

GAZİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ



BM495 BİLGİSAYAR PROJESİ I
Yazılım Projelerine Yönelik Proje Yönetim Yazılımı Geliştirilmesi

Literatür Taraması Raporu

Mert UYĞUR - 201180753

Mücahid Bilal KESKİN - 191180759

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	1
1. GİRİŞ	2
2. ARAŞTIRMALAR	2
2.1. Scoro	2
2.1.1 Projeler	2
2.1.2 Görev Yönetimi	4
2.1.3 Raporlar	5
2.2. Atlassian Jira Software	6
2.2.1 Proje Çeşitleri	6
2.2.1.1 Kanban Project	6
2.2.1.2 Scrum Project	7
2.2.2 Raporlar	7
2.2.2.1 Çevik Raporlar	7
2.2.2.2 Tahmin ve Yönetim Raporları	8
2.2.2.3 Sorun Analiz Raporları	8
2.2.3 Git Entegrasyonu	8
2.3 Proje Yönetiminde Yapay Zeka	8
3. SONUÇ	9
KAYNAKLAR	10

1. GİRİŞ

Yazılım projeleri, dünya genelindeki mühendislik projeleri içerisinde başarısızlıkla sonuçlanmaya en yatkın projelerdir. Bunun birçok sebebi vardır, fakat bu projelerin başarısızlıkla sonuçlanmasındaki en büyük faktör risk yönetiminin başarısızlığıdır.

Risk yönetimini gerçekleştirmek, proje ve takım hakkında her detayı bilmeyi ve her olasılığı hesaplayabilmeyi gerektirir. Yazılım projesindeki risk yönetimi takımı sıklıkla şu soruyu sormalıdır: Hangi risk etkenleri dikkatlerini daha çok çekmektedir ve risk etkenlerinin fazla olduğunda, hangi stratejiler risk azaltıcı olarak daha fazla etkilidir? [1]

Proje yönetimi projenin kontrolünü, planlanan zaman ve bütçe sınırları içinde tamamlamasını sağlamaktadır [10]. Bu kapsamda, ele alınacak bütün problemler zaman ve bütçe göz önüne alınacak çözülecektir. Bu kavramlar dikkate alınarak geliştirilen bir yazılım, yazılım ekiplerinin kullanmak isteyeceği bir yazılım olacaktır.

2. ARAŞTIRMALAR

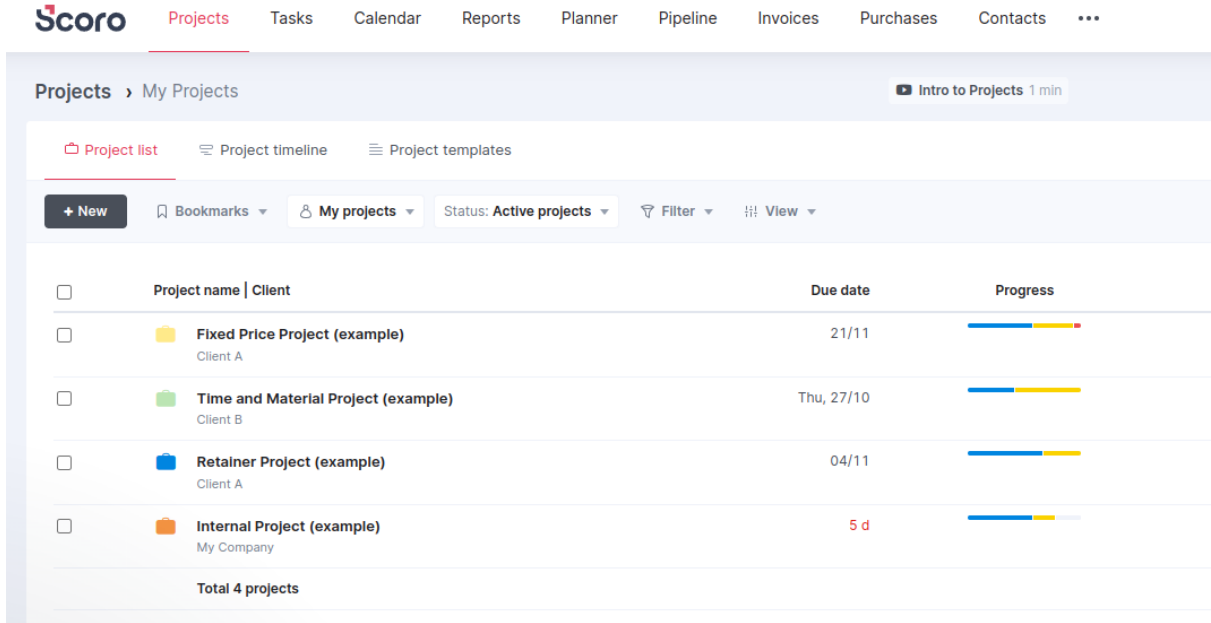
Yazılım projelerinin geliştirilmesinde kullanılan birçok uygulama kullanılmakla birlikte hepsi farklı bir soruna çözüm olacak hayatımıza girmiştir. Yazılım projesi kavramı birçok şeyi ifade edebileceğinden, bunları metotlara indirgemek daha doğru olacaktır. Her yazılım geliştirme metodu farklı bir yaklaşım geliştirmiştir. Dinamik tekniklerin kullanıldığı “Çevik” yaklaşım ve dinamik ve statik yöntemlerin bir arada kullanıldığı “Şelale” metodları [9] yazılım geliştirmeye farklı yaklaşımları gerektirmektedir. Buradaki yazılım geliştirme uygulamalarını incelerken de hangi metodların bu uygulamaları kullanırken daha faydalı olacağı konusunda çıkarımlar yapmak önemli olacaktır.

