

İSTANBUL TEKNİK ÜNİVERSİTESİ
Bilgisayar ve Bilişim Fakültesi

Çevik Çalışma Ortamında Kalite Güvence Mühendisliği

STAJ
Yunus Güngör
150150701

Yaz/2018

İstanbul Teknik Üniversitesi
Bilgisayar ve Bilişim Fakültesi
STAJ RAPORU

Akademik Yıl: 2017-2018

Staj yapılan dönem: ☒Yaz ☐Bahar ☐Güz

Öğrenci ile ilgili bilgiler

Adı ve Soyadı: Yunus Güngör

Öğrenci Numarası: 150150701

Bölüm: Bilgisayar Mühendisliği

Program: Bilgisayar Mühendisliği (%30 İngilizce)

E-posta Adresi: yunusgnr@gmail.com

(Cep) Tel No: 05436866535

ÇAP öğrencisi misiniz? ☐ Evet (ÇAP yaptığınız Fakülte/Bölüm: _____)

☒ Hayır

Mezuniyet
durumunda mısınız? ☐ Evet

☒ Hayır

Yaz okulunda ders ☐ Evet (Ders sayısı: __)

alıyor musunuz?

☒ Hayır

Öğrencinin çalıştığı kurum ile ilgili bilgiler

İsmi: Sosya Plus Bilgi Bil. Tek. Dan. Siz. Tic. A.Ş.

Birimi: Kalite Güvence

Web Adresi: <https://useinsider.com/>

Kısa Adresi: Yeşilce Mah. Emektar Sok. No:5 Kat:1-3-4 Kağıthane / İstanbul

Yetkili kiři ile ilgili bilgiler

Bölümü: Kalite Güvence
Unvanı: Kalite Güvence Müdürü
Adı ve Soyadı: Ceyhun Alyeřil
(Kurumsal) E-posta: ceyhun@useinsider.com
(Kurumsal) Tel. No.: +90-212-2707779

Yapılan iř ile ilgili bilgiler

Staj yeri ☒Türkiye
☐Yurtdıřı
Staj bařlangıç tarihi 10.08.2018
Staj bitiř tarihi 14.09.2018
Stajda alıřılan net **gün** sayısı 20
Staj süresince sigortanız var mıydı? ☒Evet, İTÜ tarafından sigortalandım.
☐Evet, kurum tarafından sigortalandım.
☐Hayır, yurtdıřı stajı yaptım.
☐Hayır.

STAJ RAPORU ONAY FORMU

İÇİNDEKİLER

| | |
|---|---|
| 1. KURULUŞ HAKKINDA BİLGİLER..... | 1 |
| 2. GİRİŞ..... | 2 |
| 3. STAJ PROJESİNİN TANIMI VE ANALİZİ..... | 2 |
| 4. SONUÇ..... | 6 |
| 5. REFERANSLAR..... | 6 |

150150701 numaralı, Yunus Güngör adlı öğrencinin, yukarıda “İçindekiler” bilgisi verilen staj raporu, görülmüş ve uygun bulunmuştur.

Formu Dolduran Firma Yetkilisinin Adı ve Soyadı: Ceyhun Alyeşil

Yetkilinin Unvanı: Kalite Güvence Müdürü

Müdür-İmza-Kaşe:



(Kurumsal) E-posta: ceyhun@useinsider.com

(Kurumsal) Tel. No.:+90 212 270 77 79

1. KURULUŞ HAKKINDA BİLGİLER

Sosya Plus Bilgi Bil. Tek. Dan. Siz. Tic. A.Ş. 2012 yılında Arda Koterin, Hande Çilingir, Sinan Toktay ve Serhat Soyuerel tarafından kurulmuş bir teknoloji şirkettir ve kendisini *Insider* olarak pazarlamaktadır. Firmaların teknolojik ortamlar üzerinde büyümesine yönelik ürünler üreten firma, kendi ürünlerini “*The First Integrated Growth Management Platform*” yani *İlk Entegre Büyüme Yönetim Platformu* adı altında birleştirmiş ve bu isimle pazarlamaktadır. Bu ürünler “*SaaS*” yani *Hizmet Şeklinde Yazılım* ilkesi ile geliştirilmektedir ve anlaşma yapılan firma yani partner’in web sitesine ya da mobil uygulamasına entegre edilerek, partnerin kullanıcılarıyla daha etkin iletişime geçmesini sağlamak ve partnerin büyümesini hızlandırmak amaçlanmaktadır. Kullanıcıları gerçek zamanlı segmentlemek için çok fazla seçenek sunabilen ürünler birlikte kullanılarak kullanıcıları belirli bir hedefe yönlendirmek amaçlanmaktadır. Aynı zamanda kullanılan ürünler kullanıcılar üzerinde *A/B testlerin* yapılmasına da olanak sağlar. *A/B testleri* veya *A/B/n testleri* kullanıcıları iki veya daha fazla topluluğa bölerek ve bu toplulukların aynı platformun farklı versiyonlarıyla etkileşmesini sağlayarak kullanıcı için optimum sistemin tasarlanmasıdır. Kullanıcılar belirli bir özelliğe veya rasgele olarak bölünebilmektedir. Platformun farklı versiyonları ise küçük renk değişikliklerinden, yenilenmiş bir tasarıma kadar her şey olabilir.

Ulusal ve uluslararası birçok girişim topluluğunda ismini duyurmuş olan *Insider* Türkiye’deki en başarılı girişimlerden biri sayılabilmektedir. İlk yatırımını *Galata Business Angels (GBA-Galata İş Melekleri)* ile sağlayan firma daha sonra *TÜBİTAK*’tan ve *Aslanoba Capital*’tan maddi yatırım almış daha sonra uluslar arası bir girişim derneği olan *Endeavor Derneği* ile işbirliği sağlamıştır (Çilingir, 2016). 16’den fazla ülkede 19’den fazla ofisi bulunan *Insider* hızla büyümesine devam etmektedir. *Insider İstanbul Ofisi* Türkiye’de bulunan iki ofisten birisidir. Türkiye’deki diğer ofis ise Ankara’da bulunmaktadır. Yazılım geliştirmelerinin tamamı *Insider İstanbul Ofisi* kapsamında yapılmaktadır ve bu ofis Sanayi Mahallesinde Matset Plaza’nın birinci, üçüncü ve dördüncü katlarında bulunmaktadır. Birinci katta toplantı odaları ve yönetim ile ilgili birimler bulunurken, dördüncü katta pazarlama ve müşteri ilişkileri ile ilgilenen departmanlar bulunmaktadır. Üçüncü kat ise tamamen yazılım geliştirme için ayrılmıştır ve bu katta 6 farklı ekip bulunmaktadır. Bunlar, Mobil Ekibi, Web Ürün Ekibi, Tasarım Ekibi, Optimus Ekibi, Tahmin Ekibi ve Kalite Kontrol Ekibi’dir. Mobil Ekibi, Android ve iOS platformlarındaki uygulamalar için ürün üretmektedirler ve bu ürünlerin entegrasyonu için birçok dil ve platform için *SDK* yani *Yazılım Geliştirme Kiti* geliştirmektedirler. Web Ürün Ekibi ise mobil ve masaüstü sitelerinde kullanılan web siteleri için ürün geliştirmektedirler. Tasarım Ekibi ise geliştirilen ürünlerin panel tasarımını, kullanıcı deneyim tasarımını ve deneyim testlerini yapmaktadır. Bu takımların yanında Optimus Ekibi Web Ürünleri için partnerlere entegrasyonda yardımcı olmaktadır ve gerekli kişiselleştirmeleri tamamlamaktadır. Tahmin Ekibi, makine öğrenme algoritmaları kullanılarak kullanıcıların, partnerlerin web sitesindeki aktivitelerine göre partnerin kullanıcılarına ürün önermesi için çalışmalarda bulunmaktadır. Kalite Güvence Ekibi ise tüm ekiplerle birlikte çalışmaktadır ve üretilen ürünlerin, güvenli, stabil ve sürdürülebilir olması için çalışmalar yapar. Staj dönemim süresince Kalite Kontrol Ekibi’nde görev aldım ve Mobil Ekibi tarafından üretilen ürünlerin kalite kontrol işlemlerini tamamladım.

