

# Yazılım Test Sürecinde Problemler ve Çözüm Önerileri

Mehmet ŞAHİNOĞLU, Mustafa SARI, Ayşegül KURT, Salih KURNAZ, Mehmet ÖZBEK,

<sup>1</sup>Yazılım Test ve Kalite Değerlendirme Merkezi, TÜBİTAK BİLGEM, Gebze, Kocaeli  
{mehmet.sahinoglu, mustafa.sari, aysegul.kurt, salih.kurnaz,  
mehmet.ozbek}@tubitak.gov.tr

**Özet.** Günümüzde, farklı sektörlerdeki ihtiyaçlar için farklı yazılım ürünleri geliştirilmektedir. Yazılım geliştirme sürecinin önemli bir aşaması da yazılım test sürecidir. Yazılım testleri ürün kalitesinin yükseltilmesi ve müşteri memnuniyetinin artırılması için kritik öneme sahiptir. Test sürecinin başarısı doğrudan projenin başarısını etkilemektedir. Yazılımlardaki çok küçük hatalar dahi can, mal kayıplarına neden olabilecek kritik sonuçlara sebebiyet verebilir. Bu sebeplerden dolayı yazılım test sürecinin etkili ve verimli geçmesi projenin istenen hedeflere ulaşmasında büyük öneme sahiptir.

Bu çalışmada test süreçlerinde karşılaşılan problemler belirli gruplarda toplanarak, bu problemlere neden olan sebepler ve bu problemler için önerilen çözüm önerileri anlatılmıştır. Karşılaşılan problemler; planlama ile ilgili problemler, yönetsel problemler, gereksinimler ile ilgili problemler, düşünsel problemler, paydaşlar arasındaki problemler ve genel test problemleri olarak gruplandırılabilir. Bu gruplar altında toplanan problemlerin sebepleri belirtilerek, çözüm yolları önerilmiştir.

**Anahtar Kelimeler.** Yazılım Test Süreci, Proje Yaşam Döngüsü, Test Sürecinde Problemler

## 1 Giriş

Yazılımın doğrulanması ve geçerlenmesi süreci, birçok alt faaliyetten oluşur. Bu faaliyetlerden biri proje yaşam döngüsü süreçlerinin sonunda meydana gelen çıktılarının doğru, tam, tutarlı ve açık olduğunun doğrulanmasıdır.

Bir diğer faaliyet ürünün teknik yeterliliğinin değerlendirilmesi ve uygun sonuçlar elde edilene kadar bu aktivitelere devam edilmesidir.

Doğrulama süreci, geliştirilen yazılımın kendisinden beklenen davranışları gerçekleştirdiğini ve istenmeyen davranışları göstermediğini ortaya koyan süreçtir. Geliştirilen yazılımın, gereksinimleri tam olarak karşıladığı ve geliştirilmenin her evresindeki çıktılarının doğru olduğuna karar verilmesi gerekir. Doğrulama sürecinin, ürünün kursosuz olarak çalışmasına katkısı büyük olduğundan, yazılım geliştirmede en önemli süreçlerden birisidir [1].

Doğrulama sürecinin temel amacı, kalitesi yüksek ürün üretebilmektedir. Doğrulama geçerleme sürecinin mümkün oldukça projelerin erken evrelerinde başlatılması olası problemlerin önünü kesme veya çözümünün daha erkene alınmasına yardımcı olacaktır [1].

Geçerleme süreci, yazılımın tam olduğunu ve kendi içinde tutarlı olduğunu gösteren bir süreçtir. Proje sırasında iş adımları ve ara çıktılar gözden geçirilip bunların kabul edilebilir olduğundan emin olunur.

Yazılımın tutarlılığını, bütünlüğünü ve doğruluğunu göstermek için yazılım geliştirme yaşam döngüsü aşamasında ve aşamaların her birinin arasında geçerleme süreci uygulanır. Ayrıca geçerleme süreci yazılım geliştirme esnasında eğitimler, standartlar, gözden geçirme ve yorumlar, geri bildirimler, kontrol listelerine göre denetimlerin yapılması gibi aşamalarla gerçekleştirilir [2].

