

Yazılım Hata Kestirimi için Örnek Bir Model

R. Burcu Karaömer

Innova Bilişim Çözümleri A.Ş. Çankaya/Ankara, Türkiye
bkaraomer@innova.com.tr

Onur Kaynak

Innova Bilişim Çözümleri A.Ş. Çankaya/Ankara, Türkiye
okaynak@innova.com.tr

Özet. İNNOVA Bilişim Çözümleri A.Ş. bir Türk Telekom Grup şirketi olup; TSE-ISO-EN ISO 9001, ISO/IEC 20000, ISO/IEC 27001 ve TS ISO/IEC 15504 SPICE Seviye 2 uluslararası sertifikasyonlarına sahiptir. Bu model ve standartlara uygun olarak uygulama geliştirme süreçleri yürütülmektedir. Uygulama geliştirme süreçlerinden yazılım güvenilirliği; yazılımın başarısını gösteren en önemli kalite faktörlerindedir. Yazılım güvenilirliği, yazılımın belirli koşullar altında, yazılımdan beklenen özellikleri yerine getirebilmesi olasılığıyken, yazılım güvenilirliğinin modellenmesi, yazılım hata kayıt verilerinin davranışlarının modellenerek yazılım güvenilirliğinin tahmin edilmesidir. Yazılım güvenilirliğinin modellenmesi, proje yönetiminin projelerinin planlamasını doğru yapabilmek için gerekli kaynakları en iyi şekilde planlaması ve müşteri ihtiyaçlarını karşılayacak ürünü en az hata içerecek şekilde müşteriye teslim etmesi açısından önemli bir unsurdur. Bu unsur dikkate alınmadığı durumda müşterinin firma ve ürüne olan güveni azalacağı gibi projenin geriye dönük kalite maliyetleri de artacaktır. Geliştirme tamamlanmış ancak aylık benzer büyüklüklerde iterasyonlar bazında ek geliştirme yapılan, yazılım ürün hattı gibi yeniden kullanımın etkin olduğu, regresyon test kümesine sahip projelerde hata değerlerinin ölçülmesi, değerlendirilip kontrol altında tutulması için yazılım güvenilirlik modellerinde özelleşmeye gitme ihtiyacı vardır. Bu çalışma kapsamında yukarıdaki özellikleri taşıyan bir projeye ait hata kayıt verileri kullanılarak bu hata kayıt verilerine en uygun yazılım güvenilirlik modeli belirlenmeye çalışılmış ve belirlenen model yardımı ile bir sonraki iterasyon da oluşabilecek hata miktarı %95 güven aralığında tahmin edilmeye çalışılmıştır.

Anahtar Kelimeler: Yazılım Güvenilirliği, Lineer Regresyon Modeli, Homojen Olmayan Poisson Süreci Goel-Okumoto Modeli, Geciktirilmiş S-Şekilli Homojen Olmayan Poisson Süreci Modeli

1 Giriş

Günümüzde her alanda hayatımıza giren yazılımlar, teknoloji ilerledikçe çeşitlenmiş ve karmaşıklaşmıştır. Teknolojinin diğer alanlarına göre daha soyut bir kavram olan yazılım ve yazılım teknolojilerinin idame edilmesi zamanla zor bir hal almış ve üzerinde çalışmalar yapılmasına neden olmuştur. Yapılan çalışmalar sonucu yazılım sektörü; yazılım geliştirme yöntemlerinden, test yöntemlerine, ölçme ve planlama yöntemlerinden, sürüm alma yöntemlerine kadar birçok alanda kendine has yöntemler geliştirerek diğer tüm sektörlerden ayrılmaktadır. Yazılım; finans, sağlık, maliye, savunma sanayii gibi sektörlerle hakim olmaya başladıkça yazılım güvenilirliğinin insan hayatındaki rolü de her geçen gün artmaktadır [1]. Ancak yazılım güvenilirliğinin nasıl ölçüleceği ile ilgili yöntemlerin yeterince tecrübe edilmemiş olması uygulanacak modellerin belirlenmesi için yeterli büyüklükte bir ölçüm havuzunun oluşmamasına neden olmaktadır. Bu çalışmada örnek olarak seçilen üç yazılım güvenilirlik modellerinin elde bulunan projenin güvenilirliğini ölçmede başarılı olup olmadıkları ve birbirlerine göre performanslarının nasıl olduğu sorusunun cevabı araştırılmıştır. Çalışma kapsamında yazılım güvenilirliğini belirlemede kullanılan yazılım güvenilirlik modellerinin karşılaştırılması ve kullanılabilirliklerinin ölçülmesi amacıyla bir durum çalışması yapılmış olup, toplanan ölçümler kurumsal veri havuzuna yerleştirilmiştir.

