



Derleme Makalesi

Bankacılık Sektöründe Yazılım Test Süreçlerinin Önemi ve Yöntemleri

İsmail Öztürk*¹

¹Fibabanka Ar-Ge Merkezi, Şişli, İstanbul, Türkiye

Öz

Anahtar Kelimeler:

Performans testi
Test otomasyonu
Manuel test
Bankacılık
Dijital banka

Günümüzde ekonomik büyümeyi finanse etmenin ve bunu sürekli hale getirmenin en etkili yolu finans sektörünün güçlü ve sağlıklı bir yapıya sahip olmasıdır. Tüketim, ticaret, yapım ve üretimin finanse edilmesinde, kurumlar ve bireyler için kredinin bulunabilirliği ve bunun devamlılığının sürdürülmesi hayati öneme sahiptir. Günümüzde finansal kurumların temel işlevi müşterilerinin ve ekonominin talep ettiği finansal hizmetleri sunmaktır. Sunduğu hizmetlerle mobil ödeme sisteminin doğru çalışmasını sağlamak olan finansal kurumlar ekonomik faaliyetin önemli bir parçasıdır. Bu kritik alanda kullanılan yazılım ürünlerinin kalitesi çok önemlidir. Yazılım ürünlerindeki hatalar maddi, manevi zararların oluşmasına, müşterilerde kuruma karşı güven ve sadakat olgularının yitirilmesine neden olarak finansal kurumların kimliğini zedeler. Günümüzde internetin kullanımının yaygınlaşmasıyla birlikte Mobil Bankacılık gibi alternatif uygulama kanalları müşteriler tarafından sıkça kullanılmaya başlanmıştır. Uygulama kanallarının kullanımının yaygınlaşmasıyla birlikte uygulamalar eleştiriler almaya başlamış ve bu alanda bir rekabet ortamı oluşmuştur. Bu rekabet, kurumların müşteriye sundukları yazılım ürünlerindeki hataların en aza indirgenmesi için yeni metotlar arayışına girmelerine sebep olmuştur. Yazılım ürünleri için test süreci, yazılım yaşam döngüsünün vazgeçilmez bir parçası haline gelmiştir.

The Importance of Software Testing Processes and Methods in the Banking Sector

Keywords:

Performance testing
Test automation
Manual testing
Banking
Digital bank

ABSTRACT

The most effective way to finance and sustain economic growth is to have a strong and healthy financial sector. The availability and maintenance of credit is vital for institutions and individuals in financing consumption, trade, construction and production. Financial institutions, whose main functions are to provide financial services needed by their customers and the economy, and to ensure the functioning of the payment system, are an important part of economic activity. The quality of software products used in such a critical area is very important. Errors in software products damage the identity of financial institutions by causing material and moral damages, and loss of trust and loyalty in customers to the institution. Today, with the widespread use of the internet, alternative application channels such as Mobile Banking have started to be used frequently by customers. With the widespread use of application channels, applications have begun to receive criticism and a competitive environment has emerged in this field. This competition has caused institutions to seek new methods to minimize the errors in the software products they offer to the customers. The testing process for software products has become an indispensable part of the software lifecycle.

*Sorumlu Yazar

** (ismail.ozturk@fibabanka.com.tr) ORCID ID 0000-0002-2968-5221

1. GİRİŞ

Yazılım testi, gereksinimler doğrultusunda yazılımın kalitesinin ölçülmesi, yazılımdaki hata oranlarının gösterilmesini, yazılımdaki hataların en az seviyeye getirilmesini, yazılımdaki hataların erken bulunmasını ve bu sayede yazılım maliyetinin düşürülmesini, müşteri memnuniyetinin en üst seviyede tutulmasını hedefleyen süreçtir. (International Software Testing Qualifications Board, 2011).

Yazılım testi aşağıdaki maddelerin sağlanması için yapılmaktadır:

- Kodlanan yazılımın gerekliliklerini yerine getirebildiğini göstermek,
- Yazılımın planlanmış olduğu gibi ilerlemesini sağlamak,
- Projeye ayrılan bütçeyi aşmadan projeyi ilerletmek,
- Yazılımı daha kaliteli hale getirmek,
- Yazılımın en az hataya sahip olmasını sağlamak,
- Yazılımın sürdürülebilir olmasını sağlamak.