Yazılım test faaliyetleri doğrulama ve geçerleme süreci içerisindeki alt aktiviteler arasında önemli aktivitelerdir.

Yazılım test sürecinin olmadığı ya da eksik olduğu durumlarda yaşanabilecek negatif durumlar; proje yönetimi ile ilgili problemler, test yönetimi ile ilgili problemler, düşünsel problemler ile ilgili problemler olarak genel başlıklarla toplanabilir. Bu çalışma kapsamında belirtilen genel durumlar detaylı bir şekilde ele alınacaktır.

## **2 Proje Yönetimi ile ilgili Problemler ve Çözüm Önerileri**

### **2.1 Proje Planlaması**

Yazılım projelerinde proje özelliklerine, ihtiyaçlarına, hedeflerine uygun olarak belirlenen yazılım geliştirme yaşam döngüsü seçimi ve bu yaşam döngüsü aşamalarının amaçlarının belirlenmesi, bu aşamalarda yapılması gereken faaliyetlerin ve bu faaliyetlerin çıktılarının belirlenmesi gerekmektedir.

Proje süreçleri içerisinde bir aşama tamamlandıktan, hedeflerine ulaşarak faaliyetler kapandıktan sonra ilerleyen dönemlerde bu aşamalara geri dönmeyen, projenin planlarından sapmasına, aynı işlerin tekrar yapılmasına, zaman, iş gücü kaybedilmesine sebebiyet vermektedir. Örneğin, projenin sistem testleri gerçekleştirilir iken, müşteri kurumdan yeni bir özellik talebi gelmesi, isterlerin tekrar düzenlenmesi, geliştirme sürecine geri dönülmesi ve test sürecinin (yineleme ve sistem testleri) yeni duruma göre tekrar değerlendirilmesini gerektirecektir. Bu problemin önüne geçilebilmesi için proje analiz ve planlama süreçlerinin paydaşlar ile birlikte belirlenmesi, onaylarının alınması ve bu planlara bağlı kalınması gerekmektedir.

### **2.2 Kaynak Planlanması**

Proje planlama aşamasında gerekli kaynaklar belirlenirken, planlanan test faaliyetleri, iş yüküne, yapılacak çalışmaların teknik gerekliliklerine göre belirlenir. Test faaliyetlerini yürütecek test mühendisi sayısı ve testler sırasında kullanılacak araçların belirlenmesi gerekmektedir. Test mühendisi sayısının gerekenden az belirlenmesi iş yükünün artmasına, yeterli detayda test yapılamamasına ve test sürecinin hedeflerine ulaşamamasına sebebiyet vermektedir. Aynı durum test faaliyetlerinde kullanılacak

araçlarının belirlenmesinde de geçerlidir. Gereken araçların tedarik edilememesi veya geç tedarik edilmesi test sürecinin verimliliğini ve etkinliğini düşürecektir.

### **2.3 Zaman Planlaması**

Yazılım gelişme yaşam döngüsünün diğer aşamalarında olduğu şekilde test çalışmalarının verimli olması ve istenen çıktılar elde edilmesi için zaman planlamasına uyulması gerekmektedir. Yazılım testlerindeki faaliyetler, planlama, test tasarımı, testlerin koşuturulması, bulunan hataların düzeltilmesi, yinleme testleri, değişikliklerin olması ve bunların dokümanite edilmesidir. Projelerde farklı sebeplerden dolayı zamansal dar boğazlara girilmesi ihtimali vardır. Proje teslim zamanının yaklaşması geliştirme sürecinin henüz tamamlanmaması sonucunda zaman kazanmak için test süreci için ayrılan zaman ötelenmekte ve test faaliyetleri planlanandan daha az bir zamana sıkıştırılmaktadır. Böylece oluşan zaman baskısı istenen detayda test yapılmasının engelleyeceği için test faaliyetlerinin kalitesini düşürmektedir.

