

## Test yöneticileri tarafından algılandığı şekliyle yazılım hata raporlarının kalitesi: endüstriyel bir vaka çalışması

Vahid GAROUSİ  
Yazılım Mühendisliği Araştırma Grubu  
Bilgisayar Mühendisliği Bölümü  
Hacettepe Üniversitesi, Ankara, Türkiye  
vahid.garousi@hacettepe.edu.tr  
Maral Yazılım Danışmanlık ve Ar-Ge  
Corporation, Calgary, Kanada

Kadir HERKİLOĞLU, Adem ÇAĞLAR, Aydın  
AKKAYA, Kemal ERGEZER, Onur SERTEL,  
Serap İDİNAK  
Test Müdürlüğü, HAVELSAN A.Ş.  
Ankara, Türkiye  
{kherkiloglu, acaglar, aakkaya,  
kergezer, osertel,  
sidinak}@havelsan.com.tr

**Özet:** Test süreç iyileştirme bağlamında yürütülen bir üniversite-sanayi işbirliği projesi kapsamında, hata raporlarının kullanımının, yararlılığının ve kalitesinin değerlendirilmesi ihtiyacı doğmuştur. Amaç hata raporlarının kalitesini değerlendirmek ve yazılım hatalarının geliştiriciler tarafından etkin ve verimli bir şekilde düzeltilebilmesini sağlayacak hata raporlarını yazabilmektir. Kaliteli hata üretim ihtiyacını karşılamak için, yazarlar yakın zaman içerisinde savunma sanayii ve bilişim alanlarında büyük bir Türk yazılım ve sistem üreticisi/entegratörü olan HAVELSAN A.Ş.'de, 38 adet yazılım geliştiriciden toplanan anket tabanlı bir araştırmayı sunmuş bulunmaktadırlar. Sunulan çalışmayı (geliştirici görüşlerine dayanan) tamamlamak için, yazarlar tarafından başka bir araştırma daha gerçekleştirilmiştir. Bu kez, geliştiriciler yerine, 5 test yöneticisinden hata raporlarının kalitesini değerlendirmek üzere görüşleri toplanmıştır. Bu makalede, test yöneticilerinin anket sonuçları analiz edilip, geliştirici görüşlerine dayanan çalışmanın sonuçları ile karşılaştırılmıştır. Her iki anketin sonuçları, hata raporlarının kalitesinin değerlendirilmesinde kullanılmış, çıkan sonuçlara göre, hata raporlarının yazılma sürecinin iyileştirilmesi hususunda çalışmalara başlanmıştır.

**Anahtar sözcükler:** Hata raporları; kalite değerlendirme; endüstriyel vaka çalışması

## Quality of software defect reports as perceived by test managers: an industrial case study

**Abstract:** In the context of an industry-academia collaboration in the scope of test process improvement, the authors noticed a need to assess the usage, usefulness and quality of defect reports. The objective was to assess the quality defect reports and also to pinpoint improvement areas in defect reports to increase developers' effectiveness in fixing defects. To address the above needs, we recently conducted a questionnaire-based opinion survey which gathered input from 38 software developers in the context of a large Turkish software and systems company providing global solu-

tions in the areas of defense and IT, and reported the results in a previous recent paper. To complement our previous study (opinions of developers), we conducted another survey, this time among a set of five test managers. We report and analyze the results of the managers' survey in the current study and compare them to the previous results (opinions of developers). Analysis of results from both surveys has helped us to assess the usage, usefulness and quality of defect reports and also to pinpoint the necessary improvement areas in defect reports, for which an improvement effort has already started.

**Keywords:** Defect reports; bug reports; quality assessment; industrial case study

## 1 Giriş

Yazılım geliştirme sürecinde, hata raporları, hataları düzeltmek adına geliştiriciler için önemli bilgiler sağlar [1]. Hataları daha hızlı bir şekilde tespit etmek ve düzeltmek için, hata raporlarının daha net ve öz yazılmaları yararlı olabilir [2].

Test süreç iyileştirme bağlamında yürütülen bir üniversite-sanayi işbirliği projesi kapsamında, hata raporlarının kullanımının, yararlılığının ve kalitesinin değerlendirilmesi için bir ihtiyaç doğmuş ve bir çalışma başlatılmıştır. Çalışmanın amacı, hata raporlarının okunma kolaylığı, faydalanılabilirlik durumu ve kalitelerinin değerlendirmesini yapmak ve değerlendirmeden çıkacak sonuçlara göre test mühendislerinin hata raporu yazma pratiklerini geliştirmektir. Çalışmanın endüstri tarafı büyük bir savunma sanayi yazılım ve sistem entegratör firması olan HAVELSAN A.Ş. dir. Söz konusu üniversite-sanayi işbirliği projesi Aksiyon-Araştırma (İngilizcede: Action-Research) metodolojisi [3] prensipleri üzerine kurulmuştur.

