

# Yazılım hatalarının atanması: bir sistematik literatür haritalaması

Bahar GEZİCİ<sup>1</sup>, Nurseda ÖZDEMİR<sup>1</sup>, Vahid GAROUSİ<sup>2,1</sup>

<sup>1</sup>Yazılım Mühendisliği Araştırma Grubu  
Bilgisayar Mühendisliği Bölümü, Hacettepe Üniversitesi, Ankara, Türkiye  
<sup>2</sup>Wageningen Üniversitesi, Hollanda

<sup>1</sup>{bhrgezici, nursedaozdemr}@gmail.com, <sup>2</sup>vahid.garousi@wur.nl

**Özet.** Büyük ölçekli yazılım projelerinde yazılım hata raporunu ilgili takıma veya ilgili yazılım geliştiriciye atama sürecini yönetmek zorlu ve zaman alan bir süreçtir. Hata atama süresini azaltmak ve doğru karar verme başarısını artırmak için çeşitli yazılım hata ataması yöntemleri geliştirilmiştir. Hata atama sürecinde otomatik karar vermeyi desteklemek için araştırmacılar ve uygulayıcılar çeşitli yöntem, araç ve süreçler sunmuşlardır. Yazılım hata ataması hakkında yapılan akademik ve teknik araştırma sayısı artmaktadır ve bu durum, mevcut çalışmalarını ve uygulamalarını sistematik olarak kategorize etme ve genel bir bakış açısı sunma ihtiyacını beraberinde getirmiştir. Mevcut çalışmalara genel bir bakış sunmak amacıyla yazılım hata ataması konusunda bu sistematik haritalama çalışması yapılmıştır. Bu çalışmada akademik literatürü taramak amacıyla çeşitli arama motorları (Google Scholar ve Scopus gibi) kullanılmıştır. Bu SH ve sonuçları; konferanslarda yayınlanmış makaleler (45 tane), dergilerde yayınlanmış makaleler (24 tane), sempozyum makaleleri (12 tane), çalıştay makaleleri (4 tane), tezler (3 tane) ve diğerlerinden (5 tane) oluşan 93 birincil kaynağa dayanmaktadır. Sonuç olarak, birincil çalışmaların yaklaşık olarak %46'sı güçlü deneysel çalışma, olup %40'ı Bayes İstatistiği makine öğrenme yöntemini kullanmıştır.

**Anahtar sözcükler:** Yazılım hatası; Hata atanması; Sistematik literatür haritalaması

## Assignment of software bugs: a systematic literature mapping study

**Abstract.** Managing the assignment of bug report to related team or related developer is challenging and time consuming issue in large-scale software projects. In order to reduce the assignment time and to increase success of right decision making automated bug assignment approaches presented. To support automated decision making on bug assignment, researchers and practitioners propound various methods, tools and processes. Number of academic and technical sources have increased and this has brought the need to systematically categorize the current studies and practices and to provide an overview. In order to give an overview of current studies, we performed a systematic

mapping study in the area of bug assignment. We searched the academic literature using several search engines (e.g., Scopus, and Google Scholar). The SM and its results are based on 93 primary studies, which were published sources like conferences (45), journals (24), symposiums (12), workshops (4) and theses (3) and others (5). As a result, approximately 46% of the primary studies are empirical studies, and 40% used Bayesian statistics as machine learning method.

**Keywords:** Bug; Bug assignment; Systematic mapping; Systematic literature mapping

## 1 Giriş

Değişim ve hata izleme sistemleri, yazılım projelerini yönetmek için vazgeçilmez unsurlardır. Hata izleme sistemleri, problem raporları ve gözlenen başarısızlıklarla ilgili ayrıntılı açıklamalara sahip olan hata raporlarını saklar. Projelerin her aşamasında, bildirilen hataların büyüklüğü doğal olarak artmaktadır. Hata raporlarının sayısındaki artışla başa çıkabilmek için her bir hata raporu analiz edilmelidir. Hata atama (triaging), hata durumunun belirlenmesi, hata önceliğinin belirlenmesi, yeni bildirilen bir hatanın daha önce rapor edilip edilmediğinin belirlenmesi, hata raporunun uygun takıma veya geliştiriciye atanması gibi çeşitli adımları içerir. Hata ayıklama odaklı sorunlarından biri de hata atamasıdır. Bildirilen hatayı çözmek için hata dağıtıcısı (triager) bu hata için en uygun geliştiriciye karar vermelidir. Hatayı uygun geliştiriciye (özellikle proje yöneticisi veya ekip lideri) atamaktan sorumlu kişi (bug triager), geliştiricilerin uzmanlık seviyesi, geliştiricilerin mevcut iş yükü ve geliştiricinin üretkenliği hakkında bilgi sahibi olması nedeniyle manuel karar verme sürecinde çok zaman harcamaktadır. Bu atama kararları, proje takviminde önemli bir role sahiptir. Çünkü atama saatler veya günler sürebilir (zaman alıcı görev) ve yeni hatalara neden olan yanlış kararlarla veya atama (yeniden atama işlemi) olayıyla sonlanabilir. Manuel hata atamasının çeşitli sebeplere dayanmasından dolayı araştırmacılar, hataların hangi geliştiricilere atanması gerektiği konusunda yarı ve tam otomatik yaklaşımlara odaklanmışlardır. Bununla beraber, birçok araştırmacı ve uygulayıcı hata atama sürecindeki zorlukları analiz etmiş ve bu alanda çeşitli yöntem, teknik, araç ve ölçütler önermişlerdir.

Bu çalışmada, hata atama konusuyla ilgili güncel çalışmaların sistematik olarak sınıflandırılması ve haritalandırılması ile ilgili genel bir bakış açısı yakalamaya çalışmaktayız. Çalışmalarımızın katkısı şöyledir: Hataların atanması konusunda çevrimiçi makale deposunda var olan birincil çalışmaları sistematik haritalama. Bu çalışmanın geri kalan kısmı ise aşağıdaki şekilde düzenlenmiştir. Bölüm 2’de, hata raporu ve hata ataması hakkında kısa bir özet verilmiştir. Bölüm 3, genel sistematik haritalama (SH) süreci, bu çalışmada izlenen hedef ve araştırma soruları da dâhil olmak üzere araştırma yöntemimizi tanımlamaktadır. Bölüm 4 ile makale seçim sürecini tartışmaktayız. 5. bölüm, haritanın yinelemeli gelişimini analiz etmektedir. Sistematik haritalamanın sonuçları Bölüm 6’da sunulmaktadır. Bölüm 7 ile SH sonuçlarının araştırmacılar ve uygulayıcılar üzerindeki etkileri özetlenip çalışmamızın geçerliliğini tehdit edecek

olasi unsurlar tartisilmaktadir. Son olarak, Bölüm 8, bu haritalamanin sonuclarini ve gelecekte yapilmasi planlanan calismalari belirtir.

## 2 Alan özeti ve ilgili calismalar

Yazilim projelerinde, eger sistem beklenmedik sonuclar veriyorsa veya yanlis davranis gosteriyoorsa, bu uygunsuzlugun nedeni hata olarak adlandirilir. Bu hatayi bulan kişi (test mühendisi gibi) hatanın çözümleyicisi (gelistirici gibi) hakkında bilgi vermelidir. Bu bilgi degisiminin resmi yolu hatayi hata raporuyla belgelemektir. Gelistiricilere hata hakkında ayrintili bilgi vermek için, hata raporlarinin önceden tanimlanmis birtakim alanlari olmalidir. Örnegin, bazı hatalar önemsizdir ve kullanicinin etkinligini bloke etmemektedir; bu nedenle bu hatalar derhal çözülmesi gereken önceliğe sahip olmamalıdır (öncelik) veya kullanicinin her eyleminden sonra bazı hatalar sık sık tekrar ederken bazı hatalarla karšılařma olasılıđı daha düşüktür (ađırlık). Hata raporunun bir diđer alanı da hatanın o anki durumunu gösteren durum alanıdır. Örnegin, bir hata ilk rapor edildiğinde, hata atayıcısı (bug triager) hatanın varlığını ve benzersizliğini doğrulamadıkça, bu hata UNCONFIRMED (onaylanmamış-dođrulanmamış) olarak işaretlenir. Eđer hata atayıcısı tarafından hatanın varlığı onaylanırsa, rapor YENİ olarak işaretlenecektir. Daha sonra bu yeni rapor, ađırlık(severity), öncelik(priority), ürün (product) ya da serbest bırakma (release) olarak sınıflandırılır. Hata atayıcısı bu hatayı kimin düzeltilmesi gerektiğine karar verirken, raporun durumu ASSIGNED (atanmış) olarak deđiştirilir. Bir hatanın geliştirici tarafından Çözölmüş (Resolved), Doğrulanmış (Verified) ya da Kapanmış (Closed) olarak işaretlenmesi, hatanın birtakim çözümlere sahip olduğunu gösterir. En olası çözüm türleri arasında Onarılmış (Fixed), Kopya (Duplicate), Worksforme (Hata tekrarlanamadı), Geçersiz (Invalid) türleri yer almaktadır. [1]

Büyük sayıdaki hata raporlarını toplamak, düzenlemek ve analiz etmek için hata izleme sistemleri kullanılır. En popüler hata izleme sistemlerinden biri 1998'de piyasaya sürölen Bugzilla'dır. Bugzilla'da sadece Firefox hataları deđil, aynı zamanda Eclipse ve diđer açık kaynak projelerinin hatalarını içeren çok sayıda hata raporu bulunmaktadır. Birçok arařtırmacının Bugzilla'yı calışmalarında kullanmasının nedeni, Bugzilla'nın açık kaynak erişiminin olması ve çok sayıda projeyi içermesidir. Bugzilla'da çeşitli hata raporları ve bu hata raporlarında hatalarla ilgili çeşitli tanımlamalar mevcuttur. Bu hataların kime atanacağı konusunda hata atayıcısının nasıl karar vereceđi önemli konulardan biridir. Hata raporlarının bir geliştiriciye atanması için hata hakkında ve geliştiriciler hakkında bilgi sahibi olunması gerekmektedir. Hata raporunun projenin hangi takımıyla ilgili olduđu deđerlendirildikten sonra geliştiricilerin iş yükü, bilgi birikimi, deneyimi vb. birçok faktör göz önüne alınarak hata atamasının yapılması gerekmektedir. Bu işlemin elle yapılması hata raporlarının sayıca az olduđu projelerde mümkündür ancak proje büyüklüğü ve hata raporu sayısı arttıkça bu işlem çok zaman alan, hatalı atama yapmaya eğimli ve yönetilemez hal almaya başlayabilir. Bunun için hata atama işleminin manuel olarak deđil otomatik olarak yapılması gerekliliđi doğmuştur. Bunun için çeşitli arařtırmacılar makine öğrenmesi, metin ma-

denciliği vb. teknikleri kullanarak hata ataması işlemini otomatikleştirmeyi ve hata atama işleminde yapılan hataları en aza indirmeyi amaçlamışlardır.

### 3 Araştırma yöntemi

Bu bölümde araştırma yöntemine genel bir bakış, amaç ve haritalama soruları ele alınmaktadır.

#### 3.1 Amaç ve haritalama soruları

Bu çalışmada kullanılan araştırma yaklaşımı Hedef-Soru-Metrik (Goal-Question-Metric (GQM)) metodolojisidir. Bu çalışmanın amacı, zorlukları gidermek ve araştırmacılar ve uygulayıcıların perspektiflerinden gelecekteki araştırmalar için alternatifleri belirlemektir. Bunun için bu alandaki mevcut yaklaşımları ve eğilimleri bulmak amacıyla hata atama konusundaki literatür sistematik olarak haritalanmış ve gözden geçirilmiştir. Bu amaca dayalı olarak aşağıdaki haritalama soruları (HS) oluşturulmuştur:

