

# INTERVIEW PREP

# MÜLAKATLARA HAZIRLIK



Batch 81



# Interview Preplerde Neler Yapacağız

- ✔ Softskill
- ✔ Java
- ✔ Selenium
- ✔ SQL
- ✔ Mock Interviewler



# Mülakatlara Bakış Açımız Nasıl Olmalı?

Mülakatı bir muharebe, bir taktik savaşı gibi görmemek lazım.

Bu şekilde düşünüldüğünde masanın karşı tarafını gözümüzde büyütmüş oluruz ve kendimizi doğru bir biçimde ifade edemeyiz.

Doğal ve içten davranmalıyız.

Bu doğallığı sağlayabilmek için önceden hikayemizi kafamızda oturtmalı, hikayemizi doğru bir biçimde aktarabilmek için uygun kelime ve cümleler seçmeli ve mutlaka önceden provasını yapmamız gerekir.

Şunu unutmamak lazım, **sizin o işe ihtiyacınız olduğu oranda onların da size ihtiyacı var.**



# Kendinizden Bahseder misiniz?

**Tekrarı olmayan tek şey ilk izleniminizdir.**

Bu soruya cevap verirken dikkat edilmesi gereken hususlar

- 1.Çok detay vermeyin
- 2.Özgeçmişinizde yer alan her şeyi karşı tarafa anlatmaya çalışmayın
- 3.Kişisel ve özel bilgilere yer vermeyin

Bu soruya 3 ana başlık altında cevap verebiliriz..

## **1- Siz Kimsiniz....**

Kendinizi profesyonel bir şekilde tanıtacak birkaç cümle kullanılabilir.

## **2- Potansiyel Hikayeniz....**

Öne çıkan yetenekleriniz, başarılarınız, çalıştığınız kurumlara katkılarınızdan bahsedebilirsiniz.

## **3- Neden Buradasınız?**

Neden bu işi tercih ettiniz? Orada bulunma heyecan ve motivasyon ve isteğinizi karşı tarafa göstermek gerekir.



# Kendinizden Bahseder misiniz?

Ben Ahmet,

X üniversitesi y fakültesi mezunuyum

Bir süre mezuniyetimle alakalı işler yaptıktan sonra , gerek IT dünyasına ilgim gerekse IT dünyasının avantajları sebebi ile bu alana yöneldim. Bu alanı çok ta sevdim. X zamandır bu alanda ilgileniyorum.

X kadar süredir yazılım testi ile ilgileniyorum. UI ve backend testleri yapıyorum. UI test için java ile birlikte Selenium Web Driver , JUNIT, TestNG ve Cucumber , Cypress, Appium toollarını kullanıyorum. Backend Tesleri için API, SQL , JDBC , Postman gibi araçlar kullanıyorum. Performans testleri için Jmeter kullanıyorum.

Çok detaycı olduğumu düşünüyorum. Bu durum beni bazen çok yorsa bile test senaryoları geliştirmemde bana büyük katkı sağlıyor.

Ben takım içi iletişime ve iş bölümüne çok önem veririm. Takım içinde arkadaşlarıma katkı sağlamaktan ve onların tecrübelerinden faydalanmaktan büyük mutluluk duyuyorum. Böylece kendi kişisel yeteneklerimizi geliştirirken, şirkete de büyük katkı sağlayabileceğimizi düşünüyorum.

Bu iş görüşmesi beni heyecanlandırdı. İş gereksinimlerinizin benim skillerime uygun olduğunu düşünüyorum. Siz de uygun görürseniz sizinle çalışmaktan mutluluk duyarım.



# SDLC Nedir?

Bu soruyu cevaplarken teoriden çok, Sdlc kültürünü anladığımızı gösteren bir cevap daha etkili olacaktır. Önce genel bir tanımı yapıp bir projede nasıl kullanıldığından bahsetmekte yarar var.

Bir uygulamanın, kullanıcıların beklentilerini karşılayacak bir şekilde piyasaya sürülebilmesi için takip edilen yazılım yaşam döngüsüdür.

