

## # TECHNO STUDY Agile Yazılım Test Projesi - Akakce TR Sprint

### \*\*1 HAFTALIK SPRINT\*\*

**\*\*URL:\*\*** [Akakce Projesi \(https://www.akakce.com/\)](https://www.akakce.com/)

1. >>> Kullanacağımız environment, canlı bir site olduğu için, lütfen sprint sonunda oluşturduğunuz kullanıcıları sistemden silin.
2. >>> Bu task için locator stratejisi olarak sadece XPATH ve CSS kullanın... 🚫🚫🚫 ID ve name kullanmıyoruz ! 🚫🚫🚫
3. >>> Cookie'leri onaylamanız veya reddetmeniz programın çalışmasını etkilemeyecek. O yüzden "Accept Cookies" 'i görmezden gelelim...

### # 1. Jira Hesabı Oluşturmak ve Proje Başlatmak

- Jira platformuna gidin ve yeni bir hesap oluşturun. Zaten bir hesabınız varsa giriş yapın. XRAY veya ZEPHYR gibi bazı eklentiler, sadece 1 aylık ücretsiz kullanım sunduğu için, 4 hafta sonra bir hesap daha tanımlamayı aklınızın bir köşesinde tutalım.
- Yeni bir proje oluşturun ve projeniz için uygun bir ad seçin, örneğin "Akakce.com Sprint 1".

### # 2. Ekip Üyelerini Davet Etmek

- Projeyi oluşturduktan sonra, "Administration" veya "User Management" bölümünden "Invite Users" linkine gidin.
- Takım üyelerinizi projenize davet edin, bu sayede projede işbirliği yapabilirler.

### # 3. User Story'lerin Oluşturulması

- "Backlog" veya "Issues" sekmesine gidin ve user story'leri Jira projenize ekleyin. Örneğin, "Kullanıcı olarak hesap oluşturmak istiyorum" gibi hikayeler ekleyin.

### # 4. Refinement (İyileştirme) ve Planlama

- Ekip, kullanıcı hikayelerini daha ayrıntılı hale getirirken "Refinement" toplantıları düzenler. Gerekliğinde zorluklarına göre user story'lere puan verir. Hatırlayacağınız üzere, fibonacci sayılarına göre (1, 2, 3, 5, 8, 13, 21, 34 ...) planning pokerde puanlama yapılıyordu. Burada elde edilen ortalama puan (estimasyon) hikayeye eklenir.
- "Sprint" veya "Board" bölümünden 1 haftalık bir sprint oluşturun ve tahminlerle puanlanmış user story'leri sprinte sürükleyerek atamalarını yapın.

### # 5. Günlük Standup (Daily Standup) Toplantıları

- Her gün düzenlenen bu toplantıda ekip üyeleri, dün ne yaptıklarını, bugün ne yapacaklarını ve karşılaştıkları engelleri (blocker) paylaşırlar.



## # 6. Sprint İçi İlerleme, gitHub ve Peer Review

1. Her ekip üyesi atanan user story'leri GitHub'da kişisel bir branch'te kodlamaya başlar.

2. İşleri bitirdikçe, "Pull Request" (PR) açar ve diğer ekip üyelerinden kod incelemesi ister (peer review). İnceleme sonrası gerekli düzenlemeler yapılır.

### 3. GITHUB KULLANIMI SIRASINDA DİKKAT EDİLECEK HUSUSLAR:

+ Her takım lideri, GitHub'da herkese açık bir proje oluşturacak ve Collaborator olarak grup arkadaşlarını ekleyecek.

+ Takım liderleri master branch'i koruma altına alacak ve grup üyeleri GitHub davetini kabul edip projeye katılacak.

+ Takım liderleri Sprint Planlama Toplantısı organize edecek. Bu toplantıda kullanıcı hikayeleri puanlanacak ve grup üyelerine atama yapılacaktır.

+ Kullanıcı hikayeleri sprint içerisine eklenerek sprint başlatılacaktır. Her üye, aldığı kullanıcı hikayesi numarasıyla GitHub projesinde bir branch oluşturacaktır. Aynı zamanda Jira'da User Story'yi "IN-PROGRESS" konumuna taşıyacaktır.

+ Her takım üyesi, kullanıcı hikayesi numarasına uygun bir sınıf içinde kod yazacaktır. Bu sayede çalışmaların önüne geçilecektir.

+ Proje üzerinde her çalışmadan önce proje güncellenecek, ardından bu güncellemeyi kendi branch'imize çekeceğiz (Örneğin: *merge master into Ayşe*).

+ Tamamlanmamış çalışmalar veya çalışmayan kodlar "PUSH" yapılmayacaktır.

+ Özellikle her "PUSH" öncesinde proje tekrar güncellenecek. Merge conflict durumunda takım lideri ile çözüm aranacak, gerekirse mentorlara danışılacaktır.

+ Commit mesajları ve PULL REQUEST açıklamaları anlamlı ve açıklayıcı olacaktır.

+ Takım liderleri PULL REQUEST'leri değerlendirecek ve uygun şekilde MASTER branch'e birleştirilmesini sağlayacaktır.

+ Takım liderleri ve mentorlar ile yakın koordinasyon sürdürülecektir.

+ Herhangi bir problem yaşandığında önce takım arkadaşlarına, ardından takım liderine danışılacak nihayet gerektiğinde mentorlara başvurulacaktır.

## # 7. Sprint İncelemesi ve Retrospektif

1. Sprint'in sonunda ekip tamamlanan çalışmayı sunar ve olası kullanıcı hikayeleri hakkında geri bildirim toplar.

2. Daha sonra sprint sürecini ve performansını değerlendirmek ve iyileştirme fırsatlarını belirlemek için "Geriye dönük" bir toplantı (retrospective) yapılır.

## # 8. Mentorluk ve Bilgi Transferi

Takım arkadaşlarınızı, mentorlarımızı etiketleyerek, Discord veya başka yollarla "Bilgi Transferi" (KT) talebi oluşturmaktan çekinmeyin. Bizler buradayız.

😊 H A P P Y C O D I N G ! ! ! 🛠️