2.1 Scoro

Yazılım geliştirme aracı denildiğinde akla gelen ilk uygulamalardan olan Scoro, gerek her şeyi kapsaması, gerek kullanışlılığı ile birçok şey öğretebilecek bir araç olarak incelenmesi tercih edilmiştir.

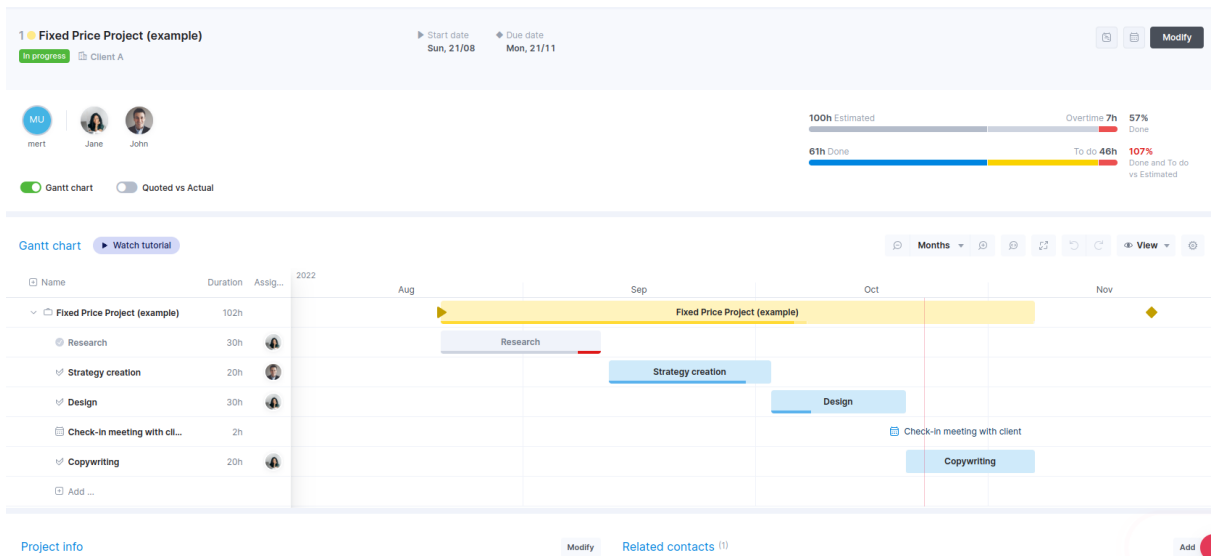
2.1.1 Projeler

Çoğu yazılım geliştirme platformunda olduğu gibi Scoro da göze çarpacak ilk yer olarak “projeler” bölümünü seçmiş. Bunun en önemli sebebi, ilk kırılımın bu noktadan gerçekleşmesi. Ayrıca projeler listesinde verilebilecek en önemli bilgiler de yer almaktadır.



Şekil 2.1: Scoro proje sayfası.