Çalışmamızın ilk adımında, hata raporların kullanılabilirlik, yararlılık ve kalitesinin geliştiriciler tarafından değerlendirilmesine odaklanılmıştır ve çalışmanın sonuçları yakın geçmişte bir makale olarak [4] yayınlanmıştır. Bu makalede, çalışmamızın ikinci adımı olarak, hata raporların kalitesi test yöneticileri tarafından değerlendirilmiş ve iki paydaş gözünden hata kalitesinin nasıl farklı algılandığı araştırılmıştır.

Makalenin bundan sonraki bölümleri şu şekilde düzenlenmiştir. Bölüm 2'de vaka (durum) tanımı ve ihtiyaç analizi özetlenmiştir. Bölüm 3 literatürdeki ilgili çalışmaları özetlemektedir. Bölüm 4'te araştırma amaç ve yöntemi sunulmaktadır. Bölüm 5'te sonuçlar ve analizler sunulmuştur. Bölüm 6'da ise sonuçlar tartışılmış, ileriye yönelik çalışmalarla ilgili yönlendirmeler paylaşılmıştır.

## 2 Vaka tanımı ve ihtiyaç analizi

### 2.1 Bağlam

Söz konusu üniversite-sanayi işbirliği projesinin endüstri tarafı olan HAVELSAN A.Ş. (Hava Elektronik Sanayii), 1982 yılında Türk Silahlı Kuvvetleri'nin yazılım mühendisliği alanındaki ihtiyaçlarının giderilmesi amacı ile kurulmuş olan bir şirkettir. Şirket merkezi Ankara'da olmakla birlikte bir çok farklı ilde ve yurtdışında ofisleri bulunmaktadır.

Yazılım mühendisliği yetenekleri açısından, HAVELSAN CMMI seviye 3 ile akredite edilmiştir. Şirketin Kalite, Test ve Süreç Yönetim Direktörlüğünde bağımsız bir Test Müdürlüğü bulunmaktadır. Test edilen yazılım sistemlerinin tipine göre, test müdürlüğü kendi içinde çeşitli test ekiplerine bölünmüştür, örneğin: (1) otomasyon uygulamaları ve görüntü işleme teknolojileri için test ekibi ve (2) yer destek sistemleri için test ekibi. Toplamda, 40'tan fazla test mühendisi çalışmaktadır. Test

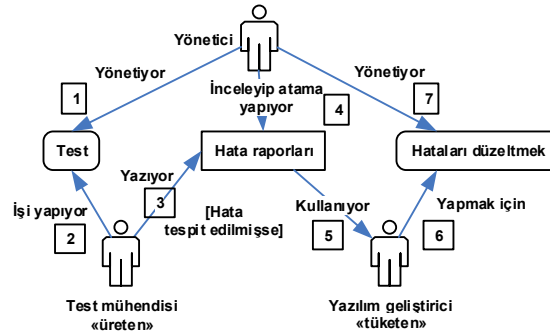
grubu; dünya çapında kullanılan “bağımsız yazılım doğrulama ve geçerleme” (İngilizcede: Independent Software Verification and Validation, ISVV) standart test yaklaşımı ve savunma ve havacılık endüstrileri tarafından çoğunlukla kullanılan, örneğin [5], prensipler üzerine kendi süreçlerini kurmuştur. Test grubu tarafından yürütülen test faaliyetlerinin neredeyse hepsi kara-kutu (black-box) test tipindedir. Firmanın test stratejisine bağlı olarak, beyaz-kutu (white-box) test faaliyetleri yazılım geliştirme ekipleri tarafından yapılmaktadırlar.

HAVELSAN tarafından geliştirilen sistemlerin tipine bağlı olarak (emniyet ve görev kritik sistemler), genel olarak şirketin tüm gruplarında ve özellikle test ekibinde, daha etkin ve verimli yazılım mühendisliği uygulamaları ve pratikleri gerçekleştirmek için sürekli çaba harcanmaktadır. Çeşitli süreç iyileştirme faaliyetleri gerçekleştirilmektedir. Bu sürekli iyileştirme çalışmalarından biri bu makalede bahsi geçen üniversite-sanayi işbirliği projesidir.

## 2.2 İhtiyaç analizi ve son çalışmanın özeti

Sürdürülen üniversite-sanayi işbirliği projesinin konularını belirlemek üzere taraflar arasında bir takım toplantılar gerçekleştirilmiştir. 7 adet konu belirlenmiş ve her konu ayrı ayrı projelendirilerek proje ekipleri oluşturulmuştur. 7 konudan bir tanesi, Test Müdürlüğü için sistematik, etkin ve verimli, Amaç-Soru-Ölçüt (İngilizcede: Goal-Question-Metric, GQM) [6]-tabanlı bir ölçüm ve iyileştirme programı kurmak olarak belirlenmiştir. Bu kapsamda, Test Müdürlüğü için büyük bir GQM ağacı geliştirilirken, test mühendisleri tarafından üretilen önemli bir çıktı (artifact) olarak hata raporları üzerinde durulmuş ve hata raporlarının kullanımının, yararlılığının ve kalitesinin değerlendirilmesi gerekliliği fark edilmiştir. Test mühendisleri hata raporlarının 'üretenleri' olarak, hataları düzeltmek için atanan geliştiriciler 'tüketici' olarak konumlandırılmışlardır. Hata raporlarının daha net ve öz yazılması, hataların daha hızlı bir şekilde tespit edilip düzeltilmesi için yararlı olabilmektedir [2].