2. FİNANS SEKTÖRÜNDE YAZILIM TESTİNİN ÖNEMİ

Finans sektörü, sektörde meydana gelen birçok değişiklikten dolayı oldukça dinamik bir sektördür. Bu dinamik sektör test etme ve kalite güvencesi kontrolleri için sürekli bir zemin hazırlamaktadır. Sektörü dinamik hale getiren bu değişimlere örnek olarak müşterilerin bankacılık, sigorta ve finans sağlayıcılarıyla etkileşim biçimindeki değişiklikler, temassız ödemenin toplu olarak benimsenmesi ve rekabetçi bir ortam sağlamayı amaçlayan düzenleyici değişiklikler sayılabilir. Yazılım ve test sektörünün en karışık uygulamaları bankacılık uygulamaları olarak kabul edilir.

Bankacılık uygulamalarının en karışık uygulamalar olmasının sebebi ise eş zamanlı birçok kullanıcının sistemi aktif olarak kullanabilmesi için sistemin bir dizi çok katmanlı işlevsel yapı ile desteklenmesidir. Bankacılık uygulamaları sahip oldukları birçok çevrimiçi bağlantılardan dolayı sürekli bir iş akışına ve aynı anda yapılan toplu işlemler nedeniyle dakikada yapılan işlem sayısı birçok alandaki uygulamaya göre çok daha fazladır.

Bu kadar fazla toplu işlemin yapıldığı ve sürekli aktif olarak çalışan bu uygulamalarda güvenlik faktörü çok önem arz etmektedir. Sürekli raporlama sistemi olduğundan dolayı her zaman iç ve dış denetimden geçmelidir. Bankacılık uygulamaları her türlü olağandışı

durumda işlevlerini yerine getirmeye devam edecek bir alt yapıya sahiptir. Banka uygulamaları güvenlik bakımından risklerin göze alınamayacak kadar hassas olduğu uygulamalardır. (Karagöz ve Molu, 2017)

3. YAZILIM TESTİ SÜRECİ

Geliştirilen yazılımların minimum seviyede hata içermesi ve gereksinimleri en iyi seviyede karşılayabilmesi için yazılım test adımları, yazılım geliştirme sürecinin en başından yerini almalıdır. Yazılım dünyasındaki hata örnekleri incelendiğinde, çıkan hataların çoğunlukla analiz ve tasarım aşamasındaki hatalardan meydana geldiği görülmektedir. Analiz ve tasarım aşamasında bulunan hatalar projenin ilk safhalarında fark edilmediğinde, projenin sonraki fazlarında bu hataların düzeltilmesi için ekstra maliyet ve zaman harcanması gerekmektedir.

Yazılım testi, yazılım geliştirme süreci ile eş zamanlı devam eden aşamalardan oluşan bir süreçten meydana gelmektedir. Bu aşamalar aşağıdaki gibi özetlenebilir:

- Test Hazırlık Süreci
- Test Planlama, Gözetim ve Kontrol
- Test Analizi ve Tasarımı
- Test Uyarılama ve Koşma
- Çıkış Kriterlerini Değerlendirme ve Raporlama
- Test Kapanış Aktiviteleri

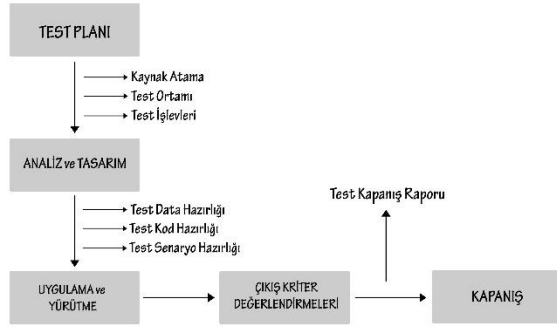
3.1. Test Hazırlık Süreci

Projeyi test edecek olan test mühendisi, öncelikle projeyi anlar ve öğrenir. Projedeki karmaşık ve risk teşkil eden kısımları tespit eder. Bu hazırlık sayesinde projede tespit edilen risklerin öncelik sırasını belirlemiş olur. Mühendisin önceden yoğunlaşacağı kısımları görmesi daha etkili çözümler yaratmasını sağlar.