Herhangi bir projenin ilk sayfasında ise, kullanıcının bilmesi gereken biraz daha detaylı bilgiler yer almakta. Bu sayfa çoğu yazılım geliştirme aracında olduğu gibi, tek sayfaya sığdırılmaya çalışılmış. Kullanıcıların proje ile ilgili bilmek istediği her şeyi tek bir sayfaya sığdırabilmek önemli bir kriter. Burada doğru bilgileri seçmek işin kritik kısmı olacaktır. Müşteri bilgilerinden, zaman aşımalarına kadar birçok bilgi bu sayfada verilebilir. Ayrıca takım üyeleri, projenin gidişatı ile ilgili grafikler ve tablolar yerleştirilebilir. Bu sayfada olması gereken, Scoro'da görülmeyen bir diğer şey ise, müşteri yorumları kısmı olabilir. Müşteri ile iletişim kuran tek kişi proje yönetici olursa, kopukluklar meydana gelebilir. O kopukluğu gidermek için müşteriyi de sisteme giriş yapıp, sistemin içine dahil eden bir anlayış, bizim uygulamamız için önemli bir geliştirme olarak görülebilir.

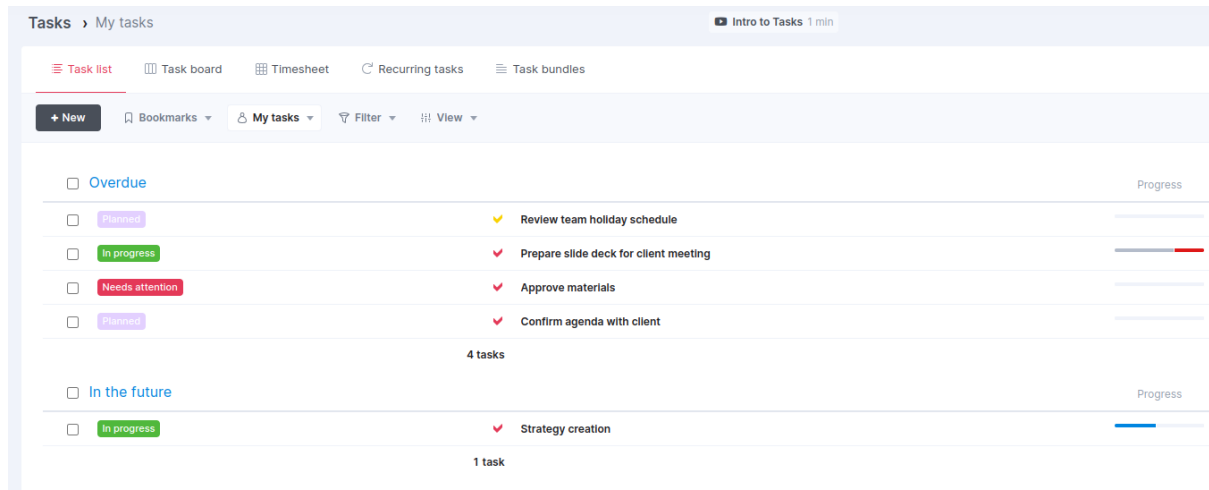


Şekil 2.2: Scoro proje ana sayfası.

2.1.2 Görev Yönetimi

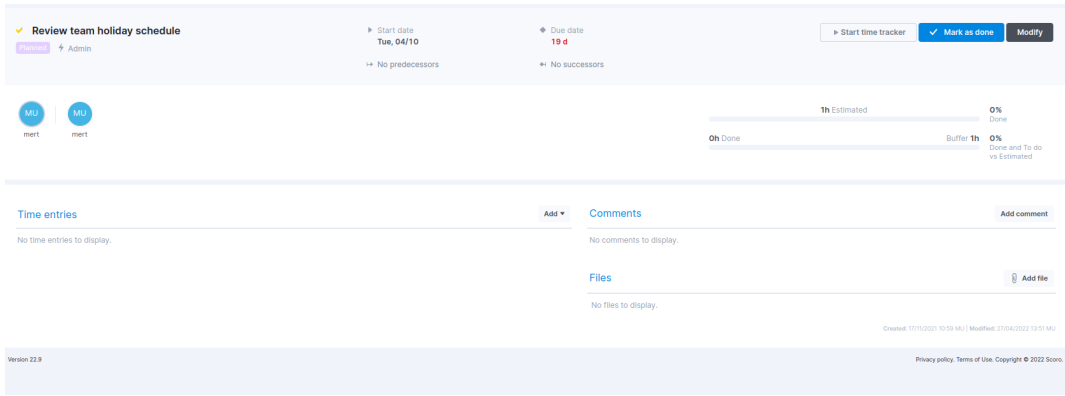
Görev yönetimi, şüphesiz, kullanışlı yapıldığı takdirde uygulamanın başarısını belirleyecek olan yegane faktörlerden biridir. Uygulamamız için üzerine çalışılacak en önemli konulardan biri olacaktır.

