzmanının nasıl karar verebileceğine ilişkin bazı uzman yorumlarını açıklayın.**

**canlı ortamda?**

Bu kritik kararlardan biridir, bu yüzden asla tek kişi veya genç adamlar tarafından alınmamıştır. Yalnızca geliştirici ve test kullanıcısı

Bu kararın alınmasına dahil olmadığında, üst yönetim periyodik olarak buna dahil olur. Yönetim testi esas olarak aşağıdakileri sağlar:

Ürün teslimatının hatasız olduğundan emin olmak için aşağıda doğrulama:

- Test uzmanı tarafından sağlanan hata raporlarını doğrulama. Hatanın nasıl çözüldüğü ve test cihazı tarafından yeniden test edilip edilmediği.
- Test eden tarafından bu belirli işlev, dokümantasyon ve alınan onay için yazılan tüm test senaryolarının doğrulanması aynı test cihazından.
- Yeni işlevlerin mevcut herhangi bir işlevi bozmamasını sağlamak için otomatik test senaryolarını çalıştırın.
- Bazen, gelişen tüm bileşenin test senaryoları tarafından kapsanmasını sağlayan test kapsamı raporunun doğrulanması yazılı.

**177. Hangi Test Teknikleri var ve bunların amacı nedir?**

Test Teknikleri esas olarak iki amaç için kullanılır: a) Kusurların belirlenmesine yardımcı olmak için, b) Test senaryolarının sayısını azaltmak.

- **Eşdeğer bölümleme**, esas olarak, aşağıda belirtilen farklı veri kümelerini tanımlayarak test senaryolarının sayısını azaltmak için kullanılır.

aynı değildir ve her veri kümesinden yalnızca bir test yürütmek

- **Sınır Değer Analizi**, izin verilen verilerin sınırlarında sistemin davranışını kontrol etmek için kullanılır.
- **Durum Geçiş Testi**, izin verilen ve verilmeyen durumları ve bir durumdan diğerine geçişleri doğrulamak için kullanılır.
- çeşitli giriş verileri
- **Çift** veya **Tüm Çiftler Testi** çok güçlü bir test tekniğidir ve esas olarak test senaryolarının sayısını azaltmak için kullanılır.
- özellik kombinasyonlarının kapsamını artırmak.

**178. Bir kusur veya hata raporuna hangi bilgiler dahil edilmelidir?**

- Kusurun kısa bir özeti
- Yeniden üretme adımları da dahil olmak üzere kusurun tam açıklaması
- Gerekirse ekran görüntüsü ekleri
- Kusurun bulunduğu ve ortaya çıktığı tarih
- Kusuru kim ihbar etti.
- Kusurun ciddiyeti ve / veya Önceliği
- Kusurun hangi bileşen atandığı.

## TEST TÜRLERİ

Kodunuzdaki değişikliklerin beklendiği gibi çalıştığından emin olmak için kullanabileceğiniz birçok farklı test türü vardır. Değil Yine de tüm testler eşittir ve burada ana test uygulamalarının birbirinden nasıl farklı olduğunu göreceğiz.

**Manuel ve otomatikleştirilmiş test**

- Yüksek düzeyde, manuel ve otomatik testler arasında ayırım yapmamız gerekiyor. Manuel testler şahsen yapılır, Uygulamaya tıklayarak veya uygun araçlarla yazılım ve API'lerle etkileşime girerek. Bu çok Birisinin bir ortam kurmasını ve testleri kendisinin gerçekleştirmesini gerektirdiği için pahalıdır ve insanlara eğilimli olabilir hata, test eden kişi test komut dosyasında yazım hataları yapabilir veya adımları atlayabilir.
- Öte yandan, otomatik testler, içinde yazılmış bir test komut dosyasını çalıştıran bir makine tarafından gerçekleştirilir. ilerlemek. Bu testler, bir sınıftaki tek bir yöntemi kontrol etmekten, kullanıcı arayüzündeki bir dizi karmaşık eylem aynı sonuçlara yol açar. Otomatik testlerden çok daha sağlam ve güvenilirdir - ancak otomatik testlerinizin kalitesi, test komut dosyalarınızın ne kadar iyi yazıldığına bağlıdır.
- Otomatik test, sürekli entegrasyon ve sürekli teslimatın önemli bir bileşenidir ve işletmenizi ölçeklendirmenin harika bir yoludur. Uygulamanıza yeni özellikler eklerken KG süreci. Ancak, ne olduğu ile bazı manuel testler yapmanın hala değeri var

aşağıda göreceğimiz gibi keşif testi olarak adlandırılır.