3.2. Test Planlama, Gözetim ve Kontrol

Test planlama aşaması test süreci bitene kadar devam eder. Test gözetimi ve test kontrol aşamaları da aynı şekilde test süreci bitene kadar devam eder. Test süreci boyunca testin plana uygunluğu kontrol edilerek plandan sapmalar tespit edilirse nedenleri tespit edilerek yeni planlar oluşturulur. (Black R, Chapter 1,9)

İncelemelerden sonra roller ve sorumluluklar belirlenir, risk analizi yapılır. Test planlaması, aşağıdaki ana görevlerden oluşmaktadır.



Şekil 1. Test Planlaması Şeması

3.2.1. Risk Analizi

Ürünün risk analizi yapılır. Risk analizi, ürünün kullanılabilirliği, tüketilebilir olması, revize ve test edilebilirliği ile alakalıdır. Risk analizi, yapılacak testin süresi hakkında kritik bir girdi sağlar. Yapılan analiz sonucunda riskin fazla olduğu kısımlara daha fazla odaklanılır. Uygulanacak teknik belirlenir ve önceliğe göre sıralamalar yapılır. Testin en önemli amaçlarından biri risklerin en aza indirilip gerekli önlemlerin alınmasıdır. (Karagöz ve Molu, 2018)

3.2.2. Test Stratejisinin Planlanıp, Uygulanması ve Çıkış Kriterlerinin Belirlenmesi

Test stratejisinin doğruluğu ve tutarlılığı değerlendirilir ve süreç boyunca kontrol edilir. Teste başlamadan önce tüm planlamaların yapılması ve görevlerin belirlenmesi gerekmektedir. Planlamalar oluşabilecek risk ve hataları öngörülebilir hale getirebilmektedir, ayrıca bir risk durumunda planlama yeniden yapılmalıdır.

Projenin müşteriye ya da üretim ortamına sunulabilmesi için testlerin belirli bir seviyede tamamlanmış olması gerekir. Çıkış kriteri, testlerin tamamlandığının bilgisinin verilmesi için gereken kriterlerdir.

Yapılan test planlama aktiviteleri her ne kadar başarılı ve hatasız olarak yapılmış olsa bile yapılan plan ve aktiviteler sürekli olarak kontrol edilmelidir. Yapılan planlama ve gerçek ilerleme karşılaştırılmalı ve planda herhangi bir değişiklik veya sapma olması durumunda proje yöneticisi ve müşteriye mevcut test durumu hakkında bilgi verilmelidir. (Hancı, 2017)

Yapılan test durumları ve kontroller sayesinde projenin amaçlarını yerine getirmek için gerekli olan yerlerde önlem alınması

kolaylaşacaktır. Test kontrolleri aşağıdaki maddelere göre yapılmalıdır:

- Testler sırasında bulunan hataların sayısı, türü ve öneminin bilinmesi. Tüm test senaryolarından kaçının tamamlandığı ve kaçının başarısız olduğunun izlenmesi gerekir,
- Yazılım projesinin yöneticisi, müşteri ve diğer paydaşlara yazılım projesinin durumu hakkında çeşitli kararlar verebilmelerine yardımcı olacak düzenli raporlar vermelidir.

Bunların haricinde uygulamanın özelinde o projeye uygun kriterler belirlenir. Örneğin bir şube mobil uygulamasında hesap adı değiştirme gibi işlemler ile finansal işlemler aynı önemi, işlem yoğunluğunu ve riski taşımaz. Finansal işlemlerin düzgün çalışması bir şube mobil uygulama projesinin test çıkış kriterlerinden biri olarak gösterilebilir.

3.3. Test Analizi ve Tasarımı

Test analizi aşaması test esasları adı verilen proje gereksinimlerinin çıkarılabileceği belgelerin analiz edilebileceği aşamadır. Test koşulları göz önünde bulundurularak, test edilecek kısımlar belirlenir. Test koşullarının oluşturulması birçok etkene bağlıdır. Koşulların tanımlanma detayı her projenin kritikliğine göre değişiklik gösterebilir.

Proje gereklilikleri ayrıntılı bir biçimde incelenir, gereklilikler küçük bölümlere ayrılır ve bu bölümlere uygun test koşulları tanımlanır. Böylece yazılım projelendirilmeye başlandığında hata oluşma riski en aza indirilmiş olur. Özellikle finans sektöründe kullanılan karmaşık işlemler içeren uygulamalarda bu aşama oldukça önem arz eder.

Test tasarımı analiz ise sonucu tanımlanan veriler ve koşullar ile tasarım tekniğinin belirlenmesi aşamasıdır. Bu aşamada yazılımın nasıl test edileceği hakkında senaryo oluşturulur.

Oluşturulan test senaryoları iki başlık altında sınıflandırılır. Bunlardan birincisi, somut olan test senaryolarıdır. Somut test senaryoları proje gerekliliklerinin açık bir şekilde tanımlandığı senaryolar olarak tanımlanır. İkincisi ise, mantık temelli test senaryolarıdır. Test süreci koşullanırken proje gerekliliklerinin kapsama alanı genişletilerek oluşturulan senaryolardır.

Oluşturulan test senaryoları açık ve anlaşılır olmalıdır. Test senaryoları risk kapsamının belirlenmesi aşamasında kritik önem taşır. (Black R, Chapter 1,11)

3.4. Test Uyarlama ve Koşma

Test uyarlama; test aşamalarının tam olarak belirlendiği ve test koşulunun gerçekleştirilmesi için gerekliliklerin sağlandığı aşamadır. Bu aşamada, test verileri oluşturulur ve test edilecek ortamın test koşumu için uygun olup olmadığı durumu kontrol edilir.

Test koşumu; test sürecinin hazırlıkları tamamlandıktan sonra test senaryolarının koşulduğu aşama olarak tanımlanır. Test koşumu sırasında karşılaşılan sonuçlarla test adımlarının gerçekleştirilmesi ile beklenen sonuçlar karşılaştırılır.

Kullanıcı ihtiyaçları, yazılım mimarisi, uygulama tasarımı, kullanıcı ara yüzleri, test verileri, test ortamı ve araçları göz önünde bulundurularak uygulanacak test teknikleri belirlenir. Projeye uygun test senaryoları hazırlanır ve projenin amaçlarına göre çeşitlendirilir. (Hancı, 2017)

3.5. Çıkış Kriterlerini Değerlendirme ve Raporlama

Test uyarlama ve koşumu aşaması tamamlandıktan sonra test sonuçlarının tespit edilip raporlandığı aşamadır. Test koşumu sonucunda elde edilen veriler ile başlangıçta öngörülen test hedeflerinin gerçekleşme oranı belirlenir ve karşılaştırılır. Test koşumu sırasında tespit edilen hatalar olursa bu hatalar önceliklerine ve önemlerine göre, bu aşamada sınıflandırılır.

Testlerin sonucunda raporlandırılan verilerden yola çıkılarak ürünün süreci hakkında kararlar verilir.

3.6. Test Kapanış Aktiviteleri

Test kapanış aşaması test süreci boyunca elde edilen çıktıların bir araya getirilerek arşivlendiği aşamadır. Arşivlenen test verileri gerektiğinde diğer benzer projelendirmelerde, mevcut projeye bakış açısı kazandırmak üzere muhafaza edilir.

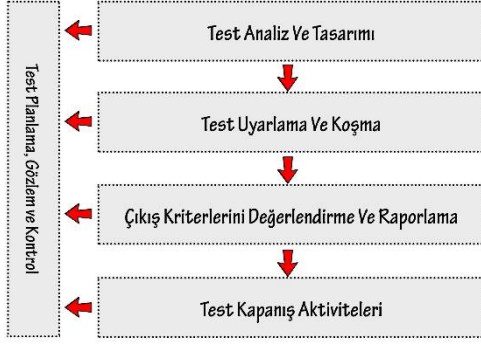
Uygulamanın belirlenen ihtiyaçları karşılayabilecek hale gelmesi, test sürecinde karşılaşılan hataların giderilmesi, kullanıcıya sunulduğunda risk oranının en az değerine indirilmesi halinde test çıkış kriterlerine uygunluğu irdelenerek “Test Kapanış Raporu” hazırlanır. Rapor hazırlandıktan sonra test sonlandırılır.

Testin süresi proje kapsamındaki karşılaşılan ve çözümlenen risklerin miktarına bağlı olarak değişmektedir. (Uzun, 2019)

Kodlamanın bitmesine yakın bir zamanda “Dinamik Test Süreci” başlar. Dinamik test süreci, uygulamada bulunan tüm hataların çözülmesini ve test sürecini sonlandırmak için ulaşılmaması gereken gereksinimlerin sağlanmasını hedefler. Dinamik test süreci içerisinde ve sonrasında genel olarak uygulanan bazı teknikler vardır. Bu teknikler şu şekilde sıralanır;

- **Birim Testi:** Dinamik test sürecinin ilk aşamasıdır. Bu aşama hataların tespit edilip düzeltilmesi açısından en önemli aşamadır.
- **Tümleyim Testi:** Uygulamadaki birbirinden farklı bileşenlerin bir arada uyum içerisinde çalışıp çalışmadığına bakılıp, test edildiği aşamadır.
- **Regresyon Testi:** Uygulamada gerekli değişiklikler yapıldıktan sonra yeniden yapılan testlerdir. Böylelikle sistemde çözülen sorunların yeniden meydana gelip gelmediği test edilir.
- **Zorlanım - Performans Testi:** Bu test “Yük Testi” ile aynı anlamda kullanılabilir. Buradaki yük kavramı uygulamadaki çok yoğun sayısal işlemler, karmaşık sorgulamalar gibi yoğun işlemlerdir. Bu işlemler sırasında sistemin işlevselliğinin ölçüldüğü test olarak adlandırılabilir.
- **Kullanıcı Kabul Testi:** Müşteri veya son kullanıcı isteklerine dayanan son test işlemidir. Aynı zamanda uygulamanın sunulduğu kişilerin uygulamayı kabul etmeden önce, uygulamanın gereksinimlerini ne derecede karşıladıkları hakkında geri dönüşlerini de belirtebilecekleri testlerdir.

Yapılan tüm testlerin sonucunda tespit edilen hataların düzeltilmesinin ardından, test hazırlık süreci tekrardan kontrol edilir. Tüm kriterler istenen düzeyde sağlanırsa test sonlandırılır. Test sonlandırıldıktan sonra uygulama kullanıcıya sunulur. Kullanıcı uygulamada bulduğu hataları veya değiştirilmesini istediği noktaları belirtir (Kullanıcı Kabul Testi). Belirtilen kısımlar yazılım ekibi tarafından gözden geçirilir ve test ekibine kontrol amaçlı sunulur. Tekrarlanan kontroller sonucu uygulama projesi, ürün aşamasına geçerek tüm kullanıcılara sunulur. Kullanıcıya sunulan uygulamanın yazılım test süreci sona ermiş olur. Böylelikle yazılım geliştirme sürecinin son basamağına gelmiş olunur. (İTÜ Bilgi İşlem Daire Başkanlığı, 2013)



Şekil 2. Yazılım Testi Süreçleri

4. YAZILIM TEST YÖNTEMLERİ VE TEST ÜRÜNLERİ

Yazılım testleri, yazılım ürünlerinin beklendiği gibi çalıştığını doğrulama işlemidir. Yazılım testini gerçekleştirmek için iki yöntem kullanılır. Bu yöntemler, manuel veya bir otomasyon aracı kullanarak yazılımı test etmektir.

4.1. Manuel Test

Manuel test bir yazılımın kalite güvence analistleri tarafından manuel olarak test edilmesidir. Geliştirilmekte olan yazılımların hatalarını tanımlamaya yardımcı olur.