Scoro, bizim uygulamamızda planladığımızın aksine, çok sade bir görev yönetimi kısmı yapmayı tercih etmiş. Görevleri, zamanı geçen ve geçmeyen olarak iki kısma ayıran Scoro, görevler hakkında da özet sayfasında çok az bilgi vermeyi tercih etmiş. Ayrıca görevler kısmında projenin tüm görevlerini değil de, giriş yapan kullanıcının görevlerini göstermekte. Bunun hiçbir sakıncası bulunmamakla birlikte, kullanıcının kendi görevleri ile birlikte tüm takımın görevlerini görmesi, aynı sayfada olunmasını sağlayacak en büyük faktör olacağından, biz daha kapsamlı bir görev yönetimi sayfası düşünüyor olacağız.



Şekil 2.3: Scoro görev paylaşımı sayfası.

Herhangi bir görevin içinde ise, bu görev üzerine çalışan kişiler, açıklamalar, belirlenen son tarih ve bu görev ile ilgili işlemler bulunuyor. Görevlerin ayrı bir sayfada açılmasının bir sakıncası bulunmamakla birlikte, bu işlem çok sayfa sayfa geçişine sebep verebilir. Görevler sayfasında açılacak bir üst sayfa ile de yapılacak bir işlem, ayrı bir sayfaya alınarak detaylandırma tercih edilmiş. Buradaki yaklaşım, projedeki görevlerin açılma sıklığına ve kapsamına göre sorgulanabilir. Eğer bir görevin bir hafta ve daha uzun sürdüğü bir yapıda çalışılıyorsa bu yaklaşım kabul edilebilir, fakat günlük görevler planlanmışsa ayrı bir sayfanın olması fazlalık olarak görülebilir.

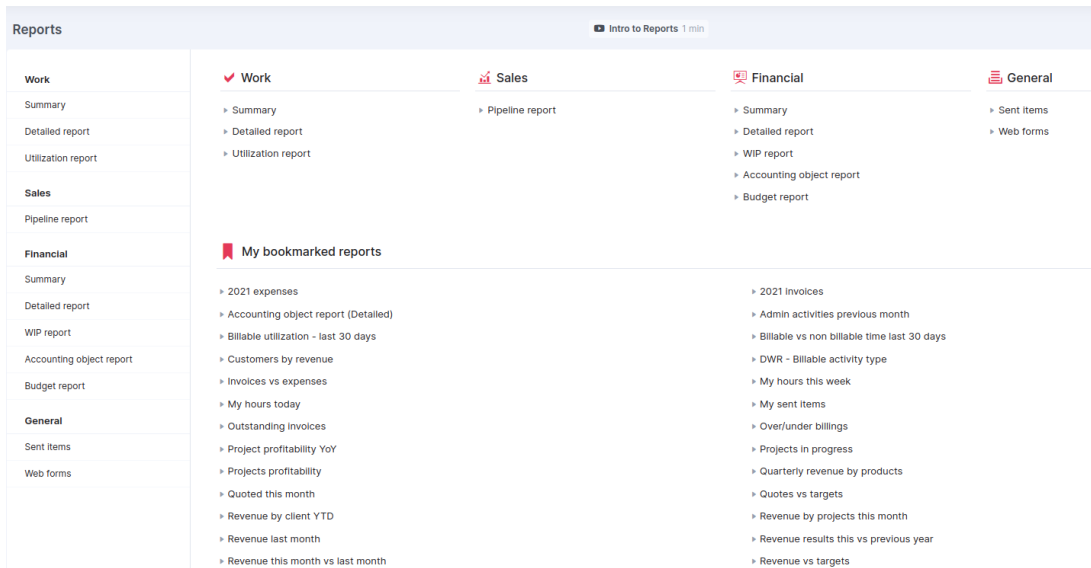


Şekil 2.4: Scoro takım sayfası

2.1.3 Raporlar

Rapor konusu, şüphesiz yazılım ekipleri ile müşterileri birleştiren en önemli konu. Scoro'yu diğer platformlardan ayıran en önemli özelliği, bu sayfa üzerine planladıkları detaylı işlemler ve bu sayfanın kullanışlılığı. Bu sayfada her takım için ayrılmış rapor bölümleri bulunmakta. Scoro bu sayfayı çoğunlukla ekip içerisinde kullanılacak bir sayfa olarak tasarlamış olsa da, düşünülecek bir diğer yöntem ise, bu sayfayı müşteri ile yazılım ekibi arasındaki köprü olarak görmektir.

