T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ÇEVİK YAZILIM GELİŞTİRME METODOLOJİSİ İLE KLİNİK ARAŞTIRMALAR İÇİN MOBİL UYGULAMA GELİŞTİRME

Mehmet ELİBOL

Enformatik Anabilim Dalı

Enformatik Programı

Danışman

Yrd. Doç. Dr. Çiğdem SELÇUKCAN EROL

II. Danışman

Prof. Dr. Yağız ÜRESİN

Haziran, 2014

İSTANBUL

Bu çalışma 13 / 06 / 2014 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı Enformatik programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi:

Yrd. Doç. Dr. Çigdem Selçukcan EROL (Danışman)
İstanbul Üniversitesi
Enformatik Bölümü

Prof. Dr. Sevinç GÜLSEÇEN
İstanbul Üniversitesi
Enfotmatik Bölümü

Prof. Dr. Erdal BALABAN
İstanbul Üniversitesi
İşletme Fakültesi

Prof. Dr. Zuhal TANRIKULU
Boğaziçi Üniversitesi

Yrd. Doç. Dr. Zerrin Ayvaz REİS
İstanbul Üniversitesi
Hasan Ali Yücel Eğitim Fakültesi

ÖNSÖZ

Bu çalışmam boyunca güler yüzü ile beni her an yönlendiren, karşılaştığım zorluklara aşmama yardımcı olan ve çalışmama sürekli destek veren benim için bir danışmanda fazlası olan değerli hocam Yrd. Doç. Dr. Çiğdem Selçukcan Erol'a teşekkür ederim.

Tezin uygulamasını geliştirme sürecinde her türlü desteğini esirgemeyen ve taleplerimi hiçbir zaman geri çevirmeyen Prof. Dr. Yağız Üresin ve Uzm. Dr. Selçuk Şen hocalarıma teşekkür ederim.

Ayrıca yurtdışında çalışmamı yürütürken bana destek veren Prof. Dr. Malgorzata Pankowska'yada teşekkür ederim.

Yazılım geliştirme sürecini kurumsal bazlı olarak nasıl gerçekleştirdikleri hakkında bilgi sağlayan Julide Arslan'a teşekkür ederim.

Enformatik bölüm başkanı Prof. Dr. Sevinç Gülseçen ve Enformatik çalışanlarına teşekkür ederim.

Marmara Üniversitesi Bilişim Merkezindeki çalışma arkadaşlarıma tez süresince bana destek oldukları ve yardımları ile yol gösterdikleri için teşekkür ederim.

Bu süreçte bana destek olan ailem ve arkadaşlarıma teşekkür ederim.

Haziran 2014 Mehmet Elibol

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **i**
İÇİNDEKİLER ..... **ii**
ŞEKİL LİSTESİ ..... **v**
SİMGE VE KISALTMA LİSTESİ ..... **vii**
ÖZET ..... **viii**
SUMMARY ..... **x**
1. GİRİŞ **1**
2. GENEL KISIMLAR **4**
2.1. KLİNİK ARAŞTIRMALAR 4
2.2. YAZILIM GELİŞTİRME METODOLOJİLERİ 6
2.3. ÇEVİK YAZILIM GELİŞTİRME METODOLOJİSİ 6
2.4. MOBİL UYGULAMA GELİŞTİRME 14
2.5. SCRUM YAZILIM GELİŞTİRME METODU 18
2.5.1. Scrum'a Genel Bakış 18
2.5.2. Scrum'daki Kavramlar 21
2.5.2.1. Koşu 21
2.5.2.2. Ürün Gereksinim Listesi 23
2.5.2.3. Koşu Gereksinim Listesi 26
2.5.2.4. Koşu İş Takip Grafiği 29
2.5.2.5. Koşu Hedefi 29
2.5.2.6. Kullanıcı Hikayeleri 29
2.5.3. Scrum Sürecindeki Roller 30
2.5.3.1. Ürün Sahibi 31
2.5.3.2. Geliştirme Ekibi 32
2.5.3.3. Scrum Uzmanı 34
2.5.4. Scrum Sürecindeki Toplantılar 36
2.5.4.1. Koşu Planlama Toplantısı 37
2.5.4.2. Günlük Scrum Toplantıları 40
2.5.4.3. Koşu Değerlendirme Toplantısı 41
2.5.4.4. Koşu Süreç Değerlendirme Toplantısı 44
2.5.5. Scrum Sürecinin Gerçek Hayatta Uygulanması 45

3. MALZEME VE YÖNTEM 51
3.1. MALZEME 51
3.1.1. Veritabanı 52
3.1.2. Servisler 52
3.1.3. Uygulama 53
3.1.4. Süreç Yönetim Yazılımları 53
3.2. YÖNTEM 54
3.2.1. Proje Sürecindeki Scrum Rolleri 56
3.2.2. Ürün Gereksinim Listesi 57
3.2.3. Kullanıcı Ekran Prototiplerinin Hazırlanması 58
3.2.4. Scrum Planlama 61
3.2.5. Birinci Koşu Planlama Toplantısı ve Birinci Koşu Gereksinim Listesi 62
3.2.6. Scrum Uzmanı tarafından İş Takip Grafiğinin Hazırlanması 66
3.2.7. Birinci Koşu Süreci 67
3.2.8. Birinci Koşu Süresince Günlük Scrum Toplantıları 69
3.2.9. Birinci Koşu Değerlendirme Toplantısı 70
3.2.10. Birinci Koşu Süreç Değerlendirme Toplantısı 71
3.2.11. İkinci Koşu Planlama Toplantısı ve İkinci Koşu Gereksinim Listesi 71
3.2.12. Scrum Uzmanı Tarafından İkinci Koşu için İş Takip Grafiğinin Hazırlanması 75
3.2.13. İkinci Koşu Süreci 75
3.2.14. İkinci Koşu Süresince Günlük Scrum Toplantıları 77
3.2.15. İkinci Koşu Değerlendirme Toplantısı 77
3.2.16. İkinci Koşu Süreç Değerlendirme Toplantısı 78
3.2.17. Üçüncü Koşu Planlama Toplantısı ve Üçüncü Koşu Gereksinim Listesi 78
3.2.18. Scrum Uzmanı Tarafından Üçüncü Koşu İçin İş Takip Grafiğinin Hazırlanması 81
3.2.19. Üçüncü Koşu Süreci 82
3.2.20. Üçüncü Koşu Süresince Günlük Scrum Toplantıları 84
3.2.21. Üçüncü Koşu Değerlendirme Toplantısı 84
3.2.22. Üçüncü Koşu Süreç Değerlendirme Toplantısı 84
3.2.23. Ürünün Yayınlanması ve Proje Sürecinin Kapatılması 84
4. BULGULAR 85
4.1. KLİNİK ARAŞTIRMALAR UYGULAMASI 85
5. TARTIŞMA VE SONUÇ 93

KAYNAKLAR 97
EKLER 100
ÖZGEÇMİŞ 101

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Yazılım projeleri başarı anketi (Williams ve diğ, 2011). 8

Şekil 2.2: Scrum Süreci (Highsmith, 2002).....21

Şekil 2.3: Koşu Planlama Toplantısında görevler için birimlerin belirlenmesi.....48

Şekil 2.4: Koşu aşamasında görevleri gerçekleştirme planı. 49

Şekil 2.5: Koşu İş Takip Grafiği örneği. 50

Şekil 3.1: Uygulama Geliştirme Sürecinde Kullanılan Araçlar.....51

Şekil 3.2: Uygulama Geliştirme Proje Süreci.....55

Şekil 3.3: Kullanıcı Karşılama ve Yönlendirme Ekranı Prototipleri. 59

Şekil 3.4: Konuları Listeleme ve Konu Detay Ekranı Prototipleri. 59

Şekil 3.5: Öğretici Sorular Listeleme ve Öğretici Soru Cevaplama Ekran Prototipleri. 60

Şekil 3.6: Öğretici Soru Geri Bildirim Ekran Prototipi. 60

Şekil 3.7: Sık Sorulan Soruları Listeleme ve Sık Sorulan Soru Detay Ekran Prototipleri. 61

Şekil 3.8: Birinci Koşu için Kullanıcı Hikayeleri.....63

Şekil 3.9: Koşu Boyunca gerçekleştirilen Görev Kartları. 65

Şekil 3.10: Koşu Boyunca gerçekleştirilen Görev Listesi. 66

Şekil 3.11: Birinci Koşu sırasında takip edilen İş Takip Grafiği.....67

Şekil 3.12: Birinci Koşu sonucunda oluşan Kullanıcı Karşılama ve Yönlendirme Ekranları. 69

Şekil 3.13: Birinci Koşu sonucunda oluşan Konuları Listeleme ve Konu Detay Ekranları. ..69

Şekil 3.14: İkinci Koşu için Kullanıcı Hikaye Kartları. 72

Şekil 3.15: İkinci Koşu boyunca gerçekleştirilen Görev Kartları.....74

Şekil 3.16: İkinci Koşu boyunca gerçekleştirilen Görev Listesi. 74

Şekil 3.17: İkinci Koşu sırasında takip edilen İş Takip Grafiği. 75

Şekil 3.18: İkinci Koşu süreci sonucunda oluşan Kullanıcı Karşılama ve Yönlendirme Ekranları. 76

Şekil 3.19: İkinci Koşu Süreci sonucunda oluşan Sık Sorulan Sorular Listeleme ve Sık Sorulan Soru Detay Ekranları. 77

Şekil 3.20: İkinci Koşu sonucunda oluşan İletişim ve Hakkında Ekranları. 77

Şekil 3.21: Üçüncü Koşu için Kullanıcı Hikaye Kartları. 79

Şekil 3.22: Üçüncü Koşu sürecinde gerçekleştirilen Görev Kartları. 80

Şekil 3.23: Üçüncü Koşu sürecinde gerçekleştirilen Görev Listesi. 81

Şekil 3.24: Üçüncü Koşu sırasında takip edilen İş Takip Grafiği. 81

Şekil 3.25: Üçüncü Koşu sonucunda oluşan Öğretici Sorular Listeleme ve Öğretici Soru Cevaplama Ekranları. 83

Şekil 3.26: Üçüncü Koşu Sonunda Oluşan Öğretici Soru Geri Bildirim Ekranları. 83

Şekil 4.1: Klinik Araştırmalar Uygulaması çalışma prensibi. 85

Şekil 4.2: Uygulama Akışı. 86

Şekil 4.3: Kullanıcı Karşılama Ekranı. 87

Şekil 4.4: Yönlendirme Ekranı. 87

Şekil 4.5: Konu Listeleme Ekranı. 88

Şekil 4.6: Konu Detay Ekranı. 88

Şekil 4.7: Öğretici Soru Listeleme Ekranı. 89

Şekil 4.8: Öğretici Soru Detay Ekranı 89

Şekil 4.9: Öğretici Soru Geri Bildirim Ekranı 90

Şekil 4.10: Sık Sorulan Sorular Listeleme Ekranı 90

Şekil 4.11: Sık Sorulan Sorular Detay Ekranı. 91

Şekil 4.12: İletişim Ekranı. 91

Şekil 4.13: Uygulama Hakkında Ekranı. 92

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar	Açıklama
ASD	**:** Adaptif Yazılım Geliştirme (*Adaptive Software Development*)
DSDM	**:** Dinamik Sistem Geliştirme Metodu (*Dynamic Systems Development Method*)
EPF	**:** Eclipse Süreç Çerçevesi (*Eclipse Process Framework*)
FDD	**:** Özelliğe Yönelik Geliştirme (*Feature-Driven Development*)
LSD	**:** Lean Yazılım Geliştirme (*Lean Software Development*)
MASAM	**:**Mobil Uygulama Geliştirme Metodu (Mobile Application SW Development Method)
TGG	**:** Test Güdümlü Geliştirme (*Test Driven Development*)
XML	**:** Genişletilebilir İşaretleme Dili (*Extensible Markup Language*)
XP	**:** Uç Programlama (*Extreme Programming*)

ÖZET

YÜKSEK LİSANS

ÇEVİK YAZILIM GELİŞTİRME METODOLOJİSİ İLE KLİNİK ARAŞTIRMALAR İÇİN MOBİL UYGULAMA GELİŞTİRME

Mehmet ELİBOL

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman: Yrd. Doç. Dr. Çiğdem SELÇUKCAN EROL

II. Danışman: Prof. Dr. Yağız ÜRESİN

Her mobil cihazın özgünlük ve kısıtlarının olması nedeniyle bu cihazlar üzerinde en performanslı çalışacak uygulamayı tasarlamak için mobil uygulama geliştirme sırasında dikkat edilmesi gereken birçok nokta bulunmaktadır. Mobil uygulama pazarında değişen müşteri istekleri ve değişen piyasa şartlarını yansıtan mobil uygulamaların en hızlı şekilde pazara çıkması önemli olmaktadır. Bunu gerçekleştirebilmek için mobil uygulama geliştirme sürecinin esnek bir yapıyla sürdürülmesi gerekmektedir.

Günümüz yazılım projelerinde, gelişen teknoloji ile birlikte ihtiyaçlar süreç içinde sürekli olarak değişmektedir. Bu süreçte kaçınılmaz değişikliklere cevap verebilmek için, çevik metodolojiler proje sürecinde esnek bir yapı sunarak, gelişen teknoloji ve değişen müşteri isteklerine göre proje sürecinin daha kolay sürdürülmesini sağlamaktadır.

Çevik metodolojilerden olan Scrum, sabit süreleri olan yinelemeleri kullanarak ürünün artırımlı olarak inşa edilmesini sağlayan bir süreç çerçevesidir. Yenilikçi mobil uygulamaların geliştirilmesinin önemli olduğu pazarda, Scrum metodunu mobil uygulama geliştirme sürecine uyarlama çalışmaları yapılmıştır.

Bu tezin amacı, klinik araştırmalar alanında ihtiyaç duyulan bir mobil uygulamanın çevik yazılım geliştirme metodolojisi kullanılarak gerçekleştirilmesidir. Bu kapsamda, çevik yazılım geliştirme metodolojilerinden birisi olan Scrum sürecindeki kavramlar detaylandırılarak, Scrum metodunun, mobil uygulama geliştirme sürecinde, nasıl uygulanabileceği hakkında bir çalışma yapılmıştır. Bu süreçte ön plana çıkan noktalar; projeye başlamak için ekiplerin nasıl oluşturulduğu, müşterinin ihtiyaçlarının nasıl tespit edildiği, süreç içinde değişen müşteri isteklerinin nasıl yönetildiği, bu süreçte şeffaflık, denetim ve adaptasyonun nasıl gerçekleştirildiği ve Scrum sürecindeki kavramlar baz

alınarak detaylandırılmıştır. Tez sonucunda geliştirilen Android tabanlı mobil uygulama Google Markette yayınlanmıştır.

Haziran 2014, 112 Sayfa.

Anahtar kelimeler: Çevik Metodolojiler, Mobil Uygulama Geliştirme, Scrum, Klinik Araştırmalar

SUMMARY

M.Sc.THESIS

MOBİLE APPLICATION DEVELOPMENT WITH AGILE SOFTWARE DEVELOPMENT METHODOLOGY FOR CLİNİCAL TRIALS

Mehmet ELİBOL

İstanbul University

Graduate School of Science and Engineering

Informatics Department

Supervisor : Asst. Prof. Dr. Çiğdem SELÇUKCAN EROL

Co-Supervisor : Prof. Dr. Yağız ÜRESİN

There are many points to take into consideration during the development of mobile application in order to design the application to work with the highest performance on these devices since every mobile device has specificities and limitations. It is very important that mobile applications reflecting the changing market conditions and customer demands in mobile application market take their places in the market in the fastest way. In order to achieve this, the process of developing mobile application should be maintained with a flexible structure.

In today's software projects, needs continuously change within the process in company with the advancing technology. In this process, agile methodologies offer a flexible structure and enable the maintenance of the project process in line with the changing customer demands and advancing technology in order to give an answer to the inevitable changes.

Scrum, one of the agile methodologies, is a process frame enabling the construction of the product incrementally by using the iterations that have fixed durations. Adapting the Scrum method to the mobile application developing process has been performed in the market where developing innovative mobile applications are important.

The aim of this thesis is to achieve a mobile application needed in the field of clinical trials by using agile software developing methodology. In this context, a study was carried out about how to apply the Scrum method in the process of developing mobile application by detailing the concepts in the process of Scrum which is one of the agile software development methodologies. The points becoming prominent in this process such as how teams are created to start the project, how customer needs are determined, how changing

customer demands are managed during the process and how transparency, inspection and adaptation are provided were detailed based on the concepts in Scrum process. Android-based mobile application developed in consequence of the thesis was published on Google Market.

June 2014, 112 Pages.

Keywords: Agile Methodologies, Mobile Application Development, Scrum, Clinical Trials

1. GİRİŞ

Yazılım sektöründe projelerin üretilmeye başlaması ile birlikte, bu proje süreçlerinde takip edilen bir çok metodoloji ortaya çıkmış ve bu metodolojiler projelerde uygulanmıştır. Zaman içerisinde ilk metodolojilerde çıkan eksiklikler ya da uygulama zorlukları nedeniyle bu metodolojiler revize edilmiş ya da yeni metodolojiler geliştirilmiştir.

Günümüzde mobil uygulama geliştirme alanında projeler üretilmeye başlanmıştır. Bu projelerin başarılı şekilde tamamlanması önemli bir noktayı teşkil etmektedir. Bu projelerin başarılı şekilde tamamlanması için müşteri ihtiyaçlarındaki değişimle birlikte, mobil akıllı cihazlarda yaşanan gelişme ve değişimlere dikkat ederek süreç içinde oluşan değişimlere uyum sağlanması gerekmektedir.

Son yıllarda ise özellikle teknoloji ve yazılım dünyasında yaşanan hızlı gelişmeler ve değişmeler neticesinde yazılım projelerini yönetmekte zorluklar ortaya çıkmaktadır. Farkı metodlar denenmiş olmasına rağmen, bu değişiklikleri en iyi şekilde yönetebilmek amacıyla çevik metodlar üzerine yoğunlaşılmıştır. Çevik metodların kullanılması ile birlikte müşteri isteklerine duyarlı ve müşteri isteklerinden gelen fikirlere göre değişim gösteren yazılım projeleri ortaya çıkmaya başlamıştır (Highsmith, 2002).

Teknolojik gelişmeler sayesinde akıllı telefonlar, beraberindeki mobil uygulamalarla günlük hayatın bir parçası haline gelmektedir. Scharff ve Verma (2010) mobil uygulama alanının; sosyal ağ, eğlence, pazarlama, haritalama, eğitim, sağlık ve tarım gibi alanlarda uygulama türlerinin çeşitleri ile hızlı şekilde geliştiğini belirtmektedir.

Mobil uygulama geliştirme, teknoloji ve yaratıcılığın önemli olduğu zorlu bir çalışma sürecidir (Scharff ve Verma, 2010). Mobil uygulama geliştirmeye başlarken dikkat edilmesi gereken noktalardan birisi bu süreçte teknolojideki gelişmelerin çok yakından ve iyi derecede takip edilmesidir. Çünkü mobil uygulama geliştirme süreci teknolojideki değişikliklere göre tekrar şekillenebilmektedir. Teknolojideki değişiklikleri takip etmekle birlikte bu süreçte yeni fikirlere açık olmak da önemli bir noktayı teşkil etmektedir.

Çünkü yeni fikirler yaratıcılığın ortaya çıkmasını sağlamaktadır. Yaratıcı fikirler de uygulamaların piyasa şartlarında güçlü olmasını sağlamaktadır.

Uygulamalar bir platform üzerinde geliştirilir ve başka mobil cihazlarda da kullanılması gerektiğinde uyarlanması lazımdır (Scharff ve Verma, 2010). Bu noktadan hareketle mobil uygulama sürecinde gelebilecek cihaz değişikliklerine göre mobil uygulama geliştirme sürecinin revize edilmesi gerekmektedir. Bu değişikliği sağlayabilmek için, proje sürecinin bu esnekliği sağlayacak bir yapıda olması gerekmektedir.

İnsanların önem verdikleri noktalardan birisi olan sağlık alanındaki bilgilere de her an her yerden ulaşmak önemli konuma gelmiştir. Bu önemli durum neticesinde, bireylerin bu alandaki bilgilere hızlı şekilde ulaşmalarını sağlamak için mobil uygulamalar geliştirilmeye başlanmıştır. Bu gelişmeler sonucunda bireyler sağlık alanındaki mobil uygulamaları kullanmaya başlamışlardır.

Mobil akıllı cihazların kullanılmaya başlaması ile birlikte, bu cihazlar üzerinde kullanılacak uygulamalar geliştirilmeye başlanmıştır. Bu uygulamaları geliştirme sürecinin gerçekleştirilmesi için, bu süreçte yaşanan kullanıcı isteklerindeki değişimle birlikte, mobil cihazlar alanında yaşanan değişimler ve gelişmelerinde dikkat edilmesi gerekmektedir.

Bu tezin amacı, mobil uygulama geliştirme sürecini detaylandırarak, müşteri isteklerindeki değişime süreç içinde yanıt verme esnekliği tanıyan bir yapıyla, klinik araştırmalar alanında ihtiyaç duyulan bir uygulama gerçekleştirilmesidir.

Bu tezin önemi, klinik araştırmalar için bir mobil uygulama geliştirirken, yeni bir alan olan mobil uygulama geliştirme alanında, proje süreçlerinin nasıl gerçekleştirildiğini ön plana çıkarmaktır.

Bu tez çalışmasında çevik metodolojiler, mobil uygulama geliştirme, çevik yöntemlerle mobil uygulama geliştirme alanında yapılan çalışmalar incelenmiş ve çevik metodolojilerle mobil uygulama geliştirme süreci birleştirilerek klinik araştırmalar alanında müşteri isteklerindeki değişimlere yanıt veren bir mobil uygulama geliştirilmesi gerçekleştirilmiştir.

Genel kısmında klinik araştırmalar hakkında bilgi verilmiş, yazılım geliştirme metodolojilerine genel olarak değinilmiş ve çevik yazılım geliştirme metodolojisinden bahsedilmiştir. Çevik yazılım geliştirmenin günümüz projeleri için neden önemli olduğu üzerinde durulurken, proje başarısı üzerindeki etkilerinden bahsedilmiştir. Projeleri başarı ile tamamlamak için değişim ve gelişimin nasıl ele alındığı ve süreç içinde nasıl uygulandığı üzerinde durulmuştur. Bununla birlikte günümüzde birçok işlem mobil uygulamalar üzerinden gerçekleştirilmektedir. Bunun sonucunda mobil uygulama geliştirme alanında başarılı projeler üretmek önemli hale gelmiştir. Mobil uygulama geliştirme alanında başarılı projeler üretmek için dikkat edilmesi gereken noktalara değinilmiştir. Mobil uygulama geliştirme alanında başarılı projeler için süreçlerin esnek olması gerektiği üzerinde durulmuştur. Bunu gerçekleştirmek için projelerin çevik metodolojiler ile yönetilmesinin gerektiği görülmüştür. Çevik metodolojilerden olan Scrum metodu ile mobil uygulama geliştirme alanını birleştiren çalışmalar incelenmiştir. Scrum metodu proje sürecinde değişime karşı son derece duyarlı olarak başarılı projeler üretilmesini öngörmektedir.

Malzeme ve Yöntem kısmında; mobil uygulama geliştirme sürecinde kullanılan malzemeler ve Scrum metodu ile mobil uygulama geliştirme sürecinin nasıl gerçekleştirildiği detaylandırılmıştır. Müşteri ihtiyaçlarının nasıl tespit edildiğinden başlanarak, Scrum Ekibinin nasıl oluşturulduğu, mobil uygulama geliştirme sürecinin Scrum metodunun ilkelerine göre yinelemeli ve artırımlı olarak nasıl gerçekleştirildiği üzerinde durulmuştur.

Bulgular kısmında klinik araştırmalar için geliştirilen mobil uygulamanın çalışma mantığı ve ekran görüntülerine yer verilmiştir.

Tartışma ve Sonuç kısmında, çevik metodolojiler ve mobil uygulama geliştirme süreci değerlendirilerek ortaya çıkan önemli noktalar üzerinde durulmuştur. Mobil uygulama geliştirme sürecinde neler yaşandığı, mobil uygulama geliştirme sürecinde nelere dikkat edilmesi gerektiği üzerinde durulmuştur.

2. GENEL KISIMLAR

2.1.KLİNİK ARAŞTIRMALAR

13 Nisan 2013 tarih ve 28617 sayılı resmi gazetede yayımlanarak yürürlüğe giren "Klinik Araştırmalar Hakkında Yönetmelik"te Klinik araştırmalar; "Bir veya birden fazla araştırma ürününün klinik, farmakolojik veya diğer farmakodinamik etkilerini ortaya çıkarmak ya da doğrulamak; advers olay veya reaksiyonlarını tanımlamak; emilim, dağılım, metabolizma ve atılımını tespit etmek; güvenliliğini ve etkililiğini araştırmak amacıyla insanlar üzerinde yürütülen çalışmalar" olarak tanımlanmıştır[1].

Dünya nüfusunun giderek artması, buna paralel olarak yaşlı nüfusun artması, yaşam koşulları ve çevresel etmenlerin değişmesi gibi birçok etmenin sonucunda sağlık sorunları da çeşitlenerek artmaya devam etmekte, yeni tedavilere ve yaklaşımlara ihtiyaç duyulmaktadır. Bu bağlamda tıp alanında gelişmenin sağlanması da ancak araştırmaların devamı ile mümkündür. Klinik araştırmalarda araştırma konusu insan olduğu için yasal, etik, teknik ve pratik birçok hususa dikkat edilmesi gerekmektedir (İlbars, 2014).

T.C Sağlık Bakanlığı Türkiye Halk Sağlığı Kurumu klinik araştırmaların çalışmalara katılan gönüllüler açısından fayda ve potansiyel risklerini aşağıdaki gibi özetlemiştir[2].

Olası Yararlar;

- Hastanın kendi sağlık durumu hakkında aktif rol alması.
- Herkesin ulaşamadığı yeni tedavi yöntemlerine ulaşabilmesi.
- Araştırma süresince uzmanlaşmış bir medikal bakım elde etmesi.
- Kendi durumunda olan diğer hastalara yardım edebilmesi olarak sıralanmıştır.

Potansiyel Riskler;

- İlaçların veya diğer tedavilerin yan(istenmeyen) etkilerinin olması.
- Tedavinin etkili olmaması.

1http://www.ttb.org.tr/mevzuat/index.php?option=com_content&view=article&id=954:klnk-aratirmalar-hakkinda-yoenetmelk&catid=2:ymelik&Itemid=33, [Ziyaret Tarihi: 25 Nisan 2014].

2http:://www.thsk.saglik.gov.tr/2013-10-01-11-00-51/halk-sagligina-yonelik-bilgiler/780-klinik-ara%C5%9Ft%C4%B1rmalar-hakk%C4%B1nda.html, [Ziyaret Tarihi: 25 Nisan 2014].

- Araştırma protokolü, tedavi bölgesine seyahatler, hastanede kalış veya kompleks doz uygulamaları gerekliliğiyle çok fazla zaman harcanması.
- Yeni tedavinin hasta için kullanılabilir olmaması olarak sıralanmıştır.

Klinik Araştırmalar Hakkında Yönetmelik uyarınca yapılacak araştırma ile ilgili olarak olası fayda ve riskler çalışmaya katılacak olan gönüllüye ayrıntılı ve anlaşılabilir şekilde açıklanmalı ve gönüllü rızasını yazılı olarak ispat eden "bilgilendirilmiş gönüllü olur formu" isimli belgenin alınması gerekmektedir[3]. Gönüllünün rızası olmadan hiçbir araştırmaya başlanamaz.

Yukarıdaki açıklamalardan da anlaşılacağı üzere klinik araştırmalar, tıbbın ilerlemesi ve yeni tedavilerin geliştirilmesine, hastaların yeni tedavilere yaygın olarak toplum kullanımına sunulmadan önce erişebilmesine, hastaların mevcut tedaviden daha üstün olabilecek tedaviye daha erken dönemde erişmesine olanak sağlamaktadır. Bunların yanında klinik araştırmalar, araştırma kapsamında yapılan tüm tetkik ve değerlendirmeler destekleyici tarafından karşılandığı için de ülke sağlık bütçesine ek maliyetler getirmemektedir. Bu konu ile ilgili olarak Klinik Araştırmalar Hakkında Yönetmelik'te "Araştırmada kullanılan her türlü araştırma ürününün, ürünlerin kullanılmasına mahsus cihaz ve malzemelerin, tetkik ve tahlillerin bedeli araştırmacı veya destekleyici tarafından karşılanır, hasta veya sosyal güvenlik kurumlan tarafından ödenmez ve bunlara ödettirilemez" açıklaması bulunmaktadır.

Dünyada farklı ülkelerde klinik araştırmalar için geliştirilmiş çeşitli mobil uygulama araçları bulunmaktadır. Ancak ülkemiz açısından İstanbul Üniversitesi İlaç Araştırmaları birimi ve İstanbul Tıp Fakültesi Tıbbi Farmakoloji Anabilim Dalı için geliştirilen bu uygulama ilk olma özelliğini taşımaktadır. Bu uygulama zaman içerisinde de gelişmelere göre güncellenecek ve içeriği genişletilecektir.

Mobil uygulama gibi araçlar ile araştırmacılar ve toplumda klinik araştırma bilincinin ve farkındalığının oluşmasına katkıda bulunulması hedeflenmektedir.

[3]http://www.ttb.org.tr/mevzuat/index.php?option=com_content&view=article&id=954:klnk-aratirmalar-hakkinda-yoenetmelk&catid=2:ymelik&Itemid=33, [Ziyaret Tarihi: 25 Nisan 2014].

2.2.YAZILIM GELİŞTİRME METODOLOJİLERİ

Gelişen teknoloji ile birlikte insanların ihtiyaçları günden güne değişmektedir. Bir kişi için bugün önemli bir ihtiyaç olan teknolojik bir araç, birkaç ay sonra ihtiyacını karşılamaz duruma gelebilmektedir. Yazılım projelerinde de benzer bir durum söz konusudur. Yazılım projelerine başlarken müşterinin ihtiyaçları tespit edilerek başlanmaktadır. Süreç içinde bu ihtiyaçlar değişebilmektedir.

1960'lı yıllardan sonra yazılım mühendisliği ile ilgilenen topluluklar tarafından birçok farklı yazılım geliştirme metodolojisi ve metodu tanıtılmış ve kullanılmıştır (Huo ve diğ, 2004). Bu süreçte yazılım geliştirme metodolojileri geleneksel yaklaşımlar ve çevik yaklaşımlar olarak iki kısma ayrılmıştır. Geleneksel geliştirme metodolojileri arasından, Şelale modeli en eski yazılım süreç geliştirme modelidir (Huo ve diğ, 2004). Şelale modelinin büyük devlet yazılım projelerine uygulanması sırasındaki iyileştirme çalışmalarına dayalı olarak Spiral model yapısı oluşmuş ve yıllar içinde gelişmiştir (Boehm, 1988). Geleneksel yaklaşımlardan şelale modelinin bir başka varyasyonu olan V-Model test süreçlerine daha çok önem vermektedir (Dennis ve diğ, 2010). Yazılım geliştirme sürecini yönetirken bu süreç içinde nesneye yönelik ilkeleri destekleyerek maksimum verimlilik sağlamak için UML gibi çeşitli metodolojiler de kullanılmaktadır (Ayvaz, 1999). Yazılım geliştirmenin daha iyi yollarını ortaya çıkarmak için Beck ve arkadaşları (2001) tarafından çevik manifesto oluşturulmuştur.

Bu metodolojilerden ve metotlardan hangisinin kullanılacağına karar vermek önemli bir aşamadır. Teknolojideki gelişme hızından dolayı, diğer metodolojilerle kıyaslandığında daha esnek yapıya sahip olan çevik yazılım geliştirme metodolojisini ön plana çıkarmaktadır.

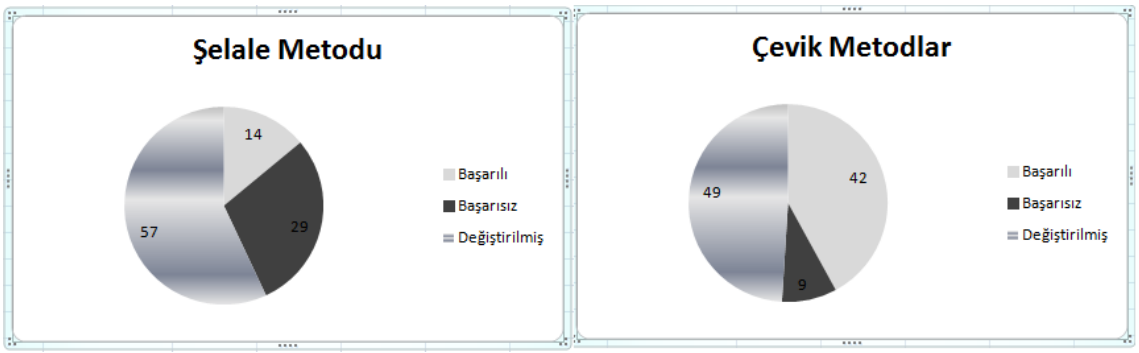
2.3. ÇEVİK YAZILIM GELİŞTİRME METODOLOJİSİ

Hızla gelişen ve değişen yazılım dünyasında, gelişmelere ve değişime bağlı olarak ihtiyaçlar değişmektedir. Örneğin; elektronik imzanın geliştirilmesinden sonra yazılım şirketleri elektronik imza ile işlem yapabilen sistemleri hayata geçirmeye başlamışlardır. Elektronik imza ile çalışan sistemler yaygınlaştıkça elektronik imza ile işlem yapan kişi sayısı da artmaya başladı. Bu süreç devam ederken, teknolojideki gelişmelerle birlikte

mobil imza kavramı ortaya çıktı. Mobil imza kavramının ortaya çıkması ile birlikte elektronik imza ile çalışan projelerin revize edilmesi gerekti. Bu noktadan hareketle yazılım projeleri başka alanlardaki projelere kıyasla değişime daha duyarlıdır. Bu açıdan ele alındığında yazılım projeleri her zaman değişime ve gelişime açık olmalıdır.

Geleneksel yaklaşımlar yeterince çalışılırsa, ürün geliştirmenin erken aşamasında gereksinimlerin tamamı tahmin edilebilirse değişimi ortadan kaldırarak maliyetin azaltılabileceğini kabul ederler. Bu sürekli değişen ve gelişen teknolojinin olduğu bir iş dünyasında geçerliliğini yitirmiştir. Değişimi ortadan kaldırmak iş koşullarına tepkisiz olmak demektir. Bu da yapılan işlerde ve projelerde iş başarısızlığı anlamına gelir (Highsmith ve Cockburn, 2001). Bundan dolayı etkili yazılım geliştirmenin en iyi yolu planlanan süreçlerden ziyade süreci deneysel şekilde uygulamaktır (Leffingwell ve Smiths,2006). Değişimin hızla gerçekleşebilmesi için yazılım projelerinin geleneksel yazılım geliştirme yöntemleri yerine değişime hızla ayak uydurabilen yöntemlerle gerçekleştirilmesi ihtiyacı doğmuştur. Bu şekilde değişime ayak uydurabilen metodlar proje başarısına doğrudan etki edebilmektedir.

Standish Grup tarafından yapılmış olan ve 2002 ile 2010 yılları arasını kapsayan çalışma ile Şelale Metodu ve Çevik Metodlar karşılaştırılarak proje başarılarına etkisi incelenmiştir. Şekil 2.1'de görüldüğü gibi; Şelale Metodu ile projelerin başarılı olma oranı %14 de kalırken, Çevik Metodlarda bu oranın % 42 olduğu görülmektedir. Bu da Çevik projelerin geleneksel projelere göre 3 kat daha fazla başarı oranına sahip olduğunu göstermektedir (Williams ve diğ, 2011).



Şekil 2.1: Yazılım projeleri başarı anketi (Williams ve diğ, 2011).

Bu verilerden sonra günümüzün sorusu; yaşam döngüsü boyunca bir projede değişim nasıl durdurabilir değil, kaçınılmaz değişiklikler nasıl işleme alınabilirdir (Highsmith ve Cockburn, 2001). Bunun için, metodolojinin temel amacı, geliştirme döngüsü boyunca uygun zamanda ihtiyaç olan ürünü sunmaktır (Flora ve Chande, 2013).