Çalışma kapsamında belirlenen proje de iterasyon bazında tespit edilen hata sayıları kayıt altına alınmıştır. Seçilen Lineer Regresyon Modeli, Homojen Olmayan Poisson Süreci Goel-Okumoto Modeli (Goel-Okumoto NHPP) ve Geciktirilmiş S-Şekilli Homojen Olmayan Poisson Süreci Modeli olmak üzere üç yazılım güvenilirlik modeli ile kayıt altına alınan hata kayıt verileri için modelleme yapılmıştır. Yapılan modellemede 8 iterasyon sonucu kayıt altına alınan hata kayıt verileri kullanılmış ve 9. İterasyon için tahmin yapılmıştır. Aynı şekilde 9. iterasyon hata kayıt verileri modele dâhil edilerek yapılan modelleme ile 10. İterasyon da oluşabilecek hata sayısı tahmin edilmiştir. Her bir model için yapılan tahminleme sonuçlarına göre modeller karşılaştırılmıştır. Bu çalışmanın, gelecekte yazılım güvenilirlik modelleri hakkında yapılacak çalışmalara yol gösterici olacağı düşünülmektedir.

2 Yazılım Güvenilirlik Modelleri

Yazılım güvenilirliği, yazılım ve yazılım bileşenlerinin belli ortamda, belirli zaman içerisinde kendisinden beklenen özellikleri yerine getirebilme olasılığıdır. Başka bir deyişle bir yazılıma ne kadar güvenilebileceğinin ölçümüdür [2]. Yazılımda ortaya çıkan hatalar yazılıma olan güveni azaltırken yazılımın kullanılabilirliğini de engellemektedir. Uzun süre hata vermeden, kendisinden beklenen özellikleri yerine getiren bir yazılım ise ona olan güveni arttırmaktadır. Bu durum, dolaylı olarak yazılımı geliştiren şirkete olan güveni de arttırmaktadır.

Yazılıma ne kadar güvenebileceğimiz yazılımda ortaya çıkan hataların sayısına, kategorisine ve yoğunluğuna bağlıdır. Günümüzde yapılan çalışmalarda, hatalar ortaya çıkmadan bir yazılımda bulunan hata miktarı tahmin edilerek yazılım güvenilirliği ölçülmeye çalışılmaktadır. Bu maksatla yazılım güvenilirlik modelleri sıklıkla kullanılan

bir yöntemdir [3]. Yazılım güvenilirlik modelleri, yazılımda önceden ortaya çıkan hataların davranışlarını inceleyerek ileride oluşabilecek hata durumlarının matematiksel olarak ifade edilmiş halleridir. Yazılım güvenilirlik modelleri, özellikle hataların oluşum zamanları veya belirli aralıklarda ortaya çıkan toplam hata sayısı gibi verileri kullanarak yazılımda gelecekte birim zamanda ne kadar hata oluşabileceğini tahmin ederler [4]. Bu şekilde farklı varsayımlar öngörülerek oluşturulmuş farklı yazılım güvenilirlik modelleri ortaya çıkmıştır. Ayrıca yapılan çalışmalarda önceden geliştirilmiş birçok yazılım güvenilirlik modeli farklı projeler için test edilmiş, birbirleri ile karşılaştırılmış ve kullanılabilirlikleri ölçülmüştür.

