

Scrum Yöntemi ile Oyun Programlama ve Süre Tahmini

Game Programming and Time Estimation with Scrum Method

Şahin MERCAN
Kocaeli Üniversitesi
Bilgisayar Mühendisliği
sahinmrcn@gmail.com
ORCID: ID 0000-0002-8646-6834

Yaşar BECERİKLİ
Kocaeli Üniversitesi
Bilgisayar Mühendisliği
ybecerikli@kocaeli.edu.tr
ORCID: ID 0000-0002-2951-7287

Öz

Son dönemlerde oyun programlama da yaşanan sorunlardan birisi olan oyun programlama gereksinimlerinin sürekli değişmesidir. Bu yüzden oyun programlarken yürüttüğümüz sürecin bu değişime ayak uyduramaması geliştirdiğimiz ürünün başarısızlıkla ya da daha yüksek maliyetlerle sonuçlanmasına yol açmaktadır. Bu nedenle son zamanlarda değişen ortam koşullarına karşı daha dinamik ve modern bir çözüm getiren çevik programlama yöntemlerinden birisi olan Scrum yöntemi ile "Balon vurma" oyunu geliştirilmiştir. Bu çalışmanın amacı Scrum yönteminin oyun programlarken nasıl kullanılabileceğine dair bir örnek göstermektir. Ayrıca çıkan sonuçları literatürdeki Scrum yönteminin başarılı ve başarısız olduğu bölümler ile ilgili karşılaştırma yapmaktır. Yapılan geliştirme sonucunda ilk başta belirtilen maliyetler ile sonda çıkan maliyetler belirli kriterlere göre karşılaştırılarak çevik yöntemlerin oyun programlamadaki başarısı ölçülmüştür. Bu karşılaştırmayı yapabilmek için projemizi belirli kısımlara ayırarak bu kısımlardaki maliyetlerin karşılaştırılmasıyla bazı sonuçlara varılmıştır. Ortaya çıkan sonuçlar ise kısaca oyunda kullanıcının odaklandığı yani ana karakterlerin olduğu kısımlardaki gereksinim oranı daha fazla değiştiğinden bu kısımdaki maliyetlerde ciddi sapmalar gözlemlenmiştir.

Anahtar Sözcükler: Çevik Yöntemler, Oyun Programlama Süreci, Oyun Proje Yönetimi, Scrum

Abstract

One of the problems experienced in game programming currently is the continuously change of game programming requirements. For this reason, the inability of the process we carry out while programming the game to keep up with this change causes the product we develop to fail or to result in higher costs. For this reason, the "Balloon shooting" game has been developed with the Scrum method, which is one of the agile programming methods that brings a more dynamic and modern solution to changing environmental conditions. The purpose of this study is to show an example of how the Scrum method can be used in game programming. In addition, the results are to compare the sections in the literature where the Scrum method is successful and unsuccessful. As a result of the development, the success of agile methods in game programming was measured by comparing the costs stated at the beginning and the costs at the end according to certain criteria. In order to make this comparison, some conclusions have been reached by dividing our project into certain parts and comparing the costs in these parts. The results, in short, have been observed in the game, since the requirement ratio in the parts where the user focuses, that is, the main characters, changes more, serious deviations have been observed in the costs in this part.

Keywords: Agile Methods, Game Programming Process, Game Project Management, Scrum

Gönderme ve kabul tarihi: 18.01.2020 - 10.05.2021

Makale türü: Araştırma

1. Giriş

1.1 Literatür Özeti

Yazılım süreçlerine başlarken ilerleyen safhalarda oluşabilecek değişim isteklerinin oluşabileceği öngörülmektedir. Bu nedenle geleneksel yöntemler yazılım süreçlerinin en başında kapsamlı bir çalışma yaparak oluşabilecek ihtiyaçların hepsini belirleyip ilerleyen safhalarda oluşabilecek değişiklikleri önlemeyi esas almaktadır. Ancak hızla değişen, gelişen ortam ve piyasa koşullarına karşın artık proje gereksinimleri daha fazla ve hızlı bir şekilde değişmektedir.

Böylece proje başında bütün ihtiyaçların eksiksiz bir şekilde belirlenmesi neredeyse imkânsız bir hal almaktadır. Daha sonraki safhalarda değişiklikler projeye ek maliyet getirir ya da projenin başarısızlıkla sonuçlanmasına yol açar. Bu nedenle kullandığımız proje yönetim modelinin bu şartlara olabildiğince uyum sağlaması beklenmektedir.

Oyun programlama da geliştiren ekip hem bu değişiklikler ile mücadele ederken, hem de oyunda eğlence faktörünü arttırmaya çalışır [17]. Bu sebeple oyun geliştirirken ortaya çıkan değişiklikler daha fazla olacaktır. Biz bu çalışmada projede oluşabilecek değişikliklere olabildiğince hızlı yanıt veren çevik yöntemlerin oyun programlamadaki başarısını ölçmeye çalıştık [17].

Proje yönetim modelinin değişimleri daha kısa sürede ve daha az maliyetle karşılayabilecek özelliklere sahip olması beklenmektedir [15]. Bu nedenle, geleneksel yazılım geliştirme yöntemlerine alternatif olarak çevik yazılım geliştirme yöntemleri ortaya çıkmıştır [1, 7]. Projenin tamamına çevik yöntemler uygulanamıyorsa, sadece gerektiği yerlerde çevik olunabilir veya çevik yaklaşımlar, projeye ve geliştirme ekibinin yapısına göre biçimlendirilebilir [6, 15, 20].

Çevik yöntem proje yönetimi çeşitleri ise Arık Geliştirme, Uyarlanabilir Yazılım Geliştirme, Scrum, Ekstrem Programlama, Kristal Yöntemleri, Özellik Güdümlü Geliştirme, Açık Kaynak Kod Geliştirme ve Dinamik Sistem Geliştirme yöntemleridir [28]. Belirtilen yöntemlerden Arık Geliştirme dışında diğerleri 2001 yılında Çevik Yazılım Geliştirme Manifestosu oluşturulduktan sonra çevik yazılım ilkeleri tanımlanmaya başlanmıştır. Bu ilkeler aşağıda verilmiştir [5].

