



BAYKAR TECHNOLOGY

YAZILIM TESTİ

BAHAR DÖNEMİ STAJ SUNUMU



YAZILIM TESTİ NEDİR?



Yazılım testi, bir yazılımin, kendisinden beklenen özellikleri karşılayıp karşılayamadığını incelemek amacıyla yapılan işlemlerdir. Bu şekilde yazılımdaki hatalar bulunup düzeltilebilir ve gereksinimlere uygun hale getirilebilir.

Yazılım Testinin Amaçları Nelerdir? Neden Gereklidir?

- Gereksinimlerin ne kadar karşılandığından emin olmak,
- Ürünün kalite düzeyine güven oluşturmak,
- Ürünün test işleminin tamamlandığını ve müşterinin beklediği şekilde çalıştığını onaylamak,
- Hataları ve kusurları bulmak ve önlemek,
- Yetersiz yazılım ve kalite riskini azaltmak,
- Yasal standartlarına uymasını sağlamak bunlardan başlıcalarıdır.



YAZILIM TEST SEVİYELERİ

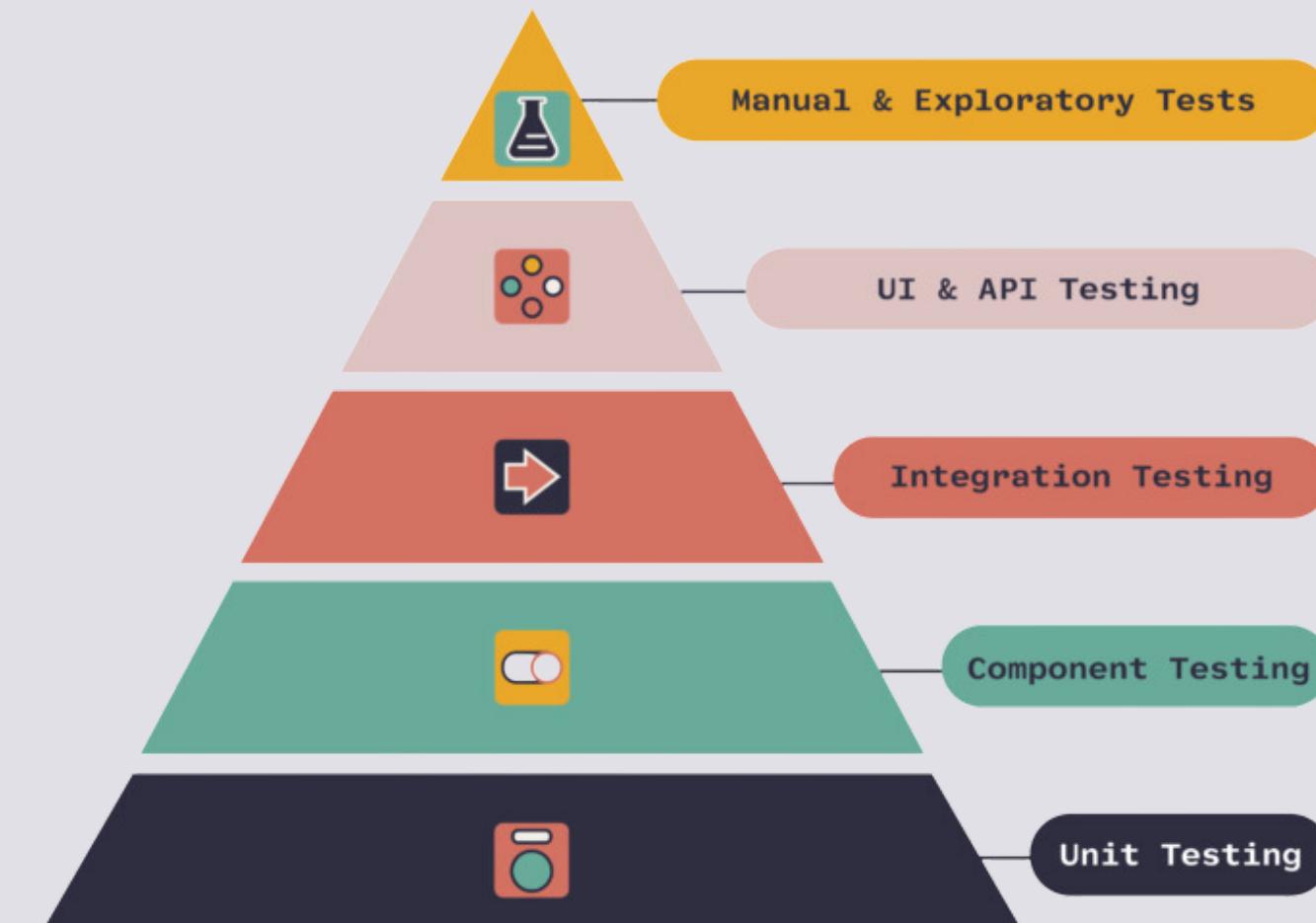


FONKSİYONEL TESTLER

- Test uzmanı sistemin nasıl çalıştığını kontrol eder.
- Kullanıcı gereksinimlerini baz alır.
- Sistem testidir.
- Uygulamanın davranışsal akışlarını içerir.
- Bu test tipi : Duman testi, Regresyon testi gibi testleri kapsar.