Çeviklik, çalkantılı iş ortamında kâr sağlamak için değişim oluşturmak ve değişime yanıt verebilme özelliklerinin ikisini kapsayan bir yetenektir (Dove, 2001). Çevikliği yazılım geliştirme sürecine göre yorumlayacak olursak yazılım geliştirme sürecinin her aşamasında değişimi kabul edilebilir olarak görmek ve bu değişimi proje sürecine uygulayabilmektir. Buradan hareketle çeviklik, değişen ihtiyaçlara göre projenin tekrar şekillendirilmesi olarak da tanımlanabilir.

Çeviklikte başarı sağlamak için genel kurallar, sınırlar ya da ölçüler bulunan bir yönerge bulunmamaktadır (Dove, 2001). Projeler çeviklik anlayışına göre sürdürülüyorsa, her zaman değişime açık olduğu anlaşılmaktadır. Çevik projelerde, projeleri bir şekil içine sokarak, sadece o şekle göre hareket etmek çeviklik anlayışı ile bağdaşmamaktadır.

Yazılım projelerinin hedeflerini kesin çizgilerle belirlemek zor olmaktadır ve takip edilecek genel adımlar bulunmasına rağmen, kesin adımlar bulunmamaktadır. Çıkan yeni bir teknoloji ile birlikte yazılım projesinin hedefi ve amacı değişebilmektedir. Genel hedefler proje başlangıcından sonuna kadar şekillenerek kesin hedeflere dönüşürler. Oluşan kesin hedeflerle çevik proje süreci tamamlanmış olur.

Çevik Yazılım Geliştirme Manifestosu Beck ve arkadaşları (2001) tarafından "bizler yazılım geliştirmenin daha iyi yollarını yazılım geliştirme sürecinde bulunarak ve

diğerlerinin yazılım geliştirme süreçlerine yardımcı olarak ortaya çıkarıyoruz" şeklinde 2001 yılında oluşturulmuştur.

Beck ve arkadaşları (2001) bu manifesto çalışması sonucunda şu değerlere ulaşmışlardır;

- Bireyler ve aralarındaki etkileşim, kullanılan araç ve süreçlerden önemlidir.
- Çalışan yazılım, kapsamlı dokümantasyondan önemlidir
- Müşteri ile işbirliği, kontrat şartlarından önemlidir.
- Değişime karşılık vermek, bir plana bağımlı kalmaktan önemlidir.

Beck ve ark.(2001) yukarıda yer alan maddelerin, sağ tarafındaki özelliklerin değerini kabul etmekle birlikte, başarılı projeler için sol taraftaki özellikleri daha değerli bulduklarını belirtmişlerdir.

Çevik Metodolojilerde çalışan yazılım ve birlikte çalışan insanların etkinliği en önemli iki konsepttir. Çalışan yazılım, müşterilerin önlerinde ne görmek istedikleri anlamına gelir (Highsmith ve Cockburn, 2001) ve modelin belirli bir türünün nasıl olacağı gibi detaylı prosedürleri tanımlamamaktadır (Ambler, 2002). Bundan dolayı yazılım geliştirici müşterinin ne istediğine odaklanarak yazılımı gerçekleştirmelidir. Yazılım geliştirme sürecinde müşteri tarafından istenmemiş ek özelliklere yazılım içine eklemeye çalışmamalıdır. Bu yazılım geliştiriciyi müşterinin esas ihtiyacı olan özelliklerden uzaklaştırabilmektedir. Örneğin müşteri bir sistem için web üzerinden kullanıcı ekleme özelliği isterse, ilk odaklanması gereken özellik yazılımın web üzerinden kullanıcı ekleme özelliğinin olmasıdır. Yazılım geliştirici bunu yapmak yerine belli bir format dosyadan kullanıcıları sisteme nasıl yükleme özelliğine odaklanmamalıdır.

Çevik metodolojiler, şirketlerin doğru ürün oluşturmasına yardımcı olmak için (Flora ve Chande, 2013) çevre ve gereksinimlerdeki değişikliklere uyum sağlamayı kolaylaştırmaktadır. Buna ek olarak süreç içindeki geliştirme tecrübelerinden yeni öğrenmeler sağlayarak yazılım geliştirme esnekliğini ve üretkenliğini artırmakta (Rahimian ve Ramsin, 2008) ve şirketlerin pazarda mümkün olduğunca rekabetçi olmasını sağlamaktadır (Flora ve Chande, 2013). Çevik metodolojileri kullanmak, ihtiyaç duyulan gereksinim ve teknolojideki değişikliklere daha hızlı yanıt verebilmeyi sağlayarak proje başarısını artırmaktadır. Bunun yanı sıra çevik projenin getirdiği esneklik ve farklılıklar nedeniyle daha çok yeni öğrenme ortaya çıkar. Bu yeni

öğrenmeleri firmalar en iyi şekilde kullanarak önemli Ar-Ge yatırımları yapabilir. Yaptığı yatırımlar sonrası rakipleri arasında gelecekte ön plana çıkacak bir firma olmanın ilk adımını atmış olur.

Firmalarda yeni yazılımlar geliştirilirken önceki yazılım geliştirme süreçlerinden elde edilen birikimler de önemli bir yer tutmaktadır. Bunun sonucunda firmalar yeni bir projeye başlarken önceki projelerdeki olumlu ve olumsuz yönleri göz önüne alarak hareket etmektedirler. Bu olumlu yönlerin en iyi şekilde proje sürecine uygulanmasını sağlayan yapı projelerin çeviklik ile sürdürülmesi ile mümkün olabilmektedir. Olumsuz yönlerin o projenin devam eden aşamalarında veya bir sonraki projede önüne geçilmesi yine çeviklik ile sağlanabilmektedir.

Beck ve ark. (2001) manifesto çalışması sonucu şu ilkelere ulaşmışlardır;

1. En önemli öncelik değerli yazılımın erken ve sürekli teslimatını sağlayarak müşteri memnuniyetini kazanmaktır.

Müşteri yazılım süreci sürerken kendine veya firmasına yarar sağlayacak ürünü son aşamada görmek yerine, ürünün yapılırken belirli aşamalarındaki halini görmek ister. Eğer bu gerçekleşirse müşteri memnuniyeti artmaktadır.

2. Değişen gereksinimler geliştirmenin son aşamasında olsa bile kabul edilebilir olmalıdır. Çevik süreçlerde müşterinin rekabet avantajı için bu değişim şarttır.

   Projenin son aşamasına gelinmiş olsa bile piyasadaki veya teknolojideki değişikliklerden dolayı, ürünün o şekilde piyasaya sürülmesi müşteri ihtiyaçlarını karşılamayacaksa, müşterinin istediği değişiklikler kabul edilebilir olmalıdır.

3. Kısa zaman aralıkları tercih edilerek sık sık çalışan yazılım müşteriye sunulmalıdır. Bu zaman aralıkları birkaç ay veya birkaç hafta şeklinde olabilir.

   Proje bittikten sonra yazılımın son şeklini müşteriye sunmak yerine, sık sık ürünün o ana kadar yapılmış çalışan kısmı müşteriye sunulmalıdır. Bu şekilde hem müşteriden geri bildirimler alınırken, hem de ürünün istenilen şekilde çalışıp çalışmadığı test edilebilmektedir. Eğer ürün bir aksaklıktan dolayı çalışmıyorsa erken aşamada tespit edilebilmektedir. Bu da yazılım geliştirme sürecinin daha sağlıklı işlemesini

sağlamaktadır. Seçilmesi gereken zaman aralığı projenin büyüklüğü ve projenin değişimlerden etkilenebilme özelliğine göre belirlenebilir.

4. İşin yapıldığı kişiler ve geliştiriciler proje süresince her gün birlikte çalışmak zorundadırlar.

Müşteri ile geliştiricinin sürekli iletişim halinde olması projenin ilk aşamasında anlaşılmayan müşteri isteklerinin geliştirme süreci içinde daha sağlıklı anlaşılmasını sağlayarak, müşterinin gerçekten istediği ürünün ortaya çıkmasını sağlamaktadır.

5. Proje motivasyonu tam sağlanmış bireyler etrafında oluşturulmalıdır. Onlara ihtiyaç duydukları ortam ve desteği vererek, işi tamamlamak için güvenmek gerekmektedir.

Proje sürecinde çalışan bir birey projeye ve yaptığı işe inanırsa, bu onun işi başarmak için motivasyonun artmasını sağlamaktadır. Bunu başarmak için proje sürecinde çalışan bireylerin ihtiyaç duydukları araçlar sağlanmalı ve yazılım geliştirme ortamı bu çerçevede oluşturulmalıdır. Örneğin yazılımcının çalıştığı bilgisayarın tepki verme süresi yazılım geliştiricinin motivasyonunu belirleyen bir unsur olmaktadır.

6. Geliştirme Ekibi içinde bilgi alışverişinde bulunmak için en verimli ve etkili yöntem yüz yüze iletişimdir.

Geliştirme ekibi bir projenin parçalarını yaparak bir ürün ortaya çıkardıkları için birbirleriyle olan iletişimleri son derece önemlidir. Bunu sağlamak için aynı projede çalışan geliştirme ekiplerinin bir arada bulunması ve proje süresince sürekli iletişim halinde olmaları önemlidir.

7. Çalışan yazılım ilerlemenin temel ölçüsüdür.

Proje süresince farklı yazılımcıların kodladığı yazılım parçalarını birleştirilmeden ve oluşan yazılımı tam olarak hiç çalıştırmadan devam eden bir süreç yazılım geliştirme açısından sağlıklı olmamaktadır. Bu şekilde geliştirilen yazılımlar da son aşamada aksaklıklar çıkabilmekte, bu da projenin başarılı şekilde tamamlanmasının önüne geçmektedir. Bu sorunu aşmak için oluşturulan yazılım parçaları belirli bir plan içinde birleştirilerek sürekli yazılımın çalışır halde olması sağlanmalıdır. Bu şekilde çalışan

yazılım ile proje sürecinin ilerlemesi projenin başarılı olacağının bir göstergesi olmaktadır.

8. Çevik süreçler sürdürülebilir geliştirmeyi destekler. Bunun için proje sürecinde yer alan tüm paydaşlar (sponsorlar, geliştiriciler ve kullanıcılar) proje süresince sabit tempoyu korumalıdır.

Örneğin; müşterinin sürekli geliştirme ekibi ile ortak çalışması gerekmektedir. Eğer müşteri projenin belirli aşamalarında geliştirme ekibi ile çalışmıyorsa ve geliştirme ekibi de bundan dolayı ürünün ilerlemesi için gerekli olan cevapları alamıyorsa, bu proje sürecini olumsuz şekilde etkileyebilmektedir.

9. Teknik mükemmellik ve tasarımın iyi yapılmasına sürekli dikkat edilmesi çevikliği artırmaktadır.

Proje süresince her aşamada tasarımın en iyi şekilde yapılması ve yazılım geliştirme sürecinde yazılım geliştirme kurallarına bağlı kalarak yazılımın geliştirilmesi ile çeviklik artmaktadır. Örneğin; yazılımsal olarak yeniden kullanılabilir yapıların oluşturulması teknik mükemmelliği arttırmaktadır. Ya da yazılımsal tasarım kalıplarının kullanılması proje sürecinde kod yönetimini kolaylaştırmaktadır.

10. Sadelik esastır. Bu da yapılması gerekli olmayan işleri proje sürecinden çıkarmakla sağlanabilmektedir.

Yazılım projelerinde müşterilerin istedikleri bazı özellikler kullanıcılar tarafından hiç kullanılmamaktadır. Bunun için müşteri istekleri iyi analiz edilerek kullanıcıların hiç kullanmayacağı özellikler proje sürecinden çıkarılmalıdır. Bu şekilde geliştirme ekibinin projede önemli olan noktalar üzerine odaklanması sağlanarak, iş saatlerinin en yüksek verimlilikte kullanılarak, müşterinin istediği ürünün daha hızlı teslim edilmesini sağlanabilmektedir.

11. En iyi mimariler, gereksinimler ve tasarımlar kendi kendine organize olan ekiplerden ortaya çıkmaktadır.

Bir ekip yukarıdan yönetilmek yerine, kendi kendine organize olarak kendi ekiplerini yönetebilirse daha başarılı işler ortaya çıkmaktadır. Örneğin; kendi kendine organize olan

ekiplerde mimariyi tartışarak belirledikleri için, daha iyi yazılım mimarileri ortaya çıkabilmektedir. Yukardan yönetimlerde yöneticilerin isteklerini gerçekleştirmek birinci öncelik olduğu için bunu başarmak zor olmaktadır.

12. Düzenli aralıklarla ekip nasıl daha etkili olabilmeyi tartışmakta ve buna göre davranışlarını ayarlamakta ve yargılamaktadır.

Ekip bazı zamanlarda kendi içindeki iletişimi ve proje sürecinde nasıl hareket ettiklerini daha net görebilmek için kendi içlerinde bir durum değerlendirmesi yapmaktadır. Bu değerlendirmede iş akışını ve süreci yavaşlatan noktalar tespit edilebilir ve daha sağlıklı ilerlemek için yapılması gerekenlere karar verilebilir. Veya ekibin performansını artıran noktalar tespit edilerek bunların daha sık yapılması sağlanabilir.

Yukarıda belirtilenleri gerçekleştirebilmek için yazılım geliştirme şirketlerinde, projeler gerçekleştirilirken birçok değişik yöntem kullanılmaktadır. Bu yöntemler kullanılırken kolaylıklar tespit edilmekte veya hatalarla karşılaşılmaktadır. Bu kolaylıklardan ve hatalardan tecrübeler oluşmaktadır. Bu yeni tecrübelere göre firmalar proje yönetiminin esnekliğinden faydalanarak projelere yeniden şekil verebilmekte ya da başlayacakları yeni projelere bu tecrübelerden yararlanarak başlamaktadırlar. Rahiman ve Ramsin'nin (2008) belirttiği gibi mobil uygulama geliştirme alanı yeni bir alan olduğu için, mobil uygulama projeleri geliştirilirken bu öğrenmelerin sayısı çok daha fazla olmaktadır. Projenin başarılı şekilde tamamlanabilmesi için bu yeni öğrenmelere göre projenin yeniden şekillendirilmesi önem arz etmektedir. Proje süreçlerinde farklı stratejilerin uygulanılmaya başlandığı günümüz iş yaşamında, bu stratejileri oluştururken göz önüne almamız gereken birçok değişken bulunmaktadır. Proje süreçlerinde önemli kavramlardan birisi de Beck ve ark.(2001) belirttiği gibi bireyler ve etkileşim kavramıdır. Bu bireylerden oluşan ekipler projemizin başarısını etkilemektedir. Bu ekiplerin müşteri ile işbirliği sonucunda değişen teknoloji şartlarına ve müşterinin değişen ihtiyaçlarına cevap veren çalışan yazılımlar ortaya çıkmaktadır. Bu değişkenleri en iyi şekilde bir arada tutarak yazılım geliştirme sürecine uygulamak için çeşitli çevik metodlar kullanılmaktadır.

Çevik yazılım geliştirme metodolojisi, aynı prensiple hareket eden farklı modelleri içermektedir. Bunlardan; "SCRUM", "Dinamik Sistem Geliştirme Metodu (*Dynamic*

Systems Development Method: DSDM)", "Kristal Method (*Crystal Methods*)", "Özelliğe Yönelik Geliştirme (*Feature-Driven Development: FDD*)", "Lean Yazılım Geliştirme (*Lean Software Development: LSD*)", "Uç Programlama (*Extreme Programming: XP*)" ve "Adaptif Yazılım Geliştirme Metodu (*Adaptive Software Development: ASD*)" yaygın olarak kullanılmaktadır (Highsmith, 2002).

2.4. MOBİL UYGULAMA GELİŞTİRME

Değişen yaşam şekilleri ile birlikte insanlar sürekli hareket halinde olmaktadır. Bu hareket halinde, sürekli ulaşılabilir olmak için veya birisine ulaşmak için mobil cihazlar taşımaya başlamışlardır. İlk zamanlarda sadece konuşma ve mesajlaşma alanındaki ihtiyaçları gideren mobil cihazlar, mobil akıllı cihazların çıkması ile birlikte farklı boyuta geçerek her alandaki ihtiyaçları karşılamak üzere kullanılmaya başlanmıştır. Bu da mobil cihaz kullanım sayısını hızlı şekilde arttırmıştır. Mobil cihaz kullanımının artması ile birlikte mobil uygulama kullanım konusunda da artış olmuş ve mobil uygulamaların gerekliliği artmıştır.

Mobil uygulama kullanımının artması ile birlikte, mobil uygulamalar her alanda kullanılmaya başlanmış, bununla birlikte her alanda mobil cihazlar üzerinde çalışan mobil uygulamalara ihtiyaç duyulmaya başlanmıştır. Bireysel olarak ve şirket bazında mobil uygulama geliştirme çalışmaları ile her alanda kullanabileceğimiz birçok mobil uygulama ortaya çıkmıştır.

Her mobil cihaz, uygulama geliştirme sırasında dikkate alınması gereken özgünlükler ve kısıtlamalar ile birlikte gelmektedir (Scharff ve Verma, 2010). Mobil uygulama geliştirirken mobil cihazın kaynaklarını en iyi şekilde kullanacak uygulamayı hazırlamak önemli olmaktadır.

Mobil cihazların cihaz boyutu, ekran boyutu ve veri giriş mekanizması için fiziksel özelliklerinin olmasının yanı sıra işlem gücü, bellek alanı ve işletim sistemi gibi teknik karakterleri de vardır (Scharff ve Verma, 2010). Bundan dolayı bir müşteri mobil uygulama talep ettiğinde, ihtiyaç analizinde mobil cihazlar için fiziksel daraltma analizi yapılmalıdır. Bu teknik risklerin hafifletilmesinde önemli rol oynamaktadır (Kaleel ve Harinshankar, 2013).

Mobil uygulama geliştirirken mobil cihazların farklı donanımsal özelliklere sahip olmasının yanı sıra bu donanımlar üzerinde çalışan Android, IOS, Windows Phone, Blackberry gibi farklı mobil platformlar olduğuna dikkat edilmelidir. Farklı donanımsal özelliklere dikkat ederek ürün geliştirmeye başlarken uygulamanın çalışacağı cihazın modelini bilmek önem kazanmaktadır. Çünkü o cihazın özelliklerini en iyi şekilde kullanacak uygulamayı hazırlamak kaynak kullanımı açısından büyük önem arz etmektedir.

Mobil ortamlar için bir yazılım sisteminin başa çıkmak zorunda olduğu sayısız zorluk bulunmaktadır (Forman ve Zahorjan, 1994; Hayes, 2002; Rahimian ve Ramsin, 2008);

- Kablosuz iletişim sorunları: Kullanılabilirliği, bağlantı kopması, bant genişliği, heterojen ağlar, güvenlik riskleri gibi hususlar.
- Hareketlilik sorunları: Adres değişimi gibi konular ve lokasyon bağımlı bilgi yönetimi gibi hususlar.
- Güvenlik Açıkları: Çeşitli standartlar, protokoller ve ağ teknolojilerinden kaynaklanabilen durumlar.
- Terminal cihazların sınırlı kapasite özellikleri: Düşük güç ile ilgili faktörler, veri bütünlüğü ile ilgili riskler, küçük ölçekli kullanıcı arabirimleri ve düşük depolama kapasitelerinin olması.
- Özel gizlilik ve özelleştirilebilir ihtiyaç durumları ortaya çıkması.
- Katı zaman ve pazar gereksinimlerinin olması.

Yukarıda açıklanan maddeler halindeki sorunları ele alırsak yazılımcıların bir mobil uygulama geliştirirken bu yazılımın kullanılacağı cihazın donanımsal ve yazılımsal özelliklerine dikkat ederek, cihaz üzerinde en performanslı şekilde çalışacak uygulamayı geliştirmeyi amaç edinmeleri gerekmektedir.

Yazılımcılar çok kaliteli bir uygulama geliştirebilir ama bu uygulama mobil cihazın özellikleri ve kaynaklarındaki kısıtlar nedeniyle mobil cihaz üzerinde tam performansla çalışmaz ise süreç boyunca en iyi şekilde geliştirilen yazılımsal özellikler kullanılamaz konuma düşebilmektedir. Örneğin; bir yazılım birimi içinde o birimde kurulu olan

kablosuz ağ teknolojisine göre çok güçlü bir veri iletişimine sahip olan bir uygulama geliştirilebilir. Ama bu uygulama gerçek hayata çıktığında mobil sistemler üzerinde kullanılan internet bant genişliğinden dolayı kullanılamaz konuma düşebilir.

Mobil uygulamaları güçlü yapan yönlerden birisi internet erişimi sayesinde uygulamaların daha etkin şekilde kullanılabilmesidir. Bazı uygulamalar, mobil cihaza yüklendikten sonra sürekli internet erişimine ihtiyaç duyarken, bazı uygulamalar sadece kurulum anında internet erişimine ihtiyaç duymaktadırlar. Sürekli internet erişimine ihtiyaç duyulan uygulamalar tasarlanırken mobil internet sağlayan şirketlerin internet sağlama özellikleri dikkate alınmalıdır. İnternet kesilmesi gibi durumlarda karşılaşılabilecek sorunlar uygulama tasarlama aşamasında göz önüne alınmalıdır.

Mobil uygulama geliştirme konusunda uygulamanın güvenliği de çok önemli bir noktayı oluşturmaktadır. Tüm işlemlerin konum bağımsız yapılmaya başladığı günümüzde, birçok işlemde mobil cihazlar üzerinden gerçekleştirilmektedir. İşlemlerin gizlilikle sürdürülmesi gerektiği ve kullanıcı bilgilerinin önemli olduğu uygulamalar geliştirilirken güvenlik konusu üzerinde durulmalıdır. Güvenlik konusunda yetersiz uygulamalar tüm özellikleri karşılasa bile güvenlik açıkları nedeniyle kullanılamaz konuma düşebilmektedir.

Bu özelliklerin bilgisayar tabanlı sistemlerde karşılanması mobil tabanlı sistemlerden daha kolay olmaktadır.

Mobil uygulamaların tasarımı, değişen ihtiyaç ve beklentilerle birlikte (Flora ve Chande, 2013), yazılım geliştirme projelerinde görünenden çok daha karmaşıktır, bu durum geliştiricileri geleneksel yazılım geliştirme metodolojilerini tekrar gözden geçirmeye zorlamaktadır (Rahimian ve Ramsin,2008). Bunun nedeni mobil sistemlerdeki bilinmezliklerden ve her an değişebilen özelliklerden dolayı yazılım geliştirme metodlarının mobil uygulama geliştirmeye göre tekrar gözden geçirilmesi zorunluluk haline gelmektedir.

Mobil uygulama geliştirmede mobil etkileşim tasarımı kullanıcı ihtiyacı tabanlı bir kullanıcı deneyimi oluşturmak için oldukça önemlidir ve mobil uygulamaların benimsenmesi ile doğrudan ilgilidir (Scharff ve Verma, 2010).

Günümüzde mobil cihazların hızla artması ile birlikte gereksinimler ortaya çıkmaktadır. Bu gereksinimler sonucu oluşan müşteri ihtiyaçlarına göre tüketicinin görüşünü yansıtan ve ivedilikle piyasaya sürülen ürünlerle, mobil uygulama şirketleri varlıklarını devam ettirebilmektedir (Jeong ve diğ, 2008). Bunun için müşteri bir mobil uygulama talep ettiğinde pazar analizini içeren ihtiyaç analizi yapılmalıdır (Kaleel ve Harinshankar, 2013). Her gün binlerce uygulamanın ortaya çıktığı pazarda ihtiyaç tespit edilerek geliştirilmeye başlayan uygulama bir başka kişi ya da kurum tarafından hazırlanarak hayata geçirilebilir. Uygulama başka firma tarafından hayata geçirildikten sonra bizim uygulamamız kullanılamaz konuma düşebilmektedir. Bu da uygulama için harcanan zaman ve paranın boşa gitmesi anlamına gelmektedir.

Bu bilgiler ışığında mobil uygulama geliştirme alanında ihtiyacı tespit ederek uygulama geliştirmeye başlamak önemli hususlardan birisidir. Bu ihtiyaçları en doğru şekilde tespit edebilmek için müşterinin ne istediğini tam olarak anlamak çok önemlidir. Hızlı gelişen ve büyüyen yazılım dünyasında ihtiyaçları tam olarak tespit edebilmenin bazı zorlukları vardır. Proje başında tespit edilen ihtiyaçlar değişen ve gelişen teknoloji ile birlikte proje sonunda değerini yitirebilmektedir ve ihtiyaç olmaktan çıkmakta hatta gereksiz konuma düşebilmektedir. Bundan dolayı müşterinin veya kullanıcıların ihtiyaçları tespit edildikten sonra müşterinin ihtiyaçlarının ürünün her aşamasında değişebileceği göz önüne alınarak sürekli müşteri ile irtibat halinde bulunmak (Beck ve ark., 2001) önem arz eden diğer bir konu olmaktadır.

Beck ve arkadaşlarının (2001) belirtiği gibi müşteri ile irtibat çok önemlidir. Mobil uygulama geliştirme alanında dikkat edilmesi gereken noktalardan birisi de müşterinin ihtiyaçları tespit edildikten sonra, müşteri ile sürekli irtibat halinde olarak ürünü en hızlı şekilde geliştirmektir. Müşteri ile sürekli irtibat halinde olunduğunda, müşterinin gerçek ihtiyaçları ürün geliştirme sürecinde daha net ortaya çıkabilmekte veya bazı ihtiyaç gibi görünen ama aslında ihtiyaç olmayan özellikler tespit edilebilmektedir. Bu şekilde ürün geliştirme süreci daha sağlıklı ilerleyebilmektedir. Ürünü en hızlı geliştirmenin önemi ise uygulama geliştirme sürecinin uzun sürmesinden kaynaklanan kayıpları en aza indirebilmektir. Uygulama geliştirme aşamasının uzun sürmesi sonucu ürün piyasaya geç çıkabilmektedir. Ürünün piyasaya geç çıkması dolayısıyla, ilk ihtiyaç olan ürün, ihtiyaç olma özelliğini kaybedebilmektedir.

Müşteri ihtiyaçlarının tespit edilmesi, değişen müşteri ihtiyaçlarının proje sürecine uygulanması, değişen teknolojilere uyum sağlanması, proje süresince ekibi sürekli etkin tutma kavramlarını ele aldığımızda mobil uygulama geliştirme alanında bu süreçleri gerçekleştirmeyi kolaylaştırmak için Mobil-D, MASAM (*Mobile Application SW Development Method - Mobil Uygulama Geliştirme Metodu*), Hibrit Metodoloji(*Hybrid Methodology*), Scrum ve Scrum Lean Six Sigma gibi metodlar bulunmaktadır (Corral ve diğ, 2013).

Tez kapsamında Scrum Çevik Geliştirme Metodu kullanılacağı için ayrıntılı olarak ana başlık altında anlatılmıştır.

2.5. SCRUM YAZILIM GELİŞTİRME METODU

2.5.1. Scrum'a Genel Bakış

Schwaber ve Sutherland tarafından tanıtılan ve hala desteklenen Scrum 1990'ların başlarından itibaren karmaşık ürün geliştirmeyi yönetmek için bir süreç çerçevesi sunmaktadır. Scrum hala Schwaber ve Sutherland tarafından desteklenmekte, Beedle'nın işbirliği ile geliştirilmektedir (Highsmith, 2002, Schwaber ve Sutherland, 2013).

Küresel mobil servis sektöründe girişimci fırsatları ortaya çıkarmak ve yeni bir alan olan mobil programlamayı tanıtma amacı taşıyan "Mobil Uygulama Geliştirme ve Girişimcilik" dersi kapsamında Scrum'un mobil sistemlere uygulanması ilk defa Scharff ve Verma (2010) tarafından bir sınıf ortamında gerçekleştirilmiş ve tanıtılmıştır. Bu şekilde Scrum kullanılarak gerçek bir müşteri için yenilikçi bir mobil uygulamanın tasarlanması ve gerçekleştirilmesinde derste kazanılan son teknik bilgiler kullanılmıştır (Scharff ve Verma, 2010).

Tekrarlamalı ve artırımlı koşuları ile karakterize edilen Scrum en biline çevik metodolojilerden birisidir (Kaleel ve Harinshankar, 2013). Yazılım geliştirme sürecini kontrol etmek ve yönetmek için kullanılan deneysel bir yönetim tekniğidir (Nicolas, 2007, Kaleel ve Harinshankar, 2013). Scrum müşteri gereksinimlerini karşılamak için (Nicolas, 2007), geliştirme faaliyetleri – gereksinimleri toplama, tasarlama, programlama – ve gerçekleştirme aşamalarından oluşan bir proje yönetim çerçevesidir. Scrum'da geliştirme süresi "Koşu" olarak adlandırılan 7-30 günlük yinelemelerden oluşan

periyotlar ile çalışan ve kaliteli yazılım ortaya çıkaracak şekildedir (Beedle ve ark., 1999, Highsmith, 2002).

Scrum'un merkezinde geliştirme sürecinde ortaya çıkacak süreçlerin tahmin edilemez olduğu fikri yer alır. Bu nedenle, esnek bir şekilde yazılım geliştirmenin nasıl olabileceği üzerine yoğunlaşmakta (Vlaanderen ve diğ, 2011) ve geliştirme aşaması için herhangi bir özel yazılım geliştirme tekniği tanımlamamaktadır (Abrahamsson, 2002). Scrum anlaşılması kolay, karmaşık olmayan ama uzmanlaşması zor bir süreçtir. Bu süreçte öngörülebilirliği arttırmak ve riski kontrol etmek için yinelemeli ve artırımlı bir yaklaşım kullanılır (Schwaber ve Sutherland, 2013). Artırımlı bir yaklaşım kullanarak ürün aşama aşama inşa edilirken bir sonraki ürün parçası için öngörülebilirliğin artmasına bağlı olarak süreç içinde risklerin kontrol edilebilirliği de artmaktadır. Örneğin; yüz tanıma ile bir işlem yapacak mobil uygulama geliştirme aşamasında, ilk olarak yüz tanıma özelliği sistem üzerinde gerçekleştirilebilir. Daha sonraki aşamada detay işlemlerin nasıl gerçekleştirileceği artırımlı bir şekilde sisteme eklenebilir.

Scrum, yinelemeli ve artırımlı süreci başarı ile uygulamak için Şeffaflık, Denetim ve Adaptasyon olmak üzere üç temel prensibi baz almaktadır (Schwaber ve Sutherland, 2013).

1. Şeffaflık: Scrum sürecindeki önemli noktalar, ürün çıktıları ile sorumlu olan herkes tarafından görülebilir olmalıdır (Schwaber ve Sutherland, 2013). Bu şekilde ekip içindeki herkesin ne yaptığı görüldüğü için ürün başarısı artmaktadır. Örneğin; bir mobil market ödeme sisteminin gerçekleştirilmesi sürecinde o marketten proje için seçilmiş kişilerin uygulama geliştirme aşamasının önemli noktaları görmeleri sağlanmalıdır.

Şeffaflık ilkesini uygulayabilmek için aşağıdakiler yapılmalıdır (Schwaber ve Sutherland, 2013);

- Ekip içinde herkesin süreci en iyi şekilde anlaşılır kılan ortak bir dil kullanması gerekmektedir.

Bu en önemli noktalardan birisidir. Eğer kullanılan dil ekip içinde herkes tarafından anlaşılmıyorsa bu proje sürecinin başarı ile bitirilmesini engelleyen bir durum

oluşturabilmektedir. Örneğin; proje sürecinde teknik olarak çalışan kişilerin Ürün Sahibine karşı sürekli teknik anlatımlar gerçekleştirmesi istenilen bir durum olmamaktadır

- Bir işin tamamlandığını gösteren "Tamamlandı" kriteri için iş yapan ekip ile işi yapılan kişinin aynı kriteri paylaşması gerekmektedir.

Belirlenen "Tamamlandı" kriterinin neye karşılık geldiği iyi tespit edilmelidir. Örneğin; bir mobil uygulama geliştirme sürecinde kullanılan "Tamamlandı" kriteri uygulamanın "pazarda yayınlanmaya hazır olduğu" yada "müşteri testleri için hazır olduğu" anlamlarından hangisine karşılık geldiği iyi tespit edilmelidir.

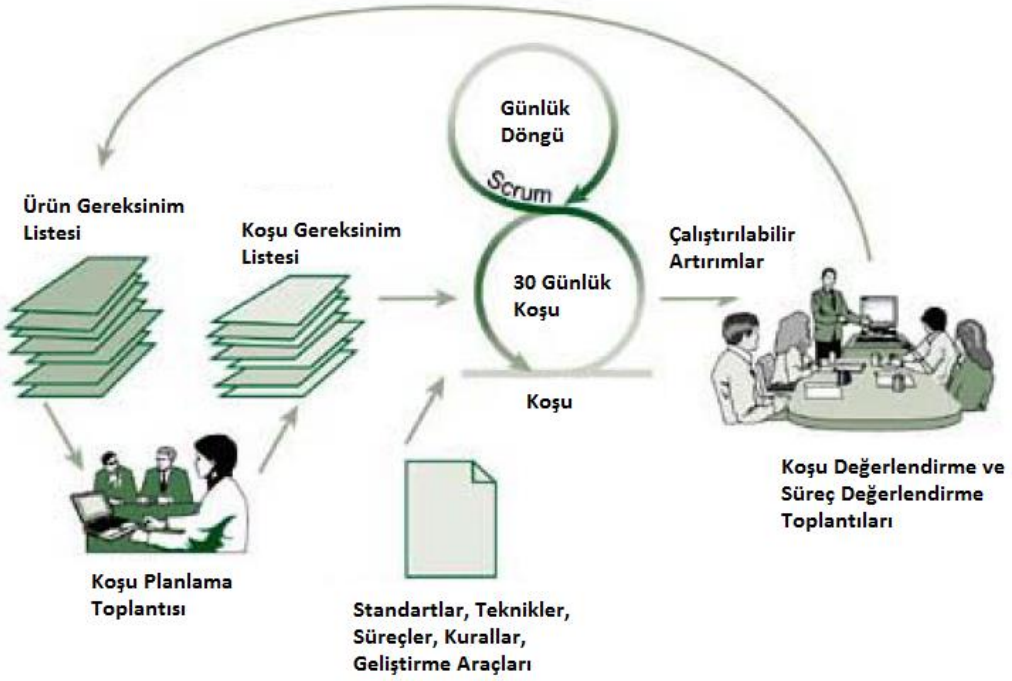
2. Denetleme: Scrum kullanıcıları istenmeyen saptamaları tespit etmek amacıyla çıktılarının ve sürecin Koşu Hedefine doğru ilerleyip ilerlemediğini sık sık kontrol etmelidir. Bu kontroller çalışmanın belirli noktalarında tecrübeli kişiler tarafından yapılarak sürecin başarısına yarar sağlamaktadır (Schwaber ve Sutherland, 2013). Scrum sürecinde bir yönetici olmadığı için ekip işi kendi kendine yönetmektedir. Bu kontroller sürecin başarısı için önemli olmaktadır.
3. Adaptasyon: Bir denetçi, kabul edilebilir limitlerin dışında bir sapma olduğunu ve bundan dolayı elde edilen ürünün kabul edilemez olacağını tespit ederse süreç veya sürecin malzemelerinin düzenlenmesi gerekir (Schwaber ve Sutherland, 2013).

Scrum, Denetim ve Adaptasyonun başarı ile uygulanması için Koşu Planlama Toplantısı, Günlük Scrum Toplantıları, Koşu Değerlendirme Toplantısı ve Koşu Süreç Değerlendirme Toplantısının yapılmasını zorunlu kılmaktadır (Schwaber ve Sutherland, 2013). Bu toplantılar ileriki aşamada Scrum Sürecindeki Toplantılar başlığı altında detaylı olarak açıklanmıştır.

Scrumla ilgili Türkçe literatür sayısının azlığı nedeni ile Türkçeleştirilen Scrum terimlerinin İngilizce karşılıkları tezin Ekler kısmında Scrum İngilizce Terimler Sözlüğü adı altında verilmektedir.