Proje yönetim platformlarının ağırlıkla bankacılık, sigortacılık ve telekomünikasyon sektörlerinde kullanılır olmasına dayanarak [8] söyleyebiliriz ki, bu tür sektörlerde iletişim çoğunlukla raporlar aracılığıyla sağlanır, her işin bir rapor çıktısı olur. buna dayanarak tasarlanması gereken bu sayfa üzerine düşünürken birden çok disiplini ve kurumsal yapıyı kapsayacak bir inşa süreci tasarlanmalıdır.



Şekil 2.5: Scoro rapor sayfası.

Rapor sayfası, yukarıda da belirtildiği gibi, müşteri ile kurulan köprü olmalıdır. Bu nedenle bu sayfa üzerinde müşterinin ihtiyaçlarını ve isteklerini de girebildiği bir bölüm açılabilir.

Müşterinin ihtiyaçları ve istekleri projenin yapısını ve gidişatını büyük bir oranda etkileyeceğinden, süresinden çok sonra öğrenilen ihtiyaçlar ve istekler projenin başarısız olmasına neden olabilir. Bu nedenle bu sayfada yazılım gereksinimlerinin belirleneceği ek bir bölüm de açılabilir. Bu bölüm, yazılım gereksinim spesifikasyonlarını belirtmekte kullanılacak olup, müşteri ile yazılım ekibi arasındaki iletişimi güçlendirmekte en önemli sayfa olacaktır. Yazılım gereksinim spesifikasyonu, müşterilerin tam olarak ne elde etmek istediklerini, yazılım ekibinin ise müşterilerin ne istediklerini gördüğü yer olacaktır. [2]

2.2 Atlassian Jira Software

Atlassian, çevik geliştirme metodu yazılım şirketleri tarafından daha yaygın olarak kullanılmaya başlandıkça popülaritesini arttırdı ve yazılım şirketleri tarafından daha yaygın olarak kullanılmaya başlandı.

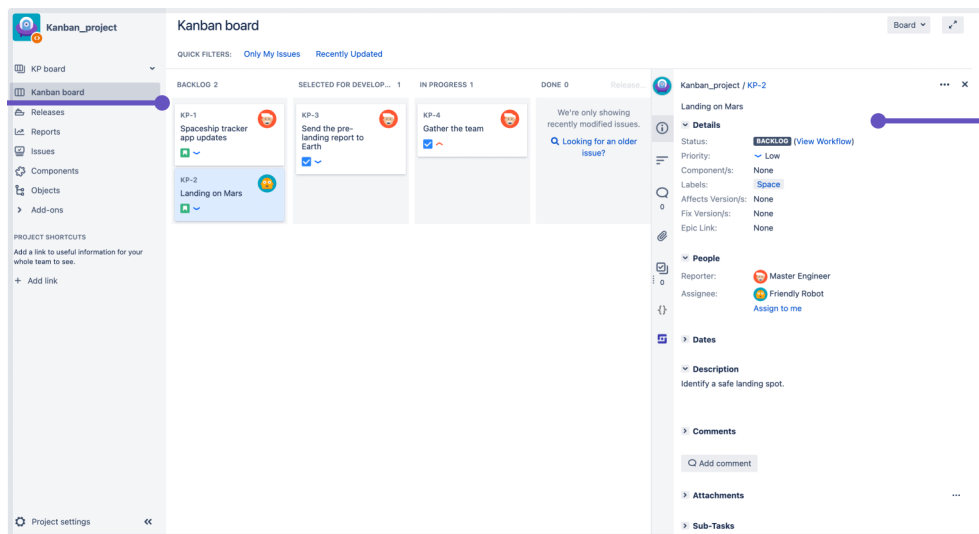
Atlassian Jira Software, sorun takibi ve proje yönetimi için kullanılan bir yazılımdır. Kullanım olarakta çevik geliştirme metodları kullanan yazılım geliştirici grupları veya organizasyonları hedef almıştır.[3]