- “Süreç ve geliştirme araçlarının yerine bireyler ve bireyler arası ilişkileri” [2]
- “Detaylı dokümantasyon yerine koşan yazılımı” [2]
- “Sözleşme görüşmeleri yerine müşteri iş birliğini” [2]
- “Plan takip etmek yerine değişikliklere ayak uydurmayı” [2]

Belirtilen 4 madde bu yöntemlerin ortaya çıktığı bildiride bulunmakta ve bu yöntemlerin etkili bir şekilde kullanılması için gerekli olan maddelerdir [2, 7].

Yukarıda belirttiğimiz çevik yöntem metodu çeşitleri birbirlerine benzese de bunları ayıran yönleri de bulunmaktadır [28]. Bu yöntemlerin birkaçı uyumlu biçimde çalışmayı hedeflerken birkaçı da projeyi geliştirmeye odaklanmıştır [7].

Bu yöntemlerin güçlü noktaları, yöntemler uygulandığında yazılım geliştirme sürecinin değişen gereksinimlere karşı daha duyarlı hale getirilebilmesidir [5]. Bu yöntemlerde, çalışan yazılım, ayrıntılı dokümantasyonun daha üzerinde tutulur; kişiler ve etkileşimler, araçlar ve süreçlerden daha önemli kabul edilir ve müşteri iş birliği, müzakere edilen sözleşmeden daha değerlidir [31].

1.2 Proje Yönetimi Araştırmaları

Çizelge-1’ de görülen Standish Group tarafından yapılan Chaos anketine göre yazılım projelerinin başarı oranı aşağıdaki gibidir;

Çizelge-1: Chaos anketine göre proje başarı oranları [32]

Yıl	Başarılı (%)	Ek Maliyet (%)	Başarısız (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

Standish Group tarafından yayınlanan 27 Nisan 2009 tarihli rapor, ilginç rakamlar içeriyor. 1985 yılından bu zamana bilgi teknolojileri sektöründeki eğilimleri tespit eden firmanın başkanı Jim Johnson 2009 Kaos Raporu’nda yaptığı açıklamada; planlanan zaman ve bütçede, tanımlanmış özellikleri ve işlevleri

gerçekleştirebilmiş olan projelerin oranının %32' ye düştüğünü belirtmiştir [32].

Projelerin %44' ü geç, bütçesini aşmış, tanımlanmış özellikleri ve işlevleri sağlamayacak şekilde tamamlanmıştır [32].

Geriye kalan %24'lük kısımdaki projeler ise tamamlanmadan iptal edilmiş veya tamamlandığı halde uygulamaya dahi alınmamıştır [32].

Başka bir anket olan Shine Teknoloji 2003 anketi sonuçlarına göre çevik yöntemlere dayalı projeler geliştirildiğinde klasik yöntemlere göre verimlilik %93 artıp, %83 iş tatmini ve %88 kalite artırım sağlanmıştır [9].

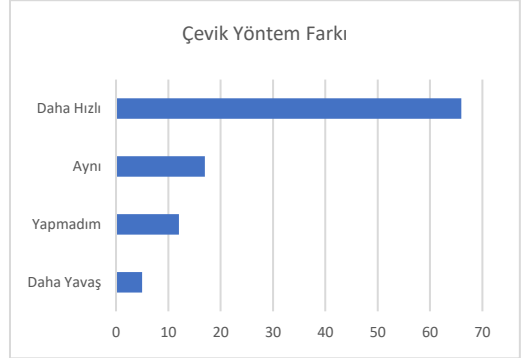
Ayrıca uygulamamızı geliştirirken çevik yöntemlerinden biri olan scrum yöntemini kullandık. Bunun sebebi ise günümüzdeki şirketlerin büyük çoğunluğu scrum yöntemini kullanmaktadır. Bunun ile ilgili yapılan anket ise aşağıda Şekil-1'de gösterilmektedir.

Çizelge-2: Organizasyonların kullandığı çevik süreçler [7]

Süreç	Oran (%)
Scrum	58
Scrum/Xp Hibrit	17
Hibrit	5
Diğerleri	4
XP	4
Bilgin yok	3
ScrumBan	3
Lean	2
Özelliğe Dayalı Geliştirme	2
Agile Up	2

Ayrıca organizasyonların çevik sürece tercih ettiklerini sonra eski durumlarına göre başarı sağlayıp sağlamadıklarına dair cevaplar Şekil-2'de görülebilir.

Buradaki sonuç ise %60'ın üzerinde daha başarılı olduğunu gösteriyor [7].



Şekil-1: Eski yöntemleri ile çevik süreçlerin farkı [7]

Yukarıdaki Şekil-1 ve Çizelge-2'de gösterilen anketler VersionOne sponsorluğunda gerçekleştirilen 2010 yılında 91 ülkede yaklaşık 2700 katılımcı ile gerçekleştirilen "State of Agile Development" isimli ankette ortaya çıkan sonuçlardır.

1.3 Çevik Yazılım Geliştirme Özellikleri

Bu yöntemin en önemli özelliği projedeki insanlara, ekipmanlar ve dokümanlardan daha fazla önem verilmesidir. [4]. Bunun nedeni geliştirmeyi yapan yazılımcılar olduğu için ayrıca geliştirmedeki ekipmanları da kullanan yazılımcılar olduğu için kişilere daha çok önem vermektedir. [11]. Bu nedenle çevik yöntemler, geliştiren kişilerin özgüvenini arttırmayı hedefler ve bu nedenle kişilere her türlü desteği vermeye önem göstermektedir [11].

Çevik yöntemler geleneksel yöntemlerin aksine kişilere önem verdiği için diğer ekipman belge vb. unsurları çok fazla dikkate almadan bir an önce geliştirmeye başlamayı hedeflemektedir. Bu sebeple geliştirme başlarken uğraşılacak doküman gibi benzeri şeyleri sadece istenilen durumlarda hazırlayarak daha hızlı ve esnek ilerlemeyi amaçlar [1, 30].