Bu çalışma kapsamında üç ayrı yazılım güvenilirlik modelinin “X” projesinde göstermiş oldukları performanslar ölçülmüş ve modeller bu performanslara göre karşılaştırılmıştır. Model seçiminde, birbirine göre farklı davranış gösteren modellerin seçilmesine özen gösterilmiş, bunun neticesinde Lineer Regresyon Modeli, Homojen Olmayan Poisson Süreci Goel-Okumoto Modeli (Goel-Okumoto NHPP) ve Geciktirilmiş S-Şekli Homojen Olmayan Poisson Süreci Modelinin (Geciktirilmiş S-Şekilleri NHPP) kullanılmasına karar verilmiştir. Seçilen bu üç modelin temel özelliği; belirli bir zamana kadar gözlemlenmesi muhtemel hataların, kümülatif toplamının ortalama değer fonksiyonunu tahmin ederken, bu üç modelin her biri bu fonksiyonun davranışını farklı bir şekilde sahip grafiklerle modellemektedir. Bu modellerin herhangi bir projede hata kayıt verilerine uygulanabilmesi için hata kayıt verilerinin her test faaliyetinden sonra düzenli olarak kayıt altına alınması ve kümülatif toplamlarının hesaplanmasına ihtiyaç vardır.

2.1 Homojen Olmayan Poisson Süreci Goel-Okumoto Modeli

Homojen Olmayan Poisson Süreci Goel-Okumoto Modeli’ne göre yazılım da var olan hatalar nedeniyle yazılımda zaman içerisinde rastgele bozulmalar yaşanır [5,6].

Bu modelin varsayımları;

- Yazılımda bulunan tüm hatalar, hataları tespit etme yönteminden bağımsızdır.
- Tespit edilen hatalar, bir sonraki teste kadar kaldırılır.
- Hataların kaldırılması sırasında yeni hata oluşmasına sebep olunmaz.
- Hataların oluşma ve tespit edilme olasılığı eşittir

Modelin ortalama değer fonksiyonu aşağıdaki gibidir;

$$m_t = a(1 - e^{-bx})$$

Fonksiyonda, a parametresi yazılımda tespit edilen hata sayısını gösterirken b parametresi hata tespit oranını göstermektedir.

2.2 Geciktirilmiş S-Şekilli Homojen Olmayan Poisson Süreci Modeli

Geciktirilmiş S-Şekilli Homojen Olmayan Poisson Süreci Modeli'ne göre yazılımı test edecek test ekibinin yazılımı tanınması zaman alacağından, hata tespit oranı test ekibinin yazılımı tanınması ile artmaya başlar [5,6].

Bu modelin varsayımları;

- Her hata birbirinden bağımsızdır ve her hatanın tespit edilme olasılığı eşittir.
- Yazılımda hata tespit etme oranı yazılımın barındırdığı hata sayısı ile doğru orantılıdır.
- Hata tespit etme olasılığı zamandan bağımsızdır.
- Yazılımda bulunan toplam hata sayısı rasgele değişkendir.
- Yazılımda bulunan herhangi bir hata herhangi bir zamanda ortaya çıkabilir.
- (i-1). ve i, hatalar arasındaki süre (i-1). hatanın zamanına bağlıdır.
- Hataların kaldırılması sırasında yeni hata oluşmasına sebep olunmaz.

Modelin ortalama değer fonksiyonu aşağıdaki gibidir;

$$m_t = a(1 - (1 + bx)e^{-bx})$$

Fonksiyonda, parametreler Homojen Olmayan Poisson Süreci Goel-Okumoto Modelindeki gibi a parametresi yazılımda tespit edilen hata sayısını gösterirken b parametresi hata tespit oranını göstermektedir.

2.3 Lineer Regresyon Modeli

Lineer Regresyon Modeli, bir bağımlı değişkenin bir ya da daha çok bağımsız değişkenle ilişkilendirilip aralarında bağıntı kurulması ve bu bağıntının çözümlendirilmesidir [7].

Bu modelin varsayımları;

- Bağımsız değişken ile bağımlı değişken arasındaki ilişki lineerdir.
- Bağımsız değişken değerleri rastgele ya da tahmin edilmiş değerler değil ölçülmüş değerlerdir.

Modelin ortalama değer fonksiyonu aşağıdaki gibidir;

$$m_t = a + bx$$

Fonksiyonda, a parametresi başlangıçta bulunan hata sayısı iken b parametresi her bir iterasyon sonucunda tespit edilebilecek ortalama hata sayısını göstermektedir.

3 Yapılan Uygulama

Bu çalışma kapsamında ilk olarak yazılım güvenilirlik modellerinin uygulanacağı pilot proje belirlenmiş ardından belirlenen projede yazılım güvenilirlik modellerinin parametreleri tahmin edilerek modeller oluşturulmuştur. Son olarak ise oluşturulan modellerin performansı analiz edilmiştir.

3.1 Çalışma Kapsamında Pilot Projenin Belirlenmesi

Çalışma kapsamında belli uygunluk seviyesinde ölçümleri düzenli olarak toplanan ve analiz edilen, geliştirilmesi tamamlanmış, aylık iterasyonlar bazında ek geliştirme yapılan, proje yönetim, tasarım, geliştirme, test, kalite ve konfigürasyon ekibine sahip aynı sektörde müşterisi olan regresyon test kümesi ve otomasyon test altyapısı olan projeye ait hata kayıt verilerine ihtiyaç duyulmuştur. Bu kapsamda İnnova Bilişim Çözümleri Telco Çözümleri bünyesinde geliştirilen aynı sektörde 7 farklı müşteriye sahip "X" Projesine ait 01.06.2015-28.03.2016 tarih aralığındaki hata kayıt verileri kullanılmıştır. "X" Projesi, proje yönetimi, tasarım, geliştirme, test, kalite güvence ve konfigürasyon yönetimi ekipleri olmak üzere 30 kişilik bir proje ekibinden oluşmakta ve farklı müşterilerden gelen talepleri aylık "Kapsam Belirleme" çalışmaları ile belirleyerek Scrum uygulama geliştirme yaşam döngüsü ile geliştirilen yazılımları müşterilerin istediği özellikler doğrultusunda teslimatını gerçekleştirmektedir.

3.2 Yazılım Güvenilirlik Modellerinin Oluşturulması

Yazılımda oluşan hata davranışlarını matematiksel olarak ifade eden yazılım güvenilirlik modelleri, her bir projeye göre uygulanabilmesi için parametrelere sahiptir. Yazılım güvenilirlik modellerinin projelerde uygulanabilmesi için ilk olarak bu parametrelerin tahmin edilmesi gerekmektedir [1]. Model parametrelerinin doğru tahmin edilmesi model performansı açısından kritik öneme sahiptir. Bu çalışma kapsamında Lineer Regresyon, Goel-Okumoto NHPP ve Geciktirilmiş S-Şekli NHPP modelleri içerisinde bulunan ve Tablo 1, Tablo 2 ve Tablo 3 de bulunan parametreler, modellenmesi yapılacak proje için Scrum uygulama geliştirme yaşam döngüsünün planlama aşamasında tahmin edilerek projeye uygulanmıştır. Parametrelerin tahmin edilmesinde Lineer Regresyon modeli için SPSS istatistiksel paket programı yardımıyla regresyon analizi yapılmış, Goel-Okumoto NHPP ve Geciktirilmiş S-Şekli NHPP modelleri için ise MATLAB programı kullanılarak uygulanan en çok olabilirlik yöntemi ile model parametreleri tahmin edilmiştir. Tahmin edilen parametreler model denklemlerine yerleştirilerek yazılım güvenilirliğini ölçmede kullanılacak model denklemleri elde edilmiştir.

