

Proje Süreci

Grooming Toplantısı ve Projenin Tanıtılması;

Her projenin başlangıcında sprint başlamadan önce yapılan Backlog Refinement da denilen toplantıya grooming toplantısı denir.

Bu toplantı genelde Product Owner önderliğinde Scrum Master ve Development Team'in katılması ile yapılır.

Başlayacağımız proje olan Medunna bir sağlık sitesi projesidir. Bununla ilgili olarak bize verilen BRD(Business Requirement Document) dokümanında müşteri istekleri doğrultusunda Product Owner tarafından oluşturulmuş 28 user story vardır.

Hastane sitemizde gezdiğinizde sitenin farklı rollerle kullanıldığını görürsünüz. Medunna sitesindeki kullanıcı rollerinden bahsedecek olursak;

- Hasta
- Doktor
- Staff
- Admin

Bu rollerin erişebildiği ve erişmediği çeşitli tablo ve alanlar mevcuttur. Bizden sitenin bu kısıtlamaları ve yetkilendirmeleri farklı kullanıcı senaryolarıyla bize verilen user story lere göre test edilmesidir. Testlerinizi yaparken kullanacağınız roller için username ve şifrelerinizi herkese verilecek olan admin rolü için username ve password'lerinizi kullanarak sistem üzerinden edinebilirsiniz.

User storylerin olduğu BRD (Business Requirement Document) dokümanı Excel tablosu olarak projedeki gruplara dağıtılacaktır. Bu dokümanı bir bulut hesabına ekleyerek takım arkadaşlarınızın ulaşımına açabilirsiniz. User Story'lerinizde istenen gereksinimleri test etme amacıyla oluşturacağınız Test Case'lerinizi bu Excel üzerinde ya da Jira'da oluşturup takibini yapabilirsiniz ama Jira üzerinden oluşturup takip edilmesi, Jira'nın da öğrenilmesi açısından tercih sebebi olması projeye katılanların yararına olur.

Her takım kendi içerisinde Scrum Team deki belli roller için birini seçmelidir. Bu roller Product Owner (PQ), Scrum Master (SM) ve Team Lead (TL) 'dir. TL ya da SM Github da ortak bir repo oluşturmalı ve birlikte hazırlayacakları framework'u bu oluşturdukları repoya push etmelidir.

Team Lead olan takım üyesinin asli görevi bu repodaki framework ün ve sonrasında buraya gönderilecek olan kodların bütünlüğünü korumaktır. Bunun sağlamak için farklı yollar izleyebilir bunlardan biri takımı organize edip repoya yapılan kod göndermelerini belli bir sıraya göre yapma olabilir. Sırası gelen takım üyesi oluşturulan bu ortak repoya push edeceği kodları yollamadan önce Framework'ün son halini ana Branch'ine geçip pull etmek sonra kendi Branch'ine geçerek kendi branchinde pull ettiği kodları merge etmek olmalıdır. Merge işlemi sonrasında framework içinde conflict oluşması durumunda kendi localinde oluşan conflict'leri düzelterek sonrasında kodlarını ortak repoya kendi branchinde push etmesidir. Team lead herkesi bu şekilde organize ederek takım arkadaşlarının ortak repoya gönderdiği kodlar içerisinde conflict oluşmasını önlemiş dolayısıyla framework bütünlüğünü sağlamış olur.

Github işlemlerinden sonra SM ya da TL; Jira'da yeni bir proje oluşturup oluşturdukları bu projeye takım arkadaşlarını Collaborate (davet) etmelidir. Bu kısımda daveti kabul eden takım üyeleri bu bölümde Backlog'da tanımlanan User Story'leri Spring Planing toplantısı yaparken puanlayıp (Puanlama

Grooming Toplantısında da yapılabilir.) Sprint'in başlamasıyla projenin izlenebilirliğini sağlamak için "To do" listesine düşen "User Story"lerinden çalıştığı User Story'yi "Progress" kısmına almalı çalışma bittikten sonrada done kısmına almalıdır. Bu sıralama düzgün yapılmadığı taktirde proje takibi için kullanılan grafikler istenilen gibi çıkmayacaktır.

Sprint Planning Toplantısı:

SPRINT süresince yapılacak işler planlanır. Scrum takımının katılımı ile gerçekleştirilir. Her takım üyesinin ya da çoğunluğun katılımı ile yapılır. Product Backlog'taki maddeler, netleştirilerek önceliklendirilir.