Çevik yöntemleri diğerlerinden ayıran husus ise değişen şartlara her şeyden daha çok değer vermesidir [11]. Geliştirici ekip müşteri beklentilerini karşılamaya çalıştığı için daha kısa sürede işin tamamlanacağı sonucuna varılabilir. Fakat ileride müşterinin beklentileri genel olarak farklılaştığından geliştirme başarısız da olabilir. Sonuç olarak geliştirme başında direkt sonuca odaklanarak müşterinin daha sonraki aşamalarındaki beklentilerini dikkate almayarak ya da bu beklentileri olumsuz bir şekilde karşılamak geliştirilenin olumsuz sonuçlanmasına yol açar [11]. Çevik yöntemler ise bu

durumları önleyecek şekilde sürekli esnek bir geliştirme sürecini temel almaktadır.

Çevik yöntemler, dinamik, bağlama özgü, ısrarla değişmeyi zorlayan ve büyüme odaklı bir yazılım geliştirme metodudur [5]. Bu durum metodun kullanımındaki başarıma ve kazanma isteği ile ilgilidir. Günümüzde, ortaya çıkan rekabetçi alanlarda başarılı olma, pazar payı ve müşteriler yazılım geliştiren şirketlerin odak noktası haline gelmiştir [8].

İnsanları etkin kullanmak, manevra kabiliyeti, hız ve maliyet tasarrufu kazandırır [5]. Örneğin birlikte çalışan birkaç tasarımcı, her birinin tek başına üreteceği tasarımdan daha iyi bir tasarım ürünü ortaya koyabilir [8].

Bireyler arası etkileşimlere bağlı olarak hızlı bilgi paylaşımı, gerektiğinde süreci değiştirmeyi kolaylaştırır [5]. Çalışan yazılımı kullanmak bizim ne kadar hızlı sonuç üretebileceğimizi ve geri bildirim alabileceğimizi ölçmemize izin verir [5]. Bireyler arasındaki sıkı etkileşim, belgelemeyi en aza indirmeye durumunun olumsuz etkilerini telafi eder [8].

Bu özellikleri aşağıdaki gibi sınıflayabiliriz.

1.3.1 Odaklanma ve Perspektif

Çevik yöntemler proje perspektifine sahiptir. Çevik yöntemlerin odak noktası proje ve proje ekibidir [21].

1.3.2 Yönetim

Çevik yöntemlerde yönetim geliştirme yapan takımın önüne çıkabilecek sorunları gidermeyi hedefleyerek takımın geliştirme yaparken takılmadan ilerleyebilmesini amaçlar [21].

1.3.3 Planlama

Çevik yöntemler sürekli değişen bir planlama yapısı ortaya koyar. Bunlar kısa süreli olanı koşu planlama şeklindedir. Diğerleri ise koşulların planlandığı genel bir planlama yapılması şeklindedir [21].

1.3.4 Öğrenme

Çevik yöntemler geliştirme yapılırken aynı anda öğrenmeyi hedefleyen bir sistemdir. Sorunun karşılaşıldığı anda araştırma yapılarak çözülmesi hedeflenir [21].

1.3.5 Değerlendirme

Çevik yöntemlerde ortaya çıkan proje sonucunda müşteriden alınan geri dönüşler üzerinden belirlenir [21].

1.3.6 Kişisel Gelişim

Çevik yöntemler kişilere önem verdiği için geliştirme konusunda deneyimli insanlardan meydana gelmesi beklenmektedir [21].

1.3.7 Proje Yaşam Döngüsü Faaliyetleri

Çevik yöntemlerde proje geliştirme aşaması ile test aşaması aynı anda yapılması istenmektedir. Bu işlemin yapılmasındaki sebep ise herhangi bir aksaklık en kısa sürede ortaya çıkması hedeflenmektedir. Bu nedenle ek maliyet ve başarısız durumların olabilecek en erken şekilde tespit edilebileceği iteratif bir yol izlenmektedir [21].

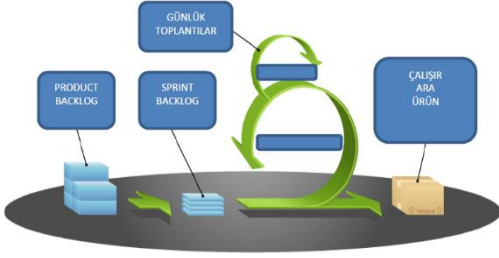
1.3.8 Tahminleme

Bu kısımda ekibin deneyimli olması beklendiğinden ötürü ilk başta belirli süreler verilmektedir. Daha sonra koşullar geçtikçe başta verilen süreler ile uygun şekilde ilerlemesi ön görülmektedir. Buradaki beklenen durum geliştirme ilerledikçe ek bir süreye ihtiyaç duyma oranını minimumda tutmaktır [21].

2. Scrum

Bu bölümde çevik yöntem geliştirme yöntemlerinden biri olan scrum yöntemi anlatılacaktır. Bu yöntemin seçilmesinin sebebi ise endüstri de çevik yöntemlerden en yaygın olarak kullanılan yöntemin scrum olmasıdır [23]. Geliştirmenin en iyi şekilde sonuçlanması için ihtiyaçları belirlemek adına takım ile müşteri arasındaki iletişimin en iyi düzeyde olduğu yöntemlerin başında geliyor [23].

Scrum geliştirme başındaki işlere çok önem vermeden geliştirme aşamasında geliştirme ekibinin oluşabilecek farklılıklara daha iyi refleks gösterebilmesine daha çok önem verir. Scrum metodu koşullardan oluşur. Bu koşulların öncelikle süreleri belirlenir ve koşu sonlarında bir sonraki koşu da neler yapılacağı belirlenir. Ayrıca her gün geliştirimdeki durumumuzu aktardığımız günlük scrum toplantılarından oluşur. Koşu bitiminde oluşturulan geliştirme müşteriyle beraber kontrolü yapılarak bir sonraki koşuya geçilir. Önemli nokta ise koşu'daki işler bitmeden sonraki koşu'ya geçilmez [12, 25]. Scrum yönteminin döngüsü Şekil-2 de gösterilmiştir.