Çalışma sırasında yazılım güvenilirlik modellerinin, yazılımda bir sonraki iterasyonda oluşabilecek hataların %95 güven aralığında alabileceği değerler ve tahmin değerinin gerçek değere yakın tahmin edilmesi amaçlanmıştır. Bu kapsamda hata kayıt verilerinin belirli bir iterasyona kadar olan verileri kullanılarak tahmin edilen model parametreleri ile modeller oluşturulmuştur. Oluşturulan modellerin sıradaki iterasyonda ortaya çıkacak hataları tahmin etmedeki performansları ölçülürken bir sonraki

iterasyonda kullanılacak modeller için yeni hata kayıt verileri de kullanılarak model parametrelerinin tahmin edilmesi güncellenmiş ve yeni modeller oluşturulmuştur. Bu döngü iki iterasyon için tekrarlanmış ve elde edilen model denklemleri Tablo 1, Tablo 2 ve Tablo 3'te verilmiştir.

Tablo 1. Lineer Regresyon Modelleri

Tahmin Dönemi	Alt Sınır Tahmin Modeli	Üst Sınır Tahmin Modeli
9. İterasyonu	$y=11,44+17,836x$	$y=45,199+24,521x$
10. İterasyonu	$y=16,338+17,040x$	$y=48,717+22,794x$

Tablo 2. Goel-Okumoto NHPP Modelleri

Kullanılan veri dönemi	Alt Sınır Tahmin Modeli	Üst Sınır Tahmin Modeli
9. İterasyonu	$y=201,687(1-e^{-0,101x})$	$y=317,493(1-e^{-0,216x})$
10. İterasyonu	$y=222,172(1-e^{-0,112x})$	$y=307,584(1-e^{-0,196x})$

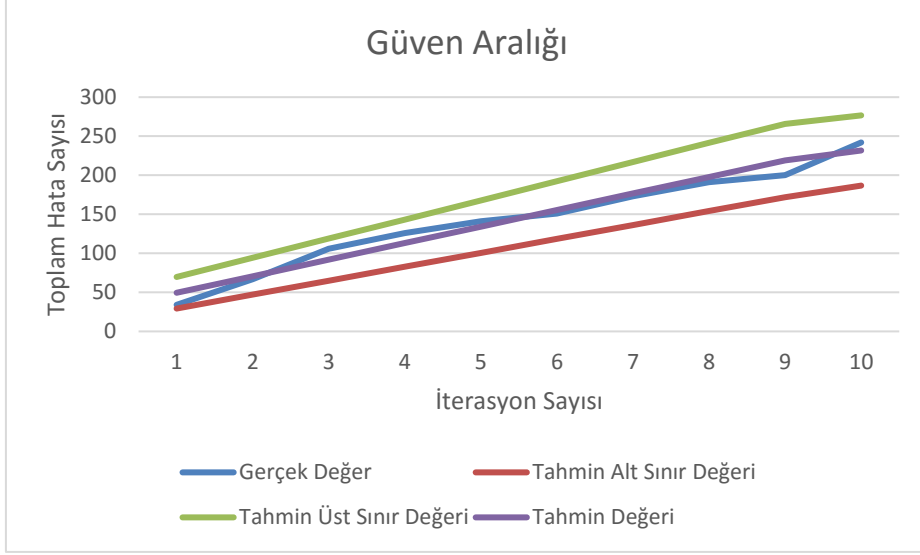
Tablo 3. Geciktirilmiş S-Şekli NHPP Modelleri

Kullanılan veri dönemi	Alt Sınır Tahmin Modeli	Üst Sınır Tahmin Modeli
9. İterasyonu	$y=162,93(1+0,465x)e^{-0,465x}$	$y=209,478(1+0,74x)e^{-0,74x}$
10. İterasyonu	$y=174,419(1+0,439x)e^{-0,439x}$	$y=218,424(1+0,675x)e^{-0,675x}$