Firmada çalışan stajyer, yarı zamanlı ve tam zamanlı çalışan sayısı sürekli artmaktadır ve 300’den fazla çalışan bulunmaktadır. “*Agile*” yani *Çevik* yazılım geliştirme yöntemleri ile çalışan firmada esnek çalışma saatleri uygulanmaktadır. Genel olarak rahat bir çalışma ortamına sahip olan şirkette yapılan projeler aksamadan ilerlemektedir. Firma çalışanlarına çok önem vermektedir ve çalışanları için sosyal aktivitelerle birlikte çalışanların kendilerini geliştirebilmesi için programlar düzenlemektedir.

2. GİRİŞ

Kalite ve güvence bölümünde mobil takım ile birlikte yaptığım ikinci zorunlu stajımda yazılım mühendisliği alanında kendimi geliştirme fırsatı buldum. Çevik geliştirme metodları nasıl uygulanır, yazılım üstünde iş birliği nasıl yapılır, kanban tahtalarının kullanımı ve ürünün geleceğini planlama hakkında çok fazla bilgi edinme imkanım oldu. Bunların yanında kurumsallaşmış ve uluslararası bir firmadaki bilgi akışı ve iletişim yapısını gözlemleme fırsatını elde etmiş oldum. Birden fazla derste öğrendiğim bilgilerimi uygulamada gösterme fırsatını yakaladığım ikinci stajımda sürekli geliştirme nasıl yapılır, servis olarak sunulan yazılım nasıl yönetilir ve bunun gibi bir çok tecrübe ve gözlem elde ettim. Ama asıl olarak testlerin nasıl yapılması gerektiğini, test sonuçlarının raporlamaların nasıl yapılması gerektiğini, testin önemini ve oluşabilecek hatalar için test senaryolarının üretilmesi üstünde kendimi çok geliştirdim. Bu stajda öğrendiğim bilgiler ve uyguladığım yöntemlerin mesleki hayatımda çok yararlı olacağını düşünüyorum.

Staj süresince üzerinde çalıştığım ürün, kullanıcılara çeşitli yollarla kampanya oluşturulmasını sağlayan bir web panel, Android ve iOS için kullanılabilen bir yazılım geliştirme kitinden oluşmaktadır. Yazılım geliştirme kitini kullanarak kendi mobil uygulamalarına ürünleri entegre eden partnerler daha sonra web paneli kullanarak kullanıcılar arasında çeşitli özelliklere göre segmentasyonlama, kampanya oluşturma ve A/B testlerini yapabilirler. Yazılım geliştirme kiti, Web Servisleri ile iletişim kurarak kullanıcıların özelliklerini aktarır ve gösterilmesi gereken kampanyaları aktive eder, aktive edilen kampanyalar ile etkileşime geçen kullanıcılar ise tekrar web servislerine bildirilerek istatistiksel veri olarak web panelinde görüntülenir.

Oluşturulabilecek kampanyalar arasında, bir çok çeşit “push” ve “in-app” kampanyaları bulunmaktadır. “push” kampanyalarında kampanya kullanıcılara bildirim gönderilerek iletilir. Bu bildirimler Android ve iOS işletim sistemlerinin kendi bildirim altyapısı kullanılarak iletilmektedir ve Android için “*Firestore Cloud Messaging*”, iOS için ise “*Apple Push Service*” hizmetleri kullanılmaktadır. Bu iki hizmet sırasıyla Google ve Apple şirketleri tarafından sunulmaktadır ve çalışma mantığı birbirine çok yakındır. Kullanıcıların cihazları için bu hizmetler tarafından oluşturulan özel bir anahtar bulunur ve bu anahtara “token” denir. Yazılım geliştirme süresinde, entegre edilen kit ile bu anahtar, bahsedilen hizmetler yardımıyla elde edilir ve web servislerine iletilir. Kullanıcılara bildirim gönderilmek istendiği zaman, Android işletim sistemi için “*Firestore Cloud Messaging*”, iOS işletim sistemi için ise “*Apple Push Service*” web servisleri ile iletişime geçilir ve bildirim gönderilmek istenen cihazların anahtarları iletilir. Bu kampanya çeşidine örnek olarak uygulama kapalıyken kullanıcıya gönderilen bir bildirim ile yılbaşı kampanyalarından haberdar etmek olabilir.