### **2.4 Görevler ile ilgili Belirsizlikler**

Projelerde planlamalar yapılırken önemli noktalardan biri de görev tanımlamaların yapılmış olmasıdır. Yazılım testlerinde farklı bağımsızlık seviyelerine göre farklı pozisyondaki kişilere farklı görevler verilebilmektedir. Test faaliyetlerinin sağlıklı ilerleyebilmesi için proje içerisinde görevlerin ve bu görevleri gerçekleştirecek insanların belirli olmaması nedeniyle, bazı işlemlerin takibi yapılamamakta ve bu işlemler tamamlanamamaktadır.

### **2.5 Yanlış Kestirimler**

Proje başlangıç aşamalarında test faaliyetleri için yapılan kestirimler iki temel dayanağa sahiptir, bunlar tecrübeye dayalı kestirimler ve istatistiksel verilere dayalı kestirimlerdir. Proje planlarının temel aldığı öngörüler takvimsel öngörüler, iş gücüne dayalı öngörüler, bütçe öngörüler ve proje risk öngörüler şeklinde sıralanabilir. Bu öngörülerin isabetli olup olmaması, test faaliyetlerinin istenen amaçlara ulaşmasında etkili olmaktadır.

### **2.6 Test Altyapısı ile ilgili Sorunlar**

Yazılım test faaliyetlerinin sağlıklı olarak devam edebilmesi için önemli noktalardan birisi de yazılım geliştirme ortamından ayrı bir test ortamının olmasıdır. Bu ortam yazılım geliştirme müdahalelerinden uzak tutulmalıdır. Burada amaç test edilen yazılım ürününün durağan bir yapıda tutulmak istenmesi, yazılımda alınan çıktıların aynı işlemler karşısında aynı kalmasını sağlamaktır.

Test edilen yazılım ürününün bahsedilen durağan tepkiler vermesini sağlamak için diğer bir zorunluluk ta ürün ile ilgili düzenli sürüm tanımlama ve sürüm kontrolünün yapılmasıdır. Bu düzen sağlanmadığı durumda yazılım sürüme ait hatalar, hata dü-

zeltmeleri (bug fix), geliřtirmeden kaynaklı eklemeler sürümler arasında net olamayacak ve sağlıklı bir test süreci iřletilemeyecektir.

### **2.7 Dokümantasyon Fazlalığı veya Eksikliği**

Yazılım test faaliyetleri, yazılım yaşam döngüsü içerisindeki diđer yazılım faaliyetleri gibi girdileri ve çıktıları için dokümantasyona ihtiyaç duymaktadır. Test sürecine girdi sağlayacak dokümanların (Gereksinimler, Plan dokümanları, Şartname vb.) yeterli detayda olmaması, süreç için gerekli verileri sağlayamadığı için süreci olumsuz etkilemesi beklenebilir.

Test faaliyetleri sonucunda çıktı olan, diđer test faaliyetlerine ve diđer yazılım geliştirme evrelerine girdi olacak dokümanların miktarı iyi belirlenmelidir. Eğer gerektiğinden fazla olur ise test faaliyetlerinin hızını düşürme riski oluşmaktadır.

Test dokümantasyonu miktarını belirlemek için IEEE 829 Standardı kullanılmalıdır. Burada Önem Derecesi kavramı (Integrity Level) öne çıkmaktadır [3].

### **2.8 Yönetim Destek Eksikliği**

Test çalışmalarının oluşturduğu doğal pozisyonlardan dolayı, yazılım geliştirme ekibi yazılım test ekibine direnç gösterebilir. Bu direnç yapılan çalışmaların verimliliğini düşürecek boyutta olabilir. Test ekibinin bu duruma gelmemesi için proje yönetiminin test ekibini desteklemesi, yaptığı çalışmaların önemine inandığını geliştirici ekip içerisinde hissettirmesi gerekmektedir. Bu desteğin sağlanamadığı durumlarda test ekibi etkinliğini kaybedebilir.