Bu toplantıda "NE" ve "NASIL" sorularının cevabı aranır. Sprintin kapsamı ve gereksinimleri bellidir ve nasıl yapılacağı konuşulur. Açık konular netleştirilir. User Story'lerin takım içerisinde puanlanıp dağılımı yapılır. Ve takım içerisinde iş bölümü yapılır. Jira da sprintin ne zaman başlatılacağı Github'da framework de neler yapılacağı hangi yapının kullanılacağı hangi Reusable metotları kullanacağı belirlenir. Günlük Dailyler planlanır. Burada alınan kararların bir günlük süreyi aşmayacak şekilde sonuçlandırılması beklenir.

Takım Kapasitesi ve Puanlama

User Story'ler, Sprint Planning toplantısında, Sprint Backlog'a alınırken takım tarafından puanlanmalıdır.

- User Story'ler okunur ve Development Team'in tüm üyeleri User Story'e puan verir.
- Verilen puanlar Fibonacci Serisindeki sayılara göre verilir. (1 1 2 3 5 8 13 21 34...)
- Her üye verdiği puanı açıklar ve tüm Team'in verdiği puanların ortalaması alınarak User Story'nin puanı belirlenmiş olur.
- Takımın verdiği puanların ortalaması alındığında çıkan sayı Fibonacci serisindeki sayılara uyarlanmalıdır. (Ortalama 7 geldi diyelim takım la konuşularak bu sayı Fibonacci serisine uyarlanarak ya 5'e ya da 8'e çıkarılmalıdır.)
- Team içindeki dev ve QA sayısına göre takım kapasitesi belirlenir.
1 point =1 günlük çalışma yani 8 saattir.

Örnek: 5 developer'in takım kapasitesi nedir?
 $5 \times 1 = 5$ (5 developer'in günlük kapasitesi 5 point)
1 haftalık kapasitesi => $5 \times 5 = 25$ point
2 haftalık kapasitesi = 50 point

Örnek: 2 QA'in takım kapasitesi nedir?
 $2 \times 1 = 2$ (2 developer'in günlük kapasitesi 2 point)
1 haftalık kapasitesi => $5 \times 2 = 10$ point
2 haftalık kapasitesi = 20 point



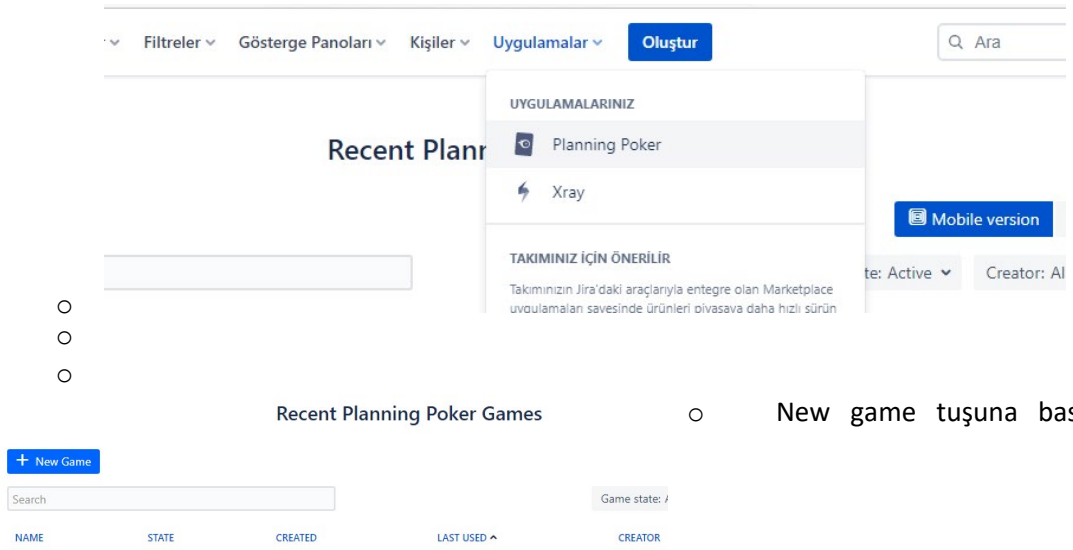
Development team'de 1 kişinin 2 haftalık sprintti yapacağı iş miktarı $5+5=10$ point'tir.