Manuel test süreci, projelendirilen yazılımın hem görünür olan hem de gizli olan hatalarını bulabildiği için en temel test süreçlerinden biri olarak bilinmektedir. Otomasyon testi yapılmadan önce, yeni geliştirilen her bir yazılım projesi için manuel test zorunludur. Manuel testin başarılı olması için, bir testçinin öncelikle gereksinimleri, yani yazılımın nasıl çalışması gerektiğini bilmesi gerekir. (Arslan,2020)

4.1.1. Manuel Olarak Yapılabilecek Testler

- **Araştırma Testi:** Keşif testidir.
- **Kullanılabilirlik Testi:** Ürünü en son haliyle kullanan kullanıcılar üzerinden gerçekleştirilen test tekniğidir.
- **Geçici Test:** Test süreci boyunca gerçekleştirilen tüm testlerin sürelerinin belirlendiği test tekniğidir.
- **Beta Testi:** Uygulamanın oluşturulan ilk yazılım sürümü sırasında uygulanan test tekniğidir.
- **GUI Yazılım Testi:** Kullanıcı ara yüzünün ve uygulamanın kullanıcılara vadettiği olanakları sunup sunmadığının kontrol edildiği test tekniğidir.

4.2. Otomatik Test (Test Otomasyonu)

Otomatik test yazılım testinde, testlerin yürütülmesini kontrol etmek için özel yazılım araçlarını kullanan bir test yöntemidir. Aynı zamanda yapılan testlerin sonuçlarını, öngörülen sonuçlarla karşılaştıran bir test yöntemidir. Tüm bunlar otomatik olarak test mühendisinin müdahalesi çok az veya hiç olmadan yapılır.

Otomasyon, manuel olarak gerçekleştirilmesi çok zor olabilecek ek testler için kullanılır. Bazı yaygın otomatik test araçları Selenium, Appium ve Test Studio'dur. (Yener vd., 2019)

4.2.1. Bazı Yazılım Geliştirme Araçları

Appium, mobil cihazlarda bulunan mobil web ve melez(hibrit) uygulamalarında istenilen testlerin en uygun şekilde yapılmasını sağlayan açık kaynaklı bir pakettir.

Selenium, kullanılan web uygulamalarının birçok platform üzerinden istenilen testlerin uygun olarak yapılmasını sağlayan açık kaynaklı bir yazılımdır. Yazılım kodu ile web tarayıcıları arasında iletişim kurulabilmesini sağlar.

MF Unified Functional Testing (UFT), daha önce QuickTest Professional (QTP) olan MF Unified Functional Testing (UFT), fonksiyonel test otomasyonu için en sık kullanılan ticari araçtır. MF UFT, masaüstü, mobil ve web platformlarında otomatik test gereksinimlerini kapsamlı olarak karşılayabilecek özellikler dizisi sunar.

Cucumber, uygulama davranışlarını geliştirme sürecini benimseyen yazılımcılar tarafından kullanılan bir yazılım paketidir. İstenilen davranışların müşteriler tarafından anlaşılabilir bir biçimde oluşturulmasını sağlar.

Jmeter, performans testlerinde sık kullanılan bir araçtır. Kullanıcıların üretilen bir web uygulamasını kullanırken sunuculardan bekledikleri talepler (web istekleri) simüle edilir. jMeter ile birden fazla kullanıcı aynı anda aynı senaryoyu çalıştırmış gibi kurgulanarak sistemde istenen büyüklükte bir yükler oluşturulabilir ve bu şekilde sistem test edilebilir.

4.2.2. Otomatik Olarak Yapılabilecek Testler

- **Gerileme (Regression) Testi:** Uygulamada yapılan değişikliklerin kontrol edildiği test tekniğidir.
- **Mantık (Sanity) Testi:** Yazılım güncellemeleri sonrasında yapılan performans testidir.

- **Tekrarlanan Testler (Repeated):** Süreç boyunca devamlı olarak yapılan test tekniğidir.
- **Performans Testi:** Sistemdeki performans ve sistem bileşenlerinin uygunluğunun kontrol edildiği test tekniğidir.
- **Yük (Load) Testi:** Programın kapasite ölçümü için yapılan test tekniğidir.

