
TECHNO STUDY Agile Yazılım Test Projesi - 2 demowebshop TR Sprint

Bir Haftalık Sprint

URL: [DemoWebShop Projesi] (<https://demowebshop.tricentis.com/>)

Projeye Genel Bakış

- Bu proje kapsamında amacımız ekte verilen user story'leri de içerecek şekilde bir otomasyon test projesinin oluşturulması ve yönetilmesidir. Proje Ekip Çalışması, Kod Yönetimi ve Test süreçlerini kapsamaktadır.

Adım 1: Hikayeler ve Görev Atamaları Oluşturma

- Verilen hikayeleri anlayın ve her hikaye için bir görev oluşturun.
- Görevleri **Jira**'da açın ve bunları **sprint toplantılarında** tartışın.
- Görevleri puanlayarak sprint toplantılarında önceliklendirin (**Refinement / Sprint Planning**).
- Ekip üyelerinin sayısına göre görevleri dağıtarak bir **sprint backlog** oluşturun.

Adım 2: Test Senaryolarını Yazma

- **Jira Xray** veya **Zephyr** eklentisi kullanarak test senaryolarını yazın.
- Her senaryo, senaryo adını, açıklamasını, adımlarını ve beklenen sonuçları içermelidir.
- Günlük toplantılarda (**daily standup / daily scrum**) test senaryolarını olgunlaştırın.

Adım 3: Kod Geliştirme ve Testlerin Yürütülmesi

- Test kodlarını yazmak için bir **GitHub** deposu oluşturun.
- Ekip Lideri ana şubeyi (**master branch**) oluşturacak ve koruma ayarlarını (**rule protection**) yapacaktır.
- Ekip üyeleri master brunch'ı yerel olarak (**local**'ine) klonlar ve kendi adlarına yeni bir dal (**branch**) oluşturur.
- **Conflict**'i en aza indirmek için kendi branch'larında test kodlarını geliştirir.
- Zaten geliştirilmiş ve çalışan web **environment**'imiz olacak:
[DemoWebMağazası](<https://demowebshop.tricentis.com/>).

Adım 4: Kod İnceleme ve Birleştirme İşlemleri

- Her üye, kendi dalında yazdığı kodu için, master yönünde bir çekme isteği (**pull request**) başlatır (*Team lead'e "bu değişikliği master'a çeker misin?" demek gibisinden...*)
- Ekip, değişiklikleri onaylaması için Ekip Liderine **pull request** gönderir ve bunları ana dala birleştirmesini ister.
- Ekip Lideri çekme isteklerini inceler (**peer review /code review**) ve gerekirse düzeltme talep eder.
- Kod incelemesi, oluşturulan çekme istekleri üzerinde gerçekleştirilir.
- Onaylanan çekme istekleri (**approved** pull requests) Ekip Lideri tarafından ana dala birleştirilir (**merge** edilir).

- Her kodlama öncesinde her üyenin 1.Uzak (remote) GitHub sunucusundaki dosyaları kontrol etmesi, 2.Yerel makinesindeki ana dalı (local > master branch) güncellemesi (update) ve 3.Ana makineyi (remote'daki master) kendi dalıyla (hakan branch'i) birleştirmesi (merge etmesi) gerekir. (merge master >info> hakan... gibi). Bu atlanmaması gereken **ÇOK ÖNEMLİ** bir adımdır. Aksi halde conflict'lerin çoğalması kaçınılmazdır!

Adım 5: Sprint Sonrası Değerlendirme ve Transferler

- Sprintten sonra bir **retrospektif** yaparak neyin iyi gittiğini ve neyin geliştirilebileceğini değerlendirin.
- **İsteğe bağlı olarak**, karmaşık veya öğrenilmemiş hikayeler bir sonraki **sprint'e** aktarılabilir.
- GitHub depo adresini (**repository**) ve **Jira board** ekran görüntüsünü **Discord** gibi iletişim kanallarında paylaşın.
- *Paylaşırken Öğretmenlerimizi ve Mentorlarımızı etiketleyelim.*

Destek ve Mentorluk

- Ekip üyeleri sorunlarla karşılaştıklarında Ekip Lideri veya grup içi iletişim kanallarından yardım isteyebilirler.
- Gerekirse **Discord** veya diğer yöntemlerle "Bilgi Transferi" (**KT**) **talepleri** oluşturun ve ekip üyelerinizi ve bizleri etiketleyin. Mentorlar olarak bilgi ve deneyimlerimizi her zaman paylaşacağımıza emin olabilirsiniz. Böylece hep beraber tecrübe kazanacağız...

Proje Bilgileri

Bu proje işbirliği yapma, kod yönetimi uygulama, süreçleri test etme ve iletişim becerilerini geliştirme fırsatı sunar. Aynı zamanda gerçek dünya projelerinde kullanılan araç ve yöntemler konusunda deneyim kazanmanıza da yardımcı olacaktır.

****Not:** Bu proje için Action sınıfının kullanılmasını istiyoruz.**

h A p p Y c O d I n G 🙄