2.5.2. Scrum'daki Kavramlar

Scrum süreci Şekil 2.2'de görüldüğü gibi belirli bir düzen ile kendi içinde Koşu, Ürün Gereksinim Listesi, Koşu Gereksinim Listesi, Koşu İş Takip Grafiği, Koşu Hedefi ve Kullanıcı Hikayeleri gibi birçok kavram barındırmaktadır.



Şekil 2.2: Scrum Süreci (Highsmith, 2002).

2.5.2.1. Koşu

Scrum'da Koşu olarak adlandırılan 1 haftadan 4 haftaya kadar sabit süreleri olan yinelemeler kullanılmaktadır (Scharff ve Verma, 2010).

Koşu aşaması, ekip üyelerinin görevleri üzerine aldığı, Koşu Hedefini başarmak için herkesin sıkı çalıştığı ve Günlük Scrum Toplantılarına katıldığı basit kurallar içermektedir. Koşu sırasında ayrıntılı planlar bulunmamaktadır, ekip üyelerinden beklenen, sonuçları gerçekleştirmek için yeteneklerini kullanmalarıdır. Scrum'u güçlü yapan, zamanı parçalara indirgemesi ve her Koşu için özelliklere "yoğunlaşması" ile her Koşuda önceliklerin değişiyor olmasıdır. Olumsuz durumlar dışında Proje Sahibi, Scrum Uzmanı veya Yönetimden herhangi birisi tarafından öncelikler Koşu sırasında değiştirilememektedir. Koşu sonunda Ürün Sahibi tüm geliştirilmiş özellikleri kabul etmeyi ya da projeye yeniden yön vermeyi seçebilmektedir, fakat bu durumda öncelikler

sabit kalmakta ve değişiklikler yeni bir Koşu içinde gerçekleşmektedir (Highsmith, 2002).

Bir aydan fazla olmamak şartıyla her Koşu bir amacı gerçekleştirmek için proje olarak kabul edilir (Schwaber ve Sutherland, 2013) ve Koşu sonunda fonksiyonel bir ürün teslim edilir (Corral ve diğ, 2013). Her Koşu sonuç olarak üretilen bir ürün veya işi oluşturmak için, bir tasarım ve esnek bir planla, ne inşa edildiğine dair tanım içermektedir. Koşular riski maksimum bir ay ile sınırlandırmaktadır (Schwaber ve Sutherland, 2013).

Her Koşunun bir proje gibi kabul edilmesi Geliştirme Ekibini sürekli aktif tutarak aşama aşama ürün inşa edilme sürecinde iş zamanı kaybını minimuma düşürmektedir. Bu da ekip içinde projeye odaklanarak yapılan iş birimini arttırmaktadır. Her Koşunun zamana göre görülebilen esnek bir planının olması, Geliştirme Ekibindeki bireylerin ne yaptıklarını görmelerine imkan vererek kendi öz örgütlenmelerini sağlamalarına yardımcı olmaktadır. Her Koşu süreci en fazla bir ay süresini kapsadığı için değişiklikler sonucu sadece bir Koşu süreci etkileneceği için firmanın bu süreçte zararı minimum olmaktadır. Her Koşu planı sadece o Koşuya yönelik hazırlandığı için, Geliştirme Ekibinden birilerinin çıkması veya Geliştirme Ekibine birilerinin katılması sadece bir Koşu süreci planını etkilemektedir. Diğer Koşu süreçleri ayrılan ekip personeline veya dahil olan ekip personeline göre yapılabilmektedir.

Koşu sırasında aşağıdakilere dikkat edilmelidir (Schwaber ve Sutherland, 2013);

- Koşu hedefini tehlikeye atacak hiçbir değişiklik yapılamaz.
- Kalite hedeflerini düşürmek kabul edilemez.
- Gerekirse Geliştirme Ekibinin kapsamı daha iyi anlaması için Ürün Sahibi ile müzakere edilmelidir ve kapsam daha anlaşılır hale getirilmelidir.

Örneğin; bir Koşu süreci başladığında o Koşunun Hedefi Koşu sonunda mobil uygulamaya yüz tanıma özelliğinin eklenmesi ise, bu Koşu Hedefini tehlikeye atacak hiçbir değişiklik yapılamaz. Bu özellik mobil uygulamaya eklenirken iş beklenenden uzun sürse bile kalite hedeflerinden ödün verilmemelidir. Geliştirme sürecinde Geliştirme Ekibinin Koşu Gereksinim Listesini gerçekleştirirken bazı ayrıntılandırılması gereken noktalar ortaya çıkarsa, bunları Ürün Sahibi ile tartışılarak yapılmalıdır.

Koşu sırasında Geliştirme Ekibi bir işin beklenenden çok büyük veya çok küçük olduğunu belirlerse, Ürün Sahibi ile Koşu Gereksinim Listesi tekrar müzakere edebilir (Schwaber ve Sutherland, 2013). Örneğin; Koşu Planlama Toplantısında dosya aktarım sisteminin kodlanması kısmında Geliştirme Ekibi bu işin, kütüphane kullanarak 4 birimlik bir iş olduğunu tahmin etmiş olsun. Ama geliştirme aşamasında farklı sistemler arasında dosya aktarımı yaparken bu kütüphanenin işe yaramadığını ve kütüphanenin ekip tarafından tekrar kodlanması gereken bir iş olduğu fark edilsin. İstenmeyen bir durum da olsa, bu noktada Ürün Sahibi ile müzakere edilerek Koşu Gereksinim Listesinde bazı değişikliklere gidilebilmektedir.

2.5.2.2. Ürün Gereksinim Listesi

Müşteri bir ihtiyaç ve bu ihtiyacı ifade eden bir hikaye ile gelmektedir. Bu hikaye ile ürün oluşturma süreci başlar. Ürün oluşturma sürecinde ilk ifade edilen gereksinimlerin ve ihtiyaçların tam olarak açık olmadığı ortaya çıkar. Bununla birlikte her zaman iyi fikirler proje süresince ortaya çıkmaktadır. Tüm bunları dikkate alarak proje süresince oluşturulan ürünün sürekli değişim içinde olduğu söylenebilir. Bunu dikkate alarak Scrum sürecinde Ürün Gereksinim Listesi kavramı üzerinde durulmuştur.

Ürün Gereksinim Listesi, gereksinimlerin bir listesidir (Schwaber, 2004) ve Scrum Metodunun merkezinde yer almaktadır (Vlaanderen ve diğ, 2011). Ürün Gereksinim Listesi, temel olarak gereksinimlerin, hikayelerin ve özelliklerin bir listesidir (Kniberg, 2007) ve üründe yapılacak herhangi bir değişiklik için gereksinimlerin kaynağıdır. Ürün Sahibi Ürün Gereksinim Listesinin kapsamından, ulaşılabilirliğinden ve önem sırasına göre sıralanmasından sorumludur (Schwaber ve Sutherland, 2013). Müşteri bir ihtiyacını belirtip, onu hikaye şeklinde anlattığında bu, ürünü geliştirmeye başlamak için yeterli bir başlangıç noktası değildir. Müşteri hikayesini anlattıktan sonra, müşterinin istekleri ve ihtiyaçları analiz edilerek müşterinin asıl gereksinimleri oluşturulur. Oluşturulan bu gereksinimler bir liste şekline dönüştürülür. Bu listedeki maddeler, ürün oluşturma sırasında ekibin doğru yolda ilerlemesini sağlayan referanslardır.

Ürün Gereksinim Listesi kapsamında, Ürün Sahibinin istekleri, Ürün Sahibinin anlayabileceği şekilde tarif edilir (Kniberg, 2007). Ürün Gereksinim Listesi hiçbir zaman tam olmayan ve ürün geliştirildikçe değişen bir yapıdadır. İlk geliştirme aşamasında bilinen ve en iyi anlaşılan gereksinimleri ortaya koyan bir listedir. Ayrıca Ürün

Gereksinim Listesi ürünün uygunluğuna, rekabete ve kullanışlılığa göre sürekli değişen dinamik bir değişim sürecindedir. Yani bir ürün var olduğu sürece, Ürün Gereksinim Listesi var olmaktadır (Schwaber ve Sutherland, 2013). Her ne kadar müşteri ihtiyaçlarını hikaye şeklinde anlatsa ve bu hikayeden bir liste oluşturulsa da müşterinin ihtiyaçları tam olarak bu listeye aktarılamaz. Çünkü müşteri, ihtiyaçlarını hikaye şeklinde anlatırken bazı önemli noktaları atlayabilir, bundan dolayı ilk aşamada sadece müşteri tarafından çok iyi ifade edilen gereksinimler üzerine odaklanılarak bu liste oluşturulmaktadır. Ürün Gereksinim Listesi müşterinin ilk anlatımında unuttuğu noktalara göre sürekli değişmekle birlikte pazar şartlarındaki değişime göre de güncellenmek zorundadır. Bu açıdan müşterinin zarara uğramaması için ürünün sunulacağı piyasadaki ani değişiklikler de Ürün Gereksinim Listesine yansıtılmak zorundadır. Ürün oluşturma aşamasında müşterinin ürünü sürekli test etmesi sağlanarak, kullanışlılık açısından ürünü kullanılmasını zorlaştırılan noktalar tespit edilerek Ürün Gereksinim Listesi buna göre güncellenmek zorundadır. Mobil uygulama dünyasında sürekli yeni konseptler ve yeni teknolojiler çıkmaktadır. Örneğin; bir uygulama geliştirme aşamasına başlarken mevcut mobil uygulama teknolojilerine göre başlanmış olsa bile, yeni bir teknoloji çıktıysa ve o teknoloji o ürünü ve müşterinin kârını etkileyecek durumdaysa Ürün Gereksinim Listesi bu değişikliklere göre güncellenmek zorundadır.

Ürün Gereksinim Listesi tüm özellikleri, fonksiyonları, gereksinimleri, geliştirmeleri ve düzeltmeleri içermektedir. Bir ürün kullanıldıkça, değer kazandıkça ve pazar geri bildirim sağladıkça, Ürün Gereksinim Listesi daha büyük ve kapsamlı bir liste şeklini almaktadır. Bu yüzden iş ihtiyaçları, piyasa şartları ve teknolojideki değişimler Ürün Gereksinim Listesinin değişmesine sebep olmaktadır. Bu açıdan Ürün Gereksinim Listesi yaşayan bir listedir (Schwaber ve Sutherland, 2013). Ürün Gereksinim Listesi ilk aşamada ürün için istenilen özellikler, bu özellikleri gerçekleştirmek için gerekli olan fonksiyonlar ve gereksinimler ve bunları gerçekleştirmek için gerekli olan geliştirmeleri içermektedir. Ürün geliştirme sürecinde oluşan değişiklikler ve geri bildirimlere göre yapılacak değişikliklerde bu listeye eklenmektedir. Bu değişikliklere göre ilk aşamada gözden kaçan noktalar, teknolojik değişiklikler ve geri bildirimler Ürün Gereksinim Listesi listesinin daha kapsamlı bir liste haline dönüşmesini sağlamaktadır. Sürekli değişikliklerden etkilenmesi ve sürekli güncellenmesi dolayısıyla Ürün Gereksinim Listesinin değişimi sürekli olarak devam etmektedir. Mobil uygulama dünyasında bu

değişimler her yeni çıkan cihazla birlikte daha hızlı gerçekleşmektedir. Örneğin; günümüzde mobil uygulamalar bir ara yüz üzerinde çalışabilecek şekilde tasarlanmaktadır. Birkaç yıl içinde Google Glass gibi teknolojilerin daha popüler olması sonucunda uygulamaların tekrar güncellenmesi gerekeceği gibi, o an devam eden projelerin Ürün İçerikleri güncellenerek proje sürecinin bir mobil ara yüz üzerinde çalışabilme özelliğinin olması ile birlikte Google Glass gibi teknolojiler üzerinde de çalışabiliyor olması hedeflenecektir.

Ürün Gereksinim Listesinin değişimi, Ürün Gereksinim Listesindeki maddeleri sıralama, tahmin yapma ve detayları ekleme şeklinde gerçekleşmektedir. Bu şekilde, Ürün Gereksinim Listesi maddelerinin değişimi, Ürün Sahibi ve Geliştirme Ekibinin işbirliği yaptığı sürekli devam eden bir süreç şeklindedir. Bu süreçte maddeler gözden geçirilir ve revize edilir. Scrum Ekibi nasıl ve ne zaman "Tamamlandı" kriterine uygun olacağına karar verir. Ürün Gereksinim Listesinin değişimi genellikle Geliştirme Ekibinin kapasitesinin 10%'nu tüketmektedir. Bununla birlikte Ürün Gereksinim Listesi maddeleri Ürün Sahibi tarafından veya Ürün Sahibinin takdiri ile herhangi bir zamanda güncellenebilmektedir (Schwaber ve Sutherland, 2013). Ürün ihtiyaçlarını en iyi bilen kişi Ürün Sahibidir. Ürün Sahibi sürekli piyasa şartlarını ve teknolojideki değişiklikleri takip ederek ürünün kendisine fayda getirip getirmeyeceğini analiz etmektedir. Piyasa şartlarındaki veya teknolojideki değişikliklere göre kendi ürünün de o piyasa şartlarına uyum sağlamasını istemektedir. Buna göre oluşturulan üründe değişiklikler yapılması kaçınılmaz olmaktadır. Bundan dolayı müşteri istediği bir zamanda Ürün Gereksinim Listesini güncelleyebilmektedir.

Dikkat edilmesi gereken bir başka nokta ise ürünü geliştiren ekibin teknoloji boyutuna daha çok hakim olmasıdır. Bundan dolayı ürünü geliştirme aşamasında Geliştirme Ekibi teknolojideki değişliklere karşı daha duyarlı olmakta ve oluşan değişiklikleri takip etmektedir. Eğer bu değişikliklerin oluşan ürünü etkileyeceği görürlerse Ürün Sahibi ile bu durumu paylaşılmalıdır. Ürün Sahibinin bu durumu değerlendirerek Ürün Gereksinim Listesinin güncellenmesi sağlanmalıdır. Yapılan değişikliklerle birlikte Ürün Gereksinim Listesindeki maddelerin önem sırası değişebilmektedir. Bunun yanı sıra Ürün Gereksinim Listesinde kapsamlı bir değişiklik olmasa bile Ürün Gereksinim Listesindeki maddelere detaylı maddeler eklemek şeklinde güncellemeler yapılabilmektedir. Bu değişikliklerden

sonra Scrum Ekibi, bu maddelerden hangilerinin ilk olarak gerçekleştirilebileceğini belirleyerek, ürün geliştirme süreci devam edilmektedir.

Ürün Gereksinim Listesindeki yüksek öncelikli maddeler düşük öncelikli olanlara göre daha net ve detaylıdır. Daha iyi tahminler daha fazla netlik ve detaya göre yapılırken, daha düşük öncelikli maddelerde daha az detaya göre yapılmaktadır (Schwaber ve Sutherland, 2013). Ürün Sahibi ilk aşamada anlatımını gerçekleştirirken kendisi için önemli olan noktalar üzerinde daha çok durmaktadır ve Ürün Sahibi için önemli olan maddeler daha net ve detaylı olmaktadır. Bundan dolayı daha net ve detaylı maddelerle ürün geliştirme süreci için daha net tahminler yapılmakta ve ekipteki herkesin tam olarak ne yapılacağını anlaması sağlanmaktadır.

Ürün Gereksinim Listesinin, Koşu Planlama Toplantısı öncesinde hazır olduğuna emin olunmalıdır (Kniberg, 2007). Ürün Gereksinim Listesindeki maddeler Koşu Planlama Toplantısı için seçilecek durumdaysa bu, Koşu içinde Geliştirme Ekibi tarafından tamamlanabilecek durumda olduğu anlamına gelmektedir. Ürün Sahibi Geliştirme Ekibinin içeriği anlamalarına yardımcı olacak şekilde etkileyebilir fakat işi yapacak olan Geliştirme Ekibi nihai tahmini yapar. Bundan dolayı Geliştirme Ekibi tüm tahminlerden sorumludur (Schwaber ve Sutherland, 2013).

2.5.2.3. Koşu Gereksinim Listesi

Her Koşu, Koşu Gereksinim Listesi olarak isimlendirilen bir dizi faaliyet içermektedir (Beedle ve ark., 1999). Koşu Gereksinim Listesi, Ürün Sahibi, Geliştirme Ekibi ve Scrum Uzmanını içeren Koşu Planlama Toplantısında oluşturulmaktadır. Bu toplantılarda Koşu boyunca yapılacak görevlerin oluşturulması sırasında Geliştirme Ekibi etkin rol oynayarak Koşu Gereksinim Listesinin oluşturulmasını sağlamalıdır (Highsmith, 2002). Koşu Gereksinim Listesi her Koşu için başlama noktasıdır (Abrahamsson, 2002). Bundan dolayı Koşu Gereksinim Listesi formatsal olarak, Koşu Planlama Toplantısı sonrasında, ilk Günlük Scrum Toplantısından önce hazırlanmış olmalıdır (Kniberg, 2007).

Koşu Gereksinim Listesi, Ürün Gereksinim Listesinin tam olarak belirlenmiş maddeleri ile beslenir (Vlaanderen ve diğ, 2011). Bununla birlikte Koşu Hedefinin gerçekleştirilmesi ve ürünün bir parçasının inşa edilmesi planıdır. Koşu Gereksinim Listesi, Geliştirme Ekibi tarafından sonraki ürün parçasında hangi işlevselliklerin olacağı

ve bu işlevsellikleri yayınlamak için "Tamamlandı" kriterine uyan hangi parçaların yapılması gerektiğini gösteren bir tahmindir. Koşu Gereksinim Listesi, Geliştirme Ekibinin "Koşu Hedefini" gerçekleştirmek için yapmaları gerekenleri tanımlayarak tüm işleri görünür olmasını sağlamaktadır (Schwaber ve Sutherland, 2013). Ürün Gereksinim Listesinin oluşturulmasından sonra belirli bir hedef doğrultusunda, Ürün Gereksinim Listesindeki maddelerden öncelikli olan veya Ürün Sahibinin öncelikli olarak yapılmasını istediği maddeler seçilir. Bu seçilen maddeler ile Koşu süresince ürünün bir parçası inşa edilir ve yayınlanır. Örneğin; bir görüntü işleme yardımı ile belirli işlemleri gerçekleştiren bir mobil uygulama gerçekleştiriliyorsa, o Koşu için seçilen maddeler kişinin yüzünü tanıma özelliğinin alt maddelerinden oluşabilir. Bu Geliştirme Ekibine ürünü yayınlamak için hangi işlevsellikleri gerçekleştirmeleri gerektiği konusunda yol gösterir. Bu şekilde Koşu Gereksinim Listesinde belirlenen ürünün bir parçası inşa edilmiş olur.

Koşu Gereksinim Listesi Günlük Scrum Toplantılarında anlaşılabilecek şekilde süreçte yapılacak değişiklik detaylarını barındıran bir plandır. Geliştirme Ekibi planı uygulamaya çalıştıkça ve Koşu Hedefini başarmak için ihtiyaç olan çalışmalar hakkında daha fazla bilgileri aldıkça, koşu boyunca Koşu Gereksinim Listesini değiştirir ve bu süreçte Koşu Gereksinim Listesi son şeklini alır (Schwaber ve Sutherland, 2013). Koşu Gereksinim Listesi, Geliştirme Ekibinin ihtiyaçları konusunda sorularına cevap verecek şekilde olmalıdır. Bu şekilde oluşan bir liste ile ekibin ne yapacaklarını sürekli görerek sürecin daha şeffaf olması sağlanmış olur. Ürün geliştirme süresince sürekli değişiklikler meydana geldiği için, Geliştirme Ekibi Koşu Hedefini gerçekleştirmek için ihtiyaç duyduğu anlarda Ürün Sahibinin onayı ile (Schwaber ve Sutherland, 2013) Koşu Gereksinim Listesini değiştirebilmektedir. Bu şekilde Koşu Gereksinim Listesinin sürekli güncel olması sağlanmış olur. Örneğin; bir mobil rezervasyon sisteminde bir Koşu Gereksinim Listesi belirlenmiş olsun, Geliştirme Ekibi süreç içinde tasarımsal olarak uygulamayı daha etkin hale getirecek bir yöntem bulursa, ürünü daha etkin hale getirmek için bunu nasıl uygulayabileceklerini tartışarak güncelleyebilir ve Koşu Gereksinim Listesinin son şekli ile gerçekleştirmeye devam edebilir. Bu süreç devam ederken Ürün Sahibi ile çalışılması veya Ürün Sahibinden onay alınması gerekmektedir.

Kural olarak Koşu sırasında Koşu Gereksinim Listesine ek olarak bir iş eklenemez (Beedle ve ark., 1999). Eğer kesinlikle yapılması gereken yeni bir işe ihtiyaç duyulursa,

Geliştirme Ekibi bunu Koşu Gereksinim Listesine eklemektedir. Bu çalışma gerçekleştirildiğinde veya tamamlandığında, tahmini kalan çalışma güncellenmektedir. Planın unsurlarından gereksiz görülenler olursa, gereksiz görülen unsurlar iptal edilmektedir. Koşu sırasında sadece Geliştirme Ekibi Koşu Gereksinim Listesini değiştirebilmektedir. Koşu Gereksinim Listesi Geliştirme Ekibinin Koşu sırasında gerçekleştirmeyi planladığı işin görünür ve gerçek zamanlı resmidir (Schwaber ve Sutherland, 2013). Örneğin; Geliştirme Ekibi servisten bir veriyi çekerek çalışmanın bir kısmını kodlama işini üzerlerine alır. Bu iş içinde 13 birimlik bir iş olduğunu öngörülebilir. Ama işi gerçekleştirme zamanı geldiğinde bu işi gerçekleştirmek için servis mantığının da değiştirilmesi gerektiğini görürlerse ve değişiklik yapmadan kodlama yapmanın mümkün olmadığını fark ederlerse, bu işlem Koşu Gereksinim Listesine eklenir ve bu iş için tahmini iş birim saati yazılarak Koşu Gereksinim Listesi güncellenir. Bundan dolayı, Koşu Gereksinim Listesi, proje planın gerçek zamanlı resmidir.

Koşu Gereksinim Listesi ekip üyelerinin Koşu boyunca uygun olacakları saatlerle birlikte isimlerini içermektedir. Bu süreçte Kullanıcı Hikayeleri, her görevi tamamlamak için kalan ve tahmin edilen süre belirlenecek şekilde görevler ve diğer görevler (hatalar ve sınırlamalar) içinde ayrıştırılmaktadır (Scharff ve Verma,2010). Oluşturulan görevler ekip üyelerinin isimlerine atanır (Vlaanderen ve diğ, 2011). Örneğin; bir mobil ödeme sisteminin gerçekleştirilmesi sırasında oluşturulan Koşu Gereksinim Listesinde kimin hangi görevi yapması gerektiği ve tahmini olarak bu görevi ne kadar sürede bitireceği yazılmaktadır.

Koşu Gereksinim Listesi, Koşu boyunca ekip üyeleri tarafından yapılacak işleri belirlemekte ve tanımlamaktadır. Bu, Ürün Gereksinim Listesinin bir alt kümesidir. Bir düzeyde, Koşu Gereksinim Listesi özellikleri tanımlarken, bir başka düzeyde bu özellikleri gerçekleştirmek için görevleri tanımlamaktadır (Highsmith, 2002). Ürün Gereksinim Listesi ürün için yapılacakların geniş bir kümesini ifade ederken, Koşu Gereksinim Listesi belli bir süreci kapsayan ve yapılması gereken görevleri ifade etmektedir. Bu nedenle Koşu Gereksinim Listesi, Ürün Sahibinin ihtiyaçlarını gerçekleştirmek için ekip üyelerinin süreç içinde ne yapması gerektiğini belirten bir görev listesi şeklinde ortaya çıkmaktadır

2.5.2.4. Koşu İş Takip Grafiği

Scrum Uzmanı, ekibin her görevin tamamlanması için verdiği saat ve gün taahhüdünü gösteren İş Takip Grafiğini ekibin ilerlemesi için bir ölçü olarak korumakta ve sürdürmektedir (Scharff ve Verma, 2010). Bu grafik süreç içinde takımın kendi kendine organize olması için önemli bir noktayı teşkil etmektedir. Geliştirme Ekibi ve Scrum Uzmanı bu grafiğe göre aksiyonlar alabilmektedir. Bu aksiyonlar, işin oluşturulan planın gerisinde kaldığı ve zamanında bitmeyecek gibi görünmesi durumunda ekstra çalışmalar yapılması şeklinde olabilmektedir. Yada düşük öncelikli Koşu Gereksinim Listesi maddelerinden bazılarını Koşu Gereksinim Listesinden çıkarmak şeklinde olabilmektedir (Kniberg, 2007).

2.5.2.5. Koşu Hedefi

Koşu Hedefi, Ürün Gereksinim Listesinin gerçekleştirilmesi için Koşu içinde uygulanacak olan ve Geliştirme Ekibine ürünü niçin geliştirdikleri konusunda rehberlik sağlayan bir amaçtır (Schwaber ve Sutherland, 2013). Bundan dolayı Koşu Hedefinin cevaplaması gereken temel soru "Niçin bu koşuyu gerçekleştiriyoruz" şeklindedir (Kniberg, 2007).

Koşu Hedefi olmadan ekip görevler üzerine aşırı odaklanabilmekte ve görevleri neden gerçekleştirdiklerinin amacından uzaklaşabilmektedir. Koşu Hedefini akılda tutmak, Koşu boyunca ortaya çıkabilecek durumlara Geliştirme Ekibinin uyum sağlaması konusunda yardımcı olmaktadır (Highsmith, 2002).

Koşu Hedefi ile Geliştirme Ekibi ayrıntılarda boğulmayarak, Koşunun asıl amacı üzerine yoğunlaşmaktadır. Örneğin; Elektronik Belge Yönetim Sistemi oluşturulan bir projede Koşu Hedefi olarak "Koşu sonunda Pdf dosyalar üzerine imza atılabilme özelliği" şeklinde bir Koşu Hedefi belirlenebilir. Bu şekilde ekibin Koşu Gereksinim Listesindeki görevleri yaparken asıl amaçlarının Koşu sonunda sistemin Pdf dosyaları üzerine imza atma özelliğinin eklenmiş olması olduğunu sürekli hatırda tutmaları sağlanmaktadır.

2.5.2.6. Kullanıcı Hikayeleri

Gereksinimler genellikle kullanıcı hikayeleri oluşturularak açıklanmaktadır. Kullanıcı Hikayeleri müşterinin anlatımı ile oluşturulan, Ürün Sahibinin ne istediğini (Scharff ve Verma, 2010) ve sonuçta görmek istediğini ifade eden (Kniberg, 2007) ürün ile

kullanıcının etkileşiminin basit bir açıklamasını içeren (Nicolas, 2007) bir liste şeklindedir. Kullanıcı Hikayeleri görevlerden oluşmamaktadır (Scharff ve Verma, 2010). Görevler Kullanıcı Hikayelerini gerçekleştirmek için gerekli olan alt işlemleri temsil etmektedir. Kullanıcı Hikayeleri çok büyük yada çok küçük olmamalıdır (Kniberg, 2007). Kullanıcı Hikayelerinin bir örneği "Kullanıcı ad, soyad, adres, telefon numarasını ve istediği abonelik türünü seçerek bir form doldurur" şeklinde olabilir (Nicolas, 2007). Kullanıcı Hikayeleri ürünün oluşturulması sırasında önemli noktalardan birisidir. Çünkü proje süresince bir çok ayrıntı Kullanıcı Hikayelerine göre belirlenmektedir. Bu nedenle ürünün ilk aşamasından son aşamasına kadar, Kullanıcı Hikayelerinin doğru şekilde oluşturulması önemlidir. Oluşturulan Kullanıcı Hikayeleri görevleri ifade etmemekte, Ürün Sahibinin süreç sonunda ne görmek istediğini ifade etmektedir. Örneğin; "mobil uygulamanın barkod okuma özelliğini gerçekleştirebiliyor olması" bir kullanıcı hikayesi olabilir. Bu Kullanıcı Hikayesinden süreç için çeşitli görevler oluşturulur. Bu görevler "Barkod okuma için uygulamanın servisle bağlantı kurması", "Barkodu okumak için kamerayı kullanacak kodlamanın yapılması" şeklinde olabilir.

2.5.3. Scrum Sürecindeki Roller

Tüm proje yönetim süreçlerinde ekip içinde bireylerin bir rolü vardır. Scrum süreci ekip içinde etkileşim ve iletişimi sağlamak için bir Scrum Ekibi oluşturulmasını öngörmektedir.

Oluşturulan Scrum Ekibinde; Ürün Sahibi, Scrum Uzmanı ve Geliştirme Ekibi olmak üzere 3 ana rol bulunmaktadır (Scharff ve Verma,2010).

Scrum Ekiplerinin kendi kendine organize olabilme ve çapraz fonksiyonel özellikleri olmalıdır. Kendi kendine organize olabilen ekipler takımı dışarıdan birilerinin yönetmesi yerine işi başarmak için en iyi yolu kendileri seçerler. Çapraz fonksiyonel takımlar, takım dışından birilerine bağımlı olmaksızın işi başarmak için ihtiyaçları olan tüm yetkinliklere sahiptir. Bu yapı esneklik, yaratıcılık ve verimliliği artırmaktadır (Schwaber ve Sutherland, 2013). Scrum esnek bir proje yönetim şekli olduğu için, Scrum Ekibinin sahip olması gereken en önemli özellik kendi kendine yönetebilme yeteneklerinin olmasıdır. Kendi kendine organize olan ekipler işi başarmak için en iyi yolu tercih ederek daha başarılı ürünlerin ortaya çıkmasını sağlamaktadır. Başarılı ürünler çıkarmak için proje süresince ihtiyaç duyulan tüm yeteneklere ekip içinden bireylerin sahip olması önemli bir

noktayı teşkil etmektedir. Örneğin; bir mobil uygulama geliştirme sürecinde uygulama içinde kullanılacak servisler, Geliştirme Ekibinin yeterliliği olmaması dolayısıyla Geliştirme Ekibi içinde kodlanmayıp, başka bir ekibe veya firmaya kodlatılmak zorunda kalınabilir. Bu durumda Geliştirme Ekibi, servislerin oluşturulmasını beklemek zorunda kalır. Bu şekilde olması Scrum süresince tercih edilen bir durum olamamaktadır. Çünkü ekip dışarıdan birilerine bağımlı olduğu için proje süresince olumsuzluklar yaşanabilmekte ve ekibin motive olamamasına sebep olmaktadır. Bu da proje sürecinin esneklik, yaratıcılık ve verimliliğini engelleyen bir durum oluşturmaktadır.

2.5.3.1.Ürün Sahibi

Ürün Sahibi, iş değerlerini başarmaktan sorumludur ve Ürün Gereksinim Listesi içinde organize edilmiş ihtiyaçların, görevlerin, kısıtların ve sistem açıklarının geliştirilmesinin ve düzenlenmesinin bir öncelik listesi ile sunulan ürünün vizyonunun (Scharff ve Verma, 2010), proje ilerledikçe bu vizyon netleşir (Schwaber,2004), sorumluluğuna sahiptir (Scharff ve Verma, 2010). Bundan dolayı Ürün Sahibinin, projenin veya ortaya çıkan sistemin etkilediği herkesin çıkarlarını temsil etme sorumluğu bulunmaktadır (Schwaber, 2004).

Ürün Sahibi bir grup değil, bir kişidir. Ürün Sahibi Ürün Gereksinim Listesi için bir grubun isteklerini temsil edebilir ama Ürün Gereksinim Listesindeki bir maddenin önceliğinin değiştirilmesinin sorumluluğu Ürün Sahibine aittir (Schwaber ve Sutherland, 2013). Proje sürecinde oluşan ürünün etkileyeceği kişi Ürün Sahibidir. Ürün Sahibi firması veya kendisi için ihtiyaç olan ürünü en iyi anlamış ve bu ürünün tüm sorumluluğunu üzerine almış kişi olmaktadır. Örneğin; üniversite için geliştirilen bir sistemde projeden birinci derecede etkilenecek birimden seçilen bir kişi Ürün Sahibi rolüne karşılık gelmektedir. Sistem Öğrenci İşleri için geliştiriliyorsa, bu sistemin geliştirilmesi sürecinde Geliştirme Ekibi ve Scrum Uzmanı ile sürekli irtibat halinde olması için seçilen kişi ürünün doğru şekilde geliştirilmesindeki tüm sorumluluğa sahiptir. Öğrenci İşleri bir grup olarak görülebilir ama Ürün Sahibi bir grup olamayacağı için, kesinlikle birim içinden seçilmiş ve istenilen projenin gerçek yaşam uygulamasına hakim birisi, yani bir muhatap Ürün Sahibi rolüne bürünmelidir.

Ürün Sahibinin Ürün Gereksinim Listesi konusundaki sorumlulukları şu şekildedir (Schwaber ve Sutherland, 2013);

- Ürün gereksinimlerini açıkça ifade etmelidir.
- Ürünün amacını ve misyonunu en iyi şekilde başarmak için Ürün Gereksinim Listesinin maddelerini öncelik sırasına göre sıralamalıdır.
- Geliştirme Ekibinin geliştirdiği işin değerini optimize etmelidir.
- Ürün Gereksinim Listesinin görünür, şeffaf ve Geliştirme Ekibinin bir sonraki aşamada ne yapacağını gösterecek şekilde açık olmasını sağlamalıdır.
- Geliştirme Ekibinin Ürün Gereksinim Listesindeki her bir maddeyi ihtiyaç duyulan seviyede anlamasını sağlamalıdır.

Ürün Sahibi firmasının, kurumunun veya kendisinin nasıl bir ürünle ihtiyacının giderileceğini ve bu ürün için gereksinimlerinin neler olduğunu detaylı şekilde ifade etmelidir. Örneğin; bir mobil öğrenci bilgi sistemi geliştirilirken öncelikli nokta öğrencilerin birinci dereceden önemli işlemlerini bu sistem üzerinden gerçekleştirmesi olmalıdır. Bundan dolayı bu ürünün geliştirilmesi ile sorumlu Ürün Sahibi birinci öncelikli gereksinimleri belirlemeli ve öğrencilerin mobil sistem üzerinden hangi işlemleri gerçekleştirmeleri gerektiğini belirlemelidir ve bu ürünün geliştirilmesi için öncelik listesi oluşturmalıdır. Örneğin; öğrencilerin "ilk etapta notlarını görmeleri, ikinci etapta mobil uygulama üzerinden ders kayıtlarını gerçekleştirebilmeli" şeklinde bir öncelik listesi olabilir. Bu şekilde ifade edilen bir gereksinimde ilk yapılması gereken öğrencilerin notlarını görebileceği mobil uygulamanın gerçekleştirilmesi olmalıdır. Bu ihtiyaç belirlendikten sonra Ürün Gereksinim Listesi netleştirilmeli ve oluşturulan Ürün Gereksinim Listesinin tüm süreç boyunca herkes tarafından görünür ve herkes tarafından anlaşılır olması sağlanmalıdır. Bu, en önemli noktalardan birisidir. Eğer Geliştirme Ekibi ve Scrum Uzmanının anlamadığı noktalar ortaya çıkarsa, bu iş sürecini yavaşlatmaktadır. Bunun yanı sıra süreç içinde anlaşılmayan maddelerin ortaya çıkma ihtimaline karşı sürekli ulaşılabilir olarak, Geliştirme Ekibinin sorularını cevaplamalıdır.

2.5.3.2.Geliştirme Ekibi

3 kişiden 15 kişiye kadar olan bir ekip olan Geliştirme Ekibi (Scharff ve Verma, 2010) işlevselliğin geliştirilmesi sorumluluğuna sahiptir (Schwaber, 2004) ve yazılımı geliştirilmektedir (Scharff ve Verma, 2010). Ürün Sahibi ve Scrum Uzmanı bu sayıya dahil değildir. 9 üyeden fazla üye içeren ekipler koordinasyona ihtiyaç duymaktadır. Üç kişiden az ekiplerde etkileşim azalmakta ve sonuca ulaşmak için üretkenlik de

yavaşlamaktadır. Geliştirme Ekibi her Koşu sonunda "Tamamlandı" kriterine uyan bir ürün ortaya çıkaran profesyonel kişilerden oluşmaktadır. Geliştirme Ekibi, kendi işlerini düzenlemek ve yönetmek için yapılandırılmış ve yetkili olmalıdır. Bu şekilde oluşan sinerji Geliştirme Ekibinin genel verimliliğini ve etkinliğini optimize etmektedir (Schwaber ve Sutherland, 2013). Projelerde çalışan kişi sayısı çoğaldıkça iletişim sorunlarından kaynaklanan birçok problem ortaya çıkabilmektedir. Bunun yanı sıra üç kişiden az ekiplerde ekip ruhu oluşamamakta ve etkileşim oluşmaması sonucu projede çalışan bireylerde motivasyon kaybı yaşanmaktadır. Bunun için Scrum süreç içinde ekiplerin belirli bir sayı içinde kalmasını öngörmüştür. Bu şekilde ekip içinde iletişimin artmasını sağlayarak, proje sürecinin tam verimle geçmesini sağlamayı öngörmektedir. Ekipler belirli bir sayı içinde kaldığı sürece ekibin kendi kendine organize olabilme yeteneği artmaktadır. Kendi kendine organize olan ekipler tüm yetkinliklerini kullanarak proje süresince kaliteli ve Ürün Sahibinin ihtiyaçlarını gideren ürünler ortaya çıkarmaktadır.