4.3. Manuel ve Otomasyon Testinin Karşılaştırılması

Tablo 1. Otomasyon ve manuel testin özellikleri

	OTOMASYON	MANUEL
Maliyet	Kullanılan araçlar ve devamlı olarak bakım gerektirmesi dolayısıyla pahalı olabilir. Fakat açık kaynaklı araçlar kullanılır ise maliyet düşer.	İnsan gücüne başvurduğu için personel maliyeti doğurur.
Verimlilik	Tekrar eden testler için uygundur.	Tekrar eden testler için uygun değildir.
Raporlama	Çeşitli uygulamaları desteklediği için otomatik bir şekilde düzenli rapor tutulabilir.	Veriler testi yapan kişi tarafından kayıt edilmelidir.
Kapsam	Sadece kodlanan kısımları kontrol eder.	En küçük ayrıntı bile test edilebileceği için daha geniş kapsamlıdır.
Zaman	Hızlıdır.	Yavaştır.
Gözlem	Kullanıcı deneyimi sağlayamaz.	Kullanıcı deneyimi sağlar.
Güvenilirlik	İnsan kaynaklı hatalar görülmez.	Manuel testler hiçbir zaman tam olarak tekrar edilemez. İnsan hataları görülebilir.
Hata Bulma Oranı	Manuel teste göre daha	Otomasyon sonucu bulunan

	fazla hata bulunmasını sağlar.	bazı hatalar bulunamayabilir.
--	--------------------------------	-------------------------------

5. SONUÇ VE ÖNERİLER

Yaşamımızın her alanında yazılıma olan ihtiyaç sürekli artmakta ve bu ihtiyaca paralel olarak yazılım proje sayıları da artmaktadır. Kaliteli yazılımların oluşturulabilmesi için projelerin iyi bir test sürecinden geçirilmesi gerekmektedir. Test sürecinden geçirilmeyen projelerin oluşturacağı risk ve sorunlar yüzünden piyasa içerisinde kaybolacağı görülmekte ve şirketler açısından saygınlık, para ve zaman kaybına sebep olacaktır. Projelerin iyi bir test sürecinden geçirilerek en az hata ile kullanıcılara sunulması için proje süresince oluşabilecek risklere karşı alınacak önlemlerde yol gösterici olacak zengin raporlamalar yapılmalıdır. Yazılım organizasyonları ar-ge çalışmalarlarıyla birlikte geliştirilerek yapılan işin kalitesi arttırılabilmektedir.

Gelişen teknoloji ve artan müşteri beklentileri, bankaların daha karmaşık ve daha fazla fonksiyonellik içeren ürünleri en hızlı şekilde üretip piyasaya sürmesini gerektirmektedir. Sektörün gerektirdiği hızlı üretim, birçok riski de beraberinde getirmektedir. Bu riskleri azaltmak için test teknikleri hakkında detay bilgilere sahip olunmalı ve bu bilgiler projelerde titizlikle uygulanmalıdır.

Mobil uygulamaların ve platformların artışı, sektörel düzenlemeler, yasal uyumluluk hedefleri ve sürekli değişen müşteri ihtiyaçları göz önüne alındığında, finans kuruluşları test aktivitelerini gerçekleştirebilmek için bulut bilişim, test otomasyonu, ileri seviye test teknikleri ve test veri yönetimi gibi teknik alanlara ve hizmetlere yönelmelidirler.

KAYNAKÇA

Arslan, O., (2020) "Manuel Test Nedir?"

Black, R. vd., "ISTQB Advanced Level Syllabus Test Manager", ISTQB, Chapter 1, 9.

Black, R. vd., ISTQB Advanced Level Syllabus Test Manager, ISTQB, Chapter 1, 11.

Certified Tester Foundation Level Syllabus", International Software Testing Qualifications Board, (2011).

Hancı, A.K. vd., "Yazılım Testi Tasarım Tekniklerinin Matematiksel Model Yaklaşımı İle Analizi", 2017.

İTÜ Bilgi İşlem Daire Başkanlığı, (2013) *Yazılım Testi Ve Test Süreçleri*.

Karagöz, A.B, Test Otomasyonunda Kritik Başarı Faktörleri, 2017.

Karagöz, A.B, Yazılım Test Süreci, Finans Sektöründeki Önemi ,2018.

Uzun, Beytullah, (2019) *Yazılım Testi Süreç Yönetimi Ve Raporlama Sisteminin Geliştirilmesi*.

Yener, Halil Ahmet vd., (2019), “Productivity Increase with Software Development and Test Automation:General Mobile.”