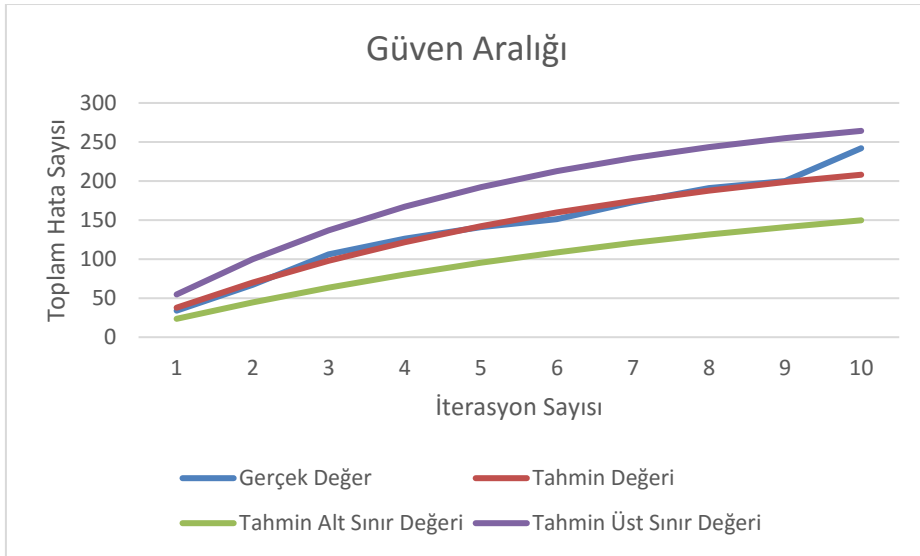
3.3 Model Performansının Analiz Edilmesi

Yazılım güvenilirlik modelleri oluşturulduktan sonra tahmin edilen modellerin gerçek hata kayıt verilerine ne kadar yakın sonuç verdiği değerlendirilmesi gerekmektedir. Literatürde tahmin modellerinin performansını değerlendirmek için pek çok yöntem vardır. Bu çalışma kapsamında elimizde bulunan hata kayıt verilerinin tahmin edilen güven aralıklarının içerisinde bulunup bulunmadığı ve tahmin değerlerinin ne kadar gerçek değere yakın sonuç verdiği değerlendirilmede kullanılmıştır. Model performanslarının değerlendirilmesi Scrum uygulama geliştirme yaşam döngüsünün gözden geçirme aşamasında JİRA hata kayıt aracı yardımıyla elde edilen gerçek hata verileri kullanılarak yapılmıştır.

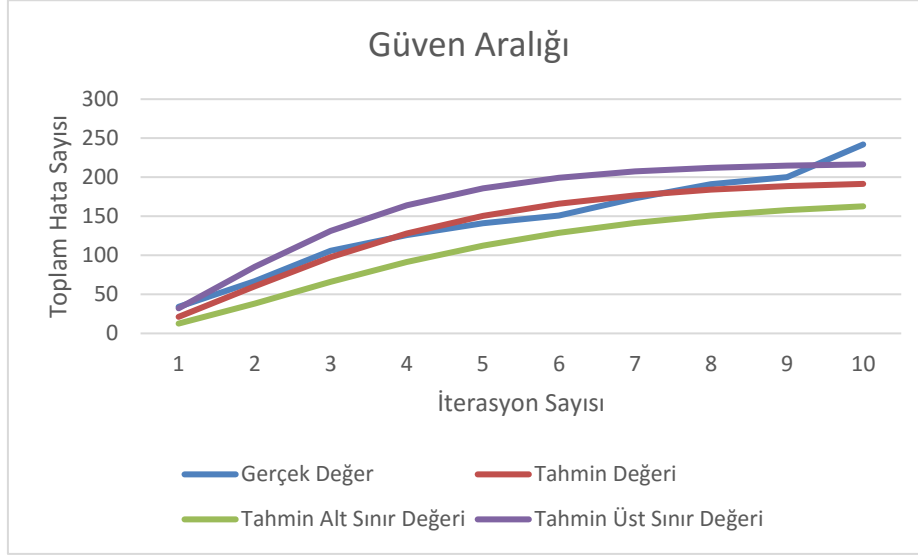
Tahmin modellerinin %95 güven aralıklarında bulunduğu değerler **Şekil 1**, **Şekil 2** ve **Şekil 3**'teki gibidir;