Şekil 1 hata raporları içeren 'üretici-tüketici' dinamiklerini göstermektedir. 1...7 arası sıra numaraları bu bağlamda gerçekleştirilen görevlerin ardışık sırasını belirlemektedir. Yukarıda bahsettiğimiz gibi, çalışmamızın ilk adımı olarak, hata raporların kullanımını, faydalılığını ve kalitelerini geliştiriciler tarafından değerlendirme üzerine odaklanılmıştır [4]. O çalışmadan sonra, söz konusu dinamığe 'yönetici' aktörü de (actor) eklenmiştir, çünkü test yöneticisi bu süreçte önemli bir role sahiptir. Örneğin: test işlemlerini yönetmek, hata raporlarını incelemek vb.



Şekil 1- Hata raporları içeren 'üretici-tüketici' dinamikleri

### 3 İlgili çalışmalar

Literatürde tekrarlı (duplicate) hatalar [7] ve hata atama problemi için [8], birçok araştırma mevcuttur. Ama, hata raporlarının kullanımı, faydaları ve kalite analizleri için daha az araştırma yapılmıştır [1, 9-12]. Etkili (yüksek kalite) hata raporu yazmak için, akademik (formal) veya gri literatürde birçok kılavuz mevcuttur, örneğin, [2, 13-15].

‘İyi bir hata raporu nasıl olur?’ başlıklı çalışma [1], iyi bir hata raporu tanımlamak için, üç büyük projenin (Apache, Eclipse ve Mozilla) geliştiricileri ve kullanıcıları arasında yapılmış bir anket sonucudur. Alınan 466 yanıtın analizi sonucunda, hata raporlarında geliştiricilerin ihtiyacı duyduğu bilgiler ile hata raporlarını yazanların verdiği bilgiler arasında bilgi uyumsuzluğu olduğu gösterilmiştir. Geliştiricilerin çoğu, en çok yararlı bulduğu bilgi olarak, hataları tekrarlama adımları, yığın izleri (stack traces) ve test durumlarını belirtmişlerdir. Ancak, hata raporlarını yazanlar, aynı bilgiler için “sağlanması en zor bilgiler” bildiriminde bulunmuşlardır.

Açık-kaynak yazılım projelerinde, geliştiriciler ve kullanıcılar arasındaki işbirliğini anlamak ve hata izleme sistemlerini iyileştirmek için, [9]’daki çalışma nitel ve nicel bir şekilde Mozilla ve Eclipse projelerden alınan 600 hata raporunu analiz etmiştir. Yazarlar hata raporlarına ait sorular ve yanıtları kategori ve proje bazında ayırmışlardır. Analizlerin sonucunda kullanıcıların rolü sadece raporlama değil, belki daha ötesi, hata raporlarının üzerinde aktif ve sürekli katılımlarının olması gerektiğini göstermiştir.

[10]’deki çalışmada, Mozilla Firefox projesi için açılmış, serbest erişimin olabildiği 27.000’den fazla yazılım hata raporunun yüzeysel özelliklerinin istatistiksel analizlerine dayanan hata raporu kalitesinin tanımlayıcı modeli sunulmuştur. Sunulan model ile bir hata raporunun belirlenen bir zaman içerisinde önceliklendirilip önceliklendirilemediği öngörülme çalışılmıştır. Yapılan analizin hata raporlama sistemlerine etkisinin olacağı belirtilmiş, hata raporu oluşturulurken vurgulanması gereken özellikler önerilmiştir.

Başka bir endüstri araştırması [11] hatayı yeniden üretmek için gerekli adımlar ile adımlar uygulandığında beklenen davranışı, hata raporlarının en önemli bilgileri olduğunu bulmuştur. Maalesef, hata raporu yazanların çoğunun bu teknik bilgidен yoksun olduğu belirtilmiş ve bu yüzden yazarlar bu süreci otomatize edecek yöntemler bulmayı önermişlerdir. Başka bir araştırma ise, aynı otomasyon sisteminin yeni hata takip sistemlerine entegre edilmesini tavsiye etmiştir.

Eclipse’teki hata raporlarının kalitesi üzerine yapılan bir diğer çalışmada [12], geliştiriciler içerisinde yığın izlerinin bulunduğu hatalar en kullanışlı olarak değerlendirilirken, içerisinde yanlış veya eksik bilgi bulunan hata raporları hataları adreslemede en az yararlı hatalar olarak bildirilmişlerdir.

İlk çalışmamızda bahsedildiği gibi [4], çalışmalar sırasında yapılan değerlendirmeler ve ilgili çalışmalar arasındaki benzerlik dikkate alındığında, [1] makalesinde sunulan hata rapor kalite ölçüğü uygun bulunmuş ve bu çalışmada kullanılmıştır.