### Farklı test türleri

#### Duman testi

- Duman testleri, uygulamanın temel işlevselliğini kontrol eden temel testlerdir. Hızlı bir şekilde uygulanmaları amaçlanmıştır ve Amaçları, size sisteminizin temel özelliklerinin beklendiği gibi çalıştığına dair güvence vermektir.
- Duman testleri, yeni bir yapı yapıldıktan hemen sonra daha pahalı testler yapıp yapamayacağınıza karar vermek için yararlı olabilir veya uygulamalarının yeni dağıtılan ortamda düzgün çalıştığından emin olmak için bir dağıtımdan hemen sonra.

#### Entegrasyon testleri

- Entegrasyon testleri, uygulamanız tarafından kullanılan farklı modüllerin veya hizmetlerin birlikte iyi çalıştığını doğrular. Örneğin, veritabanıyla etkileşimi test etmek veya mikro hizmetlerin beklendiği gibi birlikte çalıştığından emin olmak. Bu tür Uygulamanın birden çok parçasının çalışır durumda olmasını gerektirdiğinden testleri çalıştırmak daha pahalıdır.

#### Gerileme testi

- Gerileme, uygulamanın değişmemiş kısımlarının yeniden test edilmesi anlamına gelir. Regresyon testi, doğrulamak için yapılan bir testtir. Yazılımdaki bir kod değişikliğinin ürünün mevcut işlevselliğini etkilemediğini.
- Bu test, ürünün yeni eklenen işlevlerle veya üründeki herhangi bir değişiklikte daha önce olduğu gibi iyi çalıştığından emin olur. mevcut özellik veya hata düzeltilmesi yapıldıktan sonra. Daha önce yürütülen test senaryoları, etkisini doğrulamak için yeniden yürütülür. değişiklik.
- Regresyon Testi, önceki test durumlarının olup olmadığını kontrol etmek için test senaryolarının yeniden yürütüldüğü bir Yazılım Testi türüdür. uygulamanın işlevselliği iyi çalışıyor ve yeni değişiklikler herhangi bir yeni hataya yol açmadı. Bu test olabilir tek bir hata düzeltilmesinde bile orijinal işlevsellikte önemli bir değişiklik olduğunda yeni bir derlemede gerçekleştirilir.
- Bir uygulamanın bir bütün olarak herhangi bir modülde veya işlevde değişiklik için test edilmesi, Gerileme Testi olarak adlandırılır. Bu Regresyon Testinde tüm sistemi kaplamak zordur, bu nedenle bu türler için tipik olarak otomasyon test araçları kullanılır. test yapmak.

## Sayfa 42

#### Fonksiyonel testler

- İşlevsel testler, bir uygulamanın iş gereksinimlerine odaklanır. Yalnızca bir eylemin çıktısını doğrularlar ve bu eylemi gerçekleştirirken sistemin ara durumlarını kontrol edin.
- Her ikisi de birden fazla bileşen gerektirdiğinden, entegrasyon testleri ile fonksiyonel testler arasında bazen bir karışıklık olur birbirleriyle etkileşim kurmak. Aradaki fark, entegrasyon testinin veritabanını sorgulayabildiğinizi doğrulayabilmesidir. işlevsel bir test ise, ürün gereksinimleri tarafından tanımlanan veri tabanından belirli bir değeri almayı bekler.

#### FONKSİYONEL TEST

- Birim Testi
- Duman testi
- Gerileme testi
- Sağlık Testi
- Entegrasyon Testi
- Kabul testleri
- GUI Testi
- Kullanılabilirlik testi
- Sistem Testi
- Alfa Testi
- Beta testi
- Kullanıcı Kabul Testi

#### FONKSİYONEL OLMAYAN TESTLER

- Performans testi
- Yük Testi
- Hacim Testi
- Stres testi
- Güvenlik Testi
- Kurulum Testi
- Penetrasyon testi
- Uyumluluk Testi
- Göç Testi
- Kurtarma testi
- Güvenilirlik Testi
- Kullanılabilirlik testi

#### Birim testleri

- Birim testleri, uygulamanızın kaynağına yakın, çok düşük seviyeli bir bireysel yöntemleri test etmekten oluşurlar ve yazılımınız tarafından kullanılan sınıfların, bileşenlerin veya modüllerin işlevleri. Birim testleri genellikle otomatikleştirmek için oldukça ucuzdur ve sürekli bir bütünleştirme sunucusu tarafından çok hızlı bir şekilde çalıştırılabilir.