Şekil-2: Scrum Yöntemi

Aşağıda ilk öncelikle scrum yönteminin tarihsel gelişim süreci ve nasıl uygulanacağı detaylı bir biçimde anlatılmıştır.

2.1 Tarihçe

Hirota Takeuchi ve Ikujiro Nonaka 1986 yılında, proje üretiminde aşamaların uyum içinde olduğu ekibin beraber konsantre olarak olabilecek en verimli şekilde geliştirme yapılabilecek şekilde yöntemler ortaya koyuldu [29]. Çevik yöntemler birden çok alanda kullanılması amacıyla ortaya çıkmış olsa da asıl hedefi değişime ve farklılıklara olabildiğince hızlı bir şekilde yanıt verebilmektir. İlk olarak adı rugby ve holistik olsa da her şey takımın birbirini kontrol eden ve geliştirmeyi bitirmeye odaklı olması şeklinde meydana gelmiştir [29].

İlk olarak “Ken Schwaber” 1990’larda bu metodu “Easel” adlı şirketinde kullandı ve ilk olarak Scrum ismi burada ortaya çıktı [24]. Daha sonra “Schwaber” ve “Sutherland” 1995 senesinde yaptıkları geliştirmede Scrum yöntemini aktardıkları bir makaleyi ele aldılar ve bu da insanlar tarafından içeriğini öğrenmeye dahil olan ilk yazı niteliğindedir [26]. Bu yöntem bu sıralarda hemen hemen her geliştirmede başvurulan iyi sonuçlar veren en önemli yöntemlerden biri olmuştur [1].

Bu yöntem firmaların birbiriyle olan yarışları sebebiyle yaptıkları geliştirmelerde daha iyi sonuçlara ulaşmak amacıyla çok sık olarak başvurulan yöntem olmaktadır. [22]. Bu konuda geliştirme işleri ile uğraşan analistler bu yöntemin diğer yöntemlere oranla daha başarılı sonuçlar elde ettiğini kabul etmektedirler [22].

Scrum yöntemi öncelikli olarak önemli olan noktaları bitirmeyi hedef alır. Ve sürekli geri bildirim sağlayarak oluşabilecek hataları veya değiştirilmesi istenen noktaların erken farkına varılmasını sağlar [15].

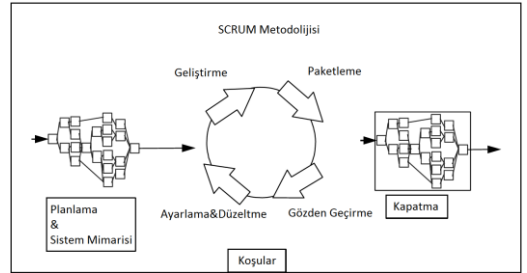
Özette geribildirimler oluşabilecek değişikliklerin zamanında öngörülmesini ve oluşabilecek sıkıntıların önceden fark edilmesini sağlar. Bu yüzden scrum değişen şartlara karşı dayanıklı ve oldukça başarılıdır [23].

2.2 Scrum Aşamaları

Scrum evreleri ilk önce geliştirmede ihtiyaç duyulan ekipmanları belirlemek ile başlar daha sonra müşteri ile genel hatlar üzerinde anlaşılır ve sonrada geliştirme evresinin yapılmasıyla devam edilir [3]. Bu evrelerden ilk olarak hazırlık evresi yapılır. Daha sonra geliştirme evresinde yinelemeli olarak ilerlenir. Her yinelemede bir ara ürün oluşturularak gerekli kontroller yapıp artımlı bir şekilde sonraki yinelemelere geçilir [3]. En sonda bitiş evresi yer alır.

Bu yöntem geliştirmelerin başladığı evre ve sonuç evresi ortasında yinelemeli geliştirme evresinin bulunduğu bir yöntemdir. Bu evreler Şekil-3’de gösterilmiştir. Planlama yani başlangıç ile kapatma yani bitiş evreleri ile yinelemeli evreler arasında düz bir şekilde devam eden yönetim mevcuttur [13]. Burada tüm evrelerde hangi işlemler yapılacağı belirli olmak zorundadır.

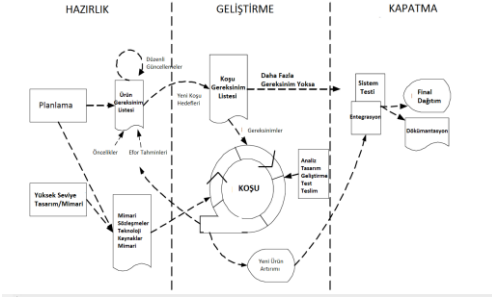
Oluşturulan her ara ürün koşullar ilerledikçe değişiklik gösterebilir [13]. Yani ürün geliştirme durumuna göre değişiklik gösterebilmektedir. [14]. Kısaca scrum işin yapımını göstermez. İşin en verimli şekilde yapılabilmesi için takımlara yetki verir [14].



Şekil-3: Scrum Aşamaları

Her koşu sonrası oluşturulan ara üründe belirlenen değişiklikler sonraki koşullarda önem derecesine göre tekrar planlanır. Bu şekilde ilerleyen koşullarda belirlenen işler değil oluşan sorunlar ve beklemede olan işler önem derecesine göre alınarak ürünün oluşturulması sağlanır [10].

Scrum aşamasının detaylı şeması aşağıda Şekil-4’de gösterilmiştir;



Şekil-4: Scrum'da süreçlerin genel görünümü

Scrum yöntemi 3 aşamadan oluşur. Bu aşamalar hazırlık aşaması, geliştirme aşaması ve kapatma aşamasından oluşur.

2.2 Hazırlık Aşaması

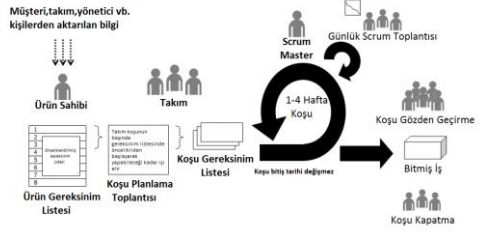
Geliştirilecek ürünün maliyet ve süre belirlemelerini de yaparak gereksinim listesinin çıkarılması aşamasıdır [22].