FONKSİYONEL OLMAYAN TESTLER

- Sistemin nasıl cevap/response verdiği kontrol edilir.
- Kullanıcı bekłentisini baz alır.
- Sistem testidir.
- Uygulamanın performansını, kontrollerini içerir.
- Bu test tipi: Yük testi, Performans testi, Güvenlik testi gibi testleri kapsar.



TEST AŞAMALARI



BİRİM TESTLERİ

- Birim testi yazılım test seviyesindeki ilk adımdır.
- Kodu yazan yazılımcı tarafından kodlanır ve koşulur.

ENTEGRASYON TESTLERİ

- Entegrasyon testleri, birimler veya sistemler arasındaki etkileşimlere odaklanır.
- Bu testlerde sistemin işleyisi değil, diğer sistemlerle nasıl entegre olduğu test edilmelidir.
- Birim ve Sistem Entegrasyonlarına ayrırlırlar.
- Riskin azaltılması, arayüzlerin kalitesine dair güvenin oluşturulması, hataların bulunması.
- Veri uyşuzrazlığı, arayüz hataları ve sistemler arası iletişim hatalarını bulur.



OTOMATİZE KULLANICI ARAYÜZ TESTLERİ

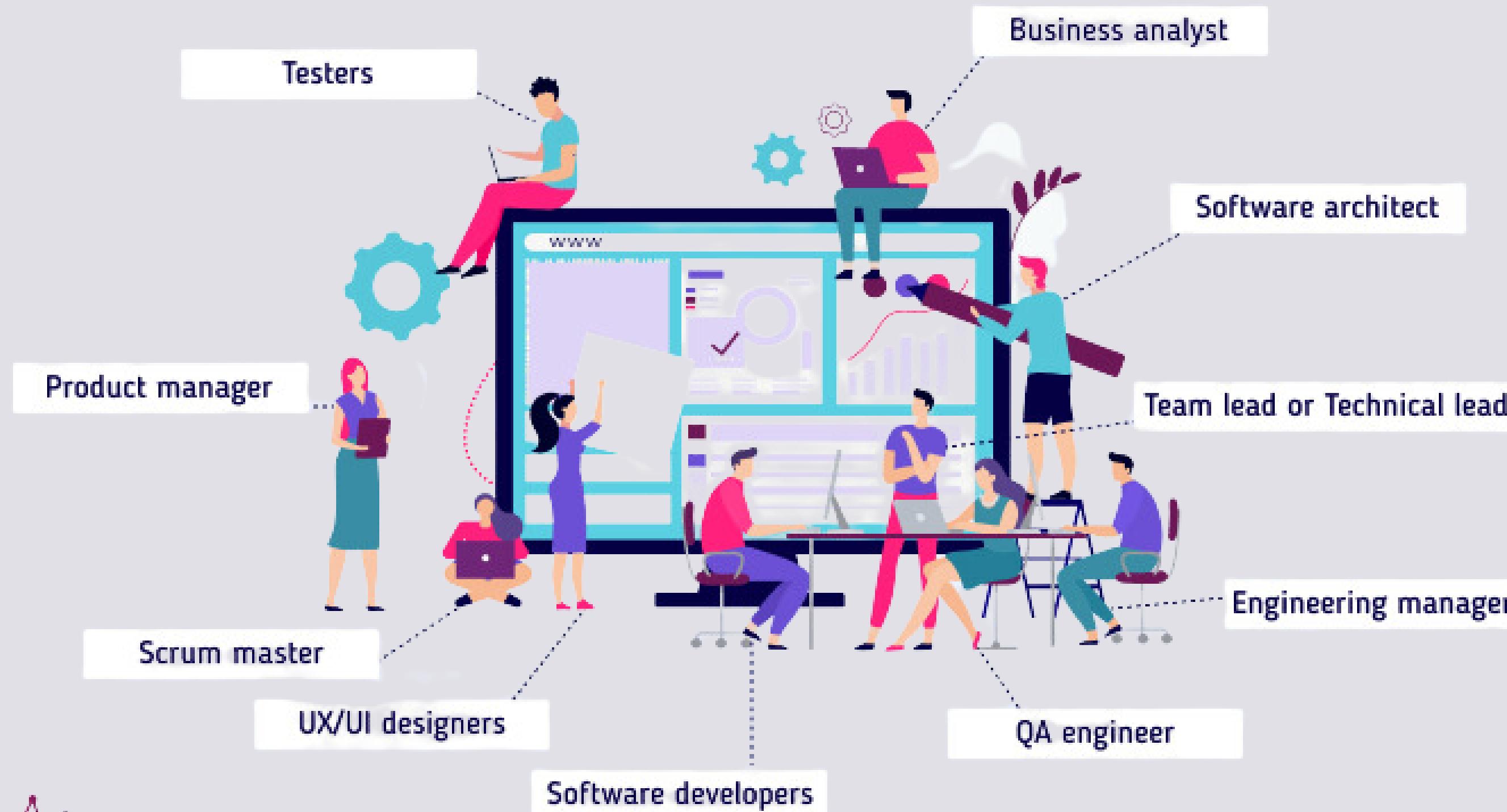
- Bu testler uygulamanızın önyüzünün doğru şekilde çalıştığını otomatize şekilde kontrol eder. Bu tip testler kodlanırken kullanıcının adımları oluşturulur ve uygulamada bu adımlar takip edilir.
- Otomatize önyüz testlerinin yazımı genel olarak daha fazla zaman alır ve sürekli bakım gerektirir. Bu sebeple mümkün olduğunda test adımı daha erken safhalarда otomatize edilmelidir.