#### Uçtan uca testler

- Uçtan uca test, eksiksiz bir uygulama ortamında yazılımla bir kullanıcı davranışını kopyalar. Bunu doğrular çeşitli kullanıcı akışları beklendiği gibi çalışır ve bir web sayfası yüklemek veya oturum açmak kadar basit veya çok daha karmaşık olabilir e-posta bildirimlerini, çevrimiçi ödemeleri vb. doğrulayan senaryolar ...
- Uçtan uca testler çok kullanışlıdır, ancak gerçekleştirilmesi pahalıdır ve otomatikleştirildiklerinde bakımı zor olabilir. Birkaç önemli uçtan uca test yapılması ve daha düşük seviyeli test türlerine (birim ve entegrasyon) daha çok güvenilmesi önerilir. testler) kırılan değişiklikleri hızlı bir şekilde belirleyebilmek için.

**Kara Kutu Testi**

- Davranışsal, opak kutu, kapalı kutu, spesifikasyona dayalı veya göz göze test olarak da bilinen kara kutu testi, hakkında fazla bir şey bilmeden bir yazılımın / uygulamanın işlevselliğini analiz eden bir Yazılım Test yöntemidir. test edilen öğenin iç yapısı / tasarımı ve girdi değerini çıktı değeriyle karşılaştırır.
- 'Davranış testi' terimi, kara kutu testi için de kullanılır. Davranışsal test tasarımı siyahtan biraz farklıdır. kutu testi tasarımı, çünkü dahili bilginin kullanılması kesinlikle yasak değil, ancak yine de önerilmez.

**Blackbox testi türleri :**

- |                         |                        |                             |
|-------------------------|------------------------|-----------------------------|
| o Fonksiyonel Testler,  | o Sınır Değer Analizi, | o Grafik Tabanlı Yöntemler, |
| o İşlevsel olmayan,     | o Karar Tablosu Testi, | o Karşılaştırma,            |
| o Eşdeğerlik Bölümleme, | o Hata Tahmin Etme,    | o Durum Geçiş testleri      |

**Sayfa 43****Beyaz Kutu Testi**

- Beyaz Kutu testi, bir uygulamanın kodunun dahili mantığı hakkındaki bilgilere dayanır.
- Cam kutu Testi olarak da bilinir. Bu tür bir işlemin gerçekleştirilmesi için dahili yazılım ve kod çalışması bilinmelidir. test yapmak. Bu testler kapsamında, kod ifadelerinin, dalların, yolların, koşulların vb. Kapsamına dayanır.
- Sonuç
- Yukarıda belirtilen Yazılım Test Türleri, testin yalnızca bir parçasıdır. Ancak yine de 100'den fazla test türleri, ancak tüm test türleri tüm proje türlerinde kullanılmamaktadır. Bu nedenle, yaygın olarak kullanılan bazı Türleri ele aldım. Çoğunlukla test yaşam döngüsünde kullanılan Yazılım Testi.
- Ayrıca, farklı organizasyonlarda kullanılan alternatif tanımlar veya süreçler vardır, ancak temel kavram aynıdır her yerde. Bu test türleri, süreçleri ve uygulama yöntemleri proje ne zaman ve ne zaman proje gereksinimler ve kapsam değişiklikleri.

**Whitebox testi türleri :**

- |                   |                     |                             |
|-------------------|---------------------|-----------------------------|
| o Birim Testi,    | o Bildirim Kapsamı, | o Grafik Tabanlı Yöntemler, |
| o Yürütme Testi,  | o Şube Kapsamı,     | o Karşılaştırma,            |
| o Mutasyon Testi, | o Yol Kapsamı,      | o Durum Geçiş testleri,     |
| o Operasyon Testi | o Güvenlik Testi,   |                             |

**Kabul testleri**

- Kabul testleri, bir sistemin iş gereksinimlerini karşılayıp karşılamadığını doğrulamak için yapılan resmi testlerdir. Hepsini gerektirirler uygulama hazır ve çalışır durumda olmalı ve kullanıcı davranışlarını kopyalamaya odaklanmalıdır. Ancak daha da ileri gidebilir ve sistemin performansı ve belirli hedefler karşılanmazsa değişiklikleri reddeder.

**Performans testi**

- Performans testleri, önemli yük altında olduğunda sistemin davranışlarını kontrol eder. Bu testler işlevsel değildir ve platformun güvenilirliğini, kararlılığını ve kullanılabilirliğini anlamak için çeşitli biçime sahip olabilir. Örneğin, olabilir çok sayıda isteği yerine getirirken yanıt sürelerini gözlemlemek veya sistemin önemli ölçüde nasıl davrandığını görmek veri.
- Performans testlerinin uygulanması ve çalıştırılması doğaları gereği oldukça maliyetlidir, ancak yeni değişiklikler olup olmadığını anlamaya yardımcı olabilirler sisteminizi bozacak.