1. Sırasıyla şunlar yapılır.
2. Gereksinim listesi çıkartılır.
3. Önem sırasına göre gereksinim listesinin derecelendirilmesi yapılır.
4. Geliştirme veya değişiklikler ile ilgili sorunlar ve problemlerin tanımlanması yapılır.
5. Görevlerin ve her bir gereksinim için yeterlilik şartlarının tanımlanması yapılır.

Bu aşama scrum'ın en önemli aşamasıdır. Bunun sebebi bu aşamada yapılacak herhangi bir eksiklik veya hata geliştirme aşamasında yüksek maliyetlere yol açabilir.

2.3 Geliştirme Aşaması

Bu evre yinelemeli aşamalardan oluşur ve Şekil-5'de gösterilmektedir. Bu aşamada bulunan kişiler yapılan işlerin takibi ile ilgilienirler yani yapılan işlere karışmazlar çünkü bu takımların sorumluluğundadır. Sadece takip ettikleri sırada bir problem gördüklerinde ekipleri bilgilendirirler [18, 19].



Şekil-5: Scrum metodolojisi geliştirme evresi

Geliştirme aşamaları koşu planlaması, günlük planlama ve koşu kapatmasıdır.

2.3.1 Koşu Planlaması

Koşular genelde 1 hafta veya 2 hafta olarak planlanır. Her koşunun başında koşu planlama toplantısı gerçekleştirilir [27].

Koşuda hazırlık aşamasında gereksinim listesinde bulunan işler önem derecesine göre alınır.

2.3.2 Günlük Planlama

Bu kısa toplantı (on beş dakika) her iş gününde genelde belirlenen saatte gerçekleştirilir ve tüm takım katılır [27]. Takımın ilerleyişini ve karşılaştıkları engelleri görmek için önemli bir fırsattır. Teker teker tüm ekip üyeleri şu soruların cevaplarını verir [27];

- Dün ne yaptım?
- Bugün ne yapacağım?
- Önümde olan engeller ve karşılaştığım sorunlar neler?

2.3.3 Koşu Kapatma

Bu günümüzde uygulamada bazı takımların es geçtiği bir pratiktir ancak bu Scrum'ı başarılı kılan etkenlerden birisidir [1]. Bu bölümde koşu boyunca alınan işlerde nelerin yapılabildiğine bakılır.

Scrum geri beslemeler ile kendini devamlı geliştiren koşulardan oluşan bir süreç yapısıdır. Bu adım bu özelliğin hayata geçirildiği en önemli ortamlardan birisidir. Bu sebeple bu aşamasın es geçilmesi bu yöntemin başarı oranını düşürecektir.

2.4 Kapatma Aşaması

Bu aşama Geliştirme aşaması bittikten sonra yapılır. Bu aşamada daha sonra projede çıkabilecek hataların daha rahat bir şekilde çözülebilmesi için bir dokümantasyon hazırlanabilir. Bu şekilde projeyi

geliştiren ekipteki kişiler değişse bile çıkan hatalara karşı hızlı bir yanıt alınabilir.

3. Uygulama

Bu bölümde Scrum metodu kullanarak oyun programlama yapılmıştır [17].

Oyunumuz Unity platformu kullanılarak C# dilinde kodlanmıştır. Masaüstü, Android ve iOS platformların da oynanabilecek şekilde yapılmıştır. Prototip üretimi yapılmamıştır. Direk oyunun yapımı işlemine geçilmiştir.

Yaptığımız oyunda bir ok bir de balon vardır. Kullanıcı oka yön vererek balonu vurmaya çalışmaktadır. Oyunda balon sürekli hareket etmektedir. Arka planda ise bir gökyüzü görseli vardır.

Balon sürekli yukarı doğru hareket etmektedir. Bunu da arka plandaki bulutların hareketinden anlaşılmaktadır. Ayrıca ok balonu vurduğu zaman balonun patlama animasyonu yapılmıştır.

Oyuna başlarken kullanıcının üç tane canı vardır. Patlatmadığı her balon için bir can azalmaktadır. Ayrıca bazı balonların içinde can bulunmaktadır. Kullanıcı bu balonları vurunca bir can kazanmaktadır. Balonu ok ile vurunca kullanıcı can kazandığına dair animasyon da yapılmıştır.

Oyunumuzun senaryosu bu şekildedir.

Scrum yöntemi 3 aşamadan oluşmaktadır. Bu bölümlerde uygulamamızda hangi işlemler yapıldı onlardan bahsedilmektedir.

3.1 Hazırlık Aşaması

Bu aşamada oyunun temel özellikleri belirlenmiştir. Bunlara göre Çizelge-3'de gereksinim listesi çıkarılmıştır.

Çizelge-3: Oyun gereksinim listesi

Özellik Adı	Açıklama
Gökyüzü	Arka planda gökyüzü görseli planlanmıştır.
Bulut	Rasgele gelecek şekilde bulut görselleri yapılmalıdır.
Oyunun durumu:	Kaç tane balon patlatıldığını kullanıcının görebilmesi için ekranın sol üst köşesinde bu görsel olacaktır.

Can	Oyunda kaç canımızın kaldığını gösterir. Ekranın sağ üst köşesinde olacaktır.
Oyun Oynanma Koşulları	Oyunun nasıl oynanması gerektiğini açıklayan kısa bir özet yazıdan oluşmaktadır. Oyuna başlamadan önce ekranda görünecektir. Oyuna başladıktan sonra kaybolacaktır.
Oyuna Başlama	Oyun oynama koşullarının hemen altında oyuna nasıl başlandığını açıklayan yazıdır.
Ok	Ekranın altında olacaktır. Kullanıcı balona doğru fırlatabilecektir.
Balon	Ekranın herhangi bir yerinde çıkabilir. Açışal ve konum olarak hareket edebilir.
Menü Tasarımı	Oyun sonlandığında ekrana çıkar. 3 tane buton vardır. Bunlar devam et, tekrar oyna ve çıkış seçenekleridir.
Bir İkonu	Her 5 balonda bir balonun içinde gelmektedir. Oyuncu bunu alırsa 1 can eklenmektedir.