SİSTEM VE KABUL TESTLERİ

- Sistem testleri, bütün bir sistemin veya ürünün davranışı ve yeteneklerine odaklanır ve genellikle sistemin gerçekleştirebileceği uçtan uca işleri ve bu işleri gerçekleştirken gösterdiği fonksiyonel olmayan davranışları ele alır.
- Riskin azaltılması
- Sistemin tamamlandığının ve beklentiği gibi çalışacağıının sağlanması
- Bir bütün olarak sistemin kalitesine dair güven oluşturulması
- Hataların bulunması ve hataların daha üst test seviyelerine veya canlıya çıkışının önlenmesi

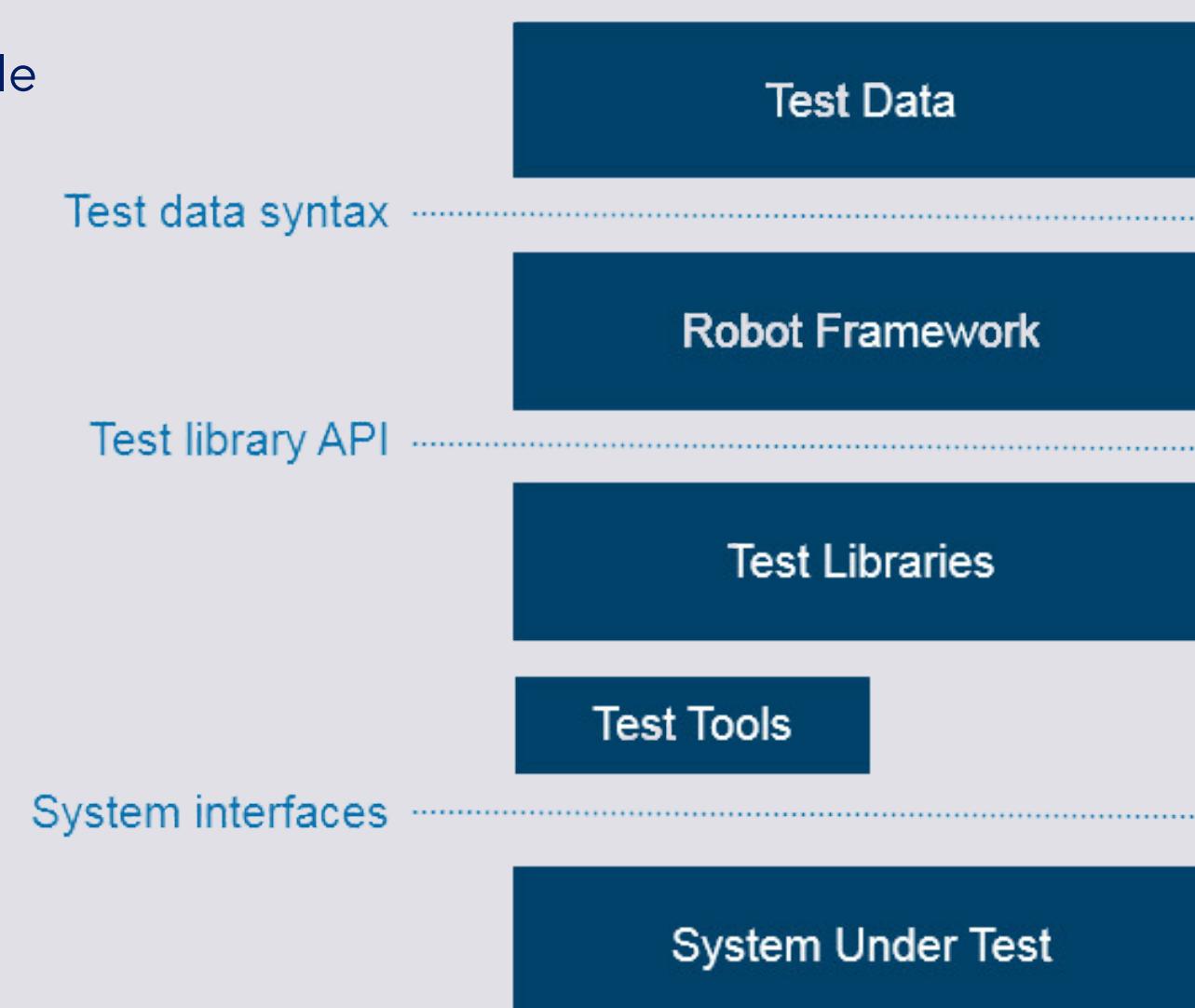
TEST AŞAMALARI

Your perfect development team



OTOMASYON TESTİ

- Farklı platformlarda test edilecek test planlarını manuel olarak daha kolay test edebilir.
- Tüm iş akışlarının, tüm alanların, tüm olumsuz senaryoların testinin zaman ve maliyetini azaltır.
- Yeniden kullanılabilir.
- İnsan müdahalesi gerektirmez.
- Test kapsamının artırılmasına yardımcı olur.
- Geliştirilmiş hata tespiti sağlar
- Daha iyi raporlama sağlar.