### 4 Araştırma amaç ve yöntemi

Bölüm 2.2’de tartışılan çalışmanın ihtiyaçlarına göre ve Amaç-Soru-Ölçüt (İngilizcede: Goal-Question-Metric, GQM) [6] hedefi şablonu kullanılarak yapılan ve bu makalede sunulan ampirik araştırmanın amacı test yöneticilerinin bakış açısından hata raporlarının kalitesini anlamak, bu görüşleri geliştiriciler tarafından verilen görüşlerle kıyaslamak [4], ve bu bulguları hata raporlarının ve raporlama uygulamalarının iyileştirilmesinde kullanmaktır.

Araştırmanın amacı belirlediği gibi, vaka çalışmamızın tipi 'keşifçi' (exploratory) olduğu üzere [16] amacımız bu kapsamda durumun ne olduğunu öğrenmek, yeni anlayışlar aramak ve gelecek iyileştirme ve araştırmalar için fikirler ve hipotezler oluşturmaktır. Makalemizde projenin amacına dayanarak, iki araştırma sorusu (ArSor, “Research Questions”) ortaya koyulmaktadır:

- ArSor 1- Yöneticilerin bakış açısından hata raporlarının kalitesi nedir ve nasıl artırılabilir?
- ArSor 2- Hata raporlarının kalitesi, test yöneticilerinin ve geliştiricilerin bakış açılarından nasıl benzerlikler ve farklıklar içermektedir?

Çalışmanın sorularını cevaplamak için, test yöneticileri arasında bir anket tasarlanıp veri toplanmıştır. [4]'te bahsedildiği gibi, anketin tasarımı çoğunlukla [1]'deki raporlanan anket baz alınarak yapılmıştır. Tablo 1 tasarlanan anketi göstermektedir. Ankette toplam 24 madde bulunmaktadır ve maddeler üç gruba bölünmüşlerdir: (1) Alanların kalitesi, (2) Alanlardaki hatalar, ve (3) Diğer sorunlar. Her test yöneticisinden her yönettiği proje için, bir kere anketi doldurması istenmiştir (kesin proje sayısı bilgi gizliliği için verilememektedir). Her madde 1-5 arası 5-noktalı Likert skalası (Likert scale) değerleri arasından seçilerek cevaplanmıştır (Tablo 2 de gösterildiği gibi). Sadece 5 adet test yöneticisi bulunduğundan, sayı azlığı nedeniyle anket örnekleme yöntemi olarak, örnekleme yöntemleri, örneğin “tabakalı örnekleme” (“stratified sampling”) [17], kullanılamamıştır.

**Tablo 1-Hata raporlarının kalitesini ölçmek için tasarlanan anket**

| Alanların kalitesi   | Alanlarda hatalar  |
|--|--|
| <ul style="list-style-type: none"> <li>• Hatayı tekrarlama adımları (Steps to reproduce)</li> <li>• Gözlemlenen davranış (Observed Behaviour)</li> <li>• Beklenen davranış (Expected Behaviour)</li> </ul> | <ul style="list-style-type: none"> <li>• Ürün adı</li> <li>• Bileşen adı</li> <li>• Versiyon numarası</li> <li>• Donanım</li> <li>• İşletim sistemi</li> <li>• Gözlemlenen davranış</li> <li>• Beklenen davranış</li> <li>• Kaynak kod örneklerinde</li> <li>• Hatayı tekrarlama adımları</li> <li>• Test durumları</li> <li>• Yığın izleri</li> <li>• Kötü gramer</li> <li>• Yapılandırılmamış metin</li> <li>• Nesir metni</li> <li>• Çok uzun metin</li> <li>• Teknik olmayan dil</li> <li>• Yazım sorunları</li> </ul> |
| Diğer sorunlar   |  |
| <ul style="list-style-type: none"> <li>• Yanlış hata önemi</li> <li>• Yanlış hata öncelikliliği</li> <li>• Tekrarlı hatalar</li> <li>• Eksik bilgiler</li> </ul>   |  |

**Tablo 2- Hata raporlarının kalite ölçmesi için kullanılan 5-noktalı Likert makyası (Likert scale)**

| Değer / value | Açıklama            | Explanation         |
|---------------|---------------------|---------------------|
| N/A           | Geçerli değil       | Not Applicable      |
| 1             | Gözlenmemiştir      | Not Observed        |
| 2             | Nadiren gözlemlenen | Rarely Observed     |
| 3             | Bazen gözlemlenen   | Sometimes Observed  |
| 4             | Sık gözlenen        | Frequently Observed |
| 5             | Daima gözlemlenen   | Always Observed     |

## 5 Sonuçlar ve analiz

ArSor 1'i cevaplamak için, Şekil 2 hazırlanmış ve yöneticilerin bakış açısından hata raporlarının kalitesi sunulmuştur. Genelde, bilgi gizliliği sağlamak için, değerlerin ortalaması alınmış ve raporlanmıştır, örneğin: Şekil 2 deki alanların kalite değerleri için, test grubu G1. Literatürde ve önceki araştırmamızda [4] görüldüğü gibi, hata raporlarında en önemli üç alan şunlardır: Hatayı tekrarlama

adımları (steps to reproduce), gözlemlenen davranış (observed behaviour), ve beklenen davranış (expected behaviour).