Gereksinim listesi belirlendikten sonra bu maddelere öncelik verilmesi ve süre belirlenmesi yapılmıştır. Bu işlem Çizelge-4'de gösterilen koşu listesinin oluşturulmasını sağlar.

Çizelge-4: Koşu listesi

Yapılacak İş	Süre	Seviye
Gökyüzü ve bulut görseli	3	Yüksek
Balon görseli	2	Yüksek
Menü görseli	2	Yüksek
Unity proje ayarları	4	Yüksek
Ekran boyutu dinamikleştirme	4	Yüksek
Ok'un açışal hareketi	2	Yüksek
Ok'un fırlatma hareketi	3	Yüksek
Ok'un balonu vurma kontrolü	5	Yüksek
Balon'un patlama animasyonu	3	Yüksek
Balon'un rasgele çıkması	2	Orta
Oyunun menü tasarımı	6	Orta
Oyun başlangıç yazıları	3	Düşük
Oyun ve can durumu görseli	3	Düşük
Oyunun masaüstü düzeltmeleri	2	Düşük
Oyunun mobil düzeltmeleri	2	Düşük

Hazırlık aşaması ile ilgili kısımlar artık tamamlanmıştır. Bu bölümden sonra geliştirme aşamasına geçilmiştir.

3.2 Geliştirme Aşaması

Bu aşama yinelenmeli aşamalardan oluşur. Koşu listesinde alınan işler önem sırasına göre yapılır. Her gün ilerleme sürecinde ne durumdayız kontrol edilir. Koşu sonunda koşu da aldığımız maddeler ile ilgili değerlendirme yapılır. Proje bitene kadar yukarıdaki kısımlar tekrar edilir.

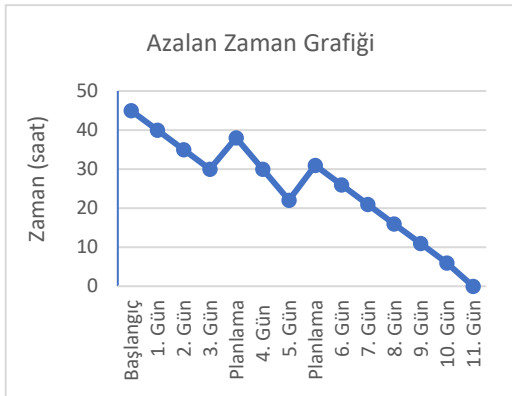
Bu aşamadan sonra Oyun da bazı geliştirmeler ön görülmüştür. Bir ikonu ekleme balonun hareket etmesi gibi bu nedenle aslında önceden yapılması gereken maddeler bu aşamadan sonra belirlendiği için bu kısımdan sonra yapılmıştır.

Çizelge-5: Geliştirme aşamasında ortaya çıkan koşu listesi

Yapılacak İş	Süre	Seviye
Can animasyonu eklenmesi	6	Yüksek
Can ikonu eklenmesi	3	Yüksek
Balon'un açılmal hareketi	3	Orta
Balon'un konumsal hareketi	5	Orta

Çizelge-5 de verilen 4 madde oyunun daha eğlenceli olması için oyuna sonradan eklenen maddelerdir. Yani ek maliyet kısmına girmektedir.

Bu bilgiler ışığında oyunumuz tamamlanmıştır.



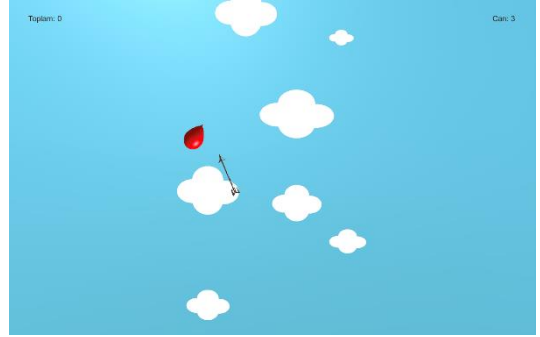
Şekil-6: Balon vurma oyun planlamasının zaman grafiği

Uygulamaya başlarken toplamda 45 saatlik bir sürede tamamlanacağı planlanmıştır. Ancak Çevik yazılımın bize verdiği esneklikten dolayı 3. ve 5. günün sonunda oyunun daha eğlenceli ve oynanabilir olması için birkaç özellik eklenmiştir. Daha sonra koşu eklenen özelliklerin önem derecesine göre tekrar

planlanmıştır. Uygulamanın zaman grafiği yukarıda Şekil-6'de gösterilmiştir.

4. Sonuç

Geliştirme tamamlandıktan sonra ortaya çıkan oyundan bir ekran görüntüsü Şekil-7'de gösterilmiştir.



Şekil-7: Oyun ekran görüntüsü

Sonuç kısmı için bizim bir önceki çalışmamızda da ek maliyet hesaplaması yapmıştık [16]. Bu oyunda da aynı şekilde hesaplamalar yaparak sonuçlar karşılaştırılacaktır.

Burada yapılan oyunu genellediğimizde 4 ana bölümden oluştuğunu söyleyebiliriz. Bu bölümler şunlardan oluşmaktadır [16].

- Oyun görsellerinin tasarımı [16].
- Oyundaki menüler ve oyun bilgilerinin yer aldığı ekranlar [16].
- Oyun ekranında oyunun daha önemsiz yan karakterlerinin oluşturduğu görseller ve senaryolar [16].
- Oyun ekranında oyunun ana karakterlerinin oluşturduğu görseller ve senaryolar [16].

4.1 Oyun Görsellerinin Tasarımı

Balon vurma oyununda kullanıcının can kazanması için 1 ikonu tasarlanmıştır. Bu tasarım için 3 saatlik ek bir süreye ihtiyaç duyulmuştur. Toplamda 14 sürenin 11 saati oyun planlama aşamasında tespit edilmiştir. Yani %27'lik bir zaman sapması yaşanmıştır.

Bir önceki uygulamada ise bu bölüm için toplamda 20 saatlik sürenin 14 saati oyun planlama aşamasında

tespit edilmiştir. Yani %43'luk bir zaman sapması yaşanmıştır [16].