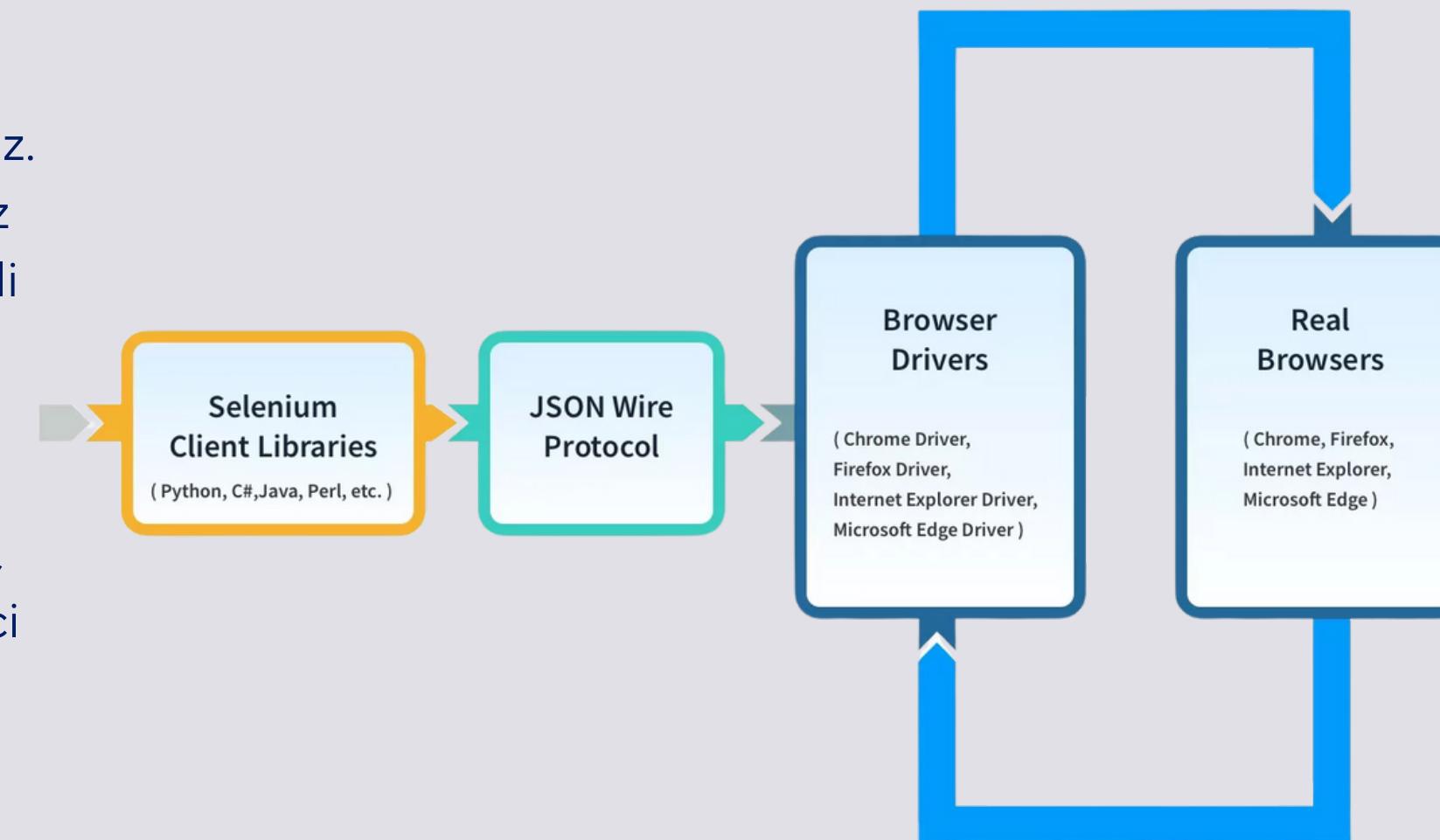
ROBOTFRAMEWORK

Robotframework ile web projelerinizin UI testlerini, API testlerini, Güvenlik Testlerini otomatik hale getirebilirsiniz. Robot framework bu testleri sizin adınıza, sizin istediğiniz zaman dilimlerinde koşar ve size geniş bir raporlama maili gönderir. Robot, python ile yazılmış bir framework'dür.

- Robot framework içerisinde Python'ın temel kütüphanelerini barındırır. Bununla birlikte selenium, appium, ssh, faker, debug, database gibi bir çok harici modül desteği sunar.

SELENIUM

- Selenium, web tarayıcılarını otomatik olarak kontrol etmek için geliştirilmiş güçlü bir araçtır. Robot Framework, Selenium Library'i kullanarak web uygulamalarını otomatik olarak test etme yeteneği kazanır. Bu sayede kullanıcıların web tarayıcıları üzerinde gerçekleştirecekleri işlemler otomatik olarak tekrarlanabilir ve test edilebilir hale gelir.



SELENIUM LIBRARY



Testler ⇒ Framework ⇒ Selenium ⇒ Tarayıcı (Web Uygulaması)



ROBOTFRAMEWORK RAPORLARI

Robot Framework, test sonuçlarını HTML formatında sunar. Bu raporlar, testlerin ayrıntılı geçmişini, başarılı ve başarısız senaryoları, hata ve uyarıları içerir. HTML raporları, kullanıcı dostu bir görünüm sunar ve test sonuçlarının görsel olarak anlaşılmasını sağlar.