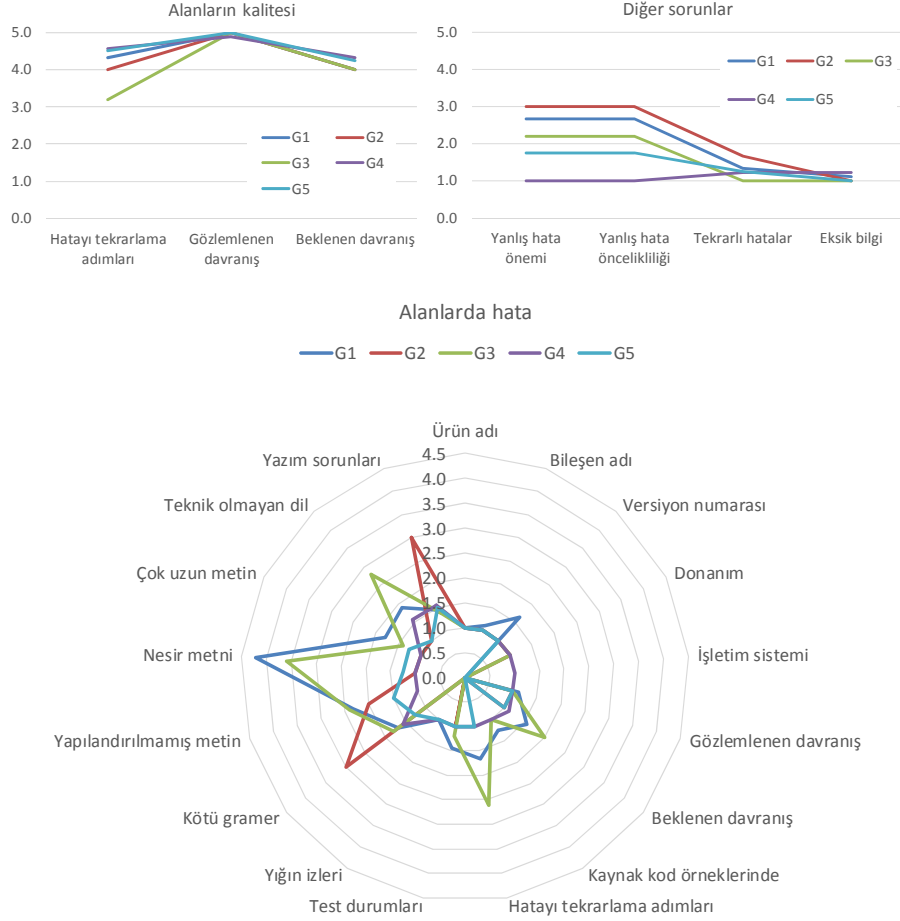
Şekil 2’de görüldüğü üzere, gözlemlenen davranış hata raporlarının hemen hepsinde mutlaka belirtildiğinden (5 test grubu arasında, dört tane 5, ‘daima gözlemlenen’, değeri ve sadece bir 4,9 değeri), yazılan hata raporlarının gözlemlenen davranışlar açısından çok kaliteli olduğu değerlendirilebilir. Tekrarlama adımları ve beklenen davranışlar açısından bakıldığında, bu değerlerin genelde (kalite notlar 3-5 arasında değişiyor) hataların içerisinde yer aldığı ve bu açıdan da hataların bu iki özellik açısından kabul edebilir bir kalitede yazıldıkları söylenebilir.

Hata raporu kalitesini etkileyen diğer hata alanlarında yapılan yanıtlara baktığımızda, beş grup arasındaki ortalama notlar 1.0 ve 2.2 aralığında değişmektedir. En çok iyileşmeye ihtiyaç duyan alanlar olarak, yazı tarzı ile ilgili alanlar raporlanmışlardır: nesir metin (prose text), kötü gramer (bad grammar), ve yapılandırılmamış metin (unstructured text). Nesir metin, dil kurallarından başka hiçbir ölçüye bağlı olmayan düz ve tabu anlatma yoludur. Sade nesir, konuşma dilinde yazılan, açık, tabii nesirdir. Hataların açılma döneminin testler koşutururken veya test oturumunun hemen ardından olduğu göz önünde bulundurulduğunda, proje gecikmelerinin en çok etkilediği faz olarak test fazının süresinin azalmasının, bu tip hataların oluşmasına bir etki yarattığı düşünülmektedir.