Sprint Günlük Toplantısı:

Günlük yapılan 15 dakika ile sınırlı bir aktivitedir. Daily Sprint’lerde her takım üyesi o gün hangi işlerin yapıldığı, bir sonraki toplantıya kadar hangi işlerin tamamlanmasının planlandığı ve varsa planlanan işlerin önündeki engellerin neler olduğunu konuşur. (Dün ne yaptım, yarın ne yapacam, beni ne engelliyor) SPRINT süresince “Burndown Chart” denilen kalan gereksinimler/geçen zaman grafiği güncellenir.

Tavsiyeler

- Takımların Framework oluşturma ve Test Case’lerindeki ortak adımları grup çalışması yaparak tamamlamaya çalışması tavsiye edilir.
- Takım içerisinde bazı testlerin önceliklendirilmesi gerekiyorsa testlerin çalışırken ki akışını bozmama adına “Depends on” yada Priority gibi metodlar kullanılabilir.
- Jira da xray kurulumunu sadece Team Lead’in yapması yeterli olur. Diğer takım üyelerinin ekstra kurulum yapmasına gerek yoktur.
- Team Lead, Team içindeki diğer elemanları Jira’da ki projeye dahil etmek için projeyi oluşturduktan sonra her takım üyesine davetiye yollaması yeterli olur.
- Jira’da proje oluşturduktan sonra tüm takım üyeleri ya da takımda sorumlu olan kişi tüm User Story’leri Product Backlog’a girmelidir.
- Poker uygulamasını çalıştırmak için alternatif bir yolda Google arama sayfasına “Jira Planning Poker” yazıp gelen ilk Atlassian sayfasına -reklam olmayan- tıklayıp Jira ya tanımladığınız email hesabıyla login olmaktır.
- Poker uygulamasını projemize entegre etmek için aşağıdaki adımları takip edin.
 - header daki uygulamalar butonuna tıklayıp “planing poker” uygulamasını çalıştırırız



- Açılan sayfada sadece game name doldurulması yeterlidir. Sonra next diyerek diğer sayfaya geçilir.

Create New Game

Game basics

Game name

Estimation field ⚠
Select an estimation field to save estimates to

Card deck

Enter comma-separated estimation values. Optionally, provide values mapping, e.g. "LABEL=VALUE"

☒ Add "?" and "coffee" cards to the deck ⓘ

Round auto start ☐ Pick next item from estimation backlog automatically

[Show advanced configuration](#)


[Next: Backlog configuration](#) [Cancel](#)

Configure Backlog

Project ▾ Type ▾ Status ▾ Assignee ▾ My Filters ▾ More + ▾ Select board and then sprint... ▾

Type to filter... 🔍

- deneme (DEN)
- TEAM83 (TEAM)
- ugur (UGUR)



No results
Modify search criteria to add issues to the game

- Proje kısmında tanımlanan proje seçilir. Type kısmında da "Story" seçilmelidir. Ardından "next start game" denilerek poker oynama başlayabilir.

Team deki diğer arkadaşlara buradaki "Share Game" butonuna basılarak verilen linki paylaşıp poker oynuna takımın dahil olması sağlanır.

Daha sonra estimate butonuna tıklanarak sırasıyla user story'ler tüm takımla puanlanır.

d: off [Become spectator](#) [Share game](#)

ESTIMATED PO

PARTICIPANT

SUAT ORU

Re-import backlog for current

EPIC	ESTIMATE
	Estimate

- Nirvana noktası olarak kullandığımız intelliJ'yi jiraya entegre edebilirsiniz. Bu şekilde tag name leri ayarlandığı zaman intelliJ de run edip çalıştırdığınız tüm test case ler proses den done düşer.

- Projede TestBaseRapor class'ında xml de test gruplarını çalıştırırken hata almamak için
tear down metodu

Driver.closeDriver();

satırı yorumla alınarak bu satırın hemen altına metoddan çıkmadan

Driver.getDriver().manage().deleteAllCookies();

satırı eklenerek webdriver her çalıştığında localdeki cookies ler silinip xml çalıştırılırken logout kaynaklı raporlama hataları giderilebilir.

- TestNG de xml de toplu testler çalıştırılırken sıkıntı yaşamamak adına TestCase'lerdeki tüm adımlar baştan sona her bir Test'te yazılmalıdır.
- TestNG'de raporlama kısmında çalışılan Class rapor Class'ına extend edilerek rapor da gözükmesi istenilen isim burada obje oluşturularak verilir. Sonrasında raporda gözükecek olan test stepleri hakkında bilgi vermek için;
extentTest.info
pass olan testler için;
extentTest.pass
fail olan testler içinde;
extentTest.fail
belirtilerek adım adım stepler yazılır.
- Testler Paralel ve CrossBrowsing Test şeklinde çalıştırılarak bu yapının nasıl çalıştığı deneyimlenmelidir.
- Sprint başlatıldıktan sonra takım üyeleri tekrar storylere puanlama yapmamalıdır. Aksi durumda burndownchart ve diğer grafiklerin akışı bozulacaktır.