REPORT			
- TEST	Check Test record present in Categories Table	00:00:00.000	PASS
Full Name:	AspNet.Check Test record present in Categories Table		
Start / End / Elapsed:	20230513 10:08:42.812 / 20230513 10:08:42.813 / 00:00:00.001		
Status:	PASS		
- KEYWORD	DatabaseLibrary.Check If Exists In Database SELECT CategoryID FROM Categories WHERE categoryName = 'Test';	00:00:00.000	
Documentation:	Check if any row would be returned by given the input 'selectStatement'. If there are no results, then this will throw an AssertionError. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.		
Start / End / Elapsed:	20230513 10:08:42.813 / 20230513 10:08:42.813 / 00:00:00.000		
10:08:42.813 [INFO] Executing : Check If Exists In Database SELECT CategoryID FROM Categories WHERE categoryName = 'Test';			
10:08:42.813 [INFO] Executing : Query SELECT CategoryID FROM Categories WHERE categoryName = 'Test';			
- TEST	Check Test1 record not present in dbo.Categories Table	00:00:00.000	PASS
Full Name:	AspNet.Check Test1 record not present in dbo.Categories Table		
Start / End / Elapsed:	20230513 10:08:42.814 / 20230513 10:08:42.814 / 00:00:00.000		
Status:	PASS		
- KEYWORD	DatabaseLibrary.Check If Not Exists In Database SELECT CategoryID FROM Categories WHERE categoryName = 'Test1';	00:00:00.000	
Documentation:	This is the negation of check_if_exists_in_database'.		
Start / End / Elapsed:	20230513 10:08:42.814 / 20230513 10:08:42.814 / 00:00:00.000		
10:08:42.814 [INFO] Executing : check If Not Exists In Database SELECT CategoryID FROM Categories WHERE categoryName = 'Test1';			
10:08:42.814 [INFO] Executing : Query SELECT CategoryID FROM Categories WHERE categoryName = 'Test1';			
- TEST	Verify Row Count is Equal to X	00:00:00.000	PASS
Full Name:	AspNet.Verify Row Count is Equal to X		
Start / End / Elapsed:	20230513 10:08:42.815 / 20230513 10:08:42.815 / 00:00:00.000		
Status:	PASS		
- KEYWORD	DatabaseLibrary.Row Count Is Equal To X SELECT CategoryID FROM Categories;, 32	00:00:00.000	
Documentation:	Check if the number of rows returned from 'selectStatement' is equal to the value submitted. If not, then this will throw an AssertionError. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.		
Start / End / Elapsed:	20230513 10:08:42.815 / 20230513 10:08:42.815 / 00:00:00.000		
10:08:42.815 [INFO] Executing : Row Count Is Equal To X SELECT CategoryID FROM Categories; 32			
10:08:42.815 [INFO] Executing : Row Count SELECT CategoryID FROM Categories;			
- TEST	Verify Row Count is Greater to Some Value	00:00:00.001	PASS
Full Name:	AspNet.Verify Row Count is Greater to Some Value		
Start / End / Elapsed:	20230513 10:08:42.816 / 20230513 10:08:42.817 / 00:00:00.001		
Status:	PASS		
- KEYWORD	DatabaseLibrary.Row Count Is Greater Than X SELECT * FROM dbo.Categories WHERE CategoryName='Test', 2	00:00:00.000	
Documentation:	Check if the number of rows returned from 'selectStatement' is greater than the value submitted. If not, then this will throw an AssertionError. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.		
Start / End / Elapsed:	20230513 10:08:42.816 / 20230513 10:08:42.816 / 00:00:00.000		
10:08:42.816 [INFO] Executing : Row Count Is Greater Than X SELECT * FROM dbo.Categories WHERE CategoryName='Test'; 2			
10:08:42.816 [INFO] Executing : Row Count SELECT * FROM dbo.categories WHERE CategoryName='Test';			
- TEST	Update record in Categories table	00:00:00.001	PASS
Full Name:	AspNet.Update record in Categories table		
Start / End / Elapsed:	20230513 10:08:42.817 / 20230513 10:08:42.818 / 00:00:00.001		
Status:	PASS		
- KEYWORD	S{output} = DatabaseLibrary.Execute Sql String Update dbo.Categories set CategoryName='Robot' where CategoryID=11;	00:00:00.001	
Documentation:	Executes the sqlString as SQL commands. Useful to pass arguments to your sql. Set optional input 'sansTran' to True to run command without an explicit transaction commit or rollback.		
Start / End / Elapsed:	20230513 10:08:42.817 / 20230513 10:08:42.818 / 00:00:00.001		
10:08:42.817 [INFO] Executing : Execute SQL String Update dbo.categories set CategoryName='Robot' where CategoryID=11;			
10:08:42.818 [INFO] S{output} = None			