SDLC ortak bir amaç çerçevesinde toplanmış takım üyeleri ile gerçekleştirilen bir takım çalışmasıdır. Her aşamasında ekibin başka bir üyesi yer alır. Bu ekip kendi içinde koordinasyonu, birlikte yaptığı toplantılarla ve proje takip programları ile sağlar. Kullanılan yazılım geliştirme metodolojisine göre işleyiş ve toplantılarda farklılıklar olabilir. Ben agile a hakimim. Bu konuda dilerseniz detay verebilirim.



# Takım yapınızdan bahseder misiniz

Burada bitirme projesi göz önünde bulundurularak cevap verilebilir. Yada gerçekte olması gereken şekilde cevap verilebilir

## **Örnek Senaryo 1 :**

Bizim takımımız 10 kişiden oluşuyordu. Bir PO, bir SM ve 6 dev ve 2 qa olarak çalışıyorduk. Diğer qa arkadaş manuel tester ben ise otomasyon tester olarak çalıştım. Agile da scrum ile çalışıyorduk. Sprintler 2 haftalıktı. Sprint başlangıcında sprint planning, sprint boyunca her sabah Daily scrum, sprint sonlarında ise sprint review ve retro toplantıları ile iletişim halinde oluyorduk. Ekip full remote bir ekip olduğu için gün içerisinde ihtiyaç duydukça birbirimizle iletişim kurabiliyorduk.

## **Örnek Senaryo 2 :**

Bizim takımımız bir qa ekibi idi. Ekte 4 tester olarak çalışıyorduk. PO ve SM da toplantılarımıza dahil oluyordu. Developerların ise kendi arasında bir takımı vardı. Aradaki iletişimi Jira ve arada yaptığımız ortak toplantılarla sağlıyorduk.



# Testing Neden Önemli/ Neden Test Ediyoruz

- Hatasız ve kullanıcı dostu uygulama oluşturmak için.
- Son kullanıcı ve müşteriye memnun etmek için.
- yüksek kaliteli ürünler oluşturmak için

## Peki Bir testerin ana sorumluluğu nedir bu durumda ?

**Test uzmanı hatayı en erken şekilde bulur ve hatanın düzeltildiğinden emin olur**

Uygulamanın hatasız olduğunu doğrulayarak kalite düzeyini yükseltir. Uygulamanın kullanıcı dostu olduğundan emin olur. Bu şekilde müşteri ve kullanıcı memnuniyeti sağlanmış olur.

Kısaca **Create value** için diyebiliriz.





# Manuel Test VS Automation Test

Manuel test, testerlar için temel olan aktivitedir.

Uygulama fonksiyonları hatadan arındırılana kadar manuel olarak test edilir.

Uygulamanın fonksiyonları stabil hale geldikten sonra, aynı fonksiyonun defalarca test edileceği durumlarda, zamandan kazanmak adına testler otomasyona dökülür.



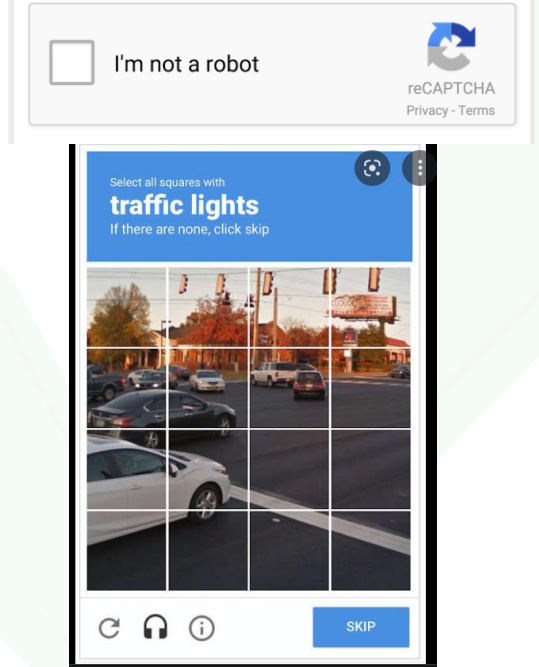
# Manuel Test VS Automation Test

## Tüm testleri otomasyona dökülebilir miyiz?

Tüm testlerin otomasyona dökülmesi mümkün değildir.  
Eğer web uygulaması test ediyorsak,

- Captcha uygulamaları otomasyona dökülemez
- Grafik ve resimler otomasyona dökülemez
- Tasarımsal hatalar otomasyona dökülemez

**Not:** Tüm bunların dışında bazı fonksiyonlar otomasyona dökülebilmesine rağmen otomasyon tercih edilmez. Bunun sebebi fonksiyonun dinamik / değişken olması ya da ürünün geliştirme aşamasında olmasından kaynaklanabilir.