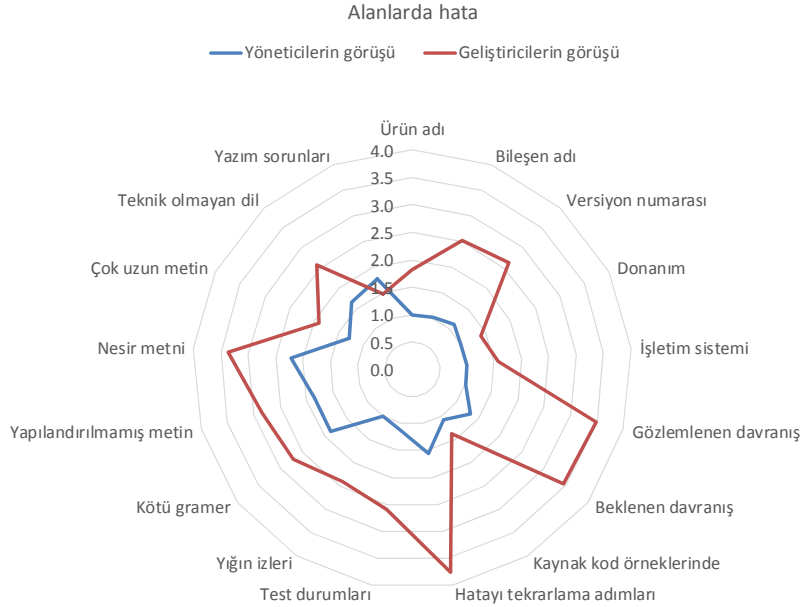
Genelde yazılım mühendisleri için teknik yazma yetenekleri çok önemli bir yetenektir [18]. Ticaret, pazarlama ve diğer sektörlerde vurgulanan deyim: “*Ne söylediğiniz çok önemli değil, ama onu nasıl söylediğiniz önemlidir*” [19], bu durum hata raporları konusunda da önemli görünmektedir. Etkili (yüksek kalite) hata raporu yazmak için, birçok online kılavuzda, örneğin [13, 14], doğru dil ve teknik kullanmanın önemi vurgulanmıştır.

Şekil 2’de, diğer sorunlara baktığımızda, hata raporlarının içinde eksik bilgiler hemen hemen yoktur. Yanlış hata öncelikliliği (priority) ve yanlış hata önemi (severity) beş test grubu arasında farklı sıklıklarla karşılaşılmaktadır. Bu değerlerin yanlışlığı test yöneticileri ve proje yönetimi için büyük sorun teşkil etmektedir çünkü ürün kalitesinin belirlenmesinde açılan hataların sayısından çok niteliği daha büyük önem taşımaktadır. Önem ve öncelik tanımlarının ekipler içerisinde yeniden anlatılması ve açılan hataların takibi için planlama yapılarak, bu problemin gelecekte oluşturacağı riskler için azaltma planı yapılmıştır. Tekrarlı hatalar (duplicate bugs) [7] nadiren görülmüştür. Bu haliyle bir sorun teşkil etmemekle birlikte, gelecekte oluşabilecek sorunun çözümü için mevcut teknik veya araçlarının kullanımı [20]’de tavsiye edilmiştir.

ArSor 2’yi cevaplamak için, Şekil 3 hazırlanmıştır. Grafikte, hata raporlarının kalitesi, test yöneticileri ve geliştiricilerin bakış açısından karşılaştırılmıştır. Şekilde, yöneticiler için gösterilen değerler, Şekil 2’deki değerlerin ortalama değerleridir. Şekil 3’de görüldüğü üzere, genelde yöneticiler hata raporlarının kalitesini geliştiricilerden daha olumlu değerlendirmektedirler. Bu ilginç gözlem için birkaç olasılık düşünebiliriz: (1) yöneticiler ile kıyasla, geliştiriciler hata raporlarının gerçek tüketicileridirler ve dolayısıyla, geliştiriciler daha objektif bir şekilde kalite değerlendirmesi yapmış olabilirler, (2) yine birinci sebepten dolayı, ürünün (hataların) sahibi olarak test yöneticileri kaliteye daha iyimser (optimistlik) bakmış olabilirler, (3) Geliştiriciler anketi doldururken, hata örneklerini yeniden incelememiş, akıllarında kalan intiba ile değerleri doldurmuş olabilirler. Test yöneticileri her projeden yeteri kadar örnekleme inceleyerek anketi doldurmuşlardır, bu durum geliştiricilerin değerlendirmesinde olması gerekenin altında bir değer oluşturmuş olabilir.



**Şekil 2- ArSor 1'e cevap: Yöneticilerin bakış açısından hata raporlarının kalitesi**



**Şekil 3- ArSor 2'e cevap: Hata raporlarının kalitesi, test yöneticileri ve geliştiriciler bakış açılarından nasıl benzerlikler ve farklıklar içermektedir**

## 6 Tartışma

### 6.1 Bulguların özeti

Bu makalede hata raporlarının kullanımının, yararlılığının ve kalitesinin belirlenmesine yönelik daha önce geliştiriciler ile yapılan ankete dayalı bir araştırmanın benzeri, test yöneticileri ile tekrarlanmış ve her iki bakış açısı kıyaslanmıştır. Bunun için 2 araştırma sorusu ile önce test yöneticilerinin bakış açısıyla hata raporlarının kalitesi ölçülmüş, sonrasında hata raporlarında sık karşılaşılan hatalar veya eksikler değerlendirilerek, bulgular geliştirici araştırmasının bulgularıyla kıyaslanmıştır. Buna göre her iki araştırma grubunun sonuçları arasında yüksek benzerlik (korelasyon) görülmüştür. Sadece 2. araştırma sorusunun (ArSor2) yanıtları incelendiğinde yine geliştiricilerin ve test yöneticilerinin değerlendirmelerinde benzerlik olmakla birlikte değer ortalamalarında farklılıklar görülmüş, bunun da farklı rollerde olmanın yaratmış olduğu değer algısından kaynaklandığı değerlendirilmektedir.