Diğer bir kampanya çeşidi ise “in-app” kampanyadır ve iletilmek istenen kampanya, uygulama içerisinde kullanıcıya bir diyalog gösterilmesi ile iletilir. Bu kampanyaların yönetimi ve kullanıcıya gösterilmesi yazılım kiti tarafından web servislerinin yardımı ile yönetilmektedir. Bu kampanya çeşidine örnek olarak uygulama açıldığında yalnızca bir kereye mahsus, kullanıcıya gösterilen bir diyalog ile yılbaşı kampanyalarından haberdar etmek olabilir.

3. STAJ PROJELERİNİN TANIMI VE ANALİZİ

Staj sürem boyunca takımdaki kalite güvence mühendisliği rolünü doldurdum. Mobil Takımıyla birlikte art arda iki farklı “sprint” koştuk ve bu “sprintlerin” planlamasından, değerlendirilmesine kadar her aşamasında bir katkı sağlayabildim. Takımdaki asıl görevim,

yapılan gelişmelerin ürüne eklenmeden önce ve eklendikten sonra gerçekleştirilmesi gereken modül testlerini ve entegrasyon testlerini gerçekleştirmek; oluşabilecek test senaryolarını hazırlayarak bunların testlerini tamamlayarak kısa bir rapor hazırlamaktı. Raporlar gayri resmi bir dille yazılmış olup, iletim kanalı olarak *Jira* adlı çevrimiçi yazılımı kullanmıştır. Bunların yanında projedeki paydaşların ürün hakkında, teknik olarak yaşadığı sorunların kaynağını tespit ederek raporlamak da benim görevlerim arasındaydı.

Sprint, *Scrum* yönteminde kullanılan bir süreci temsil etmektedir. Uyguladığımız *Scrum* yönteminde her gün güne başlamadan önce bir toplantı, *sprint* sonunda ve başında bir toplantı ve *sprintin* ortasında bir toplantı yaparak, ilerlemeyi takip etmeyi ve geri dönüşler ile gelişmenin devamlılığı sağlanmaktadır. Belirlenmiş bir yazılım geliştirme sürecini temsil eden *sprint* terimi değerlendirmelerin ve planlamaların yapılması için baz alınan süreyi temsil etmektedir. Mobil takımıyla, benim staj sürecim boyunca 2 haftalık 2 *sprint* süreci geçirdik. Yaptığımız toplantılarda yalnızca teknik konular değil, takım içi iletişimimizi, ürünün ekonomik ve pazarlama açısından durumunu da konuştuk. Yalnızca mühendislerin bulunmadığı aynı zamanda bir nevi proje yöneticisi görevi gören *ürün sahibi*, partnerlerle iletişim halinde olan ve ürünü kullanan çalışanlarla ve tasarım ekibinden bir kişiyle birlikte, *sprint* başında ve sonunda yaptığımız toplantılarda, görevlerin belirlenmesi, belirlenen görevlerin önceliği, eklenebilecek özellikler ve üründe bulunan hataları konuştuk. *Sprint* ortasında yaptığımız toplantılarda ise görevlerin durumunu ve önceliğini, eklenmesi ya da çıkarılması gereken görevleri tartıştık. Günlük gerçekleşen toplantılar ise yarım saati geçmeyen kısa toplantılar halinde yalnızca mühendislerin katılımıyla gerçekleştirildi. Bu toplantılarda yaptığımız ve o gün yapacağımız görevlerin üstünden kısa bir şekilde geçtik. *Scrum* yönteminin gerekliliği olan bu toplantılar ve süreçler, ürün sahibi ve scrum yöneticisi tarafından gözlenmekte ve yönetilmektedir.