2.2.1 Proje Çeşitleri

Jira yazılımında proje yönetimi platformu oluşturmak için hazır konfigürasyonu olan iki seçeneğiniz bulunur: Kanban ve Scrum. Bu iki geliştirme metodundan birini seçtiğinizde taslak olarak proje yönetim arayüzünü oluşturur. Bunun dışında geliştiriciler kendi ihtiyaçları doğrultusunda bu arayüzü düzenleyebilir.[4]

2.2.1.1 Kanban Project

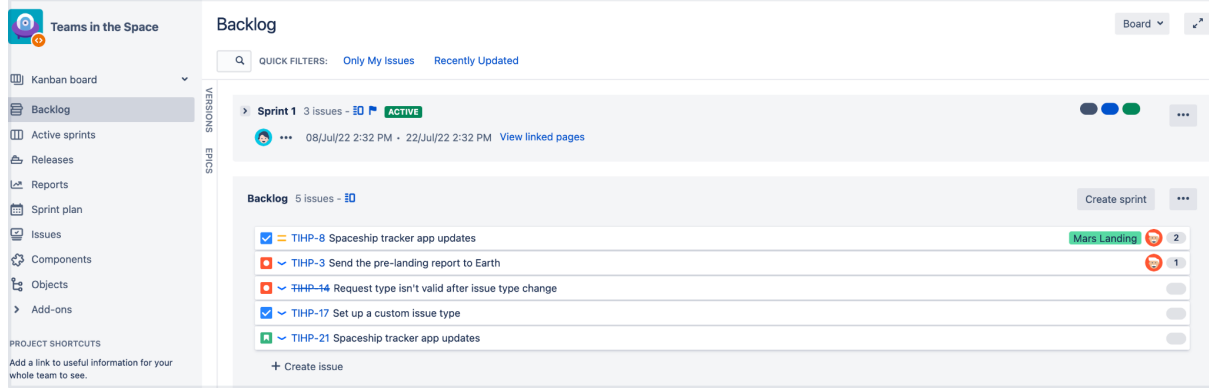
Kanban arayüzünde(Şekil 2.1) iş parçacıklarının gruplandırıldığı “Kanban board” adında bir sekme bulunur. Bu sekmede iş parçacıkları oluşturulabilir, bu iş parçacıklarına; ilgili kişi, etiket, yorum, öncelik gibi pek çok detay yerleştirilebilir. Bu iş parçacıkları ihtiyaç duyulduğunda diğer iş kümelerine yönlendirilebilir.



Şekil 2.6: Jira Kanban tahtası.

2.2.1.2 Scrum Project

Çevik geliştirme için kullanılan arayüzdür. Kanban projesinden farklı olarak Backlog için ayrı bir sekme(Şekil 2.2) ve “sprint”lerin yönetildiği ve görüntülediği “Active Sprints” sekmesi vardır. Bu sekmede iş parçacıkları sprintlere ve kişilere atanır.