# Test Caseleri Nasıl Yazarsınız?

- 1- Requirement** : Gereksinimleri çok iyi anlarım
- 2- Acceptance Criteria** : Test case lerin tüm kabul kriterlerini kapsamasına dikkat ederim
- 3- Kara Kutu Test Teknikleri** : Mümkün olduğunca geniş bir alanı kapsayacak min sayıda test case yazmak için test tekniklerini kullanırım.

## Ne gibi teknikler kullanıyorsunuz?

- 1-Denklik Paylarına Ayırma --Equivalence Partitioning (EP)
- 2-Sınır Değer Analizi --Boundary Value Analysis
- 3-Karar Tablosu Testi --Decision Table Testing
- 4-Kullanım Durum Senaryosu Testi --Use Case Testing
- 5-Durum Geçiş Testi --State Transition Testing



# Test Ortamlarınız Nelerdir?

Uygulamada 4 tane enviroment var.

- 1- Dev enviroment : Bu ortam developerların kullandığı ortamdır. Developerlar kendi birim testlerini tamamladıktan sonra uygulama qa senviromenta aktarılır.
- 2- QA enviroment : Bu ortam testerların testlerini yürüttüğü ortamdır. Hata ile karşılaşıldığında developera bilgi verilir. Developer düzeltmesini dev enviromentta yaptıktan sonra ilgili fonksiyon tekrar qa enviromenta aktarılır
- 3- Stage (Pre-prod) enviroment : testerlar testlerini tamamlayıp, uygulama buglarından arındırıldığında uygulama bu ortama aktarılır. Bu ortam canlı ortamın bir kopyasıdır. UAT ( Kullanıcı kabul testi) Testleri bu ortamda yapılır
- 4- Prod enviroment : UAT den başarı ile geçen uygulamanın aktarıldığı ortamdır. Son kullanıcının kullandığı ortamdır.

**NOT : Ortamları ayağa kaldırmak DevOps ın sorumluluğundadır...**



# Test Execution İşlemi (Testleri Run Etme , Çalıştırma)

## Manuel yürütülen testler

---

- Test takibi excel ile yapılıyor olabilir
- Testler jira ile takip ediliyor olabilir.

## Otomasyon ile yürütülen testler

---

- Testler localde koşulur. Problem yoksa kodlar githuba gönderilir.
- Testler sunucu üzerinde yada bulutta Jenkins aracılığı ile koşulabilir.



# Developer Kendi Kodunu Test Ediyorsa

## Neden QA'e İhtiyaç Var?

Developerlar whitebox test yapar, testi yaparken kodlarının davranışını baz alır. Bakış açısı kodun davranışına odaklandığı için kullanıcı gözünden değerlendiremeyebilir.

Developerlar yakın körlüğü yaşayabilir.( Bir konu üzerinde çok çalıştığınızda hataları göremeyebilirsiniz )

Testerlar ise black box test yaparlar, uygulamanın kodları ile ilgilenmezler. Tamamen gereksinim temelli , kullanıcı bakış açısı ile testlerini yazarlar

Yaşınız

Developer bakış açısı bu kısmın numeric olmasıdır. ( 20 )  
Ancak kullanıcı yazı ile de yazabilir. ( yirmi )



# Ne tür testler yapıyorsunuz?

UI ve backend te api ve database testleri yapabiliyorum.

Integration, smoke, regression, E2E testleri koşuyorum.  
Bu testleri manuel ya da otomasyonla gerçekleştirebilirim.

## **Bu tür testlere örnek verebilir misiniz**

Bizim için kritik öneme sahip testleri smoke test kapsamında çalıştırıyoruz. Bu testler her gün çalıştırılır.

Regression da tüm testlerin toplu bir şekilde çalıştırılmasını sağlıyoruz. Böylede yeni eklenen özelliklerin herhangi bir kısma etki edip etmediğini test etmiş oluyoruz.