Jira adı verilen çevrimiçi yazılım, proje yönetimi için kullanıcılarına birçok metrik ölçüm aracı, özelleştirilebilen *kanban tahtaları* ve kullanıcıların rahatça görevleri takip edebilmesi için gelişmiş bir arayüz sunmaktadır. *Kanban tahtası* üzerinde görevlerin ve bu görevlerin durumlarının bulunduğu bir tahtadır. Bizim kullandığımız *Kanban tahtası* “To-Do”, “In Progress”, “Code Review”, “Testing”, “UAT” yani “User Acceptance Test” ve “Done” adımlarını içermektedir. Stajım süresince “Testing” aşamasına gelen görevlerin testlerini tamamlamak ve yapılan testlerle ilgili kısa bir rapor hazırlayarak *Jira* yazılımında görev ile ilgili yere bu raporu ekleyerek görevimi gerçekleştirdim. Eğer test senaryolarının birisini geçemeyen bir görev olursa bu görevi tekrar “To-Do” aşamasına göndermek de benim görevlerim arasındaydı. Verilen görevler ise yeni eklenen bir özellik veya bu özelliğin bir parçası, daha önce eklenen özelliklerden birinde oluşan bir hata yani “bug” şeklinde olabilmektedir.

Geliştirme sürecinde yaptığım görevlerin yanında, entegrasyonu tamamlamış partnerler için *duman testleri* ve *entegrasyon testlerini* tamamlayıp rapor çıkarmak da benim görevlerim arasındaydı. *Duman testleri*, bir sistemin en basit fonksiyonlarının test edildiği testlerdir. Bu durumda uygulamanın açılması ve uygulamadaki sayfaların doğru bir şekilde yüklenmesi yeterlidir. Entegrasyon testleri ise uygulama içinde entegrasyon ile eklenen ve bizim ekibimizin geliştirdiği özelliklerin testlerini kapsamaktadır. Bu tip testleri *sprint*’den bağımsız olarak tamamladım.

Test ettiğim görevleri, detaylı test raporlarını ve ürünün teknik detaylarını gizlilik sözleşmesi nedeniyle staj raporumda paylaşamıyorum.

3.1 Birinci Sprint

Bu iki haftalık süreçte, birçok görevin testlerini gerçekleştirdim. İlk fark ettiğim durum, test raporlarının ne kadar önemli olduğuydu. İlk başlarda yazdığım raporların yetersiz olması sebebiyle iletişim problemleri yaşadık ve test senaryolarının doğru aktarılamaması sonucu birçok test senaryosu yanlış anlaşıldı. Daha sonradan test raporlarımı, izlenecek adımlar, beklenen sonuç, şu an gerçekleşen durum olarak yeniden düzenlediğimde daha hızlı ve akıcı bir iletişim kurarak, daha hızlı bir işleyiş sağladık. İzlenecek adımlar bölümü testi gerçekleştirirken yapılması gereken her şeyi tek tek ve detaylı olarak aktaran bölümdür. Beklenen sonuç, ürünün doğal akışı veya kapsamı gereğince beklenen sonuçtur. Bu sonuç bir hata mesajı, herhangi bir mesaj, yeni bir durum, yeni bir sayfa vb.... gibi olabilir. Gerçekleşen durum ise beklenen sonuçtan farklı olarak karşılaştığımız hata mesajı, herhangi bir mesaj, yeni bir durum, yeni bir sayfa vb.... olabilir. Entegrasyon testi yaparken kullanılan raporları ise partnerlerin isteğine ve kendi ihtiyaçlarıma göre düzenledim. Ayrıca ilk *sprint*'de ürünün yapısını daha iyi kavrayarak, birden fazla dilde geliştirilmiş bir yazılım geliştirme kitini, birden fazla *mikroservisi* ve yazılım panelini yakından inceleme fırsatını elde etmiş oldum.

Mikroservis, arka planda çalışan ve kullanıcının görmediği ancak ürünün bazı fonksiyonlarının gerçekleşmesini sağlayan, gerektiğinde tek başına çalışabilen bir sistemdir. *Mikroservislerin* gerçekleştirdiği fonksiyonlar, ürünün bazı fonksiyonları veya ürünün fonksiyonlarının gerçekleşmesine yardımcı olan fonksiyonlar olabilir.

3.2 İkinci Sprint

Stajimin sonraki iki haftalık sürecinde, ilk iki haftadan öğrendiklerimi kullanarak takımla birlikte daha iyi çalışmaya başladım. Daha sonra uyguladığım test senaryoları sonucunda, üründe bulunan diğer hataların ortaya çıkmasını sağladım. Bulduğum bu minör hataları raporlayarak, daha sonraki *sprint*'lerde tamamlanması için *Jira* üzerinde görevlerini oluşturdum. Ayrıca bu *sprint*'de yalnızca web teknolojilerini ve *mikroservisleri* içeren panel dışında, Android ve iOS platformlarında da test yaparak, yazılım bilgilerimi daha da genişlettim. Bu platformlarda kullanılan *Java* ve *Swift* gibi dillerde ve *CocoaPods*, *Gradle* gibi geliştirici araçlarında kendimi geliştirme fırsatı buldum.