### 6.2 Geçerliliğe tehditler ve onları ortadan kaldırmak

Bu bölümde, standart bir kontrol listesi temel alınarak [21], hazırlanmış çalışmamıza sınır teşkil edebilecek olası geçerliliğe tehditlerden ve bunları nasıl azalttığımızdan, ortadan kaldırmaya çalıştığımızdan bahsedilecektir. Dört tip olası geçerliliğe tehdit dikkate alınmıştır: içsel geçerlilik (internal validity), yapısal geçerlilik (construct validity), sonuç geçerlik (conclusion validity) ve dış geçerlik (external validity).



**İçsel geçerlilik:** İç geçerlilik ile bir çalışma ve çıkarılan verilere dayalı bir nedensel sonuç garanti edilir ve sunulan bilimsel çalışmaların bir özelliğidir. Bu çalışmada iç geçerliliğine bir tehdit seçim yanlılığıdır (yani, araştırmamıza katılan yöneticilerin rastlantısal olmamasıdır). Bölüm 4'de açıkladığı gibi, firmada, sayı olarak, sadece 5 adet test yöneticisi bulunmaktadır. Anketi doldurmak için, sadece onlar davet edilmiştir. Dolayısıyla, bu konuda başka bir yöntem yürütülemezdi.

**Yapısal geçerlilik:** Bu geçerlilik, çalışmanın nesnelere gerçekten çalışmanın arkasındaki teoriyi temsil etme ölçüsü ile ilgilidir. Diğer bir deyişle, konu aslında çalışmamızda gerçek dünyada veya firmada hata raporlarının kalitesinin doğru ölçülüp ölçülmediği ile ilgilidir. Kullanılan yaklaşım (anket) literatürdeki diğer anket çalışmalarına benzemektedir. Her soru için oylar sayılmış ve istatistiksel analizler yapılmıştır. Literatüre göre, oylama verilerine dayalı sonuçlar, belli bir ölçüde, çalışma kapsamında şirket uygulayıcılarının çoğunluğunun görüşlerini yansıtmaktadır.

**Sonuç geçerlilik:** Bir çalışmanın sonuç geçerliliği, onun sonuçlarının titiz ve tekrarlanabilir uygulama ile ulaşılabilir olup olmadığı ile ilgilidir. Bu çalışmada, hata raporlarının kalitesi niteliksel olarak ölçülmüş ve analiz edilmiştir. İki araştırma sorusu (ArSor, "Research Questions") ortaya konulmuş ve her ArSor için, 5'li Likert ölçeği kullanılarak anket tasarlanıp veri toplanmıştır. İstatistiksel analizler ile araştırma soruları cevaplanmıştır. Böylece, çalışmanın tüm sonuçları toplanan verilere bağlanmışlardır.

**Dış geçerlilik:** Dış geçerlilik bir çalışmanın sonuçlarının genelleştirilebilir olma ölçüsü ile ilgilidir. Bölüm 4'te açıklandığı gibi, firmada, sayı olarak, sadece 5 adet test yöneticisi var olduğundan, anketi doldurmak için daha çok kişiyi davet edemedik. Dolayısıyla, dış geçerlilik açısından sonuçlarımız sadece sözü geçen test ekiplerine aittirler ve çalışmanın sonuçları diğer test takımlarına ve firmalara genelleştirilemezler. Ama en azından, yazılım mühendisliğinde önemli olan ampiriklik ve kanıt olma açısından, sonuçlarımız değerli bir katkı sayılabilirler.

## 7 Sonuç ve gelecek çalışmalar

Test süreç iyileştirme bağlamında yürütülen bir üniversite-sanayi işbirliği projesi kapsamında, hata raporlarının kullanımı, yararlılığı ve kalite değerlendirmesi için bir ihtiyaç oluşmuştur. Çalışmamızın ilk adımı olarak, hata raporlarının okunması, kullanımı ve kalitelerinin geliştiriciler tarafından değerlendirilmesi üzerine odaklanılmış ve ilk çıktısı yakın geçmişte yayınlanmış bir makale olarak [4] sunulmuştur. Daha önceki çalışmayı (geliştiriciler görüşleri) tamamlamak için, bu makalede beş test yöneticisinden, hata raporlarının kalitesini değerlendirmek üzere görüşleri toplanmıştır. Makalede test yöneticilerin anket sonuçları analiz edilmiş ve önceki sonuçlarla (geliştirici görüşleriyle) karşılaştırılmıştır. Her iki anketin sonuçları kullanılarak, hata raporlarının kalitesi değerlendirilmiş ve hata raporların yazılmasında gerekli iyileştirmeler yapılabilmesi için, bir çalışma başlatılmıştır.