ROBOTFRAMEWORK RAPORLARI



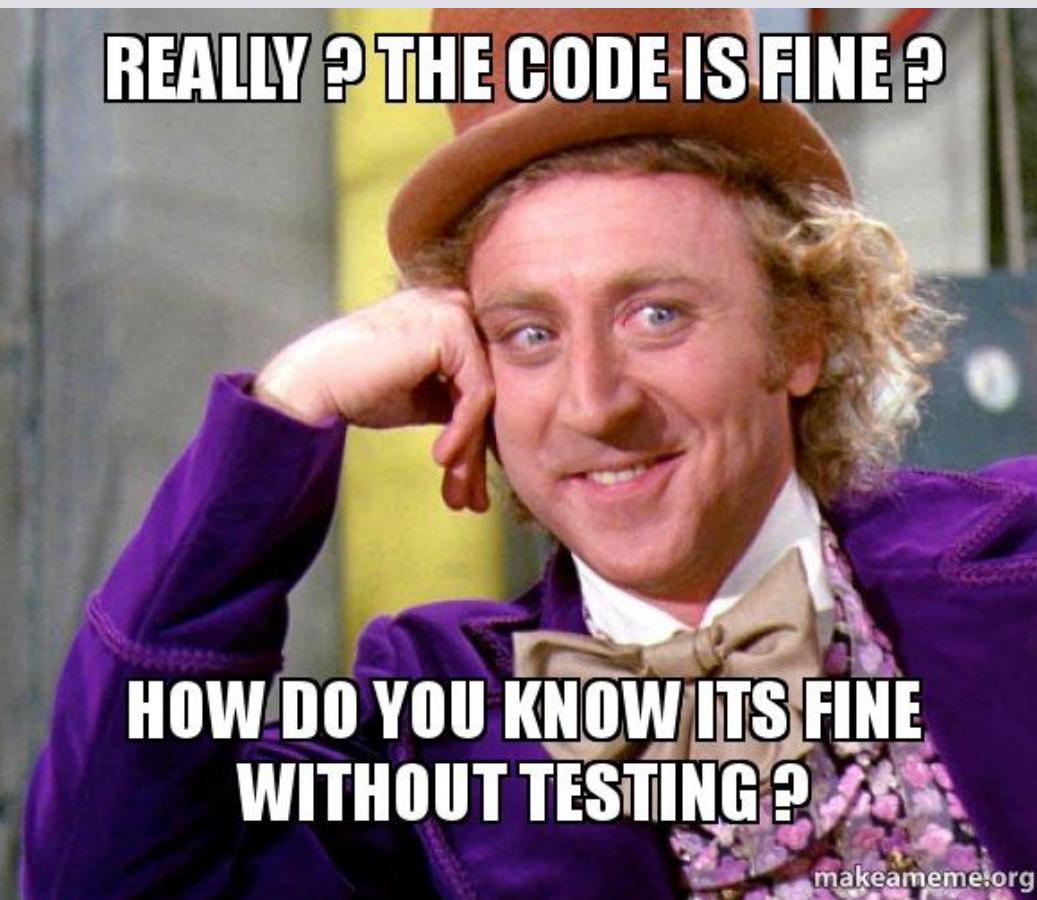
VERİTABANI TESTİ

DatabaseLibrary, Robot Framework ile veritabanı işlemlerini gerçekleştirmek için kullanılan bir kütüphanedir. Bu kütüphane, çeşitli veritabanı sistemleriyle (örneğin, MySQL, PostgreSQL, Oracle, SQL Server vb.) etkileşim kurmak ve veritabanı işlemlerini otomatikleştirmek için kullanılır.

```
1  *** Settings ***
2  Library    DatabaseLibrary
3  Library    OperatingSystem
4
5
6  Suite Setup      Connect To Database  ${DBConnect}    ${DBName}    ${DBUser}    ${DBPass}    ${DBHost}    ${DBPort}
7  Suite Teardown    Disconnect From Database
8
9  *** Variables ***
0  ${DBConnect}    pyodbc
1  ${DBName}        NORTHWIND
2  ${DBUser}        sa
3  ${DBPass}        12341234
4  ${DBHost}        localhost
5  ${DBPort}        49820
```

YAZILIM TESTİNİN ÖNEMİ

Araştırma raporlarına göre Türkiye'de firmaların test süreçlerine harcadığı zaman ve bütçe, toplam proje maliyetlerinin %7'sine karşılık geliyor. Dünya ortalamasının çok altında olan bu yaklaşımın sonucu olarak, test süreçlerine proje içinde gerektiği kadar önem verilmediği veya zaman ayrılmadığı için uygulamalarda "kalite problemleri ile sıkılıkla karşılaşmaktadır" sonucu ortaya çıkıyor.