Şekil 1. Lineer Regresyon Modeline Göre Güven Aralığı



Şekil 2. Goel-Okumoto NHPP Modeline Göre Güven Aralığı



Şekil 3. Geciktirilmiş S-Şekilli NHPP Modeline Göre Güven Aralığı

Tablo 4. Gerçek Değer ve Tahmin Değeri Arasındaki Fark

Model	9. İterasyon için Tahmin ve Gerçek Değer Arasında Bulunan Fark	10. İterasyon için Tahmin ve Gerçek Değer Arasında Bulunan Fark	Tahmin ve Gerçek Değer Arasındaki Ortalama Fark
Lineer Regresyon Modeli	18,932	10,302	14,617
Goel-Okumoto NHPP	1,361	33,907	17,634
Geciktirilmiş S-Şekilli NHPP	11,434	50,496	30,965

Şekil 1, Şekil 2 ve Şekil 3'te görüldüğü gibi Lineer Regresyon, Goel-Okumoto NHPP ve Geciktirilmiş S-Şekilli NHPP yazılım güvenilirlik modelleri ile %95 güven aralığı tahminleri yapılmıştır. Bu işlem sırasında elde bulunan on iterasyon verisi arasında ilk olarak sekiz iterasyon verisi model parametrelerinin belirlenmesi ve model oluşturmak için kullanılmıştır. Bu modellerle dokuzuncu iterasyon da oluşabilecek hata sayısı tahmin edilmiştir. Daha sonra dokuzuncu iterasyon hata kayıt verisi de model tahmin etmede kullanılarak yapılan modelleme ile onuncu iterasyon da oluşabilecek hata sayısı için tahmin yapılmıştır. Tablo 4 incelendiğinde ise 9. ve 10. iterasyonlar için tahmin edilen ve gerçekleşen hata sayıları arasındaki farklar hesaplanmıştır.

İlk olarak dokuzuncu iterasyon için yapılmış olan tahminleri ve dokuzuncu iterasyon sonucunda oluşan toplam hata sayısı incelenirse, gerçekleşen toplam hata sayısının her

üç modelde belirtmiş olan güven aralıkları içerisinde bulunduğu gözlemlenmiştir. Bu bakımdan gerçekleşen hata kayıt verisinin tahminlerde belirtilmiş olan tahmini güven aralıklarının içerisinde bulunduğu görülmüştür. Modellerin yapmış oldukları tahminler ve gerçekleşen hata sayıları arasındaki farklar incelendiğinde ise Goel-Okumoto NHPP modelinin gerçek değere daha yakın tahmin bilgisi verdiği söylenebilir. Bu açıdan Geciktirilmiş S-Şekli NHPP ve Lineer Regresyon modeli sırasıyla Goel-Okumoto NHPP modelini takip etmektedir.

Onuncu iterasyon için yapılan alt ve üst sınır tahminleri ve gerçekleşen toplam hata sayısı incelendiğinde ise gerçekleşen toplam hata sayısının Lineer Regresyon modeli ve Goel-Okumoto NHPP modelinin yapmış oldukları alt ve üst sınırlar içerisinde bulunduğu görülebilir. Ancak bununla birlikte Geciktirilmiş S-Şekli NHPP modelinin yapmış olduğu üst sınır tahmininin gerçekleşen toplam hata miktarından az olduğu gözlemlenmektedir. Bu durum projede Geciktirilmiş S-Şekli NHPP yazılım güvenilirlik modelinin kullanılmayacağını belirtmektedir. Onuncu iterasyon için diğer iki modelin tahmin edilen ve gerçekleşen hata sayısı incelendiğinde ise Lineer Regresyon modelinin Goel-Okumoto modeline göre gerçeğe daha yakın tahmin yaptığı görülür.