Gelecekte yapılacak çalışma planı olarak; bu çalışmanın devamında, hata raporların yazılmasında gerekli iyileştirme çalışmalarının sürdürülmesi ve bir süre sonra her iki anketin gelecekte yeniden tekrarlanması planlanmaktadır. Böylece, önce/sonra (before/after) durumları analiz edilebilecektir.

## Kaynaklar

- [1] N. Bettenburg, S. Just, A. Schroter, C. Weiss, R. Premraj, and T. Zimmermann, "What makes a good bug report?," presented at the Proceedings of the ACM SIGSOFT International Symposium on Foundations of software engineering, 2008.

- [2] J. D. Strate and P. A. Laplante, "A Literature Review of Research in Software Defect Reporting," *IEEE Transactions on Reliability*, vol. 62, pp. 444-454, 2013.
- [3] P. S. M. d. Santos and G. H. Travassos, "Action research use in software engineering: An initial survey," in *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*, 2009, pp. 414-417.
- [4] V. Garousi, E. G. Ergezer, and K. Herkiloğlu, "Usage, usefulness and quality of defect reports: an industrial case study," in *International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2016.
- [5] N. Silva and R. Lopes, "10 Years of ISVV: What's Next?," in *IEEE International Symposium on Software Reliability Engineering Workshops*, 2012, pp. 361-366.
- [6] V. S. Rini and E. Berghout, *The Goal/Question/Metric Method*: McGraw-Hill Education, 1999.
- [7] N. Bettenburg, R. Premraj, T. Zimmermann, and K. Sunghun, "Duplicate bug reports considered harmful: really?," in *IEEE International Conference on Software Maintenance*, 2008, pp. 337-345.
- [8] M. R. Karim, G. Ruhe, M. M. Rahman, V. Garousi, and T. Zimmermann, "An Empirical Investigation of Single-objective and Multi-objective Evolutionary Algorithms for Developer's Assignment to Bugs," *In Press, Journal of Software: Evolution and Process*, 2016.
- [9] S. Breu, R. Premraj, J. Sillito, and T. Zimmermann, "Information needs in bug reports: improving cooperation between developers and users," in *Proceedings of the Conference on Computer-supported Cooperative Work*, 2010, pp. 301-310.
- [10] P. Hooimeijer and W. Weimer, "Modeling bug report quality," in *Proceedings of IEEE/ACM international conference on Automated Software Engineering*, 2007, pp. 34-43.
- [11] E. I. Laukkanen and M. V. Mantyla, "Survey Reproduction of Defect Reporting in Industrial Software Development," in *International Symposium on Empirical Software Engineering and Measurement*, 2011, pp. 197-206.
- [12] N. Bettenburg, S. Just, A. Schroter, C. Wei, R. Premraj, and T. Zimmermann, "Quality of bug reports in Eclipse," in *Proceedings of the OOPSLA workshop on eclipse technology eXchange*, 2007, pp. 21-25.
- [13] 35 Mozilla contributors, "Bug writing guidelines," [https://developer.mozilla.org/en-US/docs/Mozilla/QA/Bug\\_writing\\_guidelines](https://developer.mozilla.org/en-US/docs/Mozilla/QA/Bug_writing_guidelines), Last accessed: Dec. 2015.
- [14] S. Tatham, "How to Report Bugs Effectively," <http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>, Last accessed: Dec. 2015.
- [15] E. Hendrickson, "Writing Effective Bug Reports," *Software testing and quality engineering magazine*, pp. 10-11, 2001.
- [16] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, pp. 131-164, 2009.
- [17] T. R. Lunsford and B. R. Lunsford, "The Research Sample, Part I: Sampling," *J. Prosthetics and Orthotics*, vol. 7, pp. 105-112, 1995.
- [18] L. Levine, L. H. Pesante, and S. B. Dunkle, "Technical Writing for Software Engineers," *Technical report*, <http://www.sei.cmu.edu/reports/90cm023.pdf>, 1991, Last accessed: May 2016.
- [19] The marketing donut, "It's not what you say, but how you say it," <http://www.marketingdonut.co.uk/marketing/pr/building-relationships-with-the-media/it-s-not-what-you-say-but-how-you-say-it>, Last accessed: May 2016.

- [20] X. Wang, L. Zhang, T. Xie, J. Anvik, and J. Sun, "An approach to detecting duplicate bug reports using natural language and execution information," presented at the Proceedings of the 30th international conference on Software engineering, Leipzig, Germany, 2008.
- [21] R. Feldt and A. Magazinius, "Validity Threats in Empirical Software Engineering Research-An Initial Survey," in *SEKE*, 2010, pp. 374-379.