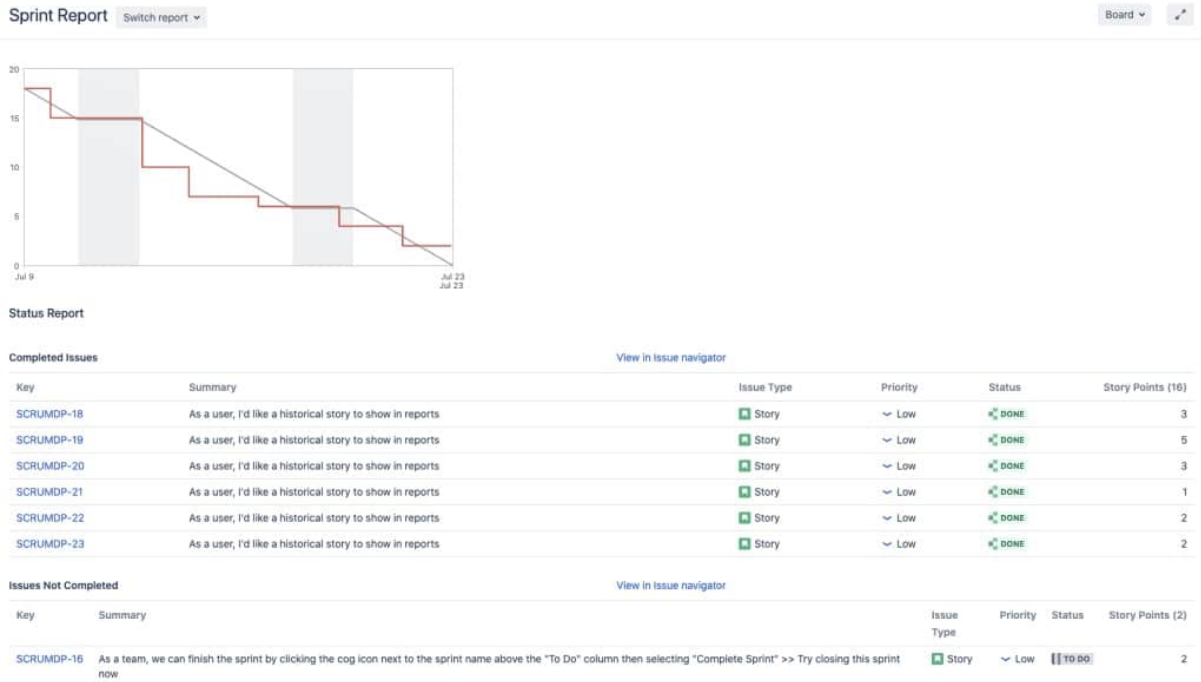
Şekil 2.7: Jira backlog sayfası.

2.2.2 Raporlar

Jira da raporlar kısmında diğer proje yönetim platformlarına göre daha çok detay içeren özellikler bulunuyor. Raporlar aracılığı ile ekiplerin performansı ve işin planlanan süreye göre ne durumda olduğu gibi pek çok bilgi canlı olarak takip edilebiliyor.[5]

2.2.2.1 Çevik Raporlar(Agile Reports)

Bir Scrum takımıysanız ve bir sprint tamamlama yolunda olup olmadığınızı anlamak istiyorsanız, Burndown/Burnup grafiklerini takip ederek iş planlamanızı düzenleyebilirsiniz. Bu grafikler, bir hedefi tamamlamak için yapılması gereken ne kadar iş kaldığını görüntüler.



Şekil 2.8: Jira Agile raporlar.

Sprint ve sürüm raporları, Burndown ve Burnup grafiklerine benzer, ancak tam tersidir. Bu raporlar, işin kalanını tamamlamak için kaç sprint kaldığını tahmin eder. Bu raporlar işin bitirme süresini hesaplamak için geçmiş verileri kullanır ve tahmin raporu oluşturur.

2.2.2.2 Tahmin ve Yönetim Raporları(Forecast and Management Reports)

Bu raporların amacı, ekip üyelerinin iş olarak aşırı yüklenmemesini sağlamaktır. Bunun yanında gerçekleştirilen işlerin tahminleriniz ile nasıl karşılaştırıldığını raporlar.

2.2.2.3 Sorun Analiz Raporları(Issue Analysis Reports)

Created vs Resolved(Oluşturulan-çözülen karşılaştırması) gibi raporlar, ekibin iş yükünü kaldırabildiğini yada kaldıramadığını gösterir. Bu kategoride aynı zamanda özelleştirilebilir grafikler oluşturabileceğiniz araçlar bulunur ve bu araçlar sayesinde daha detaylı veya istekler doğrultusunda farklı çıkarımların yapılabileceği raporlar elde edilebilir.

2.2.3 Git Entegrasyonu

Jira bize git nesnelerini referans etme olanağı sağlıyor. Bu özellik sayesinde git “branch”ine bir commit attığınızda bunu Jira da bir issue ya referans gösterebiliyorsunuz bunun yanında “repository”i Jira projenize bağlayarak repo’nuzdaki hareketleri Jira üzerinden takip edebiliyorsunuz. Bu özellikler geliştiricilere hem daha düzenli bir geliştirme ortamı hemde vakit kazancı sağlayabilir.[6]

2.3 Proje Yönetiminde Yapay Zeka

Yapay zeka tabanlı araçlar, toplantı planlama, hatırlatıcılar, günlük güncellemeler ve diğer yönetim görevleri gibi işlevleri devralabilir. Bu, proje yöneticilerinin ve ekip üyelerinin daha üst düzey, karmaşık faaliyetleri planlayabilmesini sağlayacaktır.