**Ad-hoc Test**

- İsmi kendisi, bu testin amaca yönelik olarak gerçekleştirildiğini, yani test senaryosuna atıfta bulunmadığını ve ayrıca bu tür testler için herhangi bir plan veya belge olmadan. Bu testin amacı kusurları bulmak ve uygulamanın herhangi bir akışını veya herhangi bir rastgele işlevselliği çalıştırarak uygulamayı bozun.
- Ad-hoc test, kusurları bulmanın gayri resmi bir yoldur ve projedeki herkes tarafından yapılabilir. Tanımlamak zor test senaryosu olmayan kusurlar, ancak bazen geçici test sırasında bulunan kusurların olmaması mümkündür. mevcut test senaryoları kullanılarak tanımlanır.

**Sınır Değer Testi**

- Bu tür testler, uygulamanın sınır düzeyindeki davranışını kontrol eder.
- Sınır değer Test, sınır değerlerinde kusur olup olmadığını kontrol etmek için gerçekleştirilir. Sınır değer testi aşağıdakiler için kullanılır: farklı bir sayı aralığını test etmek. Her aralık için bir üst ve alt sınır vardır ve test yapılır. bu sınır değerleri.
- Test, 1'den 500'e kadar bir test aralığı gerektiriyorsa, Sınır Değer Testi 0, 1, 2'deki değerler üzerinde gerçekleştirilir, 499, 500 ve 501.

---

## Sayfa 44

### Keşif testi

- Keşif Testi, test ekibi tarafından gerçekleştirilen gayri resmi testlerdir. Bu testin amacı, uygulama ve uygulamada var olan kusurları aramak. Bazen bu test sırasında büyük keşfedilen kusur sistem arızasına bile neden olabilir.
- Keşif testi sırasında, hangi akışı test ettiğinizi ve daha önce hangi aktiviteyi yaptığınızı takip etmeniz önerilir. belirli akışın başlangıcı.

### Testlerinizi nasıl otomatikleştirebilirsiniz?

- Bir kişi yukarıda bahsedilen tüm testleri uygulayabilir, ancak bunu yapmak çok pahalı ve verimsiz olacaktır. Gibi insanlar, çok sayıda eylemi tekrarlanabilir ve güvenilir bir şekilde gerçekleştirme kapasitemiz sınırlıdır. Ama bir makine bunu kolayca yapabilir ve giriş / şifre kombinasyonunun 100. kez şikayet etmeden çalıştığını test eder.
- Testlerinizi otomatikleştirmek için, önce testlerinizi kendinize uygun bir test çerçevesi kullanarak programlı olarak yazmanız gerekir. uygulama. [PHPUnit](#) , [Mocha](#) , [RSpec](#) , PHP, JavaScript ve Ruby için kullanabileceğiniz test çerçeveleri örnekleridir. sırasıyla. Her dil için birçok seçenek var, bu yüzden biraz araştırma yapıp sormanız gerekebilir sizin için en iyi çerçevenin ne olacağını öğrenmek için geliştirici toplulukları.
- Testleriniz terminalinizden komut dosyası aracılığıyla yürütüldüğünde, bunların bir Bamboo gibi sürekli entegrasyon sunucusu veya Bitbucket Pipelines gibi bir bulut hizmeti kullanın. Bu araçlar

- 44 -

---

## Sayfa 45

# JAVA

### 1. Java Sanal Makinesi

- JVM, derlenmiş java sınıfı dosyaları için bir çalışma zamanı ortamı olan Java Sanal Makinesi anlamına gelir.

### 2. JavaScript ve Java aynı mı?

- Java bir OOP programlama dilidir, Java Script ise bir OOP kodlama dilidir.
- Java, JavaScript kodu yalnızca bir tarayıcıda çalıştırılırken sanal bir makinede veya tarayıcıda çalışan uygulamalar oluşturur.
- JavaScript kodunun tamamı metin içindeyken Java kodunun derlenmesi gerekir.
- Farklı eklentiler gerektirirler.

### 3. Java Runtime Environment

- JRE, bir Java programını çalıştırmak için ihtiyacımız olan şeydir ve JVM'nin çalışma zamanında kullandığı kitaplıklar ve diğer dosyaları içerir.