İki oyunu karşılaştırdığımızda birinde %43 değerinde %27'lik bir süre sapması olmuştur. Ortalama ise %35'lik bir süre sapması olmuştur.

4.2 Oyun Menüleri ve Bilgileri

Balon vurma oyununda geliştirilirken herhangi bir değişikliğe ihtiyaç duyulmamıştır. Toplamda 14 saatlik sürenin tamamı oyun başında planlanmıştır. Zaman sapması %0'dır.

Bir önceki uygulamada ise toplamda 18 saatlik sürenin 16 saati oyun planlama aşamasında tespit edilmiştir. Yani %13'lik bir zaman sapması olmuştur [16].

İki oyunu karşılaştırdığımızda birinde %13 değerinde %0'lık bir süre sapması olmuştur. Ortalama ise %7'lik bir süre sapması olmuştur.

4.3 Oyun Yan Karakterleri Yapımı

Balon vurma oyununda başlangıçta 4 saatlik bir süre planlanmıştır. Fakat oyunu geliştirirken Oyuna gerçeklik katmak için ekrandaki balonu açısız hareket ettirmek istedik. Bunun için ise 3 saatlik bir ek süre planlanmıştır. Toplamda 7 saatlik süre harcanmıştır. Yani yaklaşık %75'lik bir süre sapması olmuştur.

Bir önceki uygulamada ise toplamda 13 saatlik sürenin 10 saati oyun planlama aşamasında tespit edilmiştir. Yani yaklaşık %30'luk bir zaman sapması olmuştur [16].

İki oyunu karşılaştırdığımızda birinde %30 değerinde %75'lik bir süre sapması olmuştur. Ortalama ise %53'lik bir süre sapması olmuştur.

Balon vurma oyununda süre sapmasının daha çok olmasının sebebi aslında oyun sonunda fark ettiğimiz balon karakterinin bazı özellikleri aslında oyunda ön plana çıkarak ana karakter grubuna daha yaklaşmasından kaynaklanmıştır.

4.4 Oyun Ana Karakterleri Yapımı

Balon vurma oyunu planlamasında 11 saatlik bir süre planlaması yapılmıştır. Ancak daha sonra oyunu eğlenceli yapmak için can ekleneceği zaman balonun içine 1 ikonunun eklenmesi, balon vurulduğunda bir ikonunun animasyonu ve son olarak da balonun konumsal hareketi yapılması planlanmıştır. Bunlar için ise 11 saatlik bir ek süre planlanmıştır. Yani

toplamda 22 saatlik bir süre harcanmıştır. Zaman sapması ise %100 olmuştur.

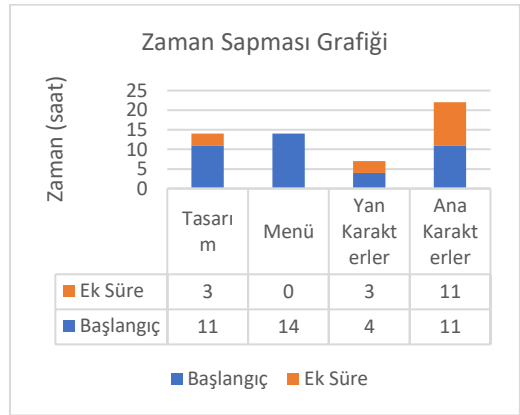
Bir önceki uygulamada ise toplamda 31 saatlik sürenin 11 saati oyun başlangıcında planlanmıştır. Yani yaklaşık %180'lik bir zaman sapması olmuştur [16].

İki oyunu karşılaştırdığımızda birinde %180 değerinde %100'lik bir süre sapması olmuştur. Ortalama ise %140'lik bir süre sapması olmuştur.

Bir önceki bölümde bahsettiğimiz gibi balon vurma oyununda bu oranın daha düşük olmasının sebebi yan karakter ve ana karakterlerin birbirine yakın olmasıdır. Ve geliştirme yapılırken yan karakterlerin ana karakter grubuna daha yaklaşmasından kaynaklanmaktadır.

4.5 Bulgular ve Öneriler

Balon vurma oyununda yukarıda belirtilen başlangıç ve daha sonradan değişen koşullara göre zaman sapması grafiği aşağıda Şekil-8'de gösterilmiştir.



Şekil-8: Balon vurma oyununun zaman sapması grafiği

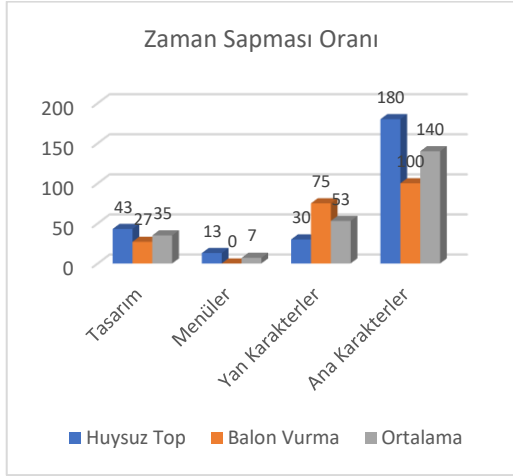
Burada çıkan sonuçlara bakıldığında oyunda daha çok değişiklik olan bölümler oyunun yan karakterleri ve ana karakterlerinin olduğu bölümlerdir. Bu iki kısım ile ilgili verilen süreler göre oyunun bitiş süresi tespiti daha çok etkilemektedir.

Yaptığımız iki oyunda en önemli fark ise yan karakterler ile ana karakterler arasında olmuştur. Burada bu iki bölüm birbiriyle iç içedir. Bu nedenle bu iki bölümü tespit ederken çok dikkat edilmelidir.

Yapılan geliřtirmelerdeki ek maliyeti daha iyi tespit edebilmek için bu örneklerin çoğaltılması gerekir. Çoğaldıkça daha net sonuçlar elde dileyebilir.

Ayrıca sadece scrum yöntemiyle değil diğey çevik programlama yöntemleriyle de bu örnekler yapılabilir.