Java nesne tabanlı bir programlama dilidir ve Android mobil platformu için kullanılan ana dil konumundadır. *Gradle* ise Android platformu için uygulama geliştirmede kullanılan, birçok derleme ve paketleme işlemini otomatik hale getirmiş hazır proje örnekleri bulunan bir yazılım üretme ve derleme aracıdır. *Swift*, *Apple* adlı firma tarafından geliştirilen ve Mac ile iOS platformlarında yazılım yazılması için kullanılan ana dil konumundadır. *CocoaPods* ise *Swift* ile yazılan projelerde kullanılan, projede kullanılan diğer yazılımlarla projenin bağımlılık durumunu yöneten bir paket yöneticisidir.

3.3 Duman ve Entegrasyon Testleri

Entegrasyonu tamamlamış olan partnerler için uyguladığım test tipi *duman testleri* ve *integrasyon testleridir*. *Duman testleri*, bir sistemin en basit fonksiyonlarının test edildiği testlerdir. *Entegrasyon testlerinde* amaç, bizim geliştirdiğimiz yazılımın partnerin uygulaması ile etkileşimi ve yazılım geliştirme kitinin doğru uygulanıp uygulanmadığını test etmektir.

Duman test senaryoları arasında, uygulamanın açılması ve web servisi ile iletişim kurulmasının test edilmesi, ürünün çalışması için temel sayılabilecek cihaz bilgilerinin alınabilmesi ve bu bilgilerin web servisine iletilmesi bulunmaktadır. Bu temel bilgiler arasında cihaza bildirim gönderebilmek için gereken anahtar yani “*token*” ve *cihazın benzersiz kimliği* gibi bilgiler bulunmaktadır. *Cihazın benzersiz kimliği* uygulamanın yüklendiği cihazı ayırt etmek için kullanılmaktadır ve her uygulama için farklı bir kimlik elde edilmektedir. Bu bilgilerin web servise iletilip iletilmediğini web servise ait olan veritabanı veya şirket içinde kullanılan özel bir “*endpoint*” ile kontrol edilebilir. Veritabanı için *SQL* sorguları kullanılmıştır. *SQL* veritabanları için kullanılan alana spesifik bir dildir. “*endpoint*” ise web servisiyle veri alış verişini yapmayı veya web servisine komut göndermeyi sağlayan, *http* gibi belirli protokoller üstüne kurulmuş olan sistem giriş ve çıkışlarıdır. Cihazdan alınan verilerin doğruluğu ve uygulamanın açılması bu yöntemlerle kontrol edilerek duman testleri tamamlanmıştır.

İntegrasyon testleri ise, partnere sunduğumuz ürünlerin hepsini test ettiğimiz testlerdir. Bu testlerin test senaryoları arasında, “*push*” kampanyası oluşturma, bu kampanya ile etkileşime geçme ve bu etkileşim sonucunun web servise yazılıp yazılmadığını kontrol etme gibi senaryolar vardır. “*in-app*” kampanyası oluşturma, bu kampanya ile etkileşime geçme ve bu etkileşim sonucunun web servise yazılıp yazılmadığını kontrol etme ve bunun benzeri test senaryoları da bulunmaktadır. Ayrıca bu senaryoları uygulamanın üç farklı durumu için de test etmek gereklidir. Bu üç durum “*önplanda*”, “*arka planda*” ve “*ölü*” durumudur. “*Önplanda*” durumu uygulamanın cihaz ekranında görüntülediği ve kullanıldığı durumdur. Uygulama donanım kaynaklarını bu durumda kullanabilir. “*Arka planda*” durumu ise uygulamanın halen donanım kaynaklarına kısıtlı bir şekilde eriştiği ancak cihaz ekranında görüntülenmediği durumdur. Bu durumdaki uygulamalar görev yöneticisinde veya uygulamalar arası geçiş ekranında görüntülenebilir. “*Ölü*” durumu ise uygulamanın tüm işlemlerinin sonlandırıldığı ve donanım kaynaklarına erişemediği durumu ifade etmektedir. Bu durumdaki uygulamalar görev yöneticisi veya uygulamalar arası geçiş ekranında görüntülenmezler.