# Bir tost makinasını nasıl test edersiniz?

(Tost makinası yerine herhangi bir ürün kullanılabilir)

Herhangi bir ürünün kalitesi, onun için hazırlanan dokumana göre değerlendirilir.

Ürünün kullanım kılavuzu ve üzerinde yazan ürün özelliklerini baz alarak testlerimizi gerçekleştirmeliyiz.

Örneğin ürün üzerinde tost makinasının plaka kalınlığı 1 mm olarak belirtilmişse, diğer makinaların 5 mm bile olsa, bu ürüne kalitesiz denemez.

Yani müşteriye belirtilen özellikleri kapsıyor olup olmadığı kalitesini belirtir.





# WhiteBox VS BlackBox

- White box testing Beyaz kutu testi, işlevselliğinin aksine bir uygulamanın iç yapılarını veya çalışmalarını test eden bir yazılım testi yöntemidir. Beyaz kutu testinde, test senaryolarını tasarlamak için sistemin iç perspektifinin yanı sıra programlama becerileri de kullanılır.
  - Unit testing(birim testi.) code temelli developerlar tarafından yapılan testlerdir. her birimi ayrı ayrı test etmek için yapılır. bu test genellikle küçük code birimleri için yapılır ve classdan daha büyük değildir.
- The black box testi aynı zamanda Functional testing olarak da adlandırılır ve verilen bilgilere dayanarak test senaryoları tasarlayan bir teknik. Black box testing sistemin iç yapısının bilinmesine gerek duyulmadan icra edilen, çoğunlukla test mühendisleri tarafından yapılan bir test tekniğidir.

NOT : Black-box testing **gecerlilik(validation)** için yapılır. white box testing genellikle **doğrulama(verification)** için yapılır.



# Defect ve Bug arasındaki fark nedir?

**Defect :** Uygulamanın geliştirme aşamasında karşımıza çıkan kusurlara denir. Bunu developerda bulabilir , testerda. Önemli olan hangi aşamada bulunmuş olmasıdır

**Bug:** Uygulama tamamlandıktan sonra bulunan hatalardır. Bu hataları testerlarda bulabilir yada son kullanıcı da bulabilir. Bu tür hatalara bug denir



# Bir bug ile karşılaştığınızda neler yaparsınız?

- Bir bug bulduğumuzda öncelikle bunun bug olduğundan emin olmamız gerekiyor.
- Bunun için testimizi tekrarlamalı, gerekirse başka tester var ise onun fikrini almamızda fayda var.
- Eğer developerlar ile aynı ortamda çalışılıyorsa, developer durumdan haberdar edilebilir. Developer bunun bug olmadığını, bug olduğunu ya da önemsiz bir durum olduğunu ifade edebilir.
- Developerin dönüşüne göre hareket edilir.
- Eğer bug olduğu konusunda emin olunmuşsa jira üzerinden bug ticket açılır ve ilgili kişiye assign edilir.
- Bu işlemleri genelde tester ekibinin lideri yapar.



## Zayıf yönlerinizden bahseder misiniz?

Burada bir özelliğimizi doğrudan zayıf gibi göstermek yerine, olumlu kısmına da değinmek gerekir

**Senaryo 1 :** İşini aşırı sahiplenene biriyim. Bu işveren açısından olumlu olabiliyorken , kendimi fazlaca yormama sebep olabilir.

**Senaryo 2 :** Mükemmeliyetçi ve detaycı biriyim. Bu temiz bir iş çıkarmama sebep olsa da bazen beni yavaşlatabiliyor.



# Güçlü yönlerinizden bahseder misiniz?

**Senaryo 1 :** Öğrenmeyi çok seviyorum. Günceli takip etmeyi, yaptığım işlerde güncel toolları kullanmayı tercih ederim. Değişikliklere çok kolay adapte olurum. Çalışma ortamında, sistemlerde güncelleme ya da değişiklikler olduğunda kolaylıkla adapte olabiliyorum.



# Java'nızı puanlar mısınız?

Ben bir puanlama yapmak yerine size neleri rahat bir şekilde kullanabildiğimi söyleyebilirim

- 1- OOP Concept
- 2-Data Structure
- 3-String Manipulation
- 4-Exceptions

Tabi ki bu cevabın yeni sorulara gebe olduğunu unutmamak gerekir. İfade ettiğimiz her şeyin açıklaması bizden istenebilir.