Ařağıda yapılan balon vurma oyunu ve bir önceki uygulamadaki oyunun süre sapmalarının karşılaştırıldığı grafiğı Şekil-9'da gösterilmiştir.



Şekil-9: İki oyunun zaman sapması karşılaştırma grafiğı

İlerleyen çalışmalarda ise bu 4 bölüm daha detaylandırılabilir. Böylece daha net sonuçlar elde edilebilir.

Ayrıca proje büyüklüğü arttıkça sonuçlar daha güvenilir olacaktır.

Çevik yöntemlerle geleneksel yöntemler karşılaştırılmak isteniyorsa eğer geleneksel yöntemlerle projeler yapıp çıkan sonuçları da karşılaştırılabiliriz.

Kaynakça

[1] Baytam, V., & Kalıpsız, O. Scrum Yazılım Geliřtirme Modeli Yönetim Aracı ScrumMApp. Beřinci Ulusal Yazılım Mühendisliğı Sempozyumu, Eylül 2011.

[2] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Kern, J. *Manifesto for agile software development*. 2001.

[3] Brito, A., & Vieira, J. September). '2TScrum' A Board Game to Teach Scrum. In Proceedings of the 31st Brazilian Symposium on Software Engineering (pp. 279-288). 2017.

[4] Cockburn, A., & Highsmith, J. *Agile software development: The people factor*. Computer, (11), 131-133, 2001.

[5] Duru, I. Çevik yöntemlerde mobil uygulama tasarımı ve gerçekleştirilmesi., 2014.

[6] Grenning, J. Launching extreme programming at a process-intensive company. IEEE Software, 18(6), 27-33, 2001.

[7] Highsmith, J. *What is agile software development?.* crosstalk, 15(10), 4-10, 2002.

[8] Highsmith, J., & Cockburn, A. *Agile software development: The business of innovation*. Computer, 34(9), 120-127, 2001.

[9] Johnson, M. *Agile methodologies: Survey results*. Victoria, Australia: Shine Technologies. 2002.

[10] Karabiyik, T., Jaiswal, A., Thomas, P., & J Magana, A. Understanding the Interactions between the Scrum Master and the Development Team: A Game-Theoretic Approach. Mathematics, 8(9), 1553, 2020.

[11] Karlıdere, T., & Kalıpsız, O. *Yazılım Mühendisliğı Projelerinde Çevik Yaklaşımların Yeri*. In EMO (Vol. 1, pp. 23-25), 2003.

[12] Kristiadi, D. P., Sudarto, F., Sugiarto, D., Sambera, R., Warnars, H. L. H. S., & Hashimoto, K., November). *Game Development with Scrum methodology*. In 2019 International Congress on Applied Information Technology (AIT) (pp. 1-6). IEEE.,2019.

[13] Landaeta R. E., Viscardi S., Tolk A., *Strategic Management of Scrum Projects an Organizational Learning Perspective*, First International Technology Management Conference, San Jose CA USA, 2011.

[14] Lee, W. L. SCRUM-X: An interactive and experiential learning platform for teaching scrum.,2016..

[15] Lippert, M., Becker-Pecbau, P., Breitling, H., Koch, J., Kornstadt, A., Roock, S., ... & Zullighoven, H. *Developing complex projects using XP with extensions*. Computer, 36(6), 67-73, 2003.

[16] Mercan, Ş., ve Becerikli, Y. *Agile Methods in Game Programming based on Scrum*. Sakarya

- Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 24(5), 882-891, 2020.
- [17] Mercan, Ş. *Oyun Programlamada Çevik*, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Ocak 2021.
- [18] Moreira, G. G., & dos Santos Marques, A. B.. *Evaluating the students' experience with the Scrum Card Game: an experience report in a Software Engineering course*. In Proceedings of the 17th Brazilian Symposium on Software Quality (pp. 344-353), 2018.
- [19] Naik, N., & Jenkins, P. August). *Relax, it's a game: Utilising gamification in learning agile scrum software development*. In 2019 IEEE Conference on Games (CoG) (pp. 1-4). IEEE. 2019.
- [20] Reifer, D. J., Maurer, F., & Erdogmus, H. *Scaling agile methods*. IEEE software, 20(4), 12-14, 2003.
- [21] Sahin, E., Keskin, I., ve Koç, H. CMMI-DEV Seviye-3 Sertifikasyonuna Sahip Bir Organizasyonda SCRUM Çevik Yazılım Geliştirme Yöntemi'nin Yazılım Geliştirme Çalışmalarında Uygulanması. In UYMS, 2013.
- [22] Schwaber, K. *Scrum development process*. In Business object design and implementation (pp. 117-134). Springer, London.,1997.
- [23] Schwaber, K., & Sutherland, J. *Scrum guide: developed and sustained*. Scrum. Org, 2009.
- [24] Sutherland, J. Agile development: Lessons learned from the first scrum. Cutter Agile Project Management Advisory Service: Executive Update, 5(20), 1-4. 2004.
- [25] Sutherland, J., Coplien, J. O. *A Scrum Book: The Spirit of the Game*. Pragmatic Bookshelf, 2019.
- [26] Sutherland, J. V., & Schwaber, K. *The SCRUM methodology*. In Business object design and implementation: OOPSLA workshop, 1995.
- [27] Sutherland, J., & Schwaber, K. *The Scrum Papers: Nuts, Bolts and Origins of an Agile Process*, 2010.
- [28] Süloğlu, S., “*Yöntem Çevik Olunca*”. 2.Ulusal Yazılım Mühendisliği Sempozyumu. <http://www.emo.2005>.
- [29] Takeuchi, H., & Nonaka, I. *The new new product development game*. Harvard business review, 64(1), 137-146, 1986.
- [30] Tekinerdogan, B. *Formalizing Agile Software Development Methods*. In Proceedings of Impact of Software Process on Quality Workshop (pp. -), 2003.
- [31] Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. *The agile requirements refinery: Applying SCRUM principles to software product management*. Information and software technology, 53(1), 58-70, 2011.
- [32] Şenkaya, E., “*Yazılım Projelerinde Başarı Anahtarları*”, CIO CLUB Bilişim Dergisi, Haziran: 54-57, 2009.