Geliştirme Ekipleri aşağıdaki karakteristik özelliklere sahiptir (Schwaber ve Sutherland, 2013) ;

- Geliştirme Ekipleri kendi kendine örgütlenmelidir. Scrum Uzmanı dahil olmak üzere herhangi birisi Ürün Gereksinim Listesindeki işin fonksiyonel olarak nasıl gerçekleştirileceğini söyleyememektedir.
- Geliştirme Ekipleri bir ürünü oluştururken bir ekip için gerekli olan tüm yetenekleri ile birlikte çapraz fonksiyonel olmalıdır.
- Scrum, Geliştirme Ekibi üyeleri için ekip üyelerinin hepsinin eşit yetkiye sahip olduğu geliştirici dışındaki bir tanımı kabul etmemektedir. Bu kuralın hiçbir istisnası yoktur.
- Scrum, Geliştirme Ekiplerinde test veya iş analizi gibi alt ekiplerin oluşturulmasını kabul etmemektedir.
- Geliştirme Ekibi üyeleri belirli alanlarda yoğunlaşmış veya özel yeteneklere sahip olabilirler fakat ürün sorumluluğu tüm Geliştirme Ekibine aittir.

Geliştirme Ekibinin en önemli özelliği öz örgütlenmeli olmasıdır. Bu, zamanı en iyi şekilde kullanarak maksimum verimlilikte işler çıkarmak anlamına gelmektedir. Örneğin; bir mobil uygulama geliştirme sürecinde, servise ulaşacak mimarinin oluşturulması ve

yapının kodlanması gerekiyorsa, bunun nasıl olacağına Geliştirme Ekibi karar vermektedir. Dışarıdan hiç kimse bu süreçte bir yapının kullanılması ve bir kodlama şeklini gerçekleştirilmesi şeklinde bir istekle gelememektedir. Bunun gerçekleştirilebilmesi için proje sürecinde ihtiyaç duyulan tüm yeteneklere Geliştirme Ekibindeki bireylerin sahip olması gerekmektedir. Geliştirme sürecinde bütün yeterliliklere sahip olan Geliştirme Ekibinde tüm bireyler süreç boyunca eşit yetki ve haklara sahiptir. Bundan dolayı Scrum süreci ekip içinde alt ekiplerin oluşturulmasını kabul etmemektedir. Gerektiğinde mobil uygulamanın bir parçasını geliştiren bir ekip üyesi mobil uygulamanın bir başka parçasını test edebilir. Süreç açısından bir ekip üyesinin bir ürünün aynı parçasını hem geliştirmesi hem de test etmesi hataların ortaya çıkmasını engellediği için kabul gören bir durum değildir. Geliştirme Ekibi bu sorumluluklarla birlikte bir ürünün yazılımsal olarak başarılı veya başarısız olmasının tüm sorumluluğunu üzerine almaktadır.

2.5.3.3. Scrum Uzmanı

Scrum Uzmanı süreç boyunca takımı yönlendirmekte ve sorunları çözmek için ekibe yardımcı olmaktadır (Scharff ve Verma, 2010).

Scrum Uzmanı, tüm Scrum sürecinden sorumludur (Schwaber, 2004), bundan dolayı ekibin Scrum teorisini, pratiklerini ve kurallarını uygulamasını sağlayarak, Scrum'un yürütülmesinden ve anlaşılmasını sağlamalıdır. Scrum Uzmanı, Scrum Ekibi için bir hizmetçi liderdir (*servant-leader*). Scrum Uzmanı ekibin dışarıdan gelen etkileşimlerden hangilerinin ekip için faydalı, hangilerinin ekip için faydasız olduğunu anlamalarını sağlayarak (Schwaber ve Sutherland, 2013) Geliştirme Ekibi, Ürün Sahibi ve müşteriler arasında oluşabilecek problemleri engelleme ve ortadan kaldırma sorumluğuna sahiptir (Schwaber,2004). Scrum Uzmanı süreç içinde kesinlikle bir yönetici değildir. Ekibin karşılaştığı problemlerin en hızlı şekilde çözülmesini sağlayarak, ekip için hizmet eden bir lider şeklindedir. Örneğin; bir mobil uygulama geliştirme sürecinin gerçekleştirilmesi sırasında Scrum Uzmanı rolündeki kişi mobil uygulama geliştirme sürecinde Scrum kurallarının uygulanmasının ve gerçekleştirilmesinin sorumluluğuna sahiptir. Ekibin mobil uygulama geliştirme sürecinde Scrum'u gerçekleştirmesi konusunda rehberlik ederek, mobil uygulama geliştirme sürecinde Scrum'un başarı ile uygulanmasını sağlamalıdır. Scrum sürecinin başarı ile uygulanması sırasında ekip dışından gelen ve

ekibi olumsuz etkileyen etkileşimlerin önüne geçme sorumluluğuna sahip olmalıdır. Örneğin; bir firma içinde Scrum ile proje geliştirilirken, ekipten sürekli bir şey istenmesinin önüne geçmelidir. Bu istekler proje ile ilgili olmayan bir veri tabanından belirli verileri çekecek sorguların yazılması, o an ilgili olunmayan bir proje ile ilgili kod düzenlemesi gibi istekler olabilmektedir.

Scrum Uzmanının Ürün Sahibi ile ilişkisi aşağıdaki şekildedir (Schwaber ve Sutherland, 2013);

- Scrum Uzmanı etkili Ürün Gereksinim Listesi yönetimi için teknikler bulmalı ve Ürün Sahibine yardımcı olmalıdır.
- Scrum Uzmanı, Scrum Ekibinin Ürün Gereksinim Listesi maddelerini açık ve net bir şekilde anlamalarına yardımcı olmalıdır.
- Ürün Geliştirme Planını kavramalıdır.
- Ürün Sahibinin işinin değerini maksimize etmek için Ürün Gereksinim Listesini nasıl düzenleneceği konusunda bilgisi olmasını sağlamalıdır.
- Çevikliğin anlaşılmasını ve uygulanmasını sağlamalıdır.
- Talep edilen veya ihtiyaç duyulan Scrum durumlarının kolaylaştırılmasını sağlamalıdır.

Örneğin; Scrum Uzmanı, bir mobil öğrenci bilgi sistemi gerçekleştirilmesi sırasında, mobil öğrenci bilgi sistemi için ihtiyaç duyulan gereksinimlerin Ürün Sahibi tarafından en iyi şekilde analiz edilerek Ürün Gereksinim Listesinin oluşturulmasının sağlanması için teknikler bulmalıdır. Zaman içinde oluşan değişikliklere göre bu Ürün Gereksinim Listesinin güncellenmesi ve en iyi şekilde yönetilmesini sağlamalıdır. İlk aşamadan itibaren ve oluşan değişikliklere göre ürün geliştirme sürecine hakim olmalıdır.

Bazı noktalarda Ürün Sahibinin proje süreci sonunda oluşan üründen kazancının daha çok olmasını sağlamak için ihtiyacı olan gereksinimleri nasıl analiz etmesi ve düzenlemesi gerektiği konusunda yardımcı olmalıdır. Örneğin; bir mobil elektronik belge yönetim sisteminin gerçekleştirilmesi sırasında önemli olan noktanın kullanıcının en çok gerçekleştirdiği işlemlerin ilk olarak mobil ortama aktarılmasını sağlamak olmalıdır. Önemli olan bu noktayı dikkat edilerek, bunları gerçekleştirmek için gerekli olan gereksinimlerin öncelikli olarak analiz edilmesi sağlanmalıdır.

Scrum Uzmanının Geliştirme Ekibi ile ilişkisi aşağıdaki şekildedir (Schwaber ve Sutherland, 2013);

- Geliştirme Ekibine öz örgütlenmeli ve çapraz fonksiyonel olma konusunda rehber olmalıdır.
- Değeri yüksek ürünler oluşturmak için Geliştirme Ekibine rehber olmalıdır.
- Geliştirme Ekibinin ilerlemesine mani olan engelleri kaldırmalıdır.
- Talep edilen veya ihtiyaç duyulan Scrum olaylarını kolaylaştırmalıdır.
- Scrum'u anlayamamış ve adapte olamamış örgütsel iş çevresinde Geliştirme Ekibine rehber olmalıdır.

Örneğin; bir mobil uygulama geliştirme sürecinde, mobil uygulama geliştirmek için tüm yetkinliklere sahip bireylerden oluşan ve kendi kendine organize olabilen bir Geliştirme Ekibi kaliteli, müşteri ihtiyaçlarını karşılayan ve piyasa şartlarında değerli bir ürün ortaya çıkarır. Bu süreçte Scrum Uzmanı, Geliştirme Ekibine, ürünün belirli bir parçasına devam etmesi için, bir verinin gelmesi gerekiyorsa, bu verinin mümkün olan en kısa sürede gelmesini sağlamalıdır.

Bu süreçte dikkat edilmesi gereken en önemli noktalardan birisi iş yaşamında tüm işlerin acil olmasıdır. Kurum, firma veya organizasyonların Scrum süreciyle proje geliştirildiğini anlamalarını sağlayarak, sürekli olarak Geliştirme Ekibini bölen işlerin gelmesini engellemeli ve bu süreçte Geliştirme Ekibine rehber olarak, bu süreçte gelebilecek olumsuzlukların önüne geçmelidir.

2.5.4. Scrum Sürecindeki Toplantılar

Hiçbir süreç ve yöntem statik değildir. Dolayısıyla geri bildirimlerin olması ve öz değerlendirmenin yapılması gerekmektedir (Leffingwell ve Smiths, 2006). Bundan dolayı proje süreçlerinde en önemli noktalardan birisi toplantıların gerçekleştirilerek yapılanlar hakkında değerlendirmelerin yapılması ve yapılacaklar hakkında programların planlanmasıdır. Yapılan toplantılar projenin o anki durumunu ortaya çıkarırken, projede o andan itibaren yapılacaklar için kararlar alınmaktadır. Bu süreçleri gerçekleştirmek için daha önce de değinildiği gibi Scrum sürecinde Koşu Planlama Toplantısı, Günlük Scrum Toplantıları, Koşu Değerlendirme Toplantısı ve Koşu Süreç Değerlendirme Toplantısı gerçekleştirilmektedir (Schwaber ve Sutherland, 2013).

2.5.4.1. Koşu Planlama Toplantısı

Ürün Sahibi, Geliştirme Ekibi ve Scrum Uzmanının katıldığı bir toplantı ile Koşu Planlama toplantısı gerçekleştirilmektedir (Highsmith, 2002). Tüm ekibin ve Ürün Sahibinin bu toplantıya katılmasının sebebi, kullanıcının her öyküsünün birbirine bağımlı üç değişken olan kapsam, önem ve tahmin içermesinden dolayıdır. Kapsam ve önem Ürün Sahibi tarafından belirlenirken, bunları gerçekleştirmek için tahmin Geliştirme Ekibi tarafından yapılır. Eğer Ürün Sahibinin toplantıya katılması mümkün olmasa, ekip içinden birisi Ürün Sahibinin vekili olarak seçilebilir. Bu toplantı sırasındaki yüz yüze diyalog ile bu üç değeri gerçekleştirmek sağlanmış olur (Kniberg, 2007). Koşu Planlama Toplantısı, geliştirme aşamasında ortaya çıkabilecek engellerin riskini azaltmak için (Kaleel ve Harinshankar, 2013) kritik bir toplantıdır (Kniberg, 2007). Kötü yapılan bir toplantı, bütün koşuyu kötü etkilemektedir (Kniberg, 2007). Koşu Planlama Toplantısında Koşu boyunca gerçekleştirilecek olan işler planlanmaktadır. Bu plan, tüm Scrum Ekibinin ortak çalışması ile oluşturulmaktadır. Koşu Planlama Toplantısı bir aylık Koşular için maksimum 8 saat şeklinde planlanmaktadır. Bir aydan daha kısa süreli Koşular için süre daha kısa olmaktadır (Schwaber ve Sutherland, 2013).

Bu toplantıların gerçekleştirilmesi sorumluluğu Scrum Uzmanına aittir ve toplantıya katılanlara toplantının amacına ulaşması için rehberlik etmelidir (Schwaber ve Sutherland, 2013). Bu toplantının amacı, birkaç haftalık Koşu boyunca rahatsız edilmeden huzur içinde çalışabilmeleri için Geliştirme Ekibine bilgi vermek ve aynı zamanda Ürün Sahibine yapılacak olanların yapılacağı konusunda güven vermektir (Kniberg, 2007). Toplantı için ortam ve zamanı sağlamak Scrum Uzmanı tarafından yapılmalıdır. Bu süreçte toplantıların amacına ulaşmasında rehberlik etmesi önemli bir noktayı teşkil etmektedir. Bazı toplantılarda Ürün Sahibi veya Geliştirme Ekibi o koşu için yapılacakları değil de, ürünün son aşamasını konuşmaya başlarlarsa, Scrum Uzmanı burada devreye girerek Koşu Planlama Toplantısının amacı dışına çıkmasını engellemesi gerekmektedir.

Bu toplantının girdisini Ürün Gereksinim Listesi, ürünün son hali, Geliştirme Ekibinin bir önceki performansı ve Koşu sırasında Geliştirme Ekibinin planladığı kapasite oluşturmaktadır. (Schwaber ve Sutherland, 2013). Örneğin; bir mobil oyun geliştirme sürecinde bu toplantının girdilerini; Ürün Gereksinim Listesinin oyun sektöründeki

değişimlere göre güncellenmiş son hali, mobil oyunun o ana kadar geliştirilmiş hali, Geliştirme Ekibinin bir önceki Koşudaki tecrübesi ve oluşturulan mobil oyun konusunda tecrübe kazanan Geliştirme Ekibinin bir sonraki Koşu için planladığı iş birim kapasitesi oluşturmaktadır.

Koşu Planlama Toplantısının çıktıları aşağıdaki şekildedir (Kniberg, 2007);

- Bir Koşu Hedefi belirlenmelidir
- Yapmayı taahhüt ettikleri görevlerle birlikte ekip üyelerinin bir listesi olmalıdır.
- Koşu Gereksinim Listesi belirlenmelidir.
- Koşu Değerlendirme Toplantısı tarihi belirlenmelidir.
- Günlük Scrum Toplantıları için yer ve zaman kesinleştirilmelidir.

Koşu Planlama Toplantısının ana amaçlarından birisi, Kullanıcı Hikayelerinden hangilerinin oluşturulacak olan Koşu Gereksinim Listesinde gerçekleştirileceğine karar vermektir (Kniberg, 2007). Bu toplantı sırasında Geliştirme Ekibi Ürün Gereksinim Listesinden yararlanılarak oluşturulmuş görevlerin belirli sayısını tamamlamak için taahhüt verir. Her bir görevi tamamlamak için gereken zaman Ürün Gereksinim Listesi kapsamında güncellenerek Koşu Gereksinim Listesi oluşturulur (Scharff ve Verma, 2010). Proje büyüklüğüne göre, birisi özellikleri seçmek, diğeri detaylı görevleri planlamak şeklinde iki toplantı gerçekleştirilebilir. Toplantı sırasında Ürün Sahibi gelecek Koşu için hangi özelliklere ihtiyacı olduğuna belirtmekte ve Geliştirme Ekibi bu özellikleri tamamlamak için gerekli görevleri hesaplamaktadır (Highsmith, 2002). Bu süreçte Geliştirme Ekibi Koşunun kaç Kullanıcı Hikayesi içereceğini karar vermekte (Kniberg, 2007) ve görev listesini belirlemekte ve Koşu sırasında ne kadarını gerçekleştirebileceklerini tahmin etmektedir (Highsmith, 2002). Bu tahminler sırasında planlama oyunu oynanmaktadır (Kniberg, 2007). Örneğin; geliştirilen proje büyük bir proje ve süreç içinde gerçekleştirilmesi gereken özellikler ve bu özellikleri gerçekleştirmek için yapılması gereken görevler çok detaylı olabilir. Bu şekilde bir projede özelliklerin belirlenmesi ve detaylı görevlerin ortaya konulması tek toplantıda mümkün olmayabilir. Bu şekilde olan bir durumda Koşu Planlama Toplantısı iki toplantı şeklinde gerçekleştirilebilir. İlk toplantıda Koşu sürecinde gerçekleştirilmesi gereken özellikler belirlenirken, ikinci toplantıda bu özellikler gerçekleştirmek için gerçekleştirilmesi gereken görevler hesaplanır. Bu süreç de şu şekilde

gerçekleştirilmektedir; İlk olarak önceden oluşturulmuş veya yeni oluşturulan Kullanıcı Hikayeleri analiz edilir. Kullanıcı Hikayelerinden bu süreçte gerçekleştirilmesi gereken özellikler belirlenir. Özelliklerin belirlenmesinden sonra bu özellikleri gerçekleştirmek için yapılması gereken görevler ortaya çıkarılır ve bu görevlerin iş birim büyüklüğü hesaplanır. İş birim büyüklüğü çeşitli zaman sınırları ile sınırlandırılır. Örneğin; bir birimlik iş, bir saate karşılık gelebilir. Bu şekilde iş birim büyüklüğü belirlendikten sonra Geliştirme Ekibi önlerindeki Koşu süreci için ne kadar iş yapabileceklerini tahmin eder.

Koşu Planlama toplantısında şu sorulara cevap verilmektedir (Schwaber ve Sutherland, 2013);

- Gerçekleştirilecek olan Koşu sonucunda hedef nedir?
- İşi başarmak için nasıl bir çalışma yapılmalıdır?

Planlama sürecinin son kısmında Koşu için amacı olan bir Koşu Hedefi belirlenmektedir. Detay özellikler ya da görevler ertelenmiş olsa bile Koşu Hedefi gerçekleştirilebilir olmalıdır (Highsmith, 2002). Bu, Koşu Planlama Toplantısının en önemli noktasını teşkil etmektedir. Örneğin; bir mobil ödeme sisteminin gerçekleştirilmesi sürecinde, Koşu için gerçekleştirilmesi gereken Koşu Hedefi, mobil uygulamanın kredi kartları ile entegrasyonun sağlanmış olması olabilir. Bu şekilde Geliştirme Ekibi detay özelliklerin gerçekleştirirken, asıl amaç olan sistemin kredi kartları ile entegrasyonunun sağlanması yönünü her zaman akıllarında tutarlar. Bu şekilde Koşu sonunda bazı detay özellikler atlanmış olsa bile, sistemin kredi kartları ile entegrasyonu sağlanmış olur.

Geliştirme Ekibi Ürün Gereksinim Listesinin maddelerini tahmin ettikten sonra Koşu sonucunda teslim edilecekler belirlenir ve Scrum Ekibi bir Koşu Hedefi oluşturur. Koşu Hedefi belirlendikten ve Koşu için Ürün Gereksinim Listesi maddeleri seçildikten sonra Geliştirme Ekibi Koşu sırasında "Tamamlandı" kavramına karşılık gelen ürünün fonksiyonel olarak nasıl inşa edileceğine karar verir. Bu karardan sonra Geliştirme Ekibi sistem tasarımı yaparak ve Ürün Gereksinim Listesinden çalışan bir ürün parçası çıkarmak için ihtiyaç olan işleri belirlemeye başlar. Bu işler farklı büyüklüklerde ve farklı eforlarda olabilir. Ancak Geliştirme Ekibi bu toplantı sırasında gelecek Koşu'da yapabileceklerine inandıkları iş miktarını tahmin eder ve bu tahminlere göre yeterince iş planlar. Bu işler bir günlük veya daha az birimlere bölünür (Schwaber ve Sutherland,

2013). Örneğin; bir mobil ödeme sisteminde kredi kartları ile entegrasyonun sağlanma özelliği Koşu sırasında sisteme eklenecekse, yapılması gereken işler belirlenir. Oluşturulan bu işler "*Bankaların servislerinin kullanılması (8 birim)*" ve "*Servisten gelen verinin işlenmesi (5 birim)*" şeklinde farklı büyüklüklerde olabilir.

2.5.4.2. Günlük Scrum Toplantıları

Günlük Scrum Toplantıları Geliştirme Ekibinin gelecek 24 saat için bir plan oluşturması ve aktivitelere senkronize olmalarını sağlamak için 15 dakikadan oluşan olaylardır. Bu toplantılar ile bir önceki toplantıdan sonra yapılan çalışmalar görülür (Schwaber ve Sutherland, 2013), yapılan iş miktarı (Nicolas, 2007) ve bir sonraki toplantıya kadar yapılacak çalışmalar tahmin edilir (Schwaber ve Sutherland, 2013). Bu şekilde sürekli olarak ekibin ilerlemelerini takip etmeleri sağlanmış olur (Abrahamsson, 2002). Bu toplantılar karmaşıklığı azaltmak için her gün aynı yer ve aynı zamanda gerçekleştirilir (Schwaber ve Sutherland, 2013). Toplantı yeri olarak ekibin çalıştığı odada Görev Kartlarının yer alığı tablonun önünde yapılmalıdır (Kniberg, 2007). Bu toplantıların sürekli yapılması ekip içindeki iletişimi artırırken, bir yönetici olmamasından kaynaklanan eksiklikler giderilerek, ekibin kendi çalışmalarını görmeleri sağlanır ve sürekli ekibin kendi kendine organize olması sağlanmış olur. Bu toplantıların kısa süreli olması ve bundan dolayı zaman kaybı yaşanmaması için her gün aynı yer ve zamanda gerçekleştirilmesi önemli bir nokta olmaktadır.

Bu toplantılar sırasında her takım üyesi kendisine üç soru sormaktadır; "Son toplantıdan sonra ne yaptım?", "Gelecek toplantıya kadar ne yapmayı planlıyorum" ve "Hedefe ulaşmak için önümde hiç engel var mı, varsa nedir?" (Scharff ve Verma,2010). Örneğin; bir mobil uygulama geliştirme sürecinde bu soruların cevabı; "son toplantıdan sonra kullanıcıların mobil uygulamaya bilgi girmeleri gereken ekran tasarımlarını yaptım", "bir sonraki toplantıya kadar bu tasarımlar üzerinden kodlamaları gerçekleştireceğim" ve "bu hedefimi gerçekleştirmek için ekip arkadaşımın oluşturduğu servisleri kullanacağım, servisler doğru çalışıyorsa bir engelim olmayacak" şeklinde olabilir.

Günlük Scrum Toplantıları Geliştirme Ekibinin Koşu Hedefine ulaşma olasılığını kolaylaştırarak (Schwaber ve Sutherland, 2013), proje sürecindeki bilgiye sosyalleşmelerini sağlar (Beedle ve ark., 1999). Bilgi sosyalleşmesi kendi kendine organize olan bir ekip yapılanması sağlar ve geliştirme süreci anlık olarak

değerlendirilmiş olur (Beedle ve ark., 1999). Scrum Uzmanı Geliştirme Ekibinin toplantıyı yapmalarını sağlar. Geliştirme Ekibi, Scrum Uzmanı rehberliğinde toplantıların Günlük Scrum Toplantıları kurallarına göre yürütülmesinden sorumludur. Scrum Uzmanı Geliştirme Ekibine toplantıların 15 dakikalık zaman sınırı içinde tamamlanması konusunda rehberlik eder. Günlük Scrum Toplantıları ekip içi iletişimi geliştirerek ekstra toplantıların gerekliliğini ortadan kaldırır ve ekip üyelerinin proje ile ilgili bilgi seviyelerinin artmasını sağlar (Schwaber ve Sutherland, 2013). Bu şekilde herkes proje sürecinde parça parça ürün oluştururken, aynı zamanda oluşan ürünün bütününü görmüş olur ve oluşturdukları ürün hakkında daha çok bilgiye sahip olurlar. Böylece ekip üyeleri oluşan ürünün bütününü anladıkları için özgüvenleri artarken, proje içinde yaptıkları iş birimi çoğalmaktadır.

2.5.4.3. Koşu Değerlendirme Toplantısı

Her Koşu sonunda Koşu Değerlendirme Toplantısı olarak adlandırılan bir süreç gerçekleştirilir (Scharff ve Verma, 2010).

Koşu Değerlendirme Toplantısı gerçekleştirilen ürünü incelemek ve ihtiyaç duyulan Ürün Gereksinim Listesini düzenlemek için her Koşu sonunda gerçekleştirilir. Bu toplantı sırasında, Scrum Ekibi ve paydaşları Koşu sürecinde yapılan işler hakkında işbirliği içinde tartışmaktadır (Schwaber ve Sutherland, 2013) ve paydaşlar Geliştirme Ekibinin ne yapıyor olduğunu öğrenmektedir (Kniberg, 2007). Ürün Gereksinim Listesindeki herhangi bir değişikliğe bağlı olarak, katılımcılar projenin değerini optimize etmek için bir sonraki süreçte yapılabilir işler hakkında işbirliği içinde tartışırlar. Bu şekilde geri bildirimi ortaya çıkarmak ve işbirliğini artırmak için yapılan işin sunumu yapılmaktadır (Schwaber ve Sutherland, 2013). Örneğin; bir mobil uygulama geliştirme sürecinde o ana kadar geliştirilmiş olan mobil uygulamanın tamamı bu toplantı sırasında sunularak incelenir ve değerlendirilir. Eğer mümkünse uygulamanın o ana kadar olan kısmı müşteriye önceden gönderilirse, toplantı sırasında alınan geri bildirimler daha verimli ve sağlıklı olabilir. Bu toplantı esnasında mobil uygulamanın o ana kadar yapılan kısmı incelenirken, bu süreçten sonra yapılacaklar da detaya girmeden konuşulur ve Ürün Gereksinim Listesinin güncellenmesi sağlanır.

Bu toplantılar bir aylık Koşular için dört saatlik, daha kısa süreli Koşular için daha kısa olacak şekilde düzenlenir. Scrum Uzmanı bu toplantıların belirlenen zaman sınırları

içinde gerçekleştirilmesini ve katılımcıların toplantının amacını anlamasını sağlar (Schwaber ve Sutherland, 2013).

Koşu Değerlendirme Toplantısı aşağıdaki unsurları içerir (Schwaber ve Sutherland, 2013);

- Katılımcılar Scrum Ekibi ve Ürün Sahibi tarafından davet edilen paydaşlardan oluşmaktadır.
- Geliştirme Ekibi, "Tamamlandı" kriterine uyan çalışmaları gösterir ve ürünün tamamlanan kısmı ile ilgili sorulara cevaplanmaktadır.
- Geliştirme Ekibi, Koşu sırasında nelerin iyi gittiğini, hangi problemlerin oluştuğunu ve bu problemlerin nasıl çözüldüğünü açıklamaktadır.
- Ürün Sahibi, Ürün Gereksinim Listesi maddelerinden "Tamamlanmış" veya "Tamamlanmamış" kriterine uyan maddeleri açıklamaktadır.
- Ürün Sahibi, son haliyle Ürün Gereksinim Listesini anlatır ve o güne kadar olan ilerlemeye dayanarak muhtemel bitiş tarihlerini projelendirmektedir.
- Bu toplantıda tüm grup gelecek aşamada ne yapılacağı konusunda işbirliği içinde tartışarak, Koşu Planlama Toplantısı için değerli bir girdi oluşturmaktadır.
- Gelecek adımda en değerli geliştirmenin yapılabilmesi için pazarın veya potansiyelin nasıl değişmiş olabileceği incelenmektedir.
- Ürünün bir sonraki beklenen sürümü için proje takvimi, bütçe, potansiyel kapasite ve pazar incelenmektedir.

Mobil uygulama geliştirme süreci açısından yukarıdaki maddeler şu şekilde özetlenebilir; bir üniversitenin öğrenci işleri için mobil uygulama geliştiriliyorsa katılımcılar, Scrum Ekibi ve Ürün Sahibi tarafından davet edilen paydaşlar olabilir. Bu paydaşlar üniversite öğrenci işleri biriminde etkin şekilde çalışan kişiler ve öğrenci işleri daire başkanı olabilir. Geliştirme Ekibi mobil uygulamanın o ana kadar oluşturulan kısmını bu toplantı esnasında sunarak, Ürün Sahibinden gelen soruları cevaplayarak, geri bildirimler alır.

Geliştirme Ekibi, Ürün Sahibi ve paydaşlarına sunumu gerçekleştirdikten sonra geçmiş Koşu sırasında yaşadıklarını belirtir. Örneğin; öğrenci işlerinden beklenen cevapların Ürün Sahibi tarafından anında Geliştirme Ekibine aktarılmasının, Geliştirme Ekibinin hızını olumlu şekilde etkilediği, ama istenilen bazı dosyaların düzenli olmaması

dolayısıyla Geliştirme Ekibinin problemler yaşadığı ve bunun bazı işlerin beklenenden daha çok zaman almasına neden olduğu gibi durumları belirtebilir.

Ürün Sahibi mobil uygulama dünyasında yaşanan gelişmeleri ve firmasının ihtiyaçlarını baz alarak Ürün Gereksinim Listesinin değişen güncellemelerle birlikte o anki halini anlatır ve Geliştirme Ekibinin o ana kadar olan performansına bağlı olarak Scrum Ekibi ile tartışarak muhtemel bir bitiş tarihi belirler.

Bu süreçlerden sonra gereksinimlerin son hali ortaya çıkarılarak, bir sonraki Koşu Planlama Toplantısı için değerli bir girdi oluşturulmuş olur.

Bu toplantı sonucu paydaşlardan önemli geribildirimler alınır (Kniberg, 2007) ve Ürün Gereksinim Listesi revize edilir (Schwaber ve Sutherland, 2013) ve yeni maddeler eklenir (Abrahamsson, 2002). Bu değişiklikle gelecek Koşu için muhtemel Ürün Gereksinim Listesi maddeleri tanımlanmış olur. Ya da Ürün Gereksinim Listesi yeni fırsatlarla karşılaşmak için tamamen değiştirilerek düzenlenebilir (Schwaber ve Sutherland, 2013).

Bu toplantı ile Geliştirme Ekibi oluşturdukları ürünü sunarak, o ana kadar yapmış oldukları çalışmayı gösterdikleri için kendilerini iyi hissetmektedirler (Kniberg, 2007).

Koşu Değerlendirme Toplantısı için dikkat edilmesi gereken noktalar aşağıdaki şekildedir (Kniberg, 2007);

- Koşu Hedefinin açık şekilde sunulması gerekmektedir.

Bu önemli bir noktadır. Koşu başında belirlenen, Ürün Sahibinin Koşu sonunda görmek istediği Koşu Hedefi açık şekilde sunulmalıdır.

- Sunum hazırlanma sürecinde çok zaman harcanmamalıdır. Çalışan kodun sunulmasına yoğunlaşılmalıdır ve sunum hızlı tempolu gerçekleştirilmelidir.

Bu toplantı öncesinde sunum hazırlama sürecinde zaman kaybını engellemek için, sunumun görsel olarak güzel olmasından ziyade, çalışan kodu gösterecek şekilde açık olmasına dikkat edilmelidir.

- Sunum esnasında “nasıl yaptıktan” ziyade “ne yaptık” üzerine odaklanılmalıdır.

Ürün Sahibi ve paydaşlar yazılımın nasıl gerçekleştirildiğinden ziyade, sonuçta oluşan ürünü görmek isterler.

2.5.4.4.Koşu Süreç Değerlendirme Toplantısı

Koşu Süreç Değerlendirme Toplantısı, Koşu Değerlendirme Toplantısından sonra ve bir sonraki Koşu Planlama Toplantısından önce yapılır. Bu toplantı sırasında ekiplerin hız ve morali değerlendirilir (Scharff ve Verma, 2010). Koşu Süreç Değerlendirme Toplantısı, Scrum Takımının kendisini değerlendirmesi için bir fırsat oluşturmakta ve sonraki Koşu sürecini iyileştirmek için plan oluşturulmasını sağlamaktadır (Schwaber ve Sutherland, 2013). Bu toplantının esas noktası, gelecek Koşuyu daha iyi nasıl gerçekleştirebilirizdir? (Kniberg, 2007).

Koşu Süreç Değerlendirme Toplantısı ile ilgili en önemli nokta bu toplantının yapıldığının emin olunmasıdır. Çünkü bazen ekipler bu toplantıyı yapmama eğiliminde olmaktadırlar (Kniberg, 2007).

Bu toplantı daha kısa süreli Koşular için süresi daha kısa olmakla birlikte bir aylık Koşular için 3 saatlik bir zaman aralığından oluşmaktadır. Scrum Uzmanı toplantının düzenlenmesini sağlamanın yanında, toplantıya süreç üzerinde sorumlu bir ekip üyesi olarak katılmaktadır (Schwaber ve Sutherland, 2013). Bu toplantı sırasında Scrum Uzmanı Koşu Gereksinim Listesini gösterir ve ekibin önemli olaylar ve kararlara dikkat ederek koşuyu özetlemesine yardımcı olur. Her bir kişi konuşması bölünmeden, süreç içinde iyi yapıldığını düşündüklerini, sürecin daha iyi nasıl yapılabileceğini ve gelecek Koşuda tamamen farklı yapmaları gereken şeylerin neler olduğunu söylemektedir. Toplantı sonunda Scrum Uzmanı gelecek Koşuda daha iyi ne yapılabileceği hakkında somut önerileri özetlemeye çalışır (Kniberg, 2007).

Koşu Süreç İyileştirme Toplantısının amacı şu şekildedir (Schwaber ve Sutherland, 2013);

- Son Koşunun insanlar, ilişkiler, süreçler ve araçlar bakımından nasıl gittiği incelenir.
- İyi giden ve potansiyel gelişmelerden önemli öğeler belirlenir ve sıralanır
- Scrum Ekibinin çalışmalarını yapabileceği şekilde iyileştirmelerin uygulanması için bir plan oluşturulur.

Örneğin; mobil uygulama geliştirme sürecinde ekip içinde çalışan bireylerin bu geliştirme sürecindeki performansları ve yeni ne gibi bilgiler öğrendikleri değerlendirilmektedir. Ekip içindeki etkileşim ve iletişim nasıl gerçekleşti ve ekip içi iletişimde herhangi bir problemle karşılaşılıp karşılaşılmadığı değerlendirilir. Scrum sürecinin uygulanmasında problemler yaşandı mı, yaşandıysa bunlar nelerdir şeklinde sorulara cevaplar aranır. Mobil uygulama geliştirme araçları nasıl kullanıldı, bu alanda çıkan yeni teknolojiler var mı, mevcut kütüphaneler ve yeni çıkan kütüphaneler proje sürecine iyi uyarlanabildi mi gibi sorulara cevap aranmaktadır. Bu sürece kadar iyi giden yönler belirlendikten sonra, oluşabilecek gelişmelere göre neler yapılacağı değerlendirilir ve bir yol haritası çıkarılır.

Scrum Uzmanı süreç çerçevesinde gelecek Koşu için geliştirme sürecini ve pratiklerini daha etkili ve eğlenceli hale getirmek için Scrum Ekibini cesaretlendirir. Scrum Ekibi Ürün kalitesini artırmak için "Tamamlandı" kriterine uyacak şekilde yöntemler planlar (Schwaber ve Sutherland, 2013).

Koşu Süreç Değerlendirme Toplantısı sonunda, Geliştirme Ekibi sonraki Koşuda uygulanacak iyileştirmeleri tespit etmektedir. Gelecek Koşuda bu gelişmelerin uygulanması, Geliştirme Ekibinin kendi kendini denetlemesi ile gerçekleştirilmektedir. Gelişmeler herhangi bir zamanda uygulanabilir olsa da, Koşu Süreç Değerlendirme Toplantısı denetim ve uyum sağlama konusunda resmi bir fırsat sağlar (Schwaber ve Sutherland, 2013). Bu toplantı gerçekleştirilmeden, ekip tekrar tekrar aynı hataları yapmaya devam edebilir (Kniberg, 2007).

2.5.5. Scrum Sürecinin Gerçek Hayatta Uygulanması

Tez çalışmasında literatür bilgisine ilave olarak 6 Mart 2014 Perşembe günü yazılım sektöründe Scrum Uzmanı olarak çalışan Julide Arslan ile görüşülerek gerçek hayatta Scrum metodunun uygulanması hakkında bilgiler alınmıştır. Bu başlık altında gerçek hayatta nasıl uygulandığına dair kişisel görüşme sonucu oluşan notlara yer verilmiştir.

Scrum uygulanabilmesi için öncelikle Scrum Ekibi oluşturulması gerekmektedir. Scrum Ekibi oluşturulduktan sonra, ekibin artık Scrum ile proje geliştirmeye başladıklarının farkında olmaları için Scrum hakkında eğitimler düzenlenmelidir. Ekibin tam olarak Scrum metodunu kavraması belirli bir süre almaktadır. Ekip Scrum Metodunu kavradıkça yaptıkları iş birimi artmaktadır. Scrum süresince saat kavramı yerine birim kavramı

kullanılmaktadır. Koşu Planlama Toplantısında bir görev oluşturulduğunda bu görevin kaç birim olduğu tespit edilmektedir.

Scrum süresince 3 önemli kavram bulunmaktadır. Bunlardan birincisi Ürün Sahibidir. Ürün Sahibi bir ürün geliştirme süresinde ihtiyaçları ve gereksinimleri en iyi şekilde anlayan ve ekibe aktaran müşteriyi temsil eden kişidir. Ürün sahibi şirketin içinden birisi veya iş yapılan şirketten birisi olmaktadır. Ürün Sahibi olarak iş yapılan şirketten birisinin olması her zaman tercih edilmektedir. Çünkü ürün ihtiyaçlarını en iyi bilen kişi o şirket içinde olan birisi olmaktadır. Eğer iş yapılan şirketin kaynak yetersizliğinden dolayı bu gerçekleşmezse, Ürün Sahibi olarak Geliştirme Ekibinden birisi seçilmektedir. Bu kişi iş yapılan şirketle sürekli iletişim halinde olarak ürünün en iyi şekilde gerçekleştirilmesinin sorumluluğuna sahiptir.

İkinci önemli kavram; Scrum Uzmanıdır. Scrum Uzmanı kesinlikle bir yönetici değildir ve ekip içinden birisi gibi çalışmaktadır. Scrum Uzmanı ekip herhangi bir problemle karşı karşıya kaldığında devreye girmektedir. Örneğin; Geliştirme Ekibi yazılım geliştirme sürecinde iş yaptıkları şirketten bir bilgiyi beklemektedir. Geliştirme Ekibinden bilgiye ihtiyacı olan kişi bu Koşuda görevlerin tamamlanabilmesi için bilgiye ihtiyaçları olduğunu belirterek Ürün Sahibinden bu bilgiyi istemektedir. Bu bilgiyi almak için iki kere e-posta gönderdikten sonra cevap gelmezse, 3. e-postayı atarak durumu Scrum Uzmanına bildirmektedir. Scrum Uzmanı da bu problemin çözülmesini sağlamak için devreye girmektedir. Scrum Uzmanı, aynı zamanda Günlük Scrum Toplantılarının gerçekleştirilmesinden de sorumludur. Bu toplantılara bir yönetici gibi değil, takım içinde çalışan birisi olarak katılmaktadır.

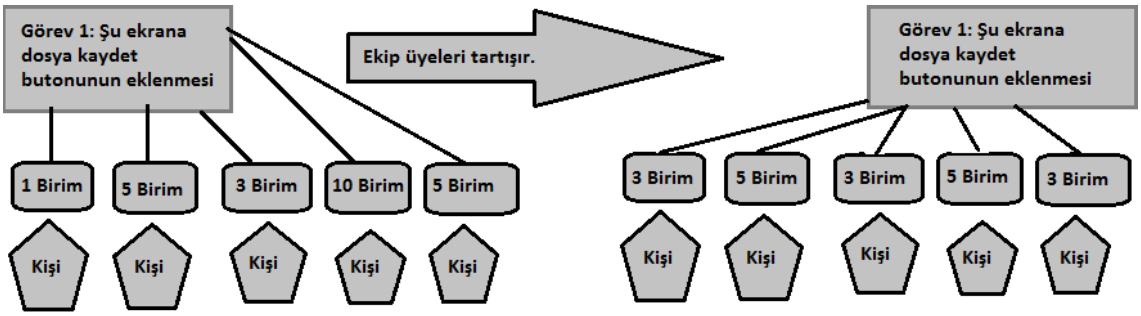
Üçüncü önemli kavram; Geliştirme Ekibidir. Geliştirme Ekibinde en önemli nokta kendi kendine yönetebilme yeteneği olan farklı niteliklere sahip bir ekip olmasıdır. Çünkü Geliştirme Ekibinde bir yönetici kavramı yoktur. Geliştirme Ekibinde herkes her işi yapabilmektedir. Örneğin; yeri geldiğinde bir yazılım geliştirici test etme işlemlerini uygulayabilmektedir. Bir test etme işlemi uygulayan kişi, eğer yazılım geliştirme konusunda yeteneği varsa yeri geldiğinde kod da yazabilmektedir.

Bir proje süreci başlarken öncelikle Ürün Sahibi ile ürün için gereksinimlerin oluşturulması sağlanmaktadır ve ürün gereksinimleri tespit edildikten sonra Ürün

Gereksinim Listesi oluşturulmaktadır. Ürün Gereksinim Listesi oluşturulduktan sonra Geliştirme Ekibi ile birlikte Koşu öncesi bu gereksinimler tekrar gözden geçirilmekte ve Koşu Planlama Toplantısına hazırlanılmaktadır.

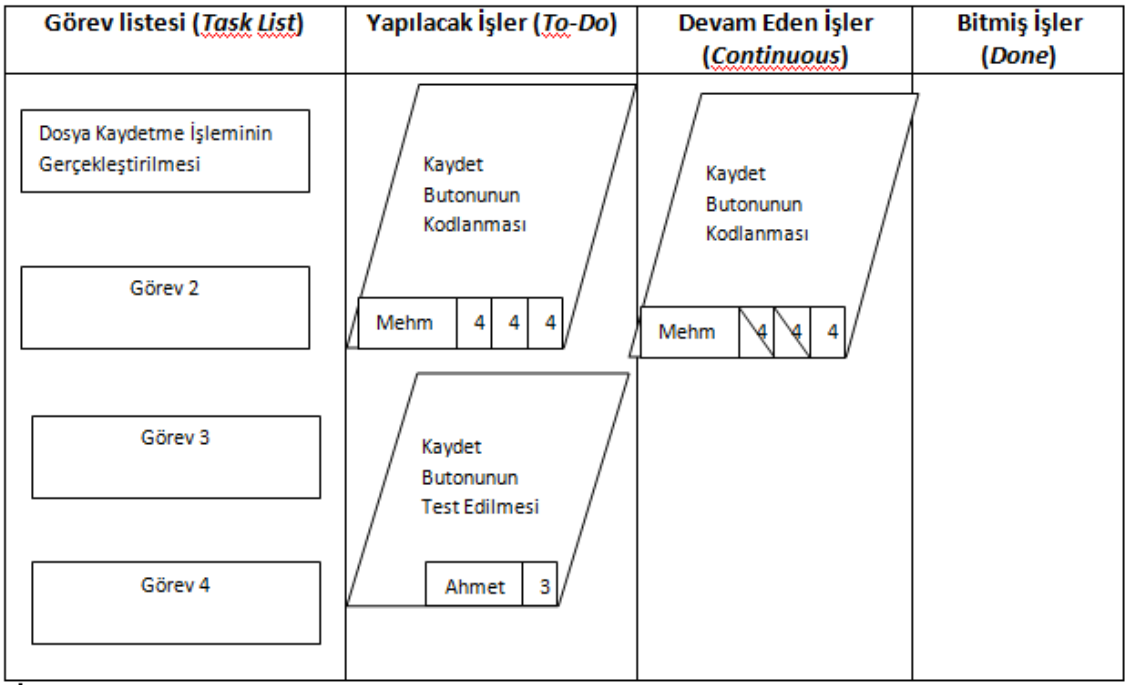
Koşu süresi genelde 2 ile 3 hafta arasında tutulmaktadır. 1 haftalık Koşular çok kısa olmakta, 3 haftadan uzun Koşular ise çok uzun olmaktadır.

Ürün Sahibi, Scrum Uzmanı ve Geliştirme Ekibinin katıldığı Koşu Planlama Toplantısında Koşu Gereksinim Listesi oluşturulmaktadır. Bu toplantı sırasında Ürün Gereksinim Listesindeki maddelerden hangilerini, öncelik veya Ürün Sahibi isteklerine göre Koşu süresince gerçekleştirilebileceği kararı verilmektedir. Görevlerin ne kadar sürede bitirileceği birim kavramı ile ifade edilmektedir. Örneğin Koşu boyunca istenen özelliklere göre görevler ayrılmaktadır. Her bir görev için tüm ekibe ne kadar birimde bitirebilecekleri sorulmaktadır. İlk görev için herkes kağıda bir birim yazarak göstermektedir. En az birim yazanla en çok birim yazana neden dolayı bu şekilde yazdıkları sorulmaktadır. Örneğin; birimi en çok yazan kişi, bu görevi sadece bir buton ekleme işlemi olarak görülebildiği ama arka planda çalışan diğer servislerinde değiştirilmesi gerekeceği, bundan dolayı uzun süren bir görev olacağını belirtmektedir. En az yazan kişi de, bu şekilde bir işlemi önceden gerçekleştirdiğini, ilk etapta arka planda çalışan servislerin de değiştirilmesi gerektiği gibi göründüğü ancak değiştirmeye gerek kalmadığını söyleyebilmektedir. Scrum Uzmanı ve Geliştirme Ekibi bunları tartıştıktan sonra ikinci kez Scrum Uzmanı bu görev için ne kadar birimde bitirebileceklerini sormaktadır. Ve ikinci kez sorulduktan sonra görev için bitirme birimi ortaya çıkarılmaktadır. Bu süreç Şekil 2.3'de ayrıntılandırılmıştır. Bu şekilde görevler oluşturulurken, Geliştirme Ekibi "bu Koşu için bu kadar görevi gerçekleştirebiliriz" dediğinde görev dağıtımı durdurulmaktadır. Bu noktada, Ürün Sahibi, acil olarak başka bir ihtiyacı belirtse dahi ekstra görev alınmamaktadır. Eğer bunun kesin gerçekleştirilmesi gerekiyorsa önceden alınmış görevlerden o görevin birimi kadar görev iptal edilerek Koşu Planlama Toplantısında o görev eklenmektedir.



Şekil 2.3: Koşu Planlama Toplantısında görevler için birimlerin belirlenmesi.

Koşu aşamasında; Koşu Planlama Toplantısı sonrası belirlenen görevler listesi beyaz bir tahta üzerine renkli kağıtlara yazılarak yapıştırılmaktadır. Bu beyaz tahta üzerinde Görev Listesinin yanındaki kısımlar Yapılacak İşler, Devam Eden İşler ve Bitmiş İşler kısımlarından oluşmaktadır. Örneğin; Şekil 2.4'de görüldüğü gibi görev listesinde "*Kaydet butonunun kodlanması*" kısmı 12 birimlik ve "*Kaydet butonunun test edilmesi*" kısmı 3 birimlik görev olarak ele alındığında "*Kaydet butonunun kodlanması*" görevi 4 4 4 şeklinde birimlere ayrılmaktadır. Yazılım Geliştirici, bu görevi Devam Eden İşler kısmına alarak kod geliştirmeye başlamaktadır. Bir gün sonunda 8 birimlik iş yapılırsa iki tane 4'ün üzerini çizerek, o işi devam eden işler kısmında bırakmaktadır. Bir sonraki gün geldiğinde diğer 4 birimlik kısmı da tamamladıktan sonra o kağıdı Devam Eden İşlerden alarak Bitmiş İşler kısmına yapıştırmaktadır. Test görevini üstlenmiş "Tester" bu işin bittiğini Bitmiş İşler Kısmında görerek, bu işle ilgili olan "*Kaydet Butonunun test edilmesi*" işlemine başlayabileceğini anlamaktadır. Bu işi Yapılacak İşler kısmından alarak Devam Eden İşler kısmına yapıştırıp test işlemine başlamaktadır. 3 birimlik test işlemini tamamladığında Bitmiş İşler kısmına yapıştırmaktadır. Bu şekilde Koşu süreci gerçekleştirilmiş olmaktadır.

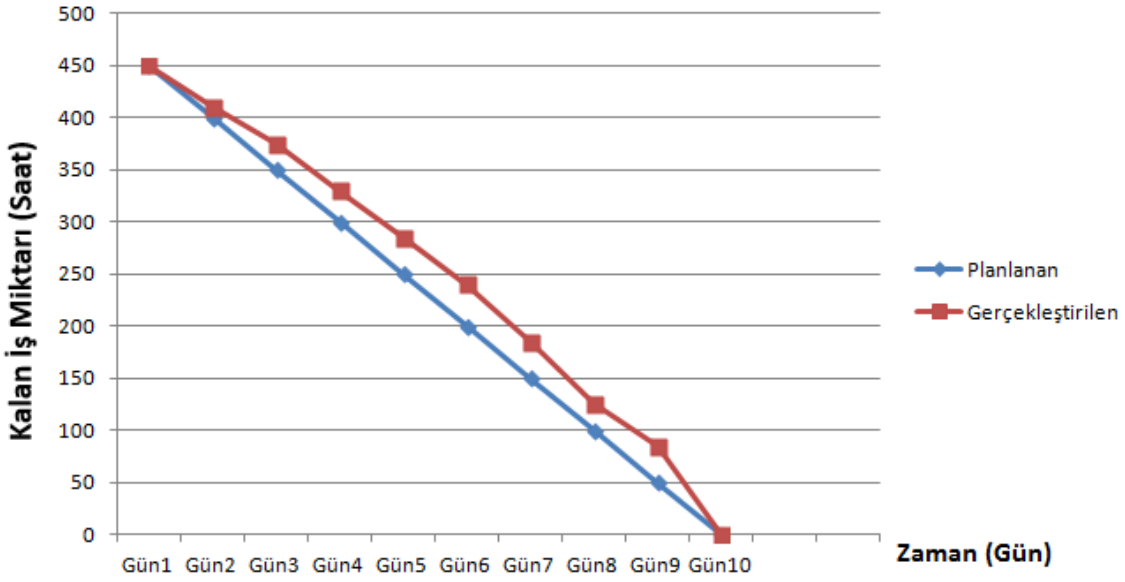


Şekil 2.4: Koşu aşamasında görevleri gerçekleştirme planı.

Koşu sırasında dikkat edilmesi gereken noktalardan biri Günlük Scrum Toplantılarının yapılmasıdır. Bu toplantıların gerçekleştirilmesinden Scrum Uzmanı sorumludur. Bu toplantılar 10 dakikayı geçmemeli ve ayakta yapılmalıdır. Gerekirse 10 dakikayı geçirmemek için zaman tutulmalıdır. Çünkü 10 dakikayı geçtiği zaman ekip üyeleri diğer ekip üyesini dinlemeyebilmektedir. Konuşma sırasında dikkati en üst düzeyde tutmak ve dinlemeyi sağlamak için kırmızı top oyunu oynanması önerilmektedir. Bu oyunda konuşma yapan kişi konuşma yaparken elinde kırmızı bir top tutmaktadır. Konuşması bittiğinde aniden ekipteki herhangi birisine bu topu atmaktadır ve konuşma yapmaya o kişi başlamaktadır. Bu toplantı sırasında Geliştirme Ekibindeki herkes yaptığı işle ilgili kısaca konuşmalıdır. Yaptığı işle ilgili problem varsa bunu belirtmelidir.

Bir Koşu sırasında işlerin ne kadarının tamamlandığının en iyi şekilde görülebilmesini sağlayan Koşu İş Takip Grafiğidir. Bu Koşu İş Takip Grafiği üzerinde her gün tamamlanan iş birimlerini tamamlayarak görevlerin ne kadarının tamamlandığı ve hedeflenen zamana göre değişim görülebilmektedir. Örneğin; Koşu süreci boyunca hedef doğrultusunda ne kadar başarılı olduğu görülebilmektedir. Koşu sürecinin sonuna yaklaşırken eğer hedeflenen süreden biraz sapma varsa, bunu zamanında tamamlamak için aksiyonlar alınmaktadır. Bu aksiyonlar mesaiye kalmak şeklinde olabilmektedir.

Şekil 2.5'de yer alan Koşu İş Takip Grafiği ilk aşamada sadece Planlanan çizgiden oluşmaktadır. 450 birimlik planlanan iş mevcuttur ve bu işin 10 günde gerçekleştirilmesi planlanmaktadır. Ekip her gün sonunda kaç birimlik iş yaptığını belirleyerek Gerçekleştirilen kısmında gerçekleştirdikleri iş kadar aşağıya doğru grafiği tamamlamaktadır. Bu grafikteki Planlanan kısmında işin planlanan şekilde devam ederse nasıl gerçekleştirileceği anlaşılırken, Gerçekleştirilen kısmında ise plana göre nasıl hareket edildiği görülmektedir.



Şekil 2.5: Koşu İş Takip Grafiği örneği.

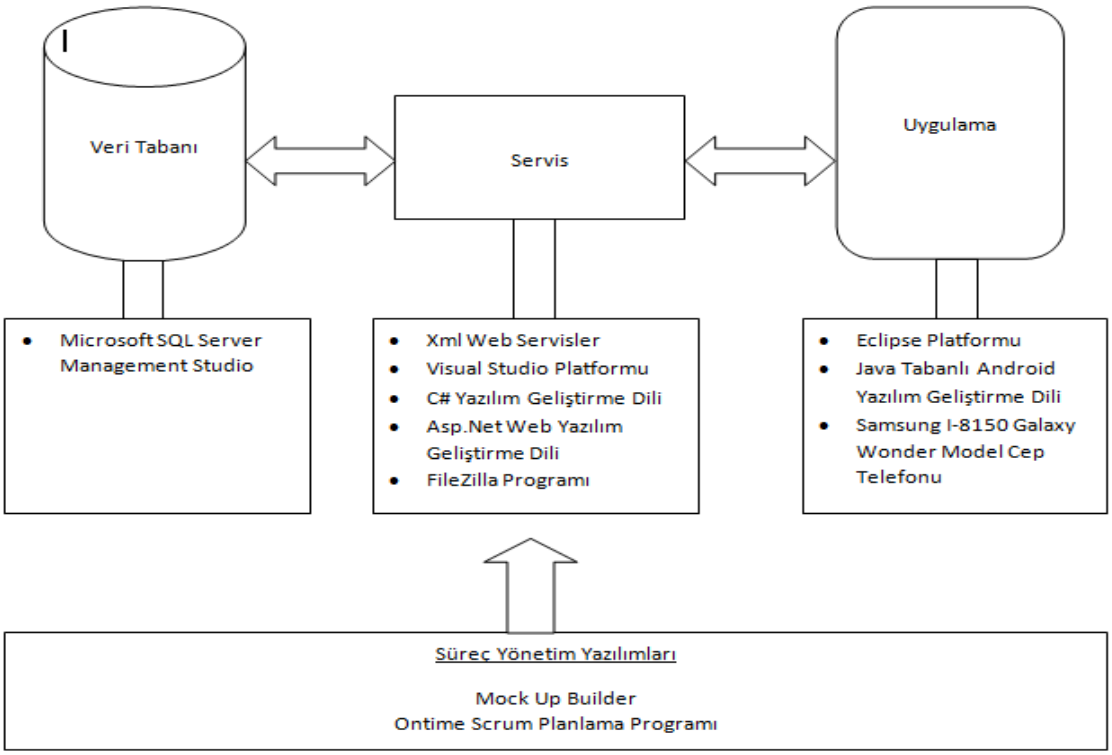
Koşu sonrası Koşu boyunca yapılan görevleri değerlendirmek için bir toplantı gerçekleştirilmektedir. Geçmiş Koşu süreci bu toplantıda değerlendirilmektedir. Scrum süresince minimum doküman oluşturmaya çalışılmaktadır. Geliştirme Ekibi öncelikli olarak görevlere odaklanarak görevi gerçekleştirmeye çalışmaktadır.

3. MALZEME VE YÖNTEM

3.1. MALZEME

Bu tez kapsamında geliştirilen mobil uygulama; verilerin sağlandığı veritabanı, veritabanı ile uygulama arasında iletişimi sağlayan servisler ve mobil uygulama kısmı olmak üzere üç adımdan oluşmaktadır.

Veri tabanı oluşturma kısmında, Microsoft SQL Server Management Studio platformu kullanılmıştır. Mobil uygulama geliştirme sürecinde Eclipse platformu üzerinde Java Tabanlı Android Yazılım Geliştirme dili kullanılmıştır. Bu iki aracın birbiri arasında iletişimi sağlayacak servisler oluşturulma aşamasında ise Visual Studio platformu üzerinde C# ve Asp.Net yazılım geliştirme dili kullanılmıştır (Şekil 3.1).



Şekil 3.1: Uygulama Geliştirme Sürecinde Kullanılan Araçlar

3.1.1. Veritabanı

Microsoft SQL Server Management Studio, Microsoft tarafından geliştirilmiş ilişkisel veri tabanı yönetimini sağlayan tümleşik bir ortamdır (Gürsoy, 2012). Tez kapsamında veri tabanı işlemleri, kolay veri tabanı yönetimi sağladığı için, Microsoft SQL Server Management Studio üzerinde gerçekleştirilmiştir.

3.1.2. Servisler

Verilere servisler aracılığıyla ile Binary, XML ve JSON olmak üzere üç yöntemle ulaşılabilmektedir. Genişletilebilir İşaretleme Dili olan XML farklı sistemler arasında veri alışverişi yapmaya yarayan standart bir dildir. Bundan dolayı günümüzde birçok sistem veri alışverişini XML formatı üzerinden yapmaktadır[4]. Tez kapsamında farklı platformlar arasında iletişimi sağladığı ve tez öğrencisinin bu alanda deneyimli olması nedeniyle XML Web Servisler kullanılmıştır.

Visual Studio, hem Microsoft hem de Android gibi diğer platformlar için uygulamaların geniş bir yelpazede oluşturulmasına yardımcı olmak amacıyla araçların ve servislerin kapsamlı bir çerçevesini içeren bir geliştirme platformudur[5]. Tez kapsamında gerçekleştirilen servisler, birçok hazır kütüphane barındırması ve C# Yazılım Geliştirme Dili ve Asp.Net Web Yazılım Geliştirme dilini desteklemesi nedeniyle Visual Studio platformu üzerinde gerçekleştirilmiştir.

C# Yazılım Geliştirme Dili, Microsoft tarafından geliştirilmiş olan ve nesne yönelik yazılım geliştirme ilkelerini destekleyen bir yazılım geliştirme dilidir[6]. Tez kapsamında servislerin kodlanma işleminde, nesne yönelik yazılım geliştirme ilkelerini desteklediği için C# Yazılım Geliştirme Dili kullanılmıştır.

Asp.Net Web Yazılım Geliştirme Dili, hazır kütüphaneleri desteklemesi nedeniyle servislerin kodlanması aşamasında kullanılmıştır.

FileZilla, FileZilla Client ve FileZilla Server'dan oluşan özgür, açık kaynak kodlu ve çoklu platform destekli bir dosya transfer programıdır[7]. Tez kapsamında yazılımın

4 http://tr.wikipedia.org/wiki/XML, [Ziyaret Tarihi: 10 Nisan 2014]
5 http://www.visualstudio.com/, [Ziyaret Tarihi: 10 Nisan 2014]
6 http://tr.wikipedia.org/wiki/C_Sharp, [Ziyaret Tarihi: 10 Nisan 2014]
7 http://tr.wikipedia.org/wiki/FileZilla, [Ziyaret Tarihi: 10 Nisan 2014]

sunucuya atılma işlemleri, ücretsiz olduğu ve kullanım kolaylığı sağladığı için FileZilla Client programı ile gerçekleştirilmiştir.

3.1.3. Uygulama

Eclipse, yazılım geliştirme yaşam döngüsü boyunca yazılımı oluşturmak, geliştirmek ve yönetmek için genişletilebilir geliştirme kütüphaneleri ve araçlarından oluşan çeşitli yazılım geliştirme dillerini destekleyen açık kaynak kodlu geliştirme üzerine yoğunlaşan, tümleşik geliştirme ortamı sağlayan bir geliştirme platformudur[8].

Tez kapsamında gerçekleştirilen Android uygulaması nesne yönelik programlama ilkelerini desteklemesi, Google'un eklentiler yayınlayarak uygulama geliştirmeyi kolaylaştırması ve Android geliştiriciler için birçok yapılandırma ayarlarını barındıran hazır paket platformu sunması nedeniyle Eclipse platformu üzerinde geliştirilmiştir. Eclipse üzerinde uygulama geliştirme sürecinde Java Tabanlı Android Yazılım Geliştirme Dili kullanılmıştır.

Uygulamanın test süreçleri, Android'in en stabil sürümü olan 2.2 Froyo işletim sistemine sahip Samsung I-8150 Galaxy Wonder Model cep telefonu üzerinde gerçekleştirilmiştir.

3.1.4. Süreç Yönetim Yazılımları

Mockup Builder aracılığı ile mobil platformlu sistemler ve web tabanlı sistemler için kolaylıkla prototip oluşturulabilmektedir[9]. Mobil platformlu tasarımları desteklemesi ve web üzerinden dosyalara her yerden ulaşma imkanı sağlaması nedeniyle tez kapsamında Mockup Builder programı kullanılarak ekran tasarımları oluşturulmuştur.

OnTime Scrum Planlama Programı, Yazılım Geliştirme Sürecini çevik yöntemlerle yönetmemizi sağlayan web tabanlı ücretli bir planlama ve yönetim aracıdır[10]. Web tabanlı yönetim ekranı bulunan OnTime Scrum Planlama Programına kayıt olunduktan sonra seçilen proje adı veya isme göre secilenisim.ontimenow.com uzantılı adres üzerinden web tabanlı olarak kullanılabilmektedir. Sistem üzerinde proje, proje süresi, Ürün Gereksinim Listesi, Koşu Gereksinim Listesi, Kullanıcı Hikayeleri ve Görev Kartları

8 http://tr.wikipedia.org/wiki/Eclipse_(yaz%C4%B1l%C4%B1m), [Ziyaret Tarihi: 10 Nisan 2014]
9 http://mockupbuilder.com/, [Ziyaret Tarihi: 10 Nisan 2014]
10 http://www.ontimenow.com/, [Ziyaret Tarihi: 10 Nisan 2014]

oluşturulabilmektedir. Proje sürecinde yer alan kişiler sisteme eklenerek, proje sürecini takip etmeleri sağlanabilmektedir. Eklenen kişiler üzerine iş atanabilmekte, işi tamamladıklarında sistem üzerinden onaylanabilmektedir.

Tez kapsamında, ücretsiz kullanım süresi tanıması ve çevik süreçleri desteklemesi nedeniyle OnTime Scrum Planlama Programı kullanılmıştır. Bundan dolayı kullanımı sırasında bazı kısıtlar oluşmuştur. Örneğin; Görev Kartları oluşturulmadan önce denemeler yapıldığı için Şekil. 3.9 ve 3.10 da görüldüğü gibi Görev Kartlarının sırası 32'den başlamıştır.

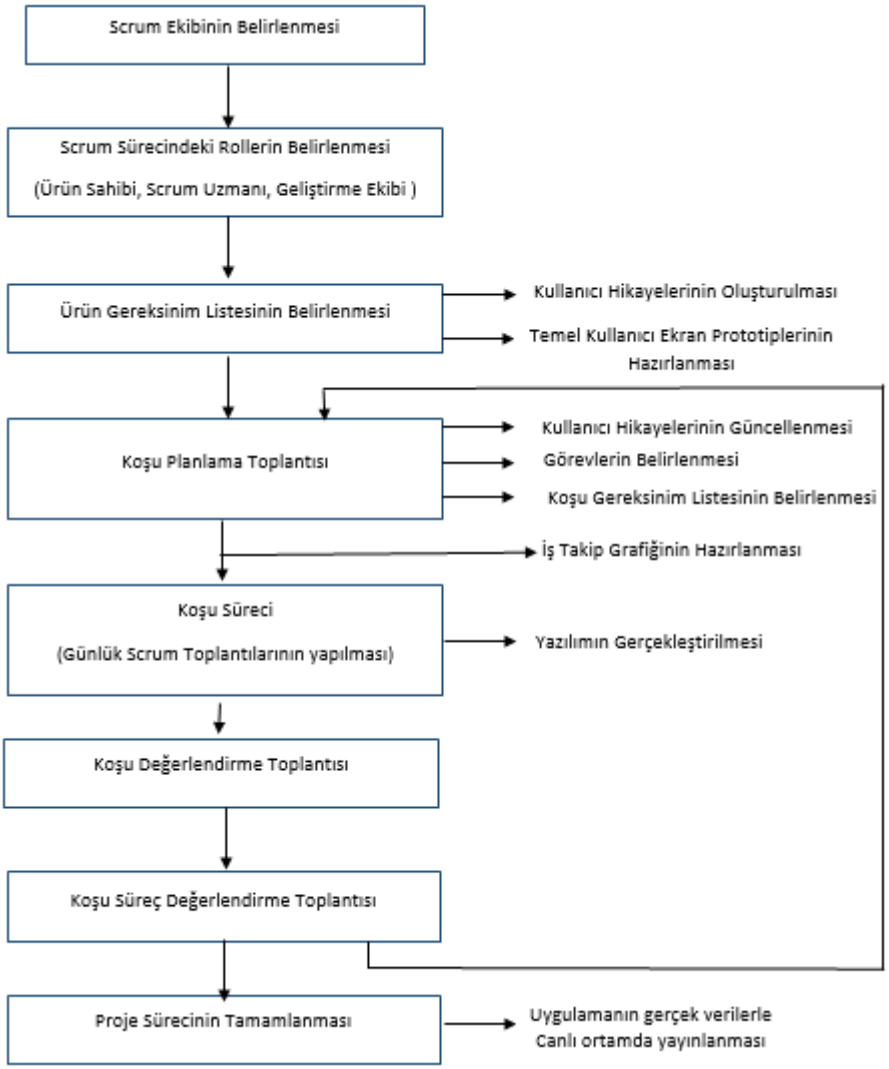
3.2. YÖNTEM

Tez çalışması kapsamında İstanbul Üniversitesi, İstanbul Tıp Fakültesi İlaç Araştırmaları Birimi için Android tabanlı bir mobil uygulama geliştirilmiştir. Bu alanda bir yazılıma ihtiyaç olduğu tespit edilerek proje süreci hakkında araştırmalara başlanmıştır. Yazılım geliştirme metodolojileri incelendiğinde, geleneksel yazılım geliştirme yöntemlerinin gelişen ve değişen teknolojiyle değişen müşteri isteklerine uyum sağlamakta ve mobil uygulama geliştirmede proje sürecinin sağlıklı ilerlemesinde sorunlar yaşandığı görülmüştür. Özellikle proje süreci başlarken ihtiyaçların tam tespit edilememesi geleneksel yazılım geliştirme metodolojilerinin proje süresince zayıf kaldığı noktalardan olmaktadır.

Gelişen ve değişen teknoloji nedeniyle ürün başında tespit edilen gereksinimlerin yanı sıra ürün ortaya çıktığı anda da gereksinimleri karşıladığı için proje süresince çevik metodolojilerin kullanılmasına karar verilmiştir. Bununla birlikte yazılım geliştirme metodları ve mobil uygulama geliştirme alanında yapılan araştırmalar da incelenmiştir. Yapılan çalışmalardan çevik yöntemler kullanan çalışmalar analiz edilerek, kullandıkları metodlar ortaya çıkarılmıştır.

Çevik Metodolojiler içinde kullanılan çeşitli metodlar olması nedeniyle bu süreçte takip edilecek çevik metod olarak; geliştirilecek mobil uygulama için yazılım geliştirme sürecinde müşteri ihtiyaçlarını en iyi analiz etmeyi sağlayan ve proje sürecinde herkesin aktif olmasına imkan veren Scrum metodu tercih edilerek, uygulama geliştirme sürecine

başlanmıştır. Bu süreçte Scrum metodunun özellikleri kullanılarak mobil uygulama geliştirme süreci tamamlanmıştır.



Şekil 3.2: Uygulama Geliştirme Proje Süreci.

Şekil 3.2'de görüldüğü gibi, Scrum metodu ile proje süreci başlarken ilk aşamada Scrum Ekibi oluşturulmuştur. Scrum Ekibinin oluşturulmasından sonra uygulama geliştirme sürecinde Ürün Sahibi, Scrum Uzmanı ve Geliştirme Ekibi'nin kimler olacağı belirlenmiştir. Bu aşamadan sonra Ürün Sahibi ile birlikte çalışılarak, Ürün Sahibinin ihtiyaçları ve analizleri değerlendirilmiş ve Ürün Gereksinim Listesi belirlenerek, temel kullanıcı ekran prototipleri hazırlanmıştır.

Sürekli güncellenen Ürün Gereksinim Listesinden öncelikli olarak hangi ihtiyaçların karşılanacağını belirlemek için Koşu Planlama Toplantıları gerçekleştirilmiş, Ürün

Sahibinin ihtiyaçlarını ifade eden Kullanıcı Hikayeleri oluşturulmuş, Kullanıcı Hikayelerine göre görevler oluşturularak Koşu Gereksinim Listesi belirlenmiştir. Koşu Gereksinim Listesinin belirlenmesinden sonra Scrum Uzmanı tarafından, yapılan işlerin takibini kolaylaştırmak ve zaman içindeki performansı görmek amacıyla İş Takip Grafiği hazırlanmıştır. Bu aşamalardan sonra Koşu Süreci başlatılmıştır. Ekibin kendi kendine organize olmasını sağlayan Günlük Scrum Toplantıları ile Koşu süreci desteklenerek yazılım ve uygulama gerçekleştirilmiştir. Bu süreç sonunda Koşu Değerlendirme Toplantısı yapılarak uygulamanın o ana kadar yapılan kısmı sunulmuş ve geri bildirimler alınmıştır. Koşu Süreç Değerlendirme Toplantısında, süreç içinde Geliştirme Ekibini olumlu veya olumsuz etkileyen yönler değerlendirilmiştir.

Koşu süreci yinelemeli olarak 3 kez gerçekleştirildikten sonra proje süreci tamamlanmış ve uygulama Google Markette yayınlanmıştır[11].

3.2.1. Proje Sürecindeki Scrum Rolleri

Sürecin başlangıç aşamasında İlaç Araştırmaları Birimi bölüm başkanı ve tez danışmanı olan Prof. Dr. Yağız ÜRESİN ile görüşmeler yapılmıştır. Yağız ÜRESİN Scrum sürecinde Ürün Sahibinin yöneticisi konumuna karşılık gelmiştir. Scrum Ekibi; tez öğrencisi Mehmet ELİBOL, tez danışmanı Yard. Doç. Dr. Çiğdem SELÇUKCAN EROL ve Ürün Sahibi olan Uzman Dr. Selçuk ŞEN'den oluşmuştur.

Ürün Sahibi rolüne; İstanbul Üniversitesi, İstanbul Tıp Fakültesi İlaç Araştırmaları biriminde Uzman Doktor olarak çalışan Selçuk ŞEN karşılık gelmektedir. Sürekli hastalarla iletişim halinde olarak hastaların nasıl bir ürüne sahip olması gerektiğini bildiğinden, Ürün Sahibi yöneticisi ile kesintisiz iletişim sağlayarak beklentilerin ve geri dönütlerin iletilmesinde önemli bir köprü oluşturmasından ve Scrum sürecinde kullanıcı ihtiyaçlarının en iyi şekilde Scrum Uzmanı ve Geliştirme Ekibine aktarılmasını sağlamasından dolayı Ürün Sahibi olarak seçilmiştir. Uygulama gereksinimlerinin öncelik listesi, toplantılar gerçekleştirilerek ve günümüz teknolojisi uzaktan haberleşme yöntemleri kullanılarak Ürün Sahibi ile görüşmeler neticesinde ortaya çıkarılmıştır.

Scrum Uzmanı rolüne ; tez öğrencisi karşılık gelmektedir. Scrum Uzmanı rolündeki tez öğrencisi endüstriyel bir bakış açısı sunarak süreçlerin, yazılımın kalitesini vurgulamıştır

[11] https://play.google.com/store/apps/details?id=com.itp.klinikuygulama

ve proje süresince ekip lideri olmamasından kaynaklanan başarısızlıkların önüne geçilmesini sağlamıştır.

Geliştirme Ekibi rolü; Scrum süresince yazılım takımlarının en az 3 kişiden oluşması gerekmektedir. Ancak tezin sınırlarından biri olarak uygulama sadece tezi hazırlayan öğrenci tarafından gerçekleştirilmiştir. Tez danışmanları süreç içerisinde zaman zaman Geliştirme Ekibinden birisi rolüne bürünerek ortaya çıkan engellerin aşılmasını sağlamış ve yazılımın ortaya çıkması sürecinde rol oynamıştır.

3.2.2. Ürün Gereksinim Listesi

Ürün Gereksinim Listesi, 4 Mart 2014 Salı günü gerçekleştirilen toplantı ile belirlenmiştir. Toplantı süresince Ürün Sahibi ihtiyaçlarını önceden hazırladığı dokümantasyon üzerinde anlatmıştır. Bu ihtiyaçlara göre Scrum Uzmanı ve Geliştirme Ekibi istenilen ürün ve özellikler hakkında notlar almıştır. Ürün sahibi ile birlikte kağıt üzerinde istenilen yazılımın bazı ekran tasarımları taslak olması açısından çizilmiştir.

Ürün Sahibi, Klinik Araştırmalar hakkında, kişileri bir mobil uygulama üzerinden bilgilendirmek istemektedir. Bu geliştirilecek olan mobil uygulamayı kullanan kişinin, mobil uygulama aracılığı ile Klinik Araştırma konuları hakkında bilgi alabilmesi, öğrendiği bilgileri test edebilmesi ve sık sorulan sorulara bu mobil uygulama üzerinden ulaşabilmesini kapsamaktadır.

Ürün Sahibi istediği ürünü genel olarak ifade ettikten sonra, detay özelliklerini anlatmıştır. Detay özelliklerini anlatması sırasında Scrum Uzmanı ve Geliştirme Ekibi sorular sorarak Kullanıcı Hikayelerinin oluşturulmasını sağlamıştır.

Kullanıcı Hikayeleri, uygulama geliştirme sürecinde güncellenerek, Koşu Gereksinim Listesine aktarılmıştır. Her Koşu sonunda güncellenen Ürün Gereksinim Listesine tekrar yer verilmemiştir. Bunun nedeni Koşu için hazırlanan ve güncellenen Kullanıcı Hikayelerinin Ürün Gereksinim Listesinin son güncellenmiş halinden seçilmiş olmasıdır. Bu kısımda Kullanıcı anlatımından oluşturulan ilk Kullanıcı Hikayelerine yer verilmiştir;

- Kullanıcı, uygulamayı ilk açtığında İstanbul Tıp Fakültesi yazan bir karşılama ekranı ile karşılanmalıdır.

- Konuları öğrenebileceği ekranda kullanıcı konuların bir listesini görmeli ve bu konular hakkında ayrıntılı bilgi almak için konu başlığına göre hareket edebilmelidir.
- Herhangi bir konuyu seçtiğinde o konuyla ilgili detaylı bilgi alabilmelidir.
- Uygulamada konuları öğrendiği bilgileri test etmek için soruların yer aldığı bir ekran yer almalıdır.
- Kullanıcı, herhangi bir soruyu seçtiğinde o sorunun detayı ve o soruyla ilgili seçenekler gelmelidir.
- Kullanıcı, herhangi bir soruda o soruyu doğru ya da yanlış yaptığını anında görebilmelidir.
- Uygulamada sık sorulan soruların olduğu bir ekran yer almalıdır.
- Kullanıcı, sık sorulan sorulardan birisini seçtiğinde hemen cevabını alabilmelidir.

3.2.3. Kullanıcı Ekran Prototiplerinin Hazırlanması

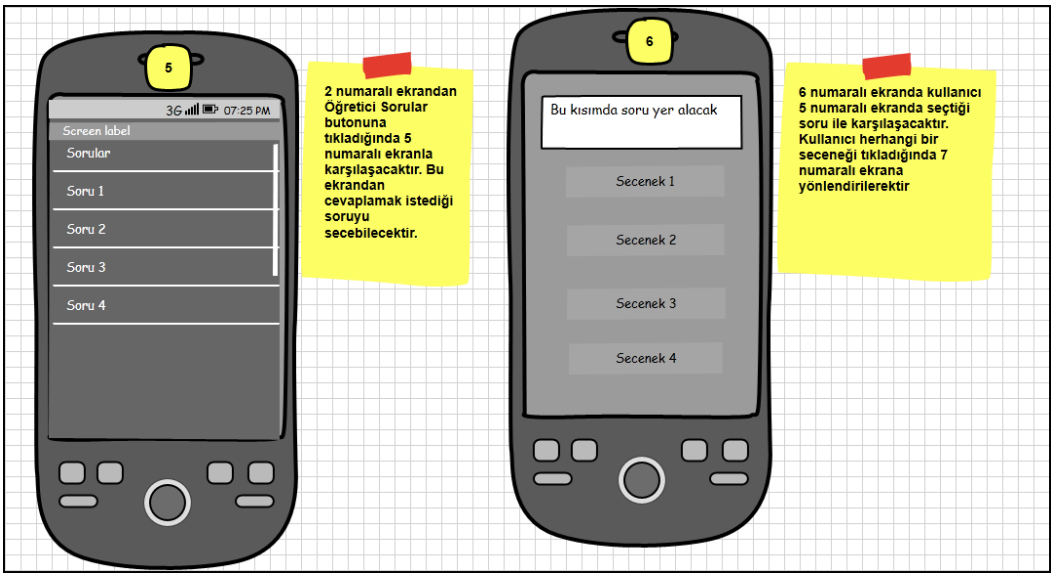
Ürün sahibinin bu isteklerine göre Geliştirme Ekibi tarafından MockUp Builder aracı kullanılarak yazılımın tasarımsal prototipi hazırlanmıştır. Bu prototipler hazırlanırken kullanıcının isteklerini en iyi temsil edecek şekilde hazırlanarak kullanıcının proje süreci başlarken ürünü net bir şekilde tasarımsal olarak görmesi sağlanmıştır. Her ne kadar ihtiyaçlar proje süresince değişecek olsa da, ürün hakkında eksikler veya yanlış anlaşılmalar ilk aşamada tespit edilerek proje sürecinin sağlıklı şekilde başlaması sağlanmıştır.



Şekil 3.3: Kullanıcı Karşılama ve Yönlendirme Ekranı Prototipleri.



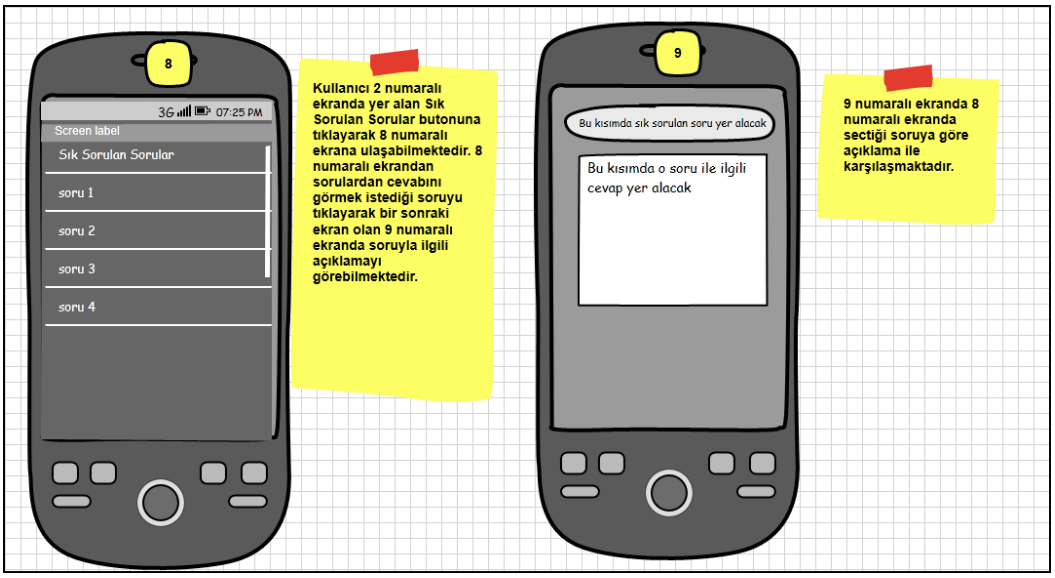
Şekil 3.4: Konuları Listeleme ve Konu Detay Ekranı Prototipleri.



Şekil 3.5: Öğretici Sorular Listeleme ve Öğretici Soru Cevaplama Ekran Prototipleri.



Şekil 3.6: Öğretici Soru Geri Bildirim Ekran Prototipi.



Şekil 3.7: Sık Sorulan Soruları Listeleme ve Sık Sorulan Soru Detay Ekran Prototipleri.

Kullanıcının, ilk giriş yaptığında karşılaşacağı ekran ve devam butonuna tıkladıktan sonra uygulama içinde seçimler yapabileceği ekran prototipleri Şekil 3.3'de, konuları seçebileceği ve konular hakkında detaylı bilgileri görebileceği ekran prototipleri Şekil 3.4'de, cevaplamak istediği soruları seçebileceği ve seçtiği soruyu cevaplandırabileceği ekran prototipleri Şekil 3.5'de, cevaplamak istediği bir soruda seçtiği seçenek ile ilgili geri bildirim aldığı ekran prototipi Şekil 3.6'da, sık sorulan sorular kısmından sık sorulan sorular listesine ulaşabileceği ve sık sorulan sorularla ilgili cevapları görebileceği ekran prototipleri ise Şekil 3.7'de görülmektedir.

3.2.4. Scrum Planlama

Koşu Planlama Toplantılarında çıkarılan görevler Scrum Uzmanı tarafından Scrum Planlama Programı üzerinde detaylandırılmıştır ve görev kartları şeklinde oluşturulmuştur. Scrum Planlama Programı Üzerinde bir proje ismi belirlenerek, bu proje içeriğinde yapılması gereken görevler Koşu Planlama Toplantılarına göre Scrum Uzmanı tarafından oluşturulmuş ve ekip üyelerinin üzerine atanmıştır. Bu süreç Koşu Planlama Toplantısı ve Koşu İçerikleri kısmında detaylandırılmıştır.

3.2.5. Birinci Koşu Planlama Toplantısı ve Birinci Koşu Gereksinim Listesi

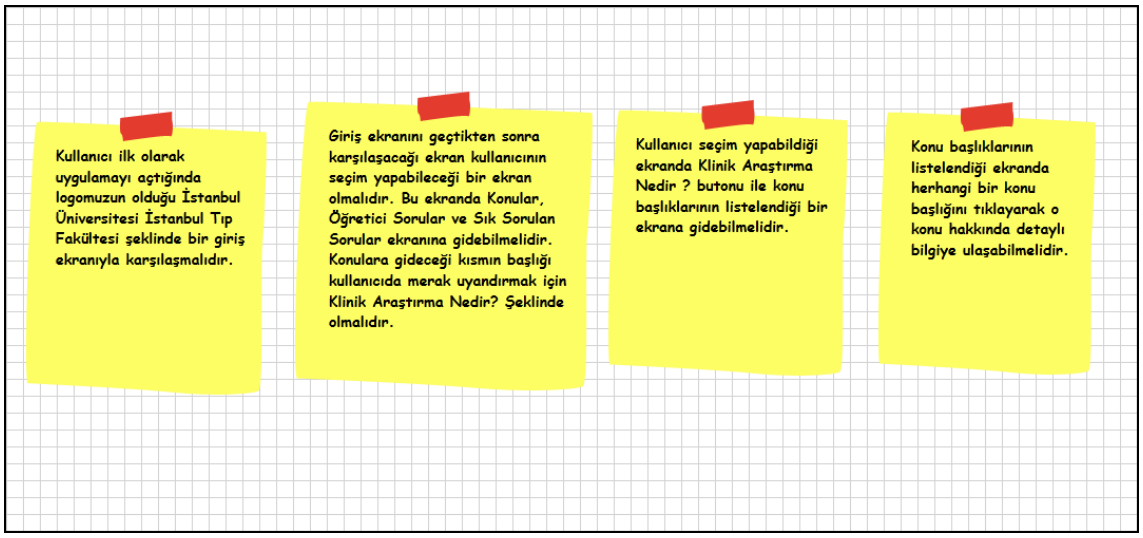
Koşu Gereksinim Listesi 10 Mart 2014 günü gerçekleştirilen toplantı ile belirlenmiştir. Bu toplantıda Koşu süresi olarak 15 gün belirlenmiş ve süreçte yapılacak işler planlanmıştır.

Bu toplantı öncesi Scrum Uzmanı ve Geliştirme Ekibi Ürün Gereksinim Listesi üzerinde çalışarak Koşu Planlama Toplantısı öncesi fikir sahibi olmuşlardır.

Ürün sahibi Kullanıcı Karşılama, Yönlendirme (Şekil 3.3) , Konuları Listeleme ve Konu Detay (Şekil 3.4) ekranlarını daha detaylandırarak anlatmıştır. Bu anlatım ile Birinci Koşu için ilk aşamada oluşturulan Ürün Gereksinim Listesindeki Kullanıcı Hikayeleri güncellenmiş ve bu hikayelerden görevler çıkarılmıştır.

Ürün Sahibi istediği özellikleri belirten Kullanıcı Hikayeleri aşağıdaki gibidir;

- Kullanıcı uygulamayı açtığında ilk olarak logonun yer aldığı İstanbul Üniversitesi İstanbul Tıp Fakültesi şeklinde bir giriş ekranıyla karşılaşmalıdır.
- Giriş ekranını geçtikten sonra karşılaşacağı ekran kullanıcının seçim yapabileceği bir ekran olmalıdır. Bu ekrandan Konular, Öğretici Sorular ve Sık Sorulan Sorular ekranına gidebilmelidir. Konulara gideceği kısmın başlığı kullanıcı da merak uyandırmak için “Klinik Araştırma Nedir?” şeklinde olmalıdır.
- Kullanıcı seçim yapabildiği ekranda “Klinik Araştırma Nedir?” butonu ile konu başlıklarının listelendiği bir ekrana gidebilmelidir.
- Konu başlıklarının listelendiği ekrandan herhangi bir konu başlığını tıklayarak o konu hakkında detaylı bilgiye ulaşabilmelidir.



Şekil 3.8: Birinci Koşu için Kullanıcı Hikayeleri.

Birinci Koşu için, Kullanıcı Hikayeleri ile oluşturulmuş Kullanıcı Hikaye Kartları Şekil 3.8'de görüldüğü şekildedir.

Kullanıcı Hikayelerini gerçekleştirmek için yapılması gereken görevler ve bu görevlerin ne kadar birimde[12] gerçekleştirileceği aşağıdaki gibi planlanmıştır;

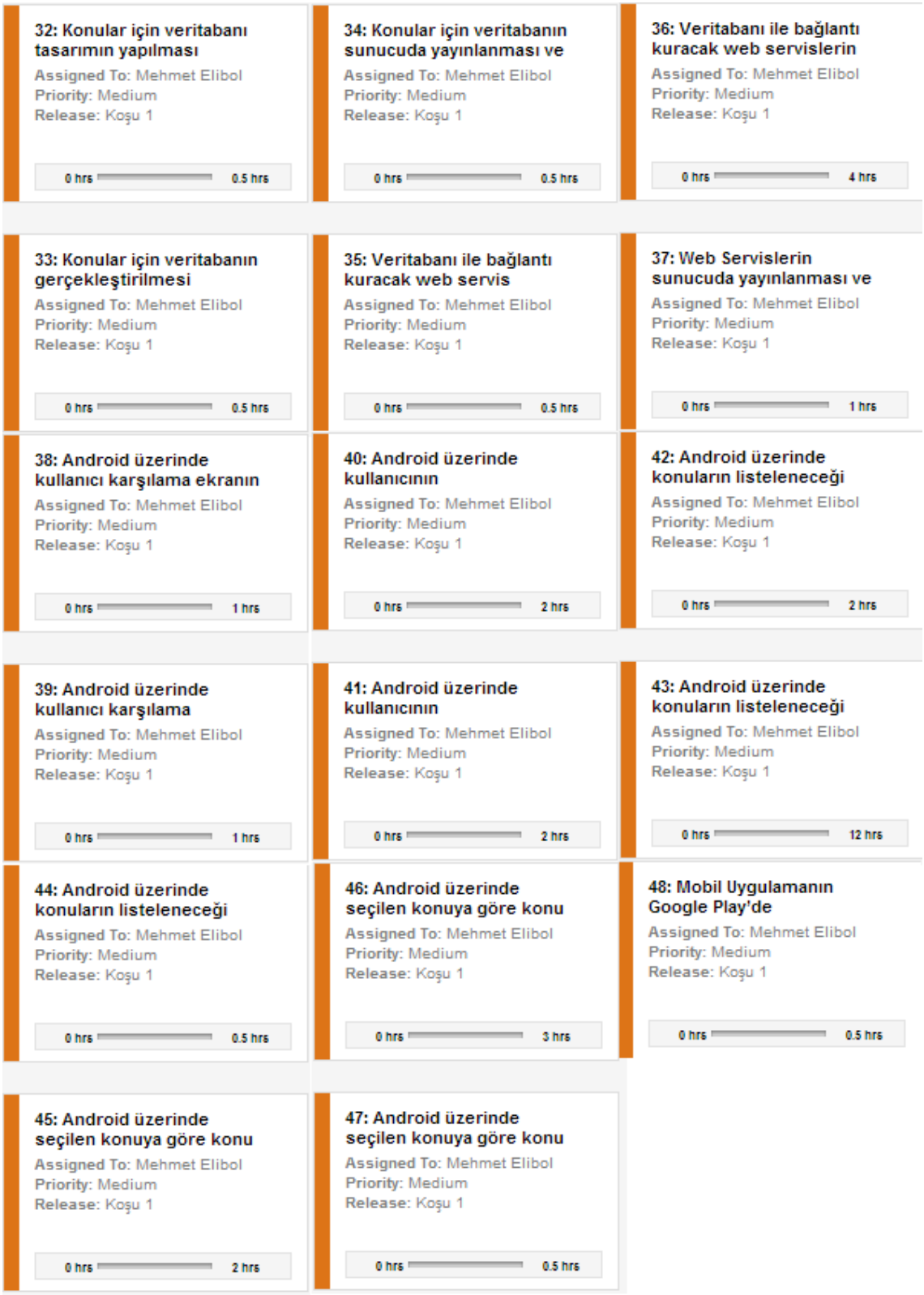
- Konular için veri tabanı tasarımının yapılması / 0,5 saat
- Konular için veri tabanının gerçekleştirilmesi/ 0,5 saat
- Konular için veri tabanının sunucuda yayınlanması ve test edilmesi / 0, 5 saat
- Veri tabanı ile bağlantı kuracak web servis tasarımının hazırlanması / 0,5 saat
- Veri tabanı ile bağlantı kuracak web servislerin kodlanması / 4 saat
- Web Servislerin sunucuda yayınlanması ve test edilmesi / 1 saat
- Android üzerinde kullanıcı karşılama ekranın tasarlanması / 1 saat
- Android üzerinde kullanıcı karşılama ekranının kodlanması ve test edilmesi / 1 saat
- Android üzerinde kullanıcının yönlendirileceği ekranın tasarlanması / 2 saat
- Android üzerinde kullanıcının yönlendirileceği ekranın kodlanması ve test edilmesi / 2 saat
- Android üzerinde konuların listeleneceği ekranın tasarlanması / 2 saat

[12] Koşu süresince iş hacmi birim kavramı ile ifade edilmektedir. Tez süresince birim kavramına karşılık olarak iş, saat olarak ifade edilmiştir.

- Android üzerinde konuların listeleneceği ekranın kodlanması / 12 saat
- Android üzerinde konuların listeleneceği ekranın test edilmesi / 0,5 saat
- Android üzerinde seçilen konuya göre konu detayının geleceği ekranın tasarlanması / 2 saat
- Android üzerinde seçilen konuya göre konu detayının geleceği ekranın kodlanması / 3 saat
- Android üzerinde seçilen konuya göre konu detayının geleceği ekranın test edilmesi / 0, 5 saat
- Mobil Uygulamanın Google Marktte yayınlanması / 0,5 saat

Kullanıcı Hikayelerine göre görevlerin oluşturulması sırasında 33,5 birimlik (saat) iş yükü oluştuğunda Scrum Uzmanı ve Geliştirme Ekibi bu süreç için bu kadar görevi gerçekleştirebileceklerini belirterek bu Koşu için görev dağıtımı tamamlanmıştır.

Koşu Planlama toplantısı sonrası gerçekleştirilmesi gereken görevler yukarıdaki şekilde belirlendikten sonra Görev Kartları (Şekil 3.9) ve Görev Listesi (Şekil 3.10) hazırlanmıştır.



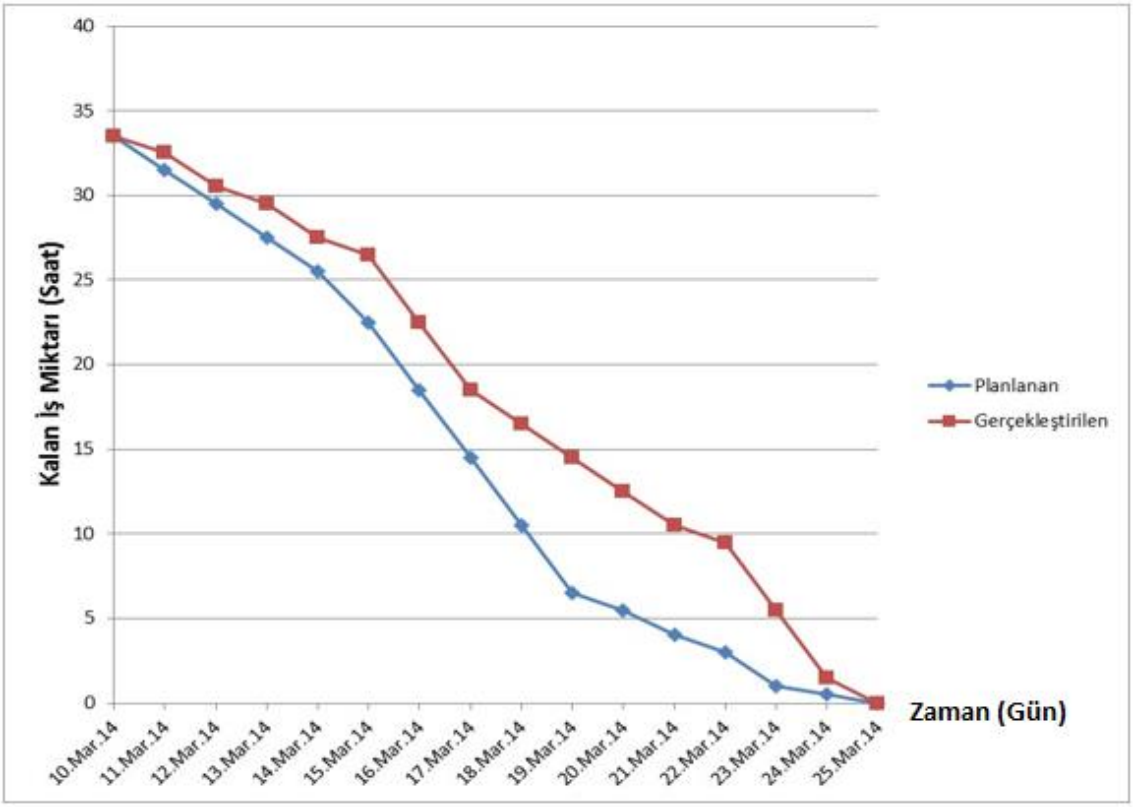
Şekil 3.9: Koşu Boyunca gerçekleştirilen Görev Kartları.

ID	Title	Start Date	Original Estimat
32	Konular için veritabanı tasarımın yapılması	10/03/2014	0.5 hrs
33	Konular için veritabanın gerçekleştirilmesi	10/03/2014	0.5 hrs
34	Konular için veritabanın sunucuda yayınlanması ve test edilmesi	10/03/2014	0.5 hrs
35	Veritabanı ile bağlantı kuracak web servis tasarımının hazırlanması	10/03/2014	0.5 hrs
36	Veritabanı ile bağlantı kuracak web servislerin kodlanması	11/03/2014	4 hrs
37	Web Servislerin sunucuda yayınlanması ve test edilmesi	13/03/2014	1 hrs
38	Android üzerinde kullanıcı karşılama ekranın tasarlanması	13/03/2014	1 hrs
39	Android üzerinde kullanıcı karşılama ekranının kodlanması ve test edilmesi	14/03/2014	1 hrs
40	Android üzerinde kullanıcının yönlendirileceği ekranın tasarlanması	14/03/2014	2 hrs
41	Android üzerinde kullanıcının yönlendirileceği ekranın kodlanması ve test edil	15/03/2014	2 hrs
42	Android üzerinde konuların listeleneceği ekranın tasarlanması	15/03/2014	2 hrs
43	Android üzerinde konuların listeleneceği ekranın kodlanması	16/03/2014	12 hrs
44	Android üzerinde konuların listeleneceği ekranın test edilmesi	19/03/2014	0.5 hrs
45	Android üzerinde seçilen konuya göre konu detayının geleceği ekranın tasarla	19/03/2014	2 hrs
46	Android üzerinde seçilen konuya göre konu detayının geleceği ekranın kodlar	21/03/2014	3 hrs
47	Android üzerinde seçilen konuya göre konu detayının geleceği ekranın test ed	23/03/2014	0.5 hrs

Şekil 3.10: Koşu Boyunca gerçekleştirilen Görev Listesi.

3.2.6. Scrum Uzmanı tarafından İş Takip Grafiğinin Hazırlanması

Scrum Uzmanı tarafından Koşu sırasında işlerin ne kadarının yapıldığının rahat bir şekilde takip edilebilmesi için Koşu başlangıcında İş Takip Grafiği oluşturulmuş ve Geliştirme Ekibi ile paylaşılmıştır. Bu Grafik üzerinde saat bazında yapılan işler çizilerek zamanlama hedefi ile ne kadar doğru orantılı olarak gidildiğinin görülmesi sağlanmıştır (Şekil 3.11).



Şekil 3.11: Birinci Koşu sırasında takip edilen İş Takip Grafiği.

İlk grafik oluşturulduğunda sadece Planlanan çizgi grafiği bulunmaktadır. Süreç içinde Geliştirme Ekibi işleri tamamladıkça gerçekleştirilen iş birimi kadar iş grafik üzerinde aşağıya yönlü bir ikinci çizgi oluşmaktadır. Bu o ana kadar gerçekleştirilen iş birimini temsil etmekte ve o andan sonra kalan işlerin görünür olmasını sağlamaktadır. Tez süresince grafikte de görüldüğü gibi hafta sonları da çalışılmıştır. Scrum süresince bu istenilen bir durum değildir ama tez öğrencisinin zaman kısıtları nedeniyle bu şekilde gerçekleştirilmiştir.

3.2.7. Birinci Koşu Süreci

Birinci Koşu süreci 10 Mart 2014 Pazartesi günü gerçekleştirilen Koşu Planlama toplantısı ile başlayarak 24 Mart 2014 Pazartesi gününe kadar 15 günlük sürede gerçekleştirilmiştir.

Scrum Uzmanı tarafından ekibin kendi kendini yönetebilmesi için İş Takip Grafiği hazırlanmıştır. Bu grafik ile Koşu süresince ekip, işlerin ne kadarını yaptığını zamana göre takip edebilmişlerdir.

Bu süreçte konular için veri tabanı tasarımı gerçekleştirilmiş ve sunucuya aktarılmıştır. Sunucu üzerindeki veri tabanına test verileri girilerek veri tabanı test edilmiş ve web servislerin oluşturulması için hazır hale getirilmiştir.

Visual Studio üzerinde Asp.Net ve C# programlama dili kullanılarak web servisler oluşturulmuştur ve oluşturulan servis yazılımı sunucuya atılarak test edilmiştir.

Ayrıca Eclipse uygulama geliştirme platformu üzerinde uygulamanın en düşük hangi Android sürümü üzerinde çalışabileceği özellikleri belirlenerek uygulama geliştirme ortamı hazırlanmıştır.

Uygulama geliştirme ortamının hazırlanmasından sonra yazılım geliştirme aşamasına geçilmiştir ve canlı ortamda çalışan servisler aracılığı ile uygulama geliştirme süreci başlatılmıştır.

Uygulama geliştirme aşamasında test süreçlerinin sağlıklı şekilde gerçekleştirilebilmesi için uygulama farklı model cihazlarda test edilerek geliştirme aşaması sürdürülmüştür. Bu şekilde birçok farklı mobil cihaz üzerinde uygulamanın nasıl çalıştığı test edilmiş ve gerekli değişiklikler yapılmıştır.

Yazılım geliştirme sürecinde, projenin ileriki aşamalarında yazılımcıların kodları yönetmesinde esneklik kazandıran ve zamanı etkin şekilde kullanmalarını sağlayan yazılımsal tekrar kullanılabilir yapılar oluşturulmuştur. Örneğin servisten veriyi çekme mantığının sınıf yapısı ile oluşturulmasıyla, yapının tekrar kullanılabilir olması sağlanmıştır.

Birinci Koşu Sırasında ortaya çıkan bazı önemli noktalar;

- Ürün Sahibinden gelen dönütler sonucunda yönlendirme ekranında İletişim ve Uygulama Hakkında seçeneklerinin olmasına karar verilmiştir. Koşu süresince ekstra iş yükü getirmediği için bu seçeneklerin tasarımsal olarak yönlendirme ekranına konulması gerçekleştirilmiş ama kodlama aşaması diğer Koşu süreçlerine bırakılmıştır.
- Uygulama giriş ekranında ilk seçilen logo yerine farklı bir logo kullanılmasına karar verilmiştir ve bunun ürün sahibi tarafından sağlanmasına karar verilmiştir.

Birinci Koşu Süresince geliştirilen mobil uygulamanın ekran görüntüleri Şekil 3.12 ve Şekil 3.13 deki gibi oluşmuştur.



Şekil 3.12: Birinci Koşu sonucunda oluşan Kullanıcı Karşılama ve Yönlendirme Ekranları.



Şekil 3.13: Birinci Koşu sonucunda oluşan Konuları Listeleme ve Konu Detay Ekranları.

3.2.8. Birinci Koşu Süresince Günlük Scrum Toplantıları

Günlük Scrum Toplantılarında ana hedef Geliştirme Ekibinin kendi içinde değerlendirme yapması şeklinde gerçekleştiği ve tez süresince Geliştirme Ekibini, genellikle tez öğrencisi temsil ettiği için bu toplantıların gerçekleştirilmesinde kısıtlar yaşanmıştır. Bazı

zamanlarda Günlük Scrum Toplantılarına Ürün Sahibi katılmıştır. Ürün Sahibinin bu toplantılara katılması sürecin daha verimli geçmesini sağlamıştır. Toplantılarda bir önceki Günlük Scrum Toplantısından sonra ne yapıldığı değerlendirilmiş, bir sonraki toplantıya kadar ne yapılacağı belirlenmiş ve bu yapılacaklar konusunda herhangi bir engel var mı, varsa bunlar nelerdir sorularına cevap aranmıştır.

Proje sürecinde Günlük Scrum Toplantıları gerek yüz yüze olsun, gerek iletişim teknolojilerinin getirdiği yeniliklerden yararlanılarak çeşitli yöntemlerle gerçekleştirilmeye çalışılmıştır. Ancak Günlük Scrum Toplantılarının bazıları gerçekleştirilememiştir. Birkaç toplantıda önemli bir nokta olan yüz yüze iletişim kaybı yaşanmıştır.

Yüz yüze gerçekleştirilmeyen toplantılar Google Talk üzerinden web cam ile görüntülü olarak veya telefonla konuşularak gerçekleştirilmiştir.

Bu toplantılar ekibin ne yaptığını görmesini sağlarken, aynı zamanda ekibin sürekli işi takip etmesi konusunda motivasyonda kalmasını sağlamıştır.

3.2.9. Birinci Koşu Değerlendirme Toplantısı

24 Mart 2014 Pazartesi günü gerçekleştirilen bu toplantı Ürün Sahibi, Scrum Uzmanı ve Geliştirme Ekibinin katılımıyla gerçekleştirilmiştir. Geliştirme Ekibi tarafından uygulamanın o ana kadar tamamlanmış kısmının sunumu yapılmış ve Ürün Sahibinden dönütler alınmıştır.

Ürün Sahibinden alınan geri bildirimler şunlardır;

- Uygulamanın ilk açıldığı ekranda, logo, belirli bir süre (3-5 saniye kadar) ekranda kaldıktan sonra Devam butonuna gerek kalmadan yönlendirme ekranına geçilebilmelidir.
- Yönlendirme ekranında “Klinik Araştırma Nedir?” butonu yerine farklı bir isimlendirme yapılmalıdır. Bu süreç içinde karar verilip bildirilecektir.
- Süreç içerisinde de belirtildiği gibi Kullanıcı Karşılama Ekranındaki logo değiştirilmelidir.

Bu toplantı sırasında bir sonraki Koşu Planlama Toplantısına hazırlık olması ve ekibin hazırlık yapması amacıyla sonraki koşuda ne yapılması gerektiği üzerinde fikir alışverişinde de bulunulmuştur.

3.2.10. Birinci Koşu Süreç Değerlendirme Toplantısı

24 Mart 2014 Pazartesi günü gerçekleştirilen bu toplantı, Scrum Uzmanı ve Geliştirme Ekibinin süreç içinde neleri iyi yaptıkları ya da neler hakkında daha dikkatli olmaları gerektiği konusunda kendilerini değerlendirdiği bir toplantı olarak gerçekleştirilmiştir.

Bu toplantı sırasında üzerinde durulan bazı noktalar aşağıdaki gibi şekillenmiştir;

- Zaman Yönetimi konusunda ekibin daha dikkatli olması gerektiği üzerinde durulmuştur.
- Bir önceki sürecin maksimum verimlilikte geçmesi sağlandığı için, bu performansın devam etmesi gerektiği üzerinde durulmuştur.
- Tasarım olarak ilk süreç başarılı bulunmuş ve bu şekilde devam edilmesinin ekibe katkı sağlayacağı görüşü üzerinde durulmuştur.

3.2.11. İkinci Koşu Planlama Toplantısı ve İkinci Koşu Gereksinim Listesi

Birinci Koşu Değerlendirme Toplantısı sırasında fikir alışverişinde bulunan Scrum Ekibi 25 Mart 2014 Salı günü gerçekleştirilen İkinci Koşu Planlama Toplantısına Sık Sorulan Sorular kısmı için hazırlıklı olarak gelmiştir. Bu toplantı internet üzerinden Google Talk aracılığıyla gerçekleştirilmiştir. Bu toplantı ile Koşu Süresi 8 gün olarak belirlenmiş ve süreçte yapılacak işler planlanmıştır.

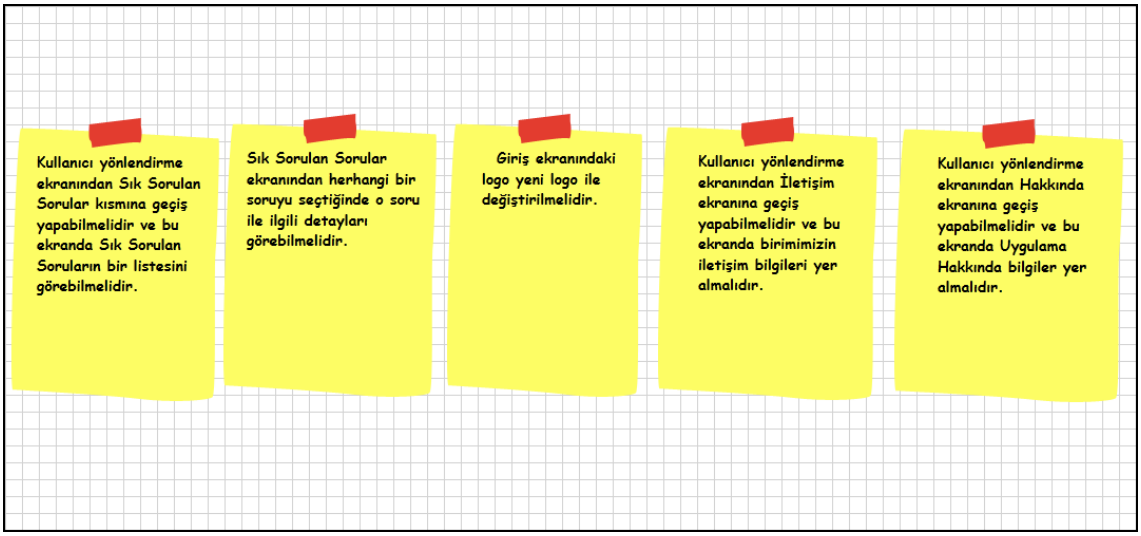
Ürün sahibi Sık Sorulan Soruları Listeleme ve Sık Sorulan Soru Detay (Şekil 3.7) ekran prototiplerini biraz daha detaylandırarak anlatmıştır. Bu anlatım ile İkinci Koşu için Kullanıcı Hikayeleri güncellenmiş ve bu hikayelerden görevler çıkarılmıştır.

Ürün Sahibinin istediği özellikleri belirten kullanıcı hikayeleri aşağıdaki gibidir;

- Kullanıcı yönlendirme ekranından Sık Sorulan Sorular kısmına geçiş yapabilmelidir ve bu ekranda Sık Sorulan Soruların bir listesini görebilmelidir.
- Sık Sorulan Sorular ekranından herhangi bir soru seçildiğinde o soru ile ilgili detaylar görülebilmelidir.

- Giriş ekranındaki logo yeni logo ile değiştirilmelidir.
- Kullanıcı yönlendirme ekranından İletişim ekranına geçiş yapılabilmelidir ve bu ekranda birim iletişim bilgileri yer almalıdır.
- Kullanıcı yönlendirme ekranından Hakkında ekranına geçiş yapılabilmelidir ve bu ekranda Uygulama Hakkında bilgiler yer almalıdır.

İkinci Koşu için, Kullanıcı Hikayeleri ile oluşturulan Kullanıcı Hikaye Kartları Şekil 3.14 de görüldüğü şekildedir.



Şekil 3.14: İkinci Koşu için Kullanıcı Hikaye Kartları.

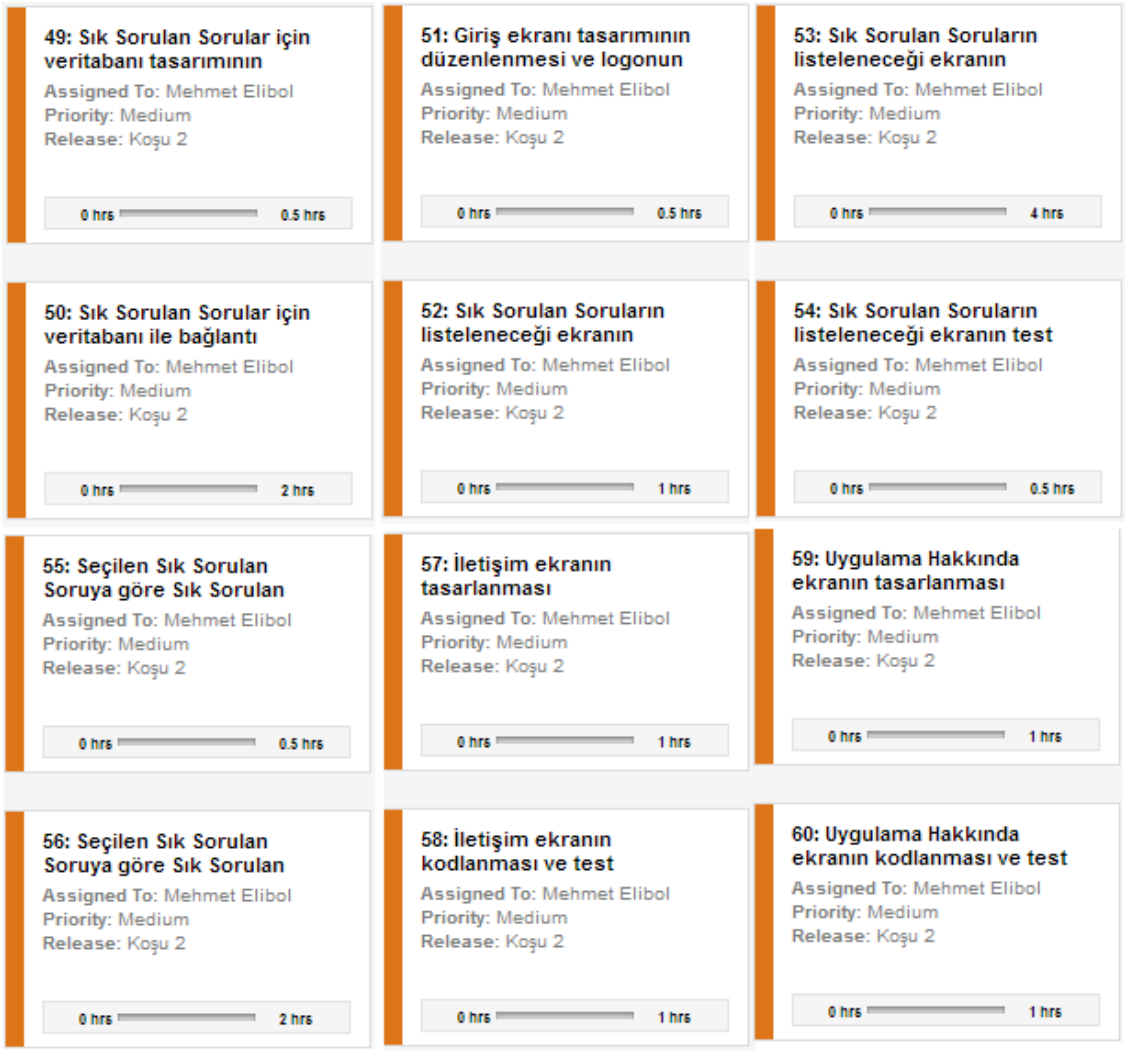
Kullanıcı Hikayelerini gerçekleştirmek için yapılması gereken görevler ve bu görevlerin ne kadar birimde gerçekleştirileceği aşağıdaki şekildedir;

- Sık Sorulan Sorular için veri tabanı tasarımının yapılması ve gerçekleştirilmesi / 0,5 saat
- Sık Sorulan Sorular için veri tabanı ile bağlantı kuracak web servislerin tasarlanması ve kodlanması / 2 saat
- Giriş ekranı tasarımının düzenlenmesi ve logonun değiştirilmesi / 0,5 saat
- Sık Sorulan Soruların listeleneceği ekranın tasarlanması / 1 saat
- Sık Sorulan Soruların listeleneceği ekranın kodlanması / 4 saat
- Sık Sorulan Soruların listeleneceği ekranın test edilmesi / 0,5 saat
- Seçilen Sık Sorulan Soruya göre Sık Sorulan Soru detayının geleceği ekranın tasarlanması / 0,5 saat

- Seçilen Sık Sorulan Soruya göre Sık Sorulan Soru detayının geleceği ekranın kodlanması ve test edilmesi / 2 saat
- İletişim ekranının tasarlanması / 1 saat
- İletişim ekranının kodlanması ve test edilmesi / 1 saat
- Uygulama Hakkında ekranının tasarlanması / 1 saat
- Uygulama Hakkında ekranının kodlanması ve test edilmesi / 1 saat

Kullanıcı Hikayelerine göre görevlerin oluşturulması sırasında 15 birimlik (saat) iş yükü oluştuğunda Scrum Uzmanı ve Geliştirme Ekibi bu süreç için bu kadar görevi gerçekleştirebileceklerini belirterek bu Koşu için görev dağıtımı tamamlanmıştır.

İkinci Koşu Planlama toplantısı sonrası gerçekleştirilmesi gereken görevler yukarıdaki şekilde belirlendikten sonra Görev Kartları (Şekil 3.15) ve Görev Listesi (Şekil 3.16) hazırlanmıştır.



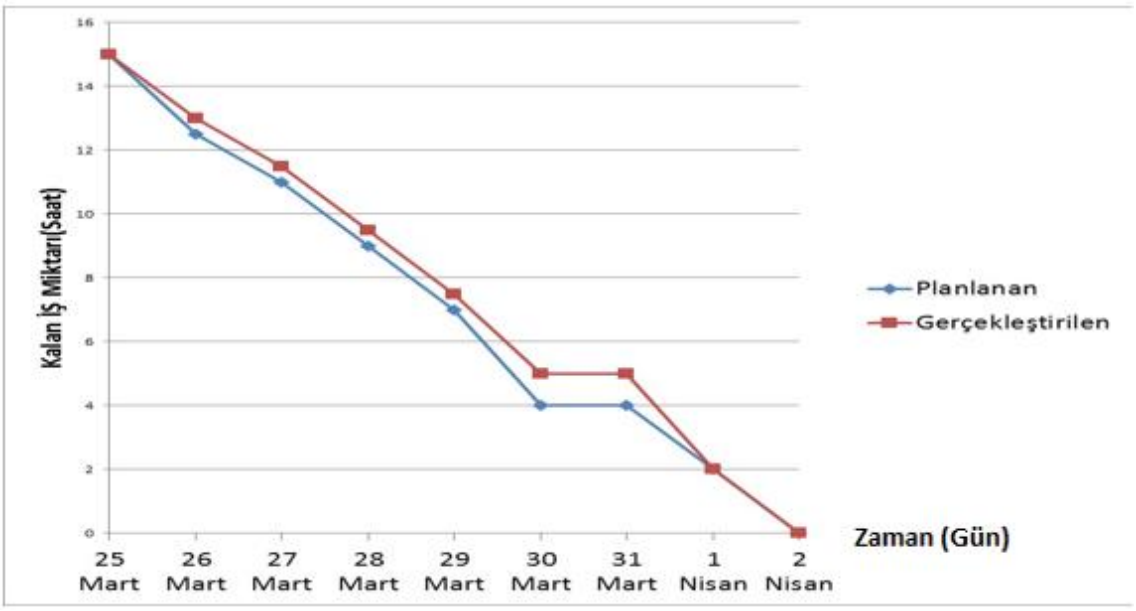
Şekil 3.15: İkinci Koşu boyunca gerçekleştirilen Görev Kartları.

ID	Title	Start Date	Original Estimat
49	Sık Sorulan Sorular için veritabanı tasarımının yapılması ve gerçekleştirilmesi	25/03/2014	0.5 hrs
50	Sık Sorulan Sorular için veritabanı ile bağlantı kuracak web servislerin tasarla	25/03/2014	2 hrs
51	Giriş ekranı tasarımının düzenlenmesi ve logonun değiştirilmesi	26/03/2014	0.5 hrs
52	Sık Sorulan Soruların listeleneceği ekranın tasarlanması	26/03/2014	1 hrs
53	Sık Sorulan Soruların listeleneceği ekranın kodlanması	27/03/2014	4 hrs
54	Sık Sorulan Soruların listeleneceği ekranın test edilmesi	29/03/2014	0.5 hrs
55	Seçilen Sık Sorulan Soruya göre Sık Sorulan Soru detayının geleceği ekranın	29/03/2014	0.5 hrs
56	Seçilen Sık Sorulan Soruya göre Sık Sorulan Soru detayının geleceği ekranın	29/03/2014	2 hrs
57	İletişim ekranın tasarlanması	31/03/2014	1 hrs
58	İletişim ekranın kodlanması ve test edilmesi	31/03/2014	1 hrs
59	Uygulama Hakkında ekranın tasarlanması	01/04/2014	1 hrs
60	Uygulama Hakkında ekranın kodlanması ve test edilmesi	01/04/2014	1 hrs

Şekil 3.16: İkinci Koşu boyunca gerçekleştirilen Görev Listesi.

3.2.12. Scrum Uzmanı Tarafından İkinci Koşu için İş Takip Grafiğinin Hazırlanması

Scrum Uzmanı tarafından İkinci Koşu sırasında işlerin ne kadarının yapıldığının rahat bir şekilde takip edilebilmesi için İkinci Koşu başlangıcında İş Takip Grafiği oluşturulmuş ve Geliştirme Ekibi ile paylaşılmıştır. İlk aşamada sadece Planlanan çizginin yer aldığı grafik üzerine, işler gerçekleştirildikçe, gerçekleştirilen iş birimleri çizilerek süreç sonunda Gerçekleştirilen çizginin oluşması sağlanmıştır. Bu şekilde grafik üzerinde saat bazında yapılan işler çizilerek zamanlama hedefi ile ne kadar doğru orantılı olarak gidildiği görülebilmiştir. Örneğin; Şekil 3.17'de görüldüğü gibi 30 Mart tarihinde planlanan şekilde herhangi bir çalışma yapılmamış ve düz çizgi oluşmuştur.



Şekil 3.17: İkinci Koşu sırasında takip edilen İş Takip Grafiği.

3.2.13. İkinci Koşu Süreci

İkinci Koşu süreci 25 Mart 2014 Salı günü gerçekleştirilen Koşu Planlama Toplantısı ile başlayarak 01 Nisan 2014 Salı günü dahil olmak üzere 8 günlük sürede gerçekleştirilmiştir.

Koşu süresince Geliştirme Ekibi İş Takip Grafiği ve Görev Kartlarını takip ederek süreci tamamlamışlardır.

İkinci Koşu sürecinde öncelikle Ürün Sahibi tarafından sağlanan logo ile giriş ekranı tasarımı değiştirilmiştir.

Sık Sorulan Sorular için veri tabanı tasarımı gerçekleştirilmiş ve test verileri ile veri tabanı test edilmiştir.

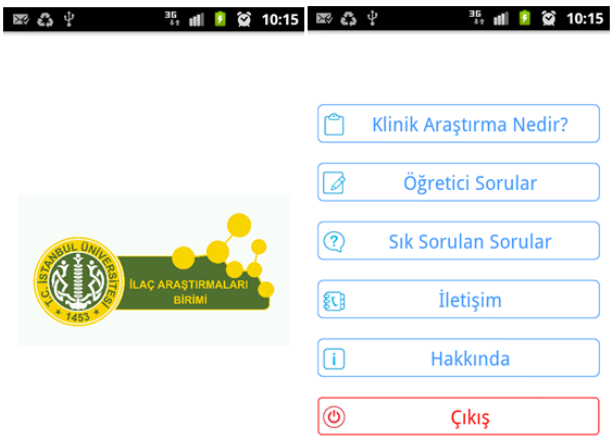
Veri tabanı oluşturulma aşamasından sonra web servisler oluşturulmuş ve sunucuya atılmıştır.

Gerçek ortamda çalışan servisler aracılığı ile uygulama aşaması gerçekleştirilmiştir.

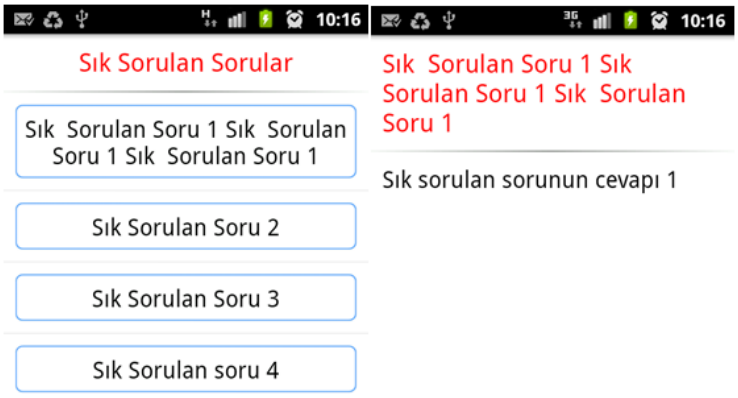
İkinci Koşu Sırasında ortaya çıkan önemli nokta şudur;

- Ekranlar üzerinde Geri ve Ana Menü butonlarının olup olmaması gerektiği Günlük Scrum Toplantıları sırasında tartışılmış ve Ürün Sahibi ile birlikte olmamasına karar verilmiştir.

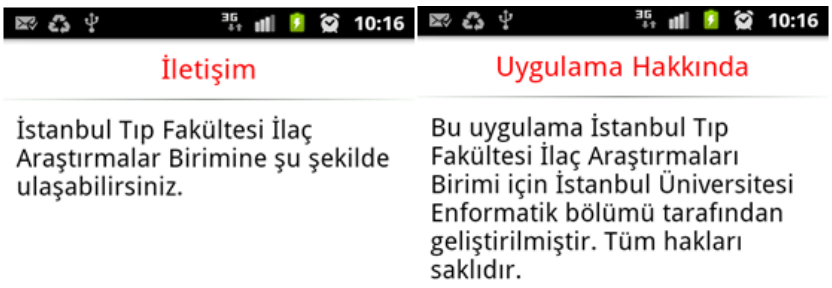
İkinci Koşu süresince geliştirilen uygulamanı ekran görüntüleri (Şekil 3.18, 3.19, 3.20) de görülmektedir.



Şekil 3.18: İkinci Koşu süreci sonucunda oluşan Kullanıcı Karşılama ve Yönlendirme Ekranları.



Şekil 3.19: İkinci Koşu Süreci sonucunda oluşan Sık Sorulan Sorular Listeleme ve Sık Sorulan Soru Detay Ekranları.



Şekil 3.20: İkinci Koşu sonucunda oluşan İletişim ve Hakkında Ekranları.

3.2.14. İkinci Koşu Süresince Günlük Scrum Toplantıları

İkinci Koşu sürecinde Günlük Scrum Toplantıları Birinci Koşu Sürecindeki Günlük Scrum Toplantılarında olduğu gibi gerçekleştirilmiştir.

3.2.15. İkinci Koşu Değerlendirme Toplantısı

1 Nisan 2014 Salı günü İkinci Koşu Değerlendirme Toplantısı gerçekleştirilmiştir. Geliştirme Ekibi tarafından istenen değişikliklerin yapıldığı ve yeni eklenen kısımlarla birlikte uygulamanın son hali sunulmuştur.

Ürün sahibi tarafından beğenilmeyen ya da değiştirilmesi istenilen özellikler not alınmıştır;

- Yönlendirme Ekranından kullanıcı, geri tuşunu kullanarak çıkamamalıdır, sadece Çıkış butonuna tıklayarak çıkabilmelidir.

Bu toplantı sırasında bir sonraki Koşu Planlama Toplantısına hazırlık olması ve ekibin hazırlık yapması amacıyla sonraki koşuda ne yapılması gerektiği üzerinde fikir alışverişinde bulunulmuştur.

3.2.16. İkinci Koşu Süreç Değerlendirme Toplantısı

1 Nisan 2014 Salı günü İkinci Koşu Süreç Değerlendirme Toplantısı gerçekleştirilmiştir ve bu toplantı sırasında üzerinde durulan bazı noktalar aşağıdaki gibi şekillenmiştir;

- Scrum süreci hakkında ekibin uzmanlaştığı ve bunu tam verimlilikle bir sonraki koşu sürecinde de gerçekleştirmeleri gerektiği üzerinde durulmuştur.
- Proje planına daha iyi uyulduğu sonucuna varılmıştır (Şekil 3.17).
- İş Takip Grafiğinin başarı ile gerçekleştirildiği ve ekibin bu şekilde devam etmesi gerektiği belirtilmiştir (Şekil 3.17).

3.2.17. Üçüncü Koşu Planlama Toplantısı ve Üçüncü Koşu Gereksinim Listesi

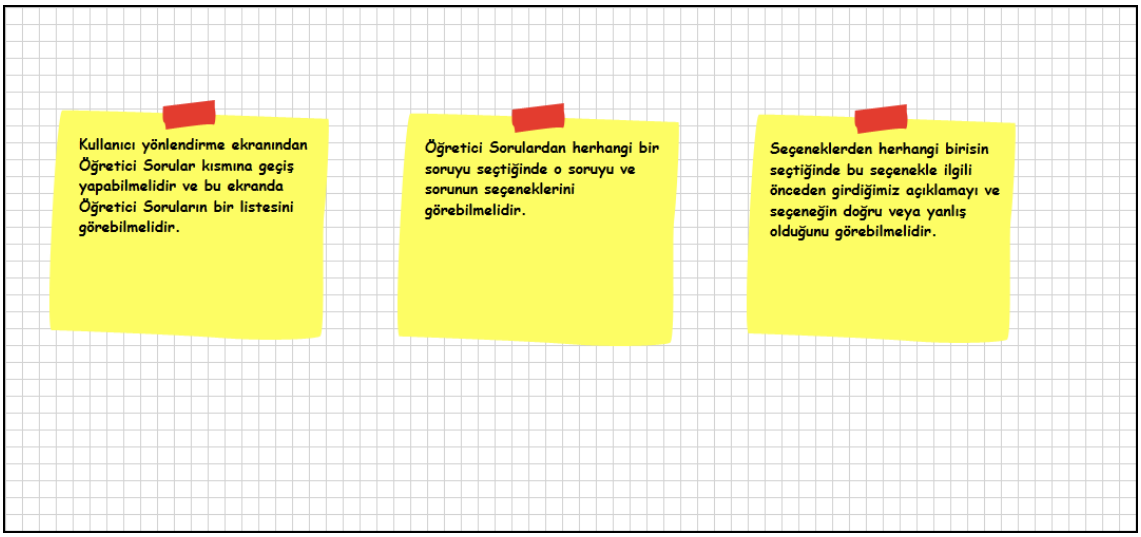
İkinci Koşu Değerlendirme Toplantısı sırasında fikir alışverişinde bulunan Scrum Ekibi 02 Nisan 2014 Çarşamba günü gerçekleştirilen Üçüncü Koşu Planlama Toplantısına Öğretici Sorular kısmı için hazırlıklı olarak gelmiştir. Bu toplantı Google Talk üzerinden gerçekleştirilmiştir.

Ürün sahibi Şekil 3.5 ve 3.6'da yer alan prototip ekranlarını biraz daha detaylandırarak anlatmıştır. Bu anlatım ile Üçüncü Koşu için Kullanıcı Hikayeleri güncellenmiş ve bu hikayelerden görevler çıkarılmıştır.

Ürün Sahibinin istediği özellikleri belirten Kullanıcı Hikayeleri aşağıdaki gibidir;

- Yönlendirme ekranından Öğretici Sorular kısmına geçiş yapılabilmelidir ve bu ekranda Öğretici Soruların bir listesi görülebilmelidir.
- Öğretici Sorulardan herhangi bir soru seçildiğinde o soru ve sorunun seçeneklerini görülebilmelidir.
- Seçeneklerden herhangi birisi seçildiğinde bu seçenekle ilgili önceden girilen açıklama ve seçeneğin doğru veya yanlış olduğu görülebilmelidir.

Üçüncü Koşu için Kullanıcı Hikayeleri ile oluşturulmuş Kullanıcı Hikaye Kartları Şekil 3.21'de görüldüğü gibidir.



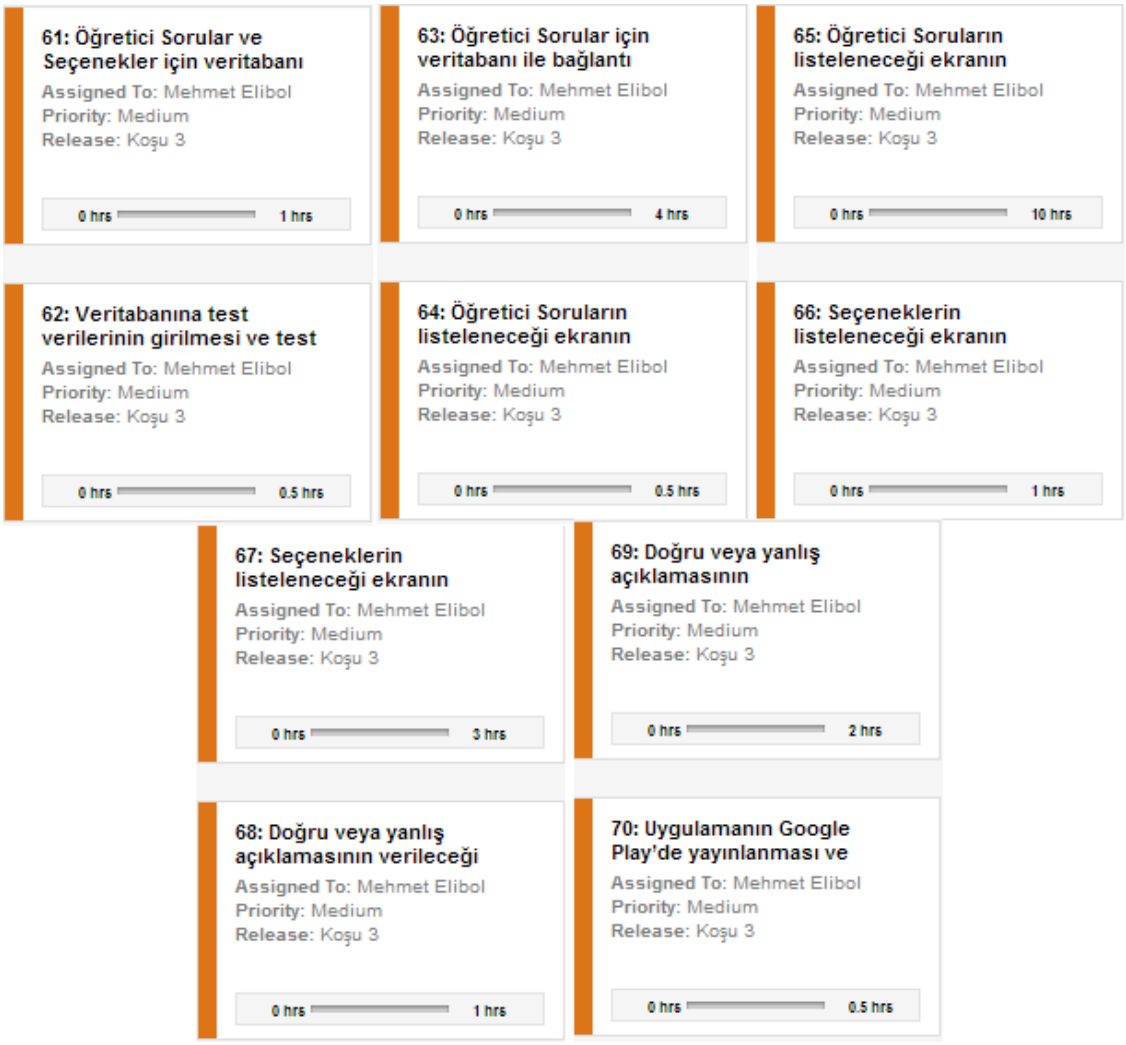
Şekil 3.21: Üçüncü Koşu için Kullanıcı Hikaye Kartları.

Kullanıcı Hikayelerini gerçekleştirmek için yapılması gereken görevler ve bu görevlerin ne kadar birimde gerçekleştirileceği aşağıdaki gibi belirlenmiştir;

- Öğretici Sorular ve Seçenekler için veri tabanı tasarımının yapılması ve gerçekleştirilmesi / 1 saat
- Veri tabanına test verilerinin girilmesi ve test edilmesi / 0,5 saat
- Öğretici Sorular için veri tabanı ile bağlantı kuracak web servislerin tasarlanması ve kodlanması / 4 saat
- Öğretici Soruların listeleneceği ekranın tasarlanması / 0,5 saat
- Öğretici Soruların listeleneceği ekranın kodlanması ve test edilmesi / 10 saat
- Seçeneklerin listeleneceği ekranın tasarlanması / 1 saat
- Seçeneklerin listeleneceği ekranın kodlanması ve test edilmesi / 3 saat
- Doğru veya yanlış açıklamasının verileceği ekranın tasarlanması / 1 saat
- Doğru veya yanlış açıklamasının gösterileceği ekranın kodlanması ve test edilmesi / 2 saat
- Uygulamanın Google Markette yayınlanması ve test edilmesi / 0,5 saat

Kullanıcı Hikayelerine göre Ürünün tamamlanması için görevlerin oluşturulması sırasında 23,5 birimlik (saat) iş yükü oluşmuştur.

Koşu Planlama Toplantısı sonrası gerçekleştirilmesi gereken görevler yukarıdaki şekilde belirlendikten sonra Görev Kartları (Şekil 3.22) ve Görev Listesi (Şekil 3.23) hazırlanmıştır.



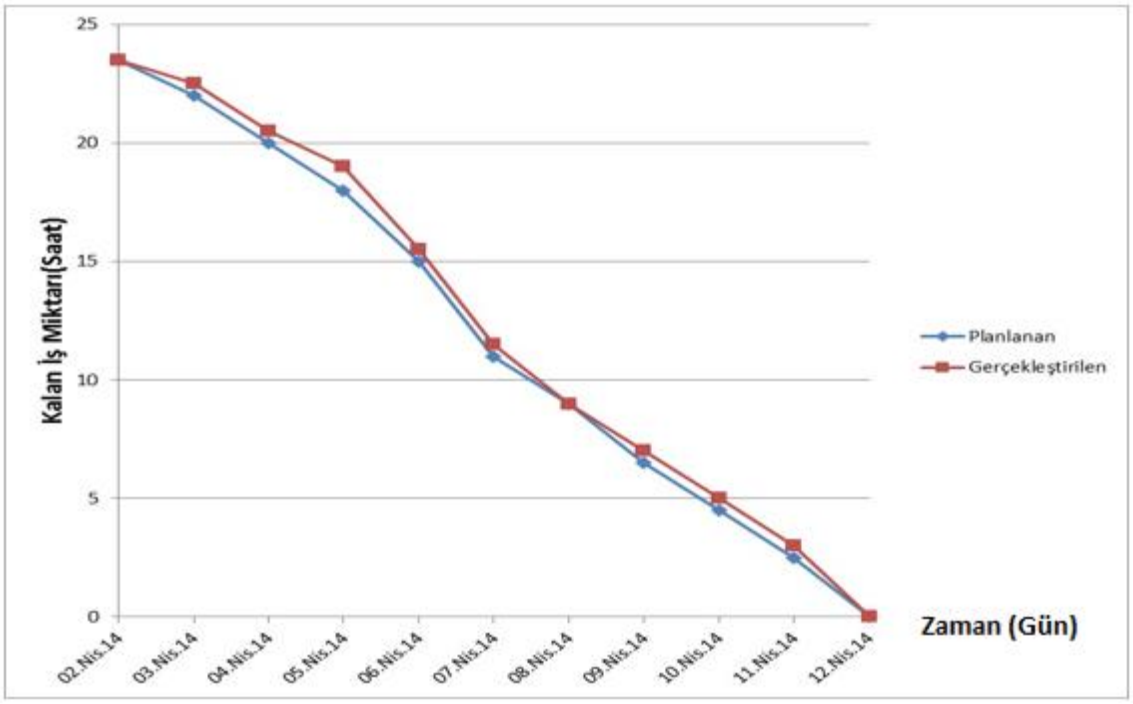
Şekil 3.22: Üçüncü Koşu sürecinde gerçekleştirilen Görev Kartları.

ID	Title	Start Date	Original Estimat
61	Öğretici Sorular ve Seçenekler için veritabanı tasarımının yapılması ve gerçek	02/04/2014	1 hrs
62	Veritabanına test verilerinin girilmesi ve test edilmesi	02/04/2014	0.5 hrs
63	Öğretici Sorular için veritabanı ile bağlantı kuracak web servislerin tasarlanma	03/04/2014	4 hrs
64	Öğretici Soruların listeleneceği ekranın tasarlanması	05/04/2014	0.5 hrs
65	Öğretici Soruların listeleneceği ekranın kodlanması ve test edilmesi	05/04/2014	10 hrs
66	Seçeneklerin listeleneceği ekranın tasarlanması	08/04/2014	1 hrs
67	Seçeneklerin listeleneceği ekranın kodlanması ve test edilmesi	09/04/2014	3 hrs
68	Doğru veya yanlış açıklamasının verileceği ekranın tasarlanması	10/04/2014	1 hrs
69	Doğru veya yanlış açıklamasının gösterileceği ekranın kodlanması ve test edili	11/04/2014	2 hrs
70	Uygulamanın Google Play'de yayınlanması ve test edilmesi	11/04/2014	0.5 hrs

Şekil 3.23: Üçüncü Koşu sürecinde gerçekleştirilen Görev Listesi.

3.2.18. Scrum Uzmanı Tarafından Üçüncü Koşu İçin İş Takip Grafiğinin Hazırlanması

Scrum Uzmanı tarafından Üçüncü Koşu için hazırlanan İş Takip Grafiği Şekil 3.24'de görülmektedir.



Şekil 3.24: Üçüncü Koşu sırasında takip edilen İş Takip Grafiği.

3.2.19. Üçüncü Koşu Süreci

Üçüncü Koşu süreci 02 Nisan 2014 Çarşamba günü gerçekleştirilen Koşu Planlama Toplantısı ile başlayarak 11 Nisan 2014 Cuma günü dahil olmak üzere 10 günlük sürede gerçekleştirilmiştir.

Üçüncü Koşu süresince Geliştirme Ekibi İş Takip Grafiği ve Görev Kartlarını takip ederek süreci tamamlamışlardır.

Öğretici Sorular ve Seçenekler için veri tabanı tasarımı gerçekleştirilmiş ve test verileri ile veri tabanı test edilmiştir.

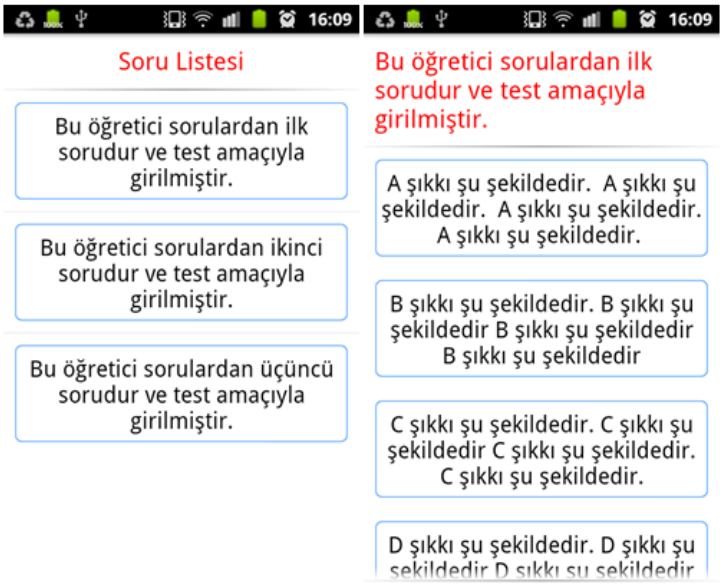
Veri tabanının oluşturulmasından sonra web servisler oluşturulmuş ve sunucuya atılmıştır.

Gerçek ortamda çalışan servisler aracılığı ile uygulama aşaması gerçekleştirilmiştir.

Uygulama geliştirme aşamasında doğru veya yanlış ile ilgili dönütün verileceği ekranın tasarımında Ürün Sahibinin geri bildirimlerine göre değişiklikler yapılmıştır. Bu ekranlarda yanlış veya doğru seçeneğe göre simgenin olması gerektiğine karar verilmiştir.

Koşu sırasında Ürün Sahibi Koşu süreci bittikten sonra uygulamanın yayınlanmamasını talep etmiştir. Gerçek verilerle uygulamanın yayınlanması istenildiği için bu görev iptal edilmiştir.

Üçüncü Koşu Süreci sonucunda geliştirilen uygulamanın ekran görüntüleri Şekil 3.25 ve 3.26'da görülmektedir.



Şekil 3.25: Üçüncü Koşu sonucunda oluşan Öğretici Sorular Listeleme ve Öğretici Soru Cevaplama Ekranları.



Şekil 3.26: Üçüncü Koşu Sonunda Oluşan Öğretici Soru Geri Bildirim Ekranları.

3.2.20. Üçüncü Koşu Süresince Günlük Scrum Toplantıları

Üçüncü Koşu sürecinde Günlük Scrum Toplantıları Birinci Koşu ve İkinci Koşu Sürecindeki Günlük Scrum Toplantılarında olduğu gibi gerçekleştirilmiştir.

3.2.21. Üçüncü Koşu Değerlendirme Toplantısı

Üçüncü Koşu sonunda 14 Nisan 2014 Pazartesi günü Geliştirme Ekibi uygulamanın son halinin sunumunu yapmıştır ve artık uygulamanın gerçek verilerle kullanılabileceğini belirtilmiştir.

Ürün Sahibinin gerçek verileri hazırladıktan sonra uygulamanın gerçek verilerle markette yayınlanmasına karar verilmiştir.

3.2.22. Üçüncü Koşu Süreç Değerlendirme Toplantısı

Son Koşudan sonra 14 Nisan 2014 Pazartesi günü Koşu Süreç Değerlendirme Toplantısı yapılmış ve Geliştirme Ekibi Ürün Sahibinin istediği ürünü tüm süreçleri en verimli şekilde kullanarak ortaya çıkardıkları fikrine varmışlardır.

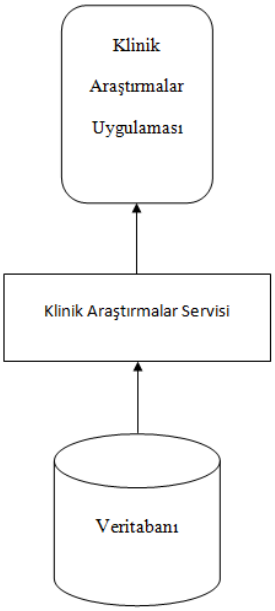
3.2.23. Ürünün Yayınlanması ve Proje Sürecinin Kapatılması

Ürün Sahibinden gelen gerçek veriler sisteme girildikten sonra, Ürün Sahibinin ve kullanıcıların gerçek verilerle sistemi test etmeleri sağlanmıştır. Gerçek verilerle kullanım sonucunda oluşan geri bildirimler sonucunda uygulama üzerinde güncellemeler yapılmıştır. Ürün Sahibinden onay alınarak uygulama Google Markette 22 Nisan 2014 tarihinde yayınlanmıştır. Markette yayınlanan uygulamanın son hali Bulgular kısmında ayrıntılı ele alınmıştır.

4. BULGULAR

4.1. KLİNİK ARAŞTIRMALAR UYGULAMASI

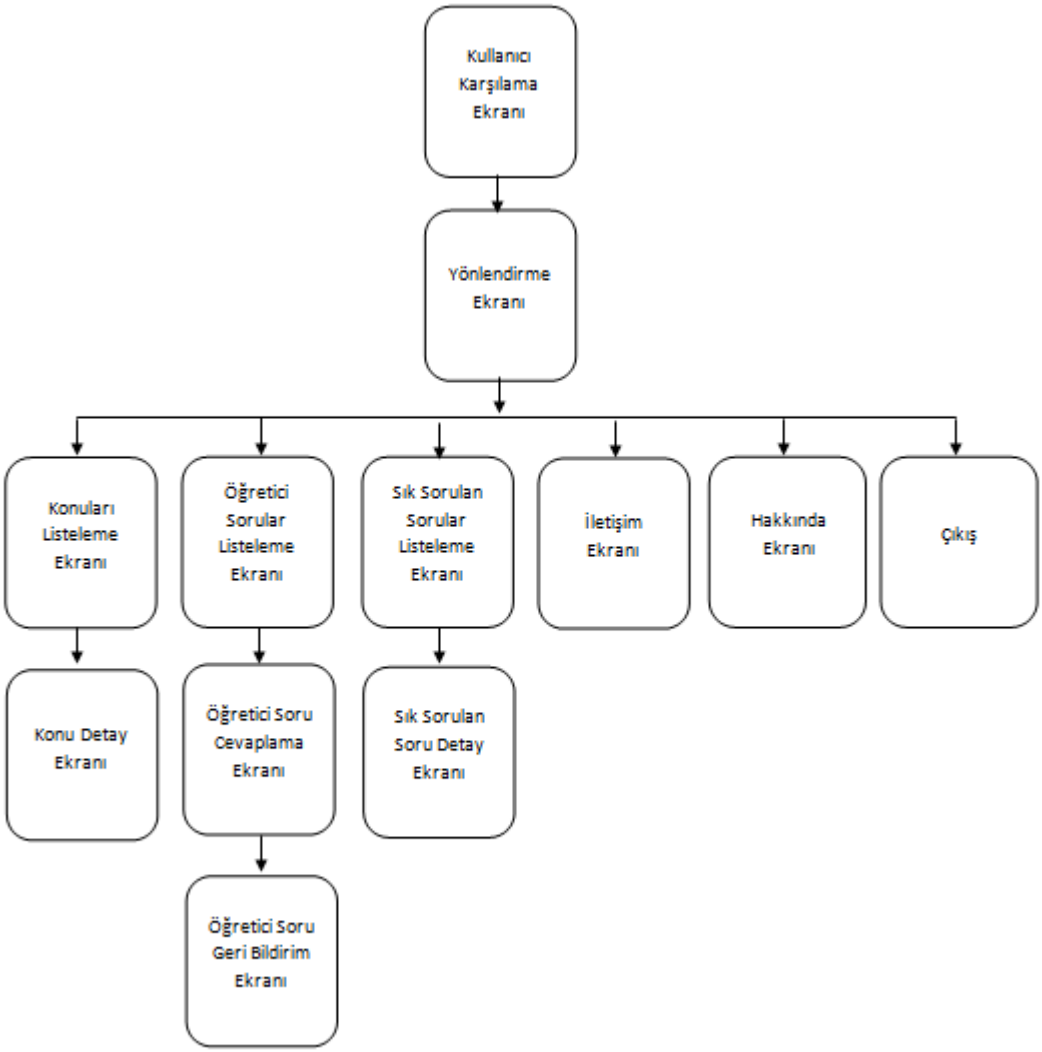
Bu tez kapsamında Çevik Metodolojilerden Scrum Metodu kullanılarak İstanbul Üniversitesi İstanbul Tıp Fakültesi İlaç Araştırmaları Birimi için servis tabanlı bir mobil uygulama geliştirilmiştir. Bu sistem ile Klinik Araştırmalar hakkında bilgi almak isteyen kişilerin bilgi sahibi olmaları amaç edinilmiştir.



Şekil 4.1: Klinik Araştırmalar Uygulaması çalışma prensibi.

Tez kapsamında gerçekleştirilen uygulama servis aracılığı ile veri tabanından veri alışverişini gerçekleştirmektedir (Şekil 4.1).

Uygulamanın 1.1 versiyonu 22 Nisan 2014 tarihinde Google Markette yayınlanmıştır. Uygulama https://play.google.com/store/apps/details?id=com.itp.klinikuygulama bağlantı ile veya Klinik Araştırmalar Uygulaması şeklinde aratılarak Google Marketten indirilebilir.



Şekil 4.2: Uygulama Akışı.

Uygulamanın çalışma akışı Kullanıcı Karşılama Ekranından başlayarak Yönlendirme Ekranı, Konuları Listeleme Ekranı, Öğretici Soruları Listeleme Ekranı, Sık Sorulan Soruları Listeleme Ekranı, İletişim Ekranı, Hakkında Ekranı, Konu Detay Ekranı, Öğretici Soru Cevaplama Ekranı, Sık Sorulan Soru Detay Ekranı ve Öğretici Soru Geri Bildirim Ekranından oluşmaktadır (Şekil 4.2).



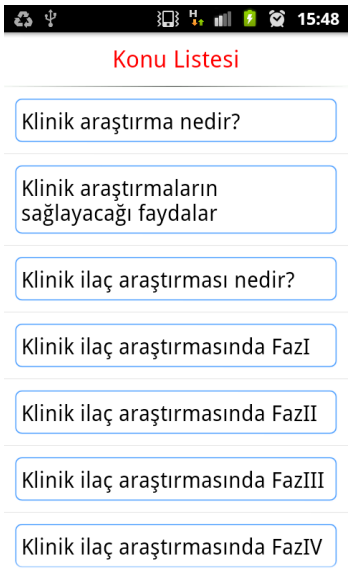


Şekil 4.3: Kullanıcı Karşılama Ekranı.

Kullanıcı, mobil akıllı cihazına yüklü uygulamayı ilk çalıştırdığında İstanbul Üniversitesi İstanbul Tıp Fakültesi İlaç Araştırmaları Biriminin logosunun yer aldığı Kullanıcı Karşılama Ekranı ile karşılanmaktadır (Şekil 4.3). Bu ekranda 3 saniye bekletilen kullanıcı, süre sonunda bir sonraki ekran olan Yönlendirme Ekranına yönlendirilmektedir (Şekil 4.3).

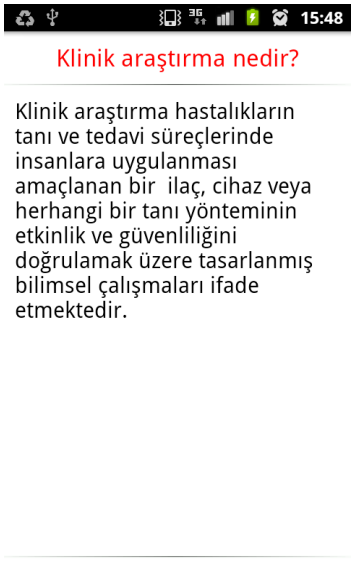


Şekil 4.4: Yönlendirme Ekranı.



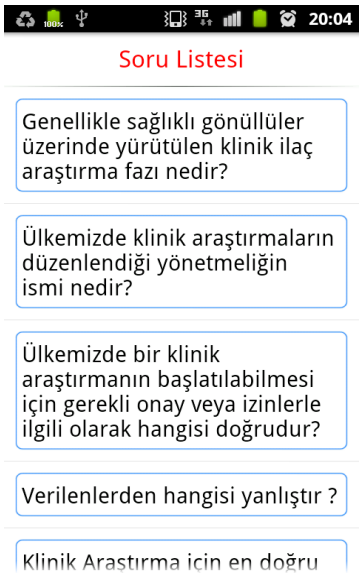
Şekil 4.5: Konu Listeleme Ekranı.

Kullanıcı, Yönlendirme Ekranından (Şekil 4.4) Klinik Araştırma butonuna tıkladığında konu veritabanında yer alan konuların listesinin yer aldığı Konu Listeleme Ekranı (Şekil 4.5) ile karşılaşmaktadır. Konu Listeleme Ekranından detaylı bilgi almak istediği konuyu seçebilmektedir.



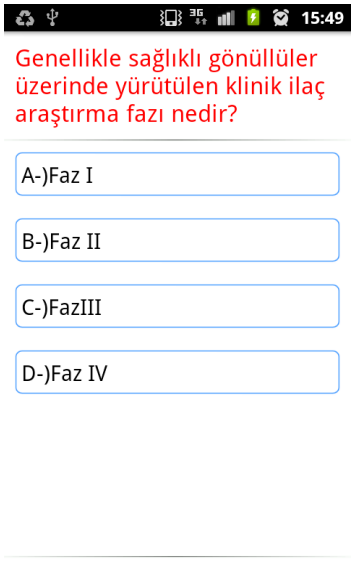
Şekil 4.6: Konu Detay Ekranı.

Kullanıcı Konu Listeleme Ekranından (Şekil 4.5) herhangi bir konuyu seçtiğinde, o konuyla ilgili detaylı bilginin yer aldığı Konu Detay Ekranına (Şekil 4.6) yönlendirilmektedir.



Şekil 4.7: Öğretici Soru Listeleme Ekranı.

Kullanıcı, Yönlendirme Ekranından (Şekil 4.4) Öğretici Sorular butonuna tıkladığında öğretici sorular veritabanında yer alan öğretici soruların listesinin yer aldığı Öğretici Soru Listeleme Ekranı (Şekil 4.7) ile karşılaşmaktadır. Bu ekrandan cevaplamak istediği soruyu seçebilmektedir.



Şekil 4.8: Öğretici Soru Detay Ekranı

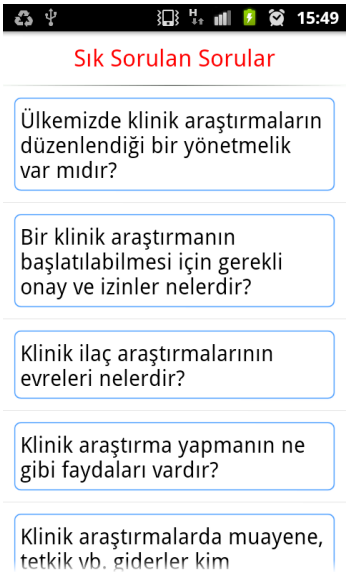
Kullanıcı, Öğretici Soru Listeleme Ekranından (Şekil 4.7) herhangi bir soruyu seçtiğinde, o soruyla ilgili seçeneklerin yer aldığı Öğretici Soru Detay Ekranına (Şekil 4.8)

yönlendirilmektedir. Kullanıcı bu ekranda soru ve şıkları detaylı şekilde görerek soruya cevap verebilmektedir.



Şekil 4.9: Öğretici Soru Geri Bildirim Ekranı

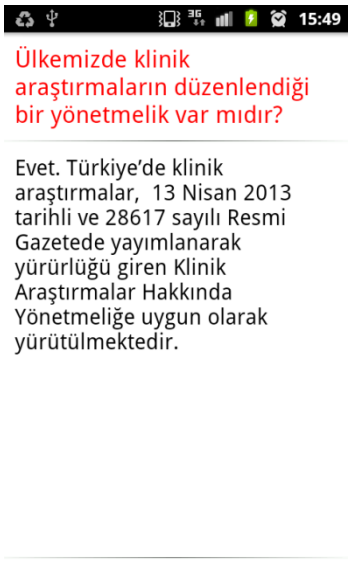
Kullanıcı, Öğretici Soru Detay Ekranından (Şekil 4.8) herhangi bir seçeneği seçtiğinde, o seçenekle ilgili geri bildirim aldığı Öğretici Soru Geri Bildirim Ekranına (Şekil 4.9) yönlendirilmektedir. Bu ekrandan kullanıcı cevabının doğru mu, yanlış mı olduğunu her seçenek için görebilmektedir.



Şekil 4.10: Sık Sorulan Sorular Listeleme Ekranı

Kullanıcı, Yönlendirme Ekranından (Şekil 4.4) Sık Sorulan Sorular butonuna tıkladığında sık sorulan sorular veritabanında yer alan sık sorulan sorular listesinin yer aldığı Sık

Sorulan Sorular Listeleme Ekranı (Şekil 4.10) ile karşılaşmaktadır. Bu ekrandan detaylı cevap almak istediği soruyu seçebilmektedir.



Şekil 4.11: Sık Sorulan Sorular Detay Ekranı.

Kullanıcı, Sık Sorulan Sorular Listeleme Ekranından (Şekil 4.10) herhangi bir soruyu seçtiğinde, o soruyla ilgili detaylı cevabın yer aldığı Sık Sorulan Sorular Detay Ekranına (Şekil 4.11) yönlendirilmektedir. Kullanıcı Sık Sorulan Sorular Detay Ekranında cevap almak istediği soruyla ilgili detaylı cevaba ulaşabilmektedir.



Şekil 4.12: İletişim Ekranı.

Kullanıcı, Yönlendirme Ekranından (Şekil 4.4) İletişim butonuna tıkladığında İletişim bilgilerinin yer aldığı İletişim Ekranı (Şekil 4.12) ile karşılaşmaktadır.



Şekil 4.13: Uygulama Hakkında Ekranı.

Kullanıcı, Yönlendirme Ekranından (Şekil 4.4) Hakkında butonuna tıkladığında uygulama hakkında bilgilerin yer aldığı Uygulama Hakkında Ekranına (Şekil4.13) yönlendirilmektedir.

Kullanıcı, Yönlendirme Ekranından (Şekil 4.4) Çıkış butonunu kullanarak uygulamadan çıkabilmektedir.

5. TARTIŞMA VE SONUÇ

Klinik araştırmalar sayesinde yeni ilaçların ve tedavilerin geliştirilmesi, yeni tedavilerin potansiyel hastalar tarafından kullanılabilmesi, hatta ülke ekonomisine katkı sağlanması mümkün olabilmektedir. Ülkemizde yapılan klinik araştırma çalışma sayısı yeterli değildir[13]. Bunun en önemli nedenleri arasında insanlarımızın klinik araştırma konusunda doğru ve yeterli bilgiye sahip olmamalarının yer aldığı düşünülmektedir. Klinik araştırma çalışma sayısının arttırılabilmesi için bilgilendirme önemli bir unsur olarak görülmektedir. Hedef kitlenin güncel bilgiye hızlı ve kolay erişebilmesi için bilginin farklı araçlar ile sunulması etkin bir yöntem olabilecektir.

Bilgi teknolojisindeki hızlı gelişmeler sayesinde, kolay ve hızlı olarak bilgiye erişmek için mobil cihazlar bireyler tarafından sıklıkla kullanılan araçlardır. Bu nedenle tez kapsamında, klinik araştırmalar alanında ihtiyaç duyulan bir mobil uygulama çevik yazılım geliştirme metodolojisi kullanılarak gerçekleştirilmiştir. Uygulama marketlerinde bu doğrultuda hazırlanan Türkçe bir uygulamaya erişilememiştir. Bu açıdan bakıldığında tez kapsamında geliştirilen bu uygulama, ülkemiz için bir ilk olma özelliğini taşımaktadır.

Bir projeye başlarken proje sürecinde nasıl hareket edileceğinin belirlenmesi gerekmektedir. Bunun için bir metodoloji belirlenmesi ve bu metodolojinin kurallarına göre hareket edilmesi önemli olmaktadır. Süreç içinde geliştirilen projeler birçok değişkenden etkilenmektedir. Highsmith ve Cockburn (2001) çalışmalarında projelerin başarılı olması için amacın, değişimi engellemek değil, bu değişime en iyi şekilde cevap vererek, projeleri maksimum verimlilikte tamamlamak olduğunu belirtmektedir. Highsmith (2002) ayrıca yazılım teknolojileri ve yazılım dünyasında yaşanan gelişmelere ayak uydurabilmek ve bu süreçte sürekli değişen müşteri ihtiyaçlarını en iyi şekilde yönetmek amacıyla çevik metodolojiler üzerine yoğunlaşılması gerektiğini vurgulamaktadır. Tez sürecinde bunları gerçekleştirmek amacıyla klinik araştırmalar için geliştirilen mobil uygulama çevik metodolojilerden olan Scrum metodu takip edilerek gerçekleştirilmiştir. Bu şekilde uygulama geliştirme sürecinde değişikliklere karşı duyarlılık oluşmuş ve değişikliklere en kısa sürede cevap verebilmek mümkün olmuştur.

[13] www.clinicaltrials.gov, [Ziyaret Tarihi: 25 Nisan 2014]

Scrum ile proje geliştirme süreci Ürün Sahibi, Scrum Uzmanı ve Geliştirme Ekibinden oluşan bir Scrum Ekibi ile gerçekleştirilmektedir. Tez öğrencisi Scrum Uzmanı ve Geliştirme Ekibi rollerini tez sürecinde yürütmüştür. Tez öğrencisinin bu iki rolü uygulamaya çalışması bir sınırlılık oluşturmuştur. Ancak bu sınırlılığın olmasına rağmen Scrum sürecinde tez öğrencisinin bu rolleri yürütmeye çalışması rollerin anlaşılmasını ve anlatılmasını kolaylaştırmıştır. Scrum sürecinde istenmeyen bu durum, tez öğrencisinin gelişimi açısından bir avantaj sağlamıştır.

Schwaber ve Sutherland (2013) Scrum Metodu proje sürecini küçük parçaları bölmekte ve bu parçaları bir proje olarak kabul etmektedir. Koşu olarak adlandırılan bu alt projelerin proje bitiş tarihleri ekiple birlikte belirlendikten sonra, işin ne kadarının bittiğinin görülmesi için İş Takip Grafiğinin oluşturulmasını zorunlu kılmaktadır. Buna uygun olarak tez sürecinde İş Bitim Grafikleri oluşturulmuş, Geliştirme Ekibinin süreç içinde, her gün işin ne kadarının yapıldığını görmeleri sağlanarak sürecin şeffaf olması sağlanmıştır. Bunun proje sürecinin takip edilmesinde son derece önemli olduğu, Scrum ile yürütülen projelerde bunun mutlaka yapılması gerektiği sonucuna varılmıştır.

Schwaber ve Sutherland (2013) Scrum Metodunda etkileşim ve iletişim kavramlarının önemi üzerinde durmaktadır. Buna uygun olarak uygulama geliştirme sürecinde Ürün Sahibi sürekli ulaşılabilir durumda olmuş, oluşan olumlu etkileşim ve pozitif iletişim ile değişikliklere karşı duyarlılık oluşmuş ve soru işareti olan noktaların hızlı şekilde çözülmesi sağlanmıştır. Bu etkileşimler sırasında iyi ve özgün fikirlerin ortaya çıkabileceği görülmüştür. Bu yeni fikirlerin de uygulama geliştirme sürecinin verimliliğini olumlu yönde etkilediği ve oluşan ürünün değerini artırdığı düşünülmektedir.

Yazılım geliştirme sürecinde önemli noktalardan birisi yazılımın süreç içinde test edilmesini sağlamaktır. Tez sürecinde Beck ve arkadaşlarının (2001) belirttiği gibi uygulama çalışır durumda tutulduğu için, Geliştirme Ekibi ve Ürün Sahibi sürekli aktif olarak kalmıştır ve bunun sonucunda uygulamanın sürekli test edilmesi sağlanmış, bu şekilde hataların anında tespit edilmesi ve çözümün hızlı şekilde bulunması sağlanmıştır. Bu sürecin farklı birçok mobil cihaz olmasından dolayı, özellikle mobil uygulama geliştirme sürecinde daha önemli olduğu düşünülmektedir.

Scrum sürecinde denetim ve adaptasyonu başarı ile gerçekleştirmek için farklı toplantıların yapılmasını zorunlu kılmaktadır (Schwaber ve Sutherland, 2013). Tez kapsamında uygulama geliştirme sürecinde bu toplantılar yüz yüze ve/veya uzaktan haberleşme yöntemleriyle gerçekleştirilmiştir. Bunun sonucunda, uzaktan haberleşme yöntemleri ile yapılan toplantıların yerine, yüz yüze iletişimin sağlandığı toplantıların daha verimli geçtiği görülmüştür. Ayrıca Ürün Sahibi ile yapılan yüz yüze toplantılar sonucunda, Ürün Sahibinin ifade etmek istediklerinin kağıt kalem kullanlarak, basit şekilde çizilmesinin, ne yapılacağının belirlenmesine ve prototipler hazırlanmasına büyük kolaylık sağladığı görülmüştür. Uzaktan haberleşme yöntemleri ile yapılan toplantılarda aynı verim sağlanamamıştır.

Ürün Gereksinim Listesinin belirlendiği ilk toplantı öncesi Scrum Uzmanı ve Geliştirme Ekibinin o alanda yapılmış çalışmaları inceleyerek, toplantıya gelmesinin bir gereklilik olduğu görülmüştür. Özellikle Scrum Uzmanı ve Geliştirme Ekibinin yabancı olduğu alanlarda bir iş geldiğinde bunun daha da önemli olabileceği düşünülmektedir. Klinik Araştırmalar için mobil uygulama geliştirme proje süreci başlamadan önce Scrum Uzmanı ve Geliştirme Ekibi bu alanda yapılan çalışmaları incelemiştir. Bu şekilde, Scrum sürecinde yer alan tüm rollerin ortak bir dili konuşması sağlanmıştır.

Mobil uygulama geliştirme sürecinde, müşteriler mobil uygulama dünyasındaki bazı kısıtları düşünmeden ne istediklerini anlatmakta ve web tabanlı sistemlerdeki gibi özellikleri sistemlere ekletmeye çalışabilmektedirler. Scrum Uzmanı devreye girerek bu durumun anlaşılmasını sağlamalıdır. Tez sürecinde bilgilerin eklenmesi sırasında, ürün sahibi bir sayfaya çok uzun bilgilerin eklenmesini istemiş, Scrum Uzmanı devreye girerek, bir sayfada ne kadar az bilgi verilirse, kullanıcıların bunu anlamasını kolaylaştırdığını belirterek, bilgilerin buna göre düzenlenmesini sağlamıştır.

Jeong ve ark belirtiği gibi (2008) tüketicinin görüşünü yansıtan ve pazar ihtiyacını karşılayan mobil uygulamaların kar sağlaması için mümkün olduğu kadar hızlı şekilde pazara çıkması önemli olmaktadır. Tez sürecinde de buna uygun olarak Çevik Metodoloji ile gerçekleştirilen mobil uygulama yaklaşık 2 ay gibi bir sürede Google Markette kullanıma sunulmuştur.

Tez sürecinin kısıtlı olmasından dolayı uygulama ve içerik paralel olarak geliştirilmiştir. Bundan dolayı gerçek verilerle çalışılmaması problem oluşturmuştur. Gerçek veriler sisteme girildikten sonra, Ürün Sahibi servisten gelen verilerden bazı önemli noktaların renklendirilerek gelmesini istemiştir. Her ne kadar çevik metodolojiler esnek bir yapıya sahip olsa da servis mantığında bunu gerçekleştirmenin uygulama yapısını değiştirecek olması nedeniyle, Ürün Sahibine bu durum açıklanarak isteği kabul edilmemiştir. Bu nedenle içeriğin uygulama geliştirmeden önce hazır olması gerektiğinin önemli olduğu kanaatine varılmıştır.

Sonuç olarak, tez kapsamında çevik metodolojilerden birisi olan Scrum kullanılarak mobil uygulama geliştirilmiştir. Değişim sürecine hızlı adaptasyon, müşteri isteklerine hızlı cevap, kısa sürede markete çıkma, test süreçlerini daha hızlı bir şekilde gerçekleştirme, uygulamanın sürekli çalışır olması, bireyler arasındaki etkileşim, Ürün Sahibi ile işbirliği sunulması gibi artılarından dolayı mobil uygulama geliştirme sürecinde Scrum Metodunun tercih edilmesi önerilmektedir. Proje süreçleri Scrum Metodu ile gerçekleştirilmiş olsa bile, ürün desteği kısmı farklı metodolojiler ile sürdürülebilir.

Bu tez projesinde elde edilen çıktıların daha büyük boyuttaki mobil projelere yol göstermesinin yanı sıra, metodoloji seçiminde kararsız kalan araştırmacılar ve Scrum ile mobil uygulama geliştirecek kişiler içinde önemli bir kaynak olduğu düşünülmektedir. Özellikle Scrum Metodunun mobil oyun sektöründe aşamalı oyunlarda uygulanmasının, pazar rekabetinde avantaj sağlayacağı düşünülmektedir. Her aşama bir Koşu sürecinde bitirilerek, bu Google Markette yayınlanabilir. Bu şekilde rekabetin olduğu pazarda uygulamanın mümkün olan en kısa sürede pazara çıkması sağlandığı için kâr etme oranı arttırılabilir. Ayrıca Scrum sürecinde her Koşu sonucunda oluşan ürün parçasının yayınlanması ile uygulamanın Google Markette arama sonucunda ön sıralarda yer alması sağlanabilir. Bu da tam rekabet piyasasında ürün açısından önemli bir avantaj olarak görülmektedir.

KAYNAKLAR

Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J., 2002, *Agile*, Vuorimiehentie, Finland, ISBN: 951-38-6009-4.

Ambler, S.W., 2002, Lessons in agility from Internet-based development, *IEEE Computer*, 19 (2), 66-73.

Ayvaz, Z., 1999, *Nesneye Yönelik Bilgisayar Destekli Bir Tasarım Paketinin Geliştirilmesi*, Doktora, İstanbul Üniversitesi Fen Bilimleri Enstitüsü.

Beck,K. ve diğ., 2001, Manifesto for Agile Software Development, http://agilemanifesto.org/, [Ziyaret Tarihi:25 Şubat 2014].

Beedle, M., Devos, M., Sharon, Y., Schwaber, K., Sutherland, J., 1999, SCRUM: An extension pattern language for hyperproductive software development, *Pattern Languages of Program, 4*, 637-651.

Boehm,B.W., 1988, A Spiral Model of Software Development and Enhancement, *Computer*, 21(5), 61-72.

Corral, L., Sillitti, L., Succi, G., 2013, Software Development Processes for Mobile Systems Is Agile Really Taking Over the Business?, *1st International Workshop on the Engineering of Mobile-Enabled Systems(MOBS 2013)*, 25 May 2013 San Francisco CA, New Jersey, IEEE, 19-24.

Dennis, A., Wixom,B.H., Roth, R.M., 2010,*Systems Analysis and Design*, Wiley, Asia, ISBN:13 978-0-470-40031-9.

Dove, R., 2001, *Response Ability:The Language, Structure, and Culture of the Agile Enterprise*, Wıley & Sons, New York, ISBN: 978-0-471-35018-7.

Flora, H.K., Chande, S.V., 2013, A Review and Anaysis on Mobile Application Developmnet Processes Using Agile Methodolojies, *International Journal of Research in Computer Science*, 3 (4), 8-18.

Forman, G.H.,Zahorjan, J., 1994, The Challenges of Mobile Computing, *Institute of Electrical and Electronics Engineers Computer*, 27(4), 38-47.

Gürsoy, İ., 2012, SQL Server Nedir ?, http://www.ismailgursoy.com.tr/sql-server-nedir/, [Ziyaret Tarihi: 10 Nisan 2014].

Hayes, I.,S., 2002, *Just Enough Wireless Computing*, Prentice Hall PTR, New Jersey, ISBN: 978-0130994615.

Highsmith, J. 1997, *Messy, exciting and anxiety-ridden:Adaptive software development*, American Programmer, Arlington.

Highsmith, J., 2002, *Agile Software Development Ecosystems*, Addison Wesley, Boston, ISBN: 0-201-76043-6.

HighSmith, J., Cockburn, A., 2001, Agile Software Development: The Business of Innovation, *Computer*, 34(9),120-127.

Huo,M., Verner,J., Zhu, L., Babar,M.A., 2004, Software Quality and Agile Methods, *COMPSAC2004*, 28-30 September 2004 Hong Kong, Hong Kong, IEEE,0-7695-2209-2,520-525.

İlbars, H., 2014, *Türkiye'de Klinik Araştırmalar*, Klinik Araştırmalar Kitabı-2014, In: Alkan, H. (ed.), İlbars, H. (ed.), Çetinkaya, N. Ö. (ed.), 1, Bilimsel Tıp Yayınevi, Ankara, 11-14.

Jeong, Y.J, Lee, J.H, Shin,G.S., 2008, Development Process of Mobile Application SW Based on Agile Methodology, *10th International Conference on Advanced Communications Technology*, 17-20 February 2008 Gangwon-Do, Gangwon-Do, Institute of Electrical and Electronics Engineers (IEEE), ISBN: 978-89-5519-136-3, 362-366

Kaleel,S.B., Harinshankar, S., 2013, Applying Agile Methodology in Mobile Software Engineering:Android Application Development and Its Challenges, *Computer Science Technical Report*, Paper 4.

Kniberg, H., 2007, *Scrum and XP from the Trenches*, C4 Media Inc,United States of America , ISBN:978-1-4303-2264-1.

Leffingwell, D., Smiths, H., 2006, *A CIO's playbook for adopting the scrum method of achieving software agility*,Rally Software Development Corporation and Ken Schwaber-Scrum Alliance, Kolorado.

Nicolas, P., 2007, *Introduction to SCRUM Agile Process for Global Software Development*, http://www.pnexpert.com/files/SCRUM_for_Global_Development.pdf, [Ziyaret Tarihi:25 Nisan 2014].

Rahimian, V.,Ramsin, R., 2008, Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach, *2nd International Conference on Research Challenges in Informations Science*, 3-6 June 2008 Marrakech, Marrakech, Institute of Electrical and Electronics Engineers (IEEE), ISBN: 978-1-4244-2273-9, 337-342.

Scharff, C., Verma, R., 2010, *Scrum to support mobile application development projectsin a just-in-time learning context*, Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering, 2010 Cape Town, New York, Association for Computing Machinery,ISBN:978-1-60558-966-4,25-31.

Schwaber, K. ,2004, *Agile Project Management with Scrum*, Microsoft Press, Washington, , ISBN:0-7356-1993-X.

Schwaber, K., Sutherland, J., 2013, *The Definitive Guide to Scrum: The Rules of the Game*, https://www.scrum.org/Scrum-Guide, [Ziyaret Tarihi:04 Nisan 2014]

Vlaanderen, K.,Jansen, S., Brinkkemper, S., Jaspers, E., 2011, The agile requirements refinery:Appling SCRUM principles to software product management, *Information and Software Technology*, 53 (1), 58-70.

Williams, L. , Brown, G., Meltzer, A., Nagappan, N., 2011, Scrum+ Engineering Practices: Experiences of Three Microsoft Teams, *ESEM '11 Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*, 22-23 September 2011 Canada, Washington, IEEE Computer Society, ISBN:978-0-7695-4604-9, 463-471.

EKLER

Scrum İngilizce Terimler Sözlüğü

Sprint: Koşu[1]

User Stories: Kullanıcı Hikayeleri[1]

Product Owner: Ürün Sahibi[1]

Development Team: Geliştirme Ekibi[1]

Product Backlog: Ürün Gereksinim Listesi[2]

Sprint Backlog: Koşu Gereksinim Listesi[2]

Sprint Planning Meeting: Koşu Planlama Toplantısı[2]

Daily Scrum Meeting: Günlük Scrum Toplantıları[2]

Sprint Demo Review Meeting: Koşu Değerlendirme Toplantısı[3]

Scrum Master: Scrum Uzmanı[4]

Sprint Goal: Koşu Hedefi[5]

Sprint Burndown Chart: İş Takip Grafiği[5]

Sprint Retrospective Meeting: Koşu Süreç Değerlendirme Toplantısı[5]

Bu tez kapsamında Scrum İngilizce terimler sözlüğünün hazırlanması esnasında yararlanılan kaynaklar aşağıda listelenmiştir;

[1]Türkiye Bilişim Derneği ve Scrum Turkey koordinatörlüğünde yürütülen çeviri çalışması "http://www.scrumturkey.com/2013/10/scrum-terimleri-ceviri-calsmas-1-asamas.html, [Ziyaret Tarihi: 25 Nisan 2014]."

[2]Baytam, V., Kalıpsız, O., 2011, Scrum Yazılım Geliştirme Modeli Yönetim Aracı: ScrumMApp, 5.Ulusal Yazılım Mühendisliği Sempozyumu, Bildiriler Kitabı,, 26-28 Eylül Ankara, Ankara, Orta Doğu Teknik Üniversitesi, 275-283.

[3]Sahin, E., Keskin, I., & Koç, H., 2013, CMMI-DEV Seviye-3 Sertifikasyonuna Sahip Bir Organizasyonda SCRUM Çevik Yazılım Geliştirme Yöntemi'nin Yazılım Geliştirme Çalışmalarında Uygulanması, Bildiriler Kitabı, 25-28 Eylül İzmir, İzmir, Ege Üniversitesi.

[4]ACM Yazılım Çözümleri tarafından hazırlanan çalışma "http://www.acm-software.com/Agile/ScrumHakkinda, [Ziyaret Tarihi: 30 Nisan 2014]."

[5]İlk 4 kaynakta yer alan terimlerin Türkçe çevirileri göz önüne alınarak, bu terimler tarafımızca yeniden çevrilmiştir.

ÖZGEÇMİŞ



Kişisel Bilgiler

Adı Soyadı	Mehmet ELİBOL
Uyruğu	T.C.
Doğum tarihi, Yeri	1989, İzmir
Telefon	0 216 414 05 45 / 1111
E-mail	mehmet.elibol@marmara.edu.tr
Web adres	www.mehmetelibol.com

Eğitim

Derece	Kurum/Anabilim Dalı/Programı	Yılı
Doktora	İ.Ü. Fen Bilimleri Enstitüsü/ /	
Yüksek Lisans	İ.Ü. Fen Bilimleri Enstitüsü/ Enformatik / Enformatik	2014
Lisans	Yıldız Teknik Üniversitesi/ Eğitim Fakültesi/ Bilgisayar ve Öğretim Teknolojileri Öğretmenliği	2011
Lise	Tire Anadolu Ticaret Meslek Lisesi/ Bilgisayar Programcılığı	2007