Bu çalışma özelinde iki iterasyon sonucuna göre gerçek hata kayıt verilerinin güven aralığı içinde olması ve tahmin edilen hata sayısı ile gerçekleşen hata sayısındaki ortalama fark incelendiğinde Lineer Regresyon modelinin diğer modellere oranla daha başarılı olduğu söylenebilir. Lineer Regresyon modelinin diğer modellere göre başarılı sonuç vermesinin sebebi diğer modellerin projede belirli olgunluğa erişildikten sonra projedeki değişimin ve dolayısıyla yeni tespit edilen hata sayısının zamanla azalacağını varsaymasıdır. Ancak projenin iteratif olarak ek geliştirmelerinin devam etmesi ve her iterasyon da yapılan değişiklikte bir azalma olmaması tespit edilen toplam hata sayısında artışa sebep olmaktadır.

Bu çalışmanın geçerliliğini etkileyebilecek faktörler olarak; çalışmada kullanılan hata kayıt verilerinin tek bir projeye ait olması, var olan birçok yazılım güvenilirlik modellerinden sadece üç modelin performansının incelenmesi ve model performanslarının yalnızca iki iterasyon sonucu için değerlendirilmesi düşünülmektedir.

4 Gelecek Çalışmalar

Bu çalışma kapsamında Lineer Regresyon, Goel-Okumoto NHPP ve Geciktirilmiş S-Şekli NHPP yazılım güvenilirlik modelleri incelenmiş ve model performansı açısından Lineer Regresyon Modelinin tahminleme bakımından gerçekleşene en yakın sonucu verdiği görülmüştür.

Çalışma kapsamında sadece ön kabul de koşulan testler sonucu bulunan hata sayısının iterasyona göre değişimi kullanılarak, hata oluşumları modellenmeye çalışılmıştır. İterasyon bazlı projede yapılan değişiklikler farklılaşacağı için her iterasyonda hata oluşumu farklılık gösterir ve her iterasyonda yapılması planlanan değişime göre hata oluşumu modellenmelidir. Bu sebeple bir iterasyon kapsamında değişebilecek kod satır sayısı, yeni geliştirilecek gereksinimlerin sayısı ve zorluğu gibi değişkenlerde eklenecek yeni model oluşturup her iterasyon başında elde edilen verilere dayanarak iterasyon boyunca oluşabilecek hata sayısının tahmin edilmesi hedeflenmektedir.

5 Kaynakça

- [1] Karaömer, R., B., "Web Yazılım Projelerinde Homojen Olmayan Poisson Süreci Yazılım Güvenilirlik Modellerinin Karşılaştırılması" Yüksek Lisans, Endüstri Mühendisliği, Hacettepe Üniversitesi, 2015.
- [2] Kurtel, K. ve Eren, Ş., "Quality Requirements for Software Architecture: Software Reliability," Journal of Computer Science and Engineering, vol. 4, pp. 75-83, 2011.
- [3] Sarı, Ö., Kalıpsız, O., "Yazılım Hata Kestirimi İçin Veri Analizi Yöntemlerinin Kullanılması" 2014 Ulusal Yazılım Mühendisliği Sempozyumu, 2014
- [4] Quyoum, A., Dar, M. U. D. ve Quadri, S. M. K., "Improving Software Reliability using Software Engineering Approach-A Review," International Journal of Computer Applications, vol. 10(5), pp. 41-47, 2010.
- [5] Pham, H., System Software Reliability London: Springer, 2006.
- [6] Sugüneş, Ö., "Yazılımda İstatiksel Süreç Kontrolü ve Güvenilirlik Kestirim Modelleri," Yüksek Lisans, İstatistik Bölümü, Ankara Üniversitesi, 2010.
- [7] Aydın, A., "İteratif Yazılım Geliştirme için Hata Tahminleme Modeli Araştırması: Bir Durum Çalışması," Yüksek Lisans, Bilgisayar Mühendisliği, Hacettepe Üniversitesi, 2014.