### **2.9 Gereksinim Yönetimi ile ilgili Problemler**

Yazılım projelerinde başarısızlıkların nedenleri incelendiğinde, gereksinimler ile ilgili sorunların en başta olduğu görülmektedir [4]. Yazılım projelerinde gereksinimler, test faaliyetlerinin ana girdisidir.

Test stratejisi, test teknikleri, önceliklendirme gibi noktaların belirlenmesinde fonksiyonel ve fonksiyonel olmayan gereksinimler incelenmektedir. Bu açıdan sistemde yer alan bir özelliğın, bir fonksiyonun gereksiniminin olmaması test sürecine negatif yansıtacağı öngörülebilir. Gereksinimler tanımlanırken sadece fonksiyonel isterlerin tanımlanması yeterli olmayacaktır. Fonksiyonel olmayan güvenlik (security), güvenilirlik (reliability), kullanılabilirlik, gibi özelliklerin de sistem içerisinde tanımlanması gerekmektedir. İsterler tanımlanırken uyulması gereken kurallar bulunduğu gibi isterleri değerlendirirken isterlerin taşınması gereken özellikler de bulunmaktadır. İsterler doğru, doğrulanabilir, tam, net, sade, tutarlı, gerçekleştirilebilir, izlenebilir olmalıdır. Gereksinimlerin proje başlangıç aşamasından itibaren tasarım, gerçekleştirme, test faaliyetleri aşamaları için izlenebilirliğinin kurulmuş olması gerekmektedir. Proje yaşam döngüsü süreçlerinde gözden geçirme (review) faaliyetleri önemli yer tuttuğu gibi isterler için de gözden geçirme çalışmaları yapılmalıdır. Yukarıda belirtilen özelliklere uygunluk durumu, izlenebilirliği de gözden geçirilmelidir. İsterlerin yazımı tamamlandıktan sonraki evrelerde müşteri talepleri, tasarımsal deęişiklik-

ler gibi sebepler ile isterlerin deęiştirilmesi söz konusu olabilir. Bu deęişikliklerin etkilerinin deęerlendirilmesi ve proje ilerleyişinde hesaba katılması gerekmektedir. Bahsedildięi gibi, gereksinimler üzerine inşa edilmiş bir projede gereksinim deęişiklik oranlarının yüksek olmaması dikkat edilmesi, incelenmesi, gerekli tedbirlerin alınması gereken bir durumdur. Bu durumların sonuçlarında test planlama, test özelliklerinin belirlenmesi faaliyetlerinin tekrar yapılacak olması test sürecini uzatacağı öngörülebilir. Birçok kişiden oluşan ve farklı fiziki ortamlarda çalışanların olduęu projelerde gereksinimlerde yapılan deęişikliklerin tüm ekip tarafında takip edilebileceęi ve gerekli uyarıların yapılabileceęi bir alt yapı kurulması dięer bir gerekliliktir.

## **2.10 Baęımsızlık Kavramı**

Yazılım projelerinde test faaliyetlerini gerçekleştiren kişi veya kişilerin farklı baęımsızlık seviyelerinde oldukları durumlar görölmektedir. Her baęımsızlık seviyesinin pozitif ve negatif yanları bulunmaktadır. Baęımsızlık seviyesi arttıkça test mühendisinin önyargı ve şartlanmışlık seviyesi azalmaktadır. Dolayısı ile başkalarının göremedięi hataları görebilirler. Baęımsız Test ekibi sistemin gereksinim ve tasarım aşamalarında etkili olarak, testlerin başarılı bir şekilde gerçekleştirilmesini sağlamaktadır. Baęımsızlık seviyesi düşükten yükseęe doęru şu şekilde sıralanabilir:

- Geliştirmeyi yapan yazılımcı kendi kodunu test edebilir,
- Geliştirme ekibinin içerisinde, ürünü kodlayan kişiden başka bir kişi test edebilir,
- Aynı firma içerisinde, test için uzmanlaşmış ayrı bir ekip tarafından testler gerçekleştirilebilir,
- Başka bir firmadan test işlerini yapacak ayrı bir ekip proje test faaliyetlerini gerçekleştirebilir.

Baęımsız test ekibi ile çalışmanın avantajları olduęu gibi, dezavantajları da sözkonusu olabilir. Tamamen baęımsız olma durumunda geliştirme ekibinden izole olunma ihtimali bulunmaktadır. Baęımsız test ekibi yazılım ekibi ile farklı fiziksel ortamlarda çalıştıklarından dolayı çalışanlar birbirlerini görmemektedir. Bu durum beraberinde, yazılım geliştirme ekibi için kalite duygusunun yitirilmesini getirebilir.

## **3 Test yönetimi ile ilgili problemler**

### **3.1 Yanlış Test Stratejisi ve Yaklaşımı**

Test stratejisi ve yaklaşımı projeye özgü bir şekilde belirlenmektedir. Test planlaması sırasında test yaklaşımına karar verilir, detaylandırılır ve bu yaklaşımdan yola çıkılarak test durumları, test aktiviteleri belirlenir. Test planlamasının ilk adımı test stratejisinin belirlenmesidir. Yanlış test yaklaşımının belirlenmesi test sürecinde istenen hedeflere ulaşılamamasına, tekrarlı işlerin yapılmasına, sürenin uzamasına neden olabilir. Kurumsal firmalarda daha çok görülen hatalardan biri de her projeye benzer test stratejisinin uygulanmak istenmesidir. Bir uygulamada başarılı olmuş bir

test stratejisi, diğ er bir projenin kendisine özgü durumundan dolayı istenen başarı seviyesinde olmayabilir.

Her projeye aynı test yöntemlerini uygulamama yaklaşımını sistematik hale getirmek için önem derecesi kavramı kullanılabilir. Burada proje bir proje özelliğine göre (risk, sonuç vb.) seviyelendirilir. Daha sonra bu özelliğe göre projenin tamamı veya modüllerine bir önem derecesi tanımlanır. Bu önem derecesi kavramı temel alınarak, test sürecindeki dokümantasyon ve faaliyetlerin derinliği miktarı belirlenir. Böylece her projenin hatta her modülün özelliklerine göre test yaklaşımı belirlenebilir.

Test yaklaşımında yapılan diğ er bir hata da test edilecek noktaların doğru belirlenmemesidir. Genelde kolay test edilen kısımlar tekrar tekrar test edilirken, test edilmesi zor ama önemli alanlara yeterli önem gösterilmemektedir. Diğ er bir durum ise test sürecinde tüm test durumları birden koşturularak, diğ erlerine göre daha önemli kısımlara aynı özen gösterilmektedir. Bu durum zamanın ve test eforunun etkin ve etkili kullanılmaması sonucuna çıkmaktadır. Bu durumda önceliklendirme yapılması gerekir. Sistemdeki karşılıklarının önemlerine dayanarak test durumlarının önceliklendirilmesi gerekmektedir.

### 3.2 Planlamanın Eksik olması

Proje planlaması yapılırken test süreçlerinin de planlaması gerekmektedir. Test planları tüm test süreçlerini, test aktivitelerini, test giriş ve çıkış kriterlerini, test yaklaşımlarını, takvimi, kabulleri ve varsayımları, test sürecinde kullanılacak araçları tanımlamalıdır. Test dokümantasyonu IEEE Std. 829 standardında belirtilmiştir [3]. Test planları tüm test seviyelerini kapsayacak şekilde yazılan bir Test Ana Planından, ve her test seviyesini (birim test seviyesi, entegrasyon test seviyesi, sistem test seviyesi) tanımlayan Test Planlarından oluşmaktadır. Planları hazırlarken yeterli detay seviyesi açıklanmadığı durumlarda belirsizlikler sorunlara sebebiyet verebilmektedir. Testlere giriş kriterleri ne zaman testlere başlanacağını, hangi durumların testlere başlanması için gerekli olduğunun belirtilmesidir. Bu konular yazılım geliştiren ekip ile daha önceden belirlenmesi gereken konulardan biridir. Test ekibi etkili test yapabilmek için bazı şartların olgunlaşmasını talep edebilir, sistemin duman (smoke) testlerini tamamlamış olması buna örnek olarak verilebilir. Bu durum test ekibi ile yazılım ekibi arasında anlaşmazlıklara sebebiyet verebilir.

Test Ana Planında (Master Test Plan) ve seviye test planlarında testlerin hangi kriterler sağlandığında tamamlanacağı bilgisi de yukarıda belirtilen sebeplerden dolayı yer almalıdır.

Test süreçlerinde, nelerin test edileceğ inin belirlenmesi yanında, nelerin test edilmeyeceğ inin de belirlenmesi ve bu konuda paydaşlar arasında anlaşmaya varılmalıdır. Bu durum da taraflar arasında problemlere sebebiyet verebilir.

### 3.3 Yetersiz Derinlikteki Testler

Test faaliyetlerinin gerçekleştirilmesi sırasında yaşanabilecek tehlikelerden birisi de yeterli seviyede test edilememesidir. Yazılımın test edilmesi sadece bir yol takip edilerek doğru çalıştığı nın gösterilmesi değildir. Eğer sadece ana akış (Happy Path) test

edilirse sistemin çalışmayan yanlarının gösterilmesi için gösterilmesi gereken çaba gerçekleştirilmemiş ve amaca ulaşamaz. Bu durumu kontrol etmek ve test faaliyetlerinin hangi seviyede yapıldığının izlenmesi için etkili yöntemlerden birisi Test Kapsam Aracı (Test Coverage Tool) kullanmaktır. Bu araçlar yazılıma ait kodları izlemektedirler. Yazılım testleri yapılırken çalıştırılan kod parçalarını belirlemekte, işlem sonucunda da test faaliyetleri sonucunda çalıştırılan kodların, tüm kodlara oranını vererek bir kapsam yüzdesi vermektedir.

### 3.4 Yetersiz Test Metrikleri

Yazılım geliştirme yaşam döngüsü sırasında tüm evrelerde olduğu gibi test süreçlerinin de izlenmesi ve bu evreler hakkında bilgi sahibi olunması, çıkarımlar yapılması ve bu çıkarımlara göre gerektiği durumlarda sürece müdahale edilmesi gerekmektedir. Ölçüm işlemi doğru metriklerin alınması ile yani doğru soruların sorulması ile başlamaktadır. Yetersiz metrikleri olan bir test süreci yeterli ölçümleri yapamayacak, dolayısı ile doğru verilere ve çıkarımlara ulaşamayacaktır. Bir projede en yaygın kullanılan test metrikleri aşağıdaki gibidir:

- Test durumları hazırlama aşamasında, test durumlarının tamamlanma yüzdesi,
- Test ortamı hazırlanmasında yapılması gereken işlerin yüzde kaçının tamamlandığı,
- Testlerin koşturulması safhasında test durumlarının ne kadarının koşturulduğu,
- Hata verileri, (hata sayısı, hata yoğunluğu, hata oranı, düzeltilen hata sayısı, düzeltilme işleminden sonra tekrarlanan testlerde başarı oranı),
- Test kapsam yüzdesi (kod üzerinde),
- Test mühendislerinin sisteme güveni,
- Takvimsel kilometre taşları,
- Test maliyeti

Örnekleri yukarıda verilen metriklerden veriler geldiğinde verileri değerlendirmek, riskler belirlemek ve bu riskler doğrultusunda kararlar almak gerekmektedir. Eğer riskleri belirleyecek, değerlendirecek ve gerekli tedbirleri alacak bir mekanizma kurulamaz ise proje takibi takip edilemez, plandan sapmalar tespit edilemez, projenin başarısını etkileyecek risklere karşı gerekli müdahaleler yapılamaz.