Sprint Rewiev Toplantısı;

Sprint bittikten sonra ya da sprintin son günü gerçekleştirilen bu toplantı Sprint sırasında üretilen çalışma özelliklerini göstermenin yanı sıra, gelecekteki sprint'ler için çalışmayı yönlendirmeye yardımcı olabilecek Ürün İş Listesi'ni ekleyebileceğiniz yararlı geribildirimleri de almak için yapılan toplantıdır.

Reel de sprint review toplantılarında bir önceki sprintin değerlendirmesi yapılır edinilen deneyimler takımla paylaşılır toplantıyı Product Owner yönetir. Toplantı genelde sprint sonucu ortaya çıkan ürünün demosunu yaparak başlar ve hangi işlerin tamamlandığı hangi ilerin neden tamamlanamadığı kısaca açıklanır. Toplantı sırasında katılımcıların yaptıkları geribildirimler önceliklendirme ve değerlendirmeler için kayıt altına alınırlar. İki haftalık bir sprint için toplantı için en fazla iki saat ayrılması önerilir.

Sprint Review toplantısının asıl amacı, paydaşlara bir durum raporu sunmak veya ürünün bitmiş kısmının sunumunu yapmak değildir. Asıl amaç; çalışan yazılımı paydaşlara göstermek, böylece ürünün bitmiş kısmıyla (increments) ilgili geri bildirim toplamak ve ilerlemeyi (Progress Report) takip etmektir. Bu toplantılarda şeffaflık çok önemlidir. Herkesin tam olarak bilgi sahibi olması, hiçbir şeyin gizlenmemesi, her şeyin açık olarak dile getirilmesi gerekir.

Sprint Retrospective Toplantısı

Bu toplantılar geliştirme ekibi, scrum master ve product owner katılımı ile yapılır. Pratikte, product ownerın katılması beklenir ama zorunlu değildir. 2 haftalık planlanan sprintler için 1–1.5 saat sürecektir. Retrospective toplantıları bir sonraki Sprint'te iş süreçlerini iyileştirmek için geçmiş Sprint'in incelendiği ve “nasıl daha iyi performans gösterebiliriz?” sorusuna cevap aranan toplantılardır.

Bu toplantılar da Sprint içerisinde iyi ya da kötü gitmiş olan her şeyi, her olayı tüm ekibin birlikte şeffaf bir şekilde masaya yatırmaları gerekir. Daha sonra iyi yaptıkları şeyleri korumak ya da kötü yaptıkları şeylerden kaçınmak için ne gibi aksiyonlar almaları gerektiğini konuşurlar. Toplantının amacı üretkenliği arttırmak adına bir sonraki Sprint'te mevcut Sprint'e göre bir iyileştirme sağlamak için alınması gereken aksiyonları belirlemek ve hangi aksiyonların öncelikli bir şekilde hemen hayata geçirilmeye çalışılacağına karar vermektir.

Peki bir retrospective toplantısını nasıl yapıyoruz;

Etkili bir Retrospective toplantısı için takım, aşağıdaki sorulara cevap vermesi gerekir:

- **Takım olarak neleri iyi başardık, zor olan nelerin üstesinden geldik?** Her bir takım üyesi yaşadığı zorlukların nasıl üstesinden geldiğini açıklayarak hem motive olacak hem de takım arkadaşlarının benzer sorunlarla baş edebilmeleri için referans olacaktır.
- **Ne tür sorunlar yaşadık?** Sprint boyunca karşılaşılan problemler, engeller ve gecikmeler belirlenir ve bunların temel sebebine odaklanılır.
- **Geliştirilmesi veya sonraki sprintlerde dikkat edilmesi gereken konular neler?** Sonraki Sprint'te karşılaşılmaması beklenen sorunlar için nasıl önlem alınabilir, neler geliştirilebilir ve mevcut problemlerin üstesinden nasıl gelinebileceği tartışılır.

Siz de bir sonraki Sprint'i nasıl geliştireceğinize hızlı bir şekilde karar vermenize yardımcı olması için Sprint geçmişe dönük gündeminizi planlarken bu soruları sorabilirsiniz.