YAZILIM TESTİNİN ÖNEMİ

YAZILIM TESTİNİN ÖNEMİ



YAZILIM TESTİNİN ÖNEMİ

YAPILAN İŞLER

road map



YAZILIM TESTİNİN ÖNEMİ

- BUG SAYISI
- PERFORMANS SORUNLARI
- KOD HATALARI
- KAYNAK YETERSIZLIGI

Kaynak Yetersizliği

8.1%

Kod Hataları

13.1%

Performans Sorunları

8.1%

BEFORE TEST

application

Bug Sayısı
70.7%

YAZILIM TESTİNİN ÖNEMİ

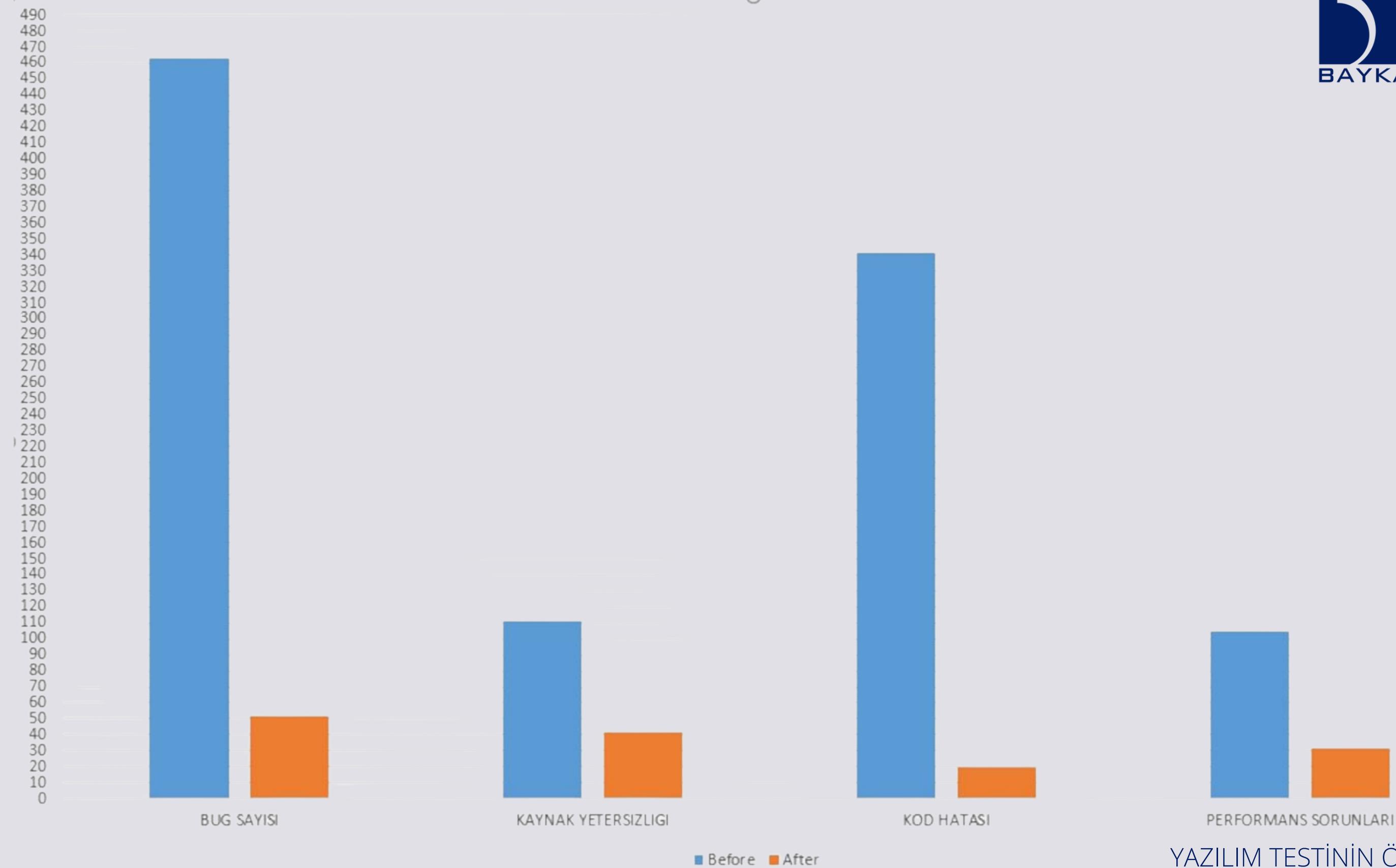


AFTER TEST

application



- PERFORMANS SORUNLARI
- KOD HATALARI
- KAYNAK YETERSIZLIĞI



YAZILIM TESTİNİN ÖNEMİ

PROJELER

- TestRail ve Confluence uygulamalarında test senaryoları yazdım.
- PostMan'da API testi gerçekleştirdim.
- Çeşitli işlevleri bulunan ReactApp oluşturup , bu fonksiyonlara unit testler yazdım.
- Veritabanıyla bağlantısı bulunan projenin Veritabanı testlerini ve Entegrasyon testlerini gerçekleştirdim.
- Clean Architecture mimarisini ve Single Page Application yaklaşımını örneklerle birlikte öğrendim.
- ASP.Net ile oluşturulan hesap makinesi, banka hesapları gibi örnek projelerin birim testlerini yazdım.
- Baykar şirketinin web sitesindeki belirli işlevlere otomasyon testi yazdım.
- RobotFramework ile birlikte Selenium kütüphanesi kullanarak web uygulamalarının otomasyon testlerini gerçekleştirdim.
- RobotFramework'de bulunan DataBase Library'yi kullanarak veritabanının uzaktan bağlantısını ve güncellemelerini gerçekleştirdim.
- YouTube'da kullanıcı arayüz testlerini koştum.

TEKNOLOJİLER



Visual Studio®



PyCharm



ROBOT
FRAME
WORK/



POSTMAN



Selenium



TEKNOLOJİLER