### 3.5 Alan Bilgisi Eksikliği

Test faaliyetleri farklı alanlarda farklı stratejiler gerektirmektedir, örneğin güvenlik-kritik bir sistemin testleri ile bir banka uygulamasının testleri farklı bilgiler, farklı alt yapı gerektirmektedir. Test mühendisinin projenin ilgili konusunda bilgiye sahip olması gerekmektedir. Test mühendisi yeterli bilgiye sahip değilse yapacağı testlerin istenen hedeflere ulaşmasından şüphe edilebilir.

### 3.6 Hataların Net Bir Şekilde Bildirilmemesi

Test süreci sırasında bulunan hataların test ekibi tarafından raporlanarak yazılım geliştirme ekibine iletilmesi, yazılım geliştirme ekibi tarafından bu hataların analiz edilerek düzeltilmesi gerekmektedir. Bu işlemlerin yapılabilmesi için bulunan hataların geliştirme ekibine doğru ve tam olarak iletilmesi kritik öneme sahiptir.

### 3.7 Test Araçları ile ilgili Problemler

Test araçları doğru kullanıldığında test faaliyetlerine yüksek katkıları olmaktadır. Bununla beraber, araçların yanlış kullanılması test araçlarından alınan verimi düşürmektedir.

Test araçları ile ilgili yapılan genel hatalar:

- Gerçekçi olmayan beklentiler,
- Gerekli sürelerin azımsanması,
- Gerekli eforun azımsanması,
- Araca gerekenden fazla güvenilmesi,
- Araç için destek alınan firmanın desteği kesmesi, şeklinde sıralanabilir.

### 3.8 Yineleme (Regresyon) Testleri ile ilgili Problemler

Yazılım test sürecinde hatalardan dolayı yapılan düzeltmeler ve yeni geliştirme faaliyetler sonucunda kodda değişiklikler yapılmaktadır. Bu değişikliklerin sistemin ilgili alanlarında beklenmedik bir hataya sebebiyet verip vermediğinin kontrolü yineleme testleri (regresyon) ile yapılmaktadır. Bu testlerin yetersiz olması durumunda sistem içerisinde beklenmedik hataların çıkması ihtimali artmaktadır.

## 4 Sonuç

Yazılım ürününün başarısı, gereksinimleri karşılaması ve müşterilerin memnuniyeti ile ölçülebilir. Bu tanımlanan başarıya ulaşmak için ürünün doğru süreçler işletilerek, gereksinimlerde tanımlanan şekilde sonuçlandırılması gerekmektedir. Bu amaç ile doğrulama ve geçerleme süreci yazılım projelerinin hedeflenen başarıya ulaşmasında kritik bir yere sahiptir. Yazılımın doğrulama ve geçerleme faaliyetleri içinde değerlendirilebilecek test faaliyetleri geliştirilen ürünün istenen özelliklere uygunluğunun kontrol edilmesi açısından oldukça önemlidir. Bu faaliyetlerin etkinliği ve etkililiği çok farklı sebeplerden dolayı düşebilir. Bu sebepler öngörülebilir ve önlem alınabilir sebeplerdir.

**Teşekkür** - Yazarlar, bu çalışmanın gerçekleştirilmesi için destek sağlayan TÜBİTAK BİLGEM Yazılım Test ve Kalite Değerlendirme Merkezi'ne teşekkür eder.



## **Kaynaklar**

1. zbek M., Kurt A., Grbz A., “Emniyet-Kritik Sistemlerin Yazılım Doğrulama Süreci  
Software Verification Process in Safety-Critical Systems”
2. Yıldırım E., Pişken M., “Orta Ölçekli Yazılım Firmaları İçin İdeal Bağımsız Doğrulama ve  
Geçerleme Organizasyon Yaklaşımı”
3. IEEE 829-2008, Standard for Software and System Test Documentation
4. The Standish Group Report, (1995)
5. IEEE 1012-2012, Standard for Software Verification and Validation