Pulse of the Profession® anketine katılanların %80'inden fazlası, kuruluşlarının yapay zekadan faydalandığını bildiriyor. PMI'nın "AI Innovators: Cracking the Code on Project Performance" raporuna göre, önümüzdeki üç yıl içinde proje yöneticileri yapay zeka kullanarak yönettikleri projelerin %23'ten %37'ye çıkmasını bekliyor.

KPMG'nin “AI Transforming the Enterprise” adlı raporuna göre, yapay zekaya yatırım yapan kuruluşlar, üretkenlikte ortalama %15'lik bir iyileşme gördüklerini bildiriyorlar. PMI'nın "AI@Work" adında bir diğer araştırma ise proje liderlerinin sıklıkla yapay zeka kullanımının ilerlemeyi izleme, dokümantasyon yönetimi, aktivite ve kaynak planlaması gibi faaliyetlere harcadıkları zamanı azalttığını gösteriyor.[7]

3.SONUÇ

Proje yönetim platformları yazılım geliştirme süreçlerine çok önemli bir rol oynuyor. Bu süreçlerin verimi kullanılan proje yönetim platformuna göre değişiklik gösteriyor. Ayrıca platformlarda kullanılan konfigürasyonlar yazılım geliştirme yöntemine göre farklı tercih ediliyor. Bu bilgiler ışığında farklı yazılım geliştirme ekiplerini müşteri kitlesi olarak benimseyecek bir proje yönetim platformunun kişiselleştirme kısmına önem verilmesi gerekiyor. İncelediğimiz araştırmalara göre yapay zekanın proje yönetiminde süreç tahmini, dökümantasyon ve kaynak planlaması gibi alanlarda kullanılıyor ve verimlilikte ciddi bir artış sağlıyor. Şu an genel çözümler(herkesin kullanımına açık platformlar) arasında tam olarak yapay zeka destekli bir proje yönetim platformu bulunmuyor. Buradaki açıklık geliştirmeyi planladığımız platformu daha farklı ve işlevsel kılıyor.

KAYNAKLAR

- [1] O. Ayhan ERDEM, Alaa E. YOUNİS “Yazılım Projelerinin Geliştirme Sürecinde Yönetim”, Bilişim Teknolojileri Dergisi, 2014
- [2] Doğan YILDIZ, “Bilgi Sistemi Yazılım Geliştirme Yaşam Döngüsü Safhalarından Gereksinim Belirleme ve Sistem Tasarımında Kalite Odaklılık: Üç Proje İncelemesi”, Bilişim Teknolojileri Dergisi, 2022
- [3] The Daily Egg, “Jira Review”
<https://www.crazyegg.com/blog/jira-review/>
- [4] Jira Software Support, “Creating Your Backlog”, Erişim: 14 Temmuz 2022
<https://confluence.atlassian.com/jirasoftwareserver/creating-your-backlog-938845071.html>
- [5] Chris COOKE, “Getting Started as a Jira Software Manager”, Erişim: 6 Mayıs 2021
<https://www.oldstreetsolutions.com/reports-in-jira-basics>
- [6] Jira Software Support, “Integrate Your Issues and Development Tools”
<https://support.atlassian.com/jira-software-cloud/docs/reference-issues-in-your-development-work/>
- [7] PMI, “5 Implications of Artificial Intelligence for Project Management”, Erişim: 9 Aralık 2022
<https://www.pmi.org/learning/publications/pm-network/digital-exclusives/implications-of-ai>
- [8] Özden GEBİZLİOĞLU ÖZVURAL, Özgür GÜN, Elif AK, “Etkin Bir Yazılım Süreç Yönetimi İçin Süreç Yönetim Aracı Seçimi”, Tübitak Bilgem BTE Bilişim Teknolojileri Enstitüsü
- [9] Doğan YILDIZ, “Bilgi Sistemi Yazılım Geliştirme Yaşam Döngüsü Safhalarından Gereksinim Belirleme ve Sistem Tasarımında Kalite Odaklılık: Üç Proje İncelemesi”, Bilgi Teknolojileri Başkanlığı, TUSAŞ, 2021
- [10] Vildan ÇORUMLU, “PROJE ÇİZELGELEME PROBLEMLERİNDE BULANIK DOĞRUSAL PROGRAMLAMA İLE PROJE TAMAMLANMA SÜRESİNİN BELİRLENMESİ VE BİR YAZILIM PROJESİNDE UYGULANMASI”, YÜKSEK LİSANS TEZİ, 2022