Duman ve İntegrasyon test yöntemleri, diğer testlere göre bu durumda daha avantajlı olduğu için tercih edilmiştir. Öncelikli olarak partner uygulamasının kaynak kodlarına sahip olmadığımız ve yalnızca derlenmiş uygulamayı kullandığımız için, yapılan testler *kara kutu testleri* olmak zorundadır. *Kara kutu testleri* yalnızca girdi ve çıktılar test edildiği, sistemin içeriğinin bilinmediği ve her modülün ayrı ayrı test edilmediği test tipidir. İlk önce duman testi yapılmasının sebebi ise, İntegrasyon testine harcanan vakiti azaltmaktır. Eğer duman testleri başarısız olursa integrasyon testlerinin başarılı olması imkansız olduğu için duman testleri başarısız olan uygulamaların testleri tamamlanmaz ve zaman kazancı sağlanmış olur. Ayrıca integrasyon testlerini tamamlamak sistemin çalışacağını garantilediği için yalnızca bu testleri tamamlamak yazılımın tamamını kapsamak için yeterli olmaktadır.

3.4 Testler Dışındaki Görevler

Uyguladığım testler dışında ürünün paydaşları ve kullanıcıları tarafından tespit edilen hataların yeniden üretilmesi ve bu hataların doğrulanması üzerine de bir kaç farklı görev tamamladım. Bu hataları yeniden ürettikten sonra hatanın tekrarlanması için gerekli adımları detaylı olarak yazarak takım arkadaşlarım için gerekirse yeni görevler oluşturdum, gerekirse iletişime geçerek hatanın giderilmesi için çalıştım. Bu hataların kontrolü sırasında web servisindeki “*endpoint*”leri ve *SQL* ile veritabanını kontrol ettim. Bunun dışında test cihazlarına bahsi geçen uygulamayı yükleyerek hatanın adımlarını tekrarladığım durumlar da oldu. Bu duruma örnek olarak, bir partnerin “*push*” kampanyası oluşturmaya rağmen kullanıcılara bildirim gitmemesi durumunu araştırırken, hem yazılım geliştirme kitinde hem de

web servisinde cihazın “token”’inin alınması ile ilgili bir hata olduğunu farkettim. Hatanın detaylarını gizlilik sözleşmesi sebebiyle paylaşamamaktayım. Hataya sebep olan senaryo ise, bir kullanıcının uygulamayı ilk defa açtığı anda bildirimlere izin vermemesi ancak daha sonradan izin vermesi idi.

4.SONUÇ

Yazılım mühendisliği ve yazılım testi alanında kendimi geliştirmemi sağlayan ikinci stajımda, çevik geliştirme metodunun, *scrum* metodunun ve *kanban* tahtalarının kullanımı hakkında tecrübe edindim. Aynı zamanda testlerin nasıl yapılması gerektiği konusunda da çok fazla bilgi ve tecrübe kazandım. Bu stajın hayatımın ilerleyen dönemlerinde mesleğimi şekillendirmem için bana çok yararı dokundu. Verimli bir ay boyunca yazılım sektörünü ve kurumsal bir firmada uygulanan çevik geliştirme metodlarını yakından tanıma fırsatı elde ettim.

5.REFERANSLAR

Çilingir, H. (2016) Röportajı CNNTürk yapmıştır, 8 Ocak 2016. Erişim: <https://www.cnnturk.com/bilim-teknoloji/teknoloji/insider-webi-kisisellestiren-teknolojiyi-sunuyor> (11.10.2018’de erişildi)

Apple Inc. (2018) Apple Geliştirici Dökümanı. Erişim: https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html#//apple_ref/doc/uid/TP40008194-CH8-SW1 (17.12.2018’de erişildi)

Google Inc. (2018) Firebase Geliştirici Dökümanı. Erişim: <https://firebase.google.com/docs/cloud-messaging/> (17.12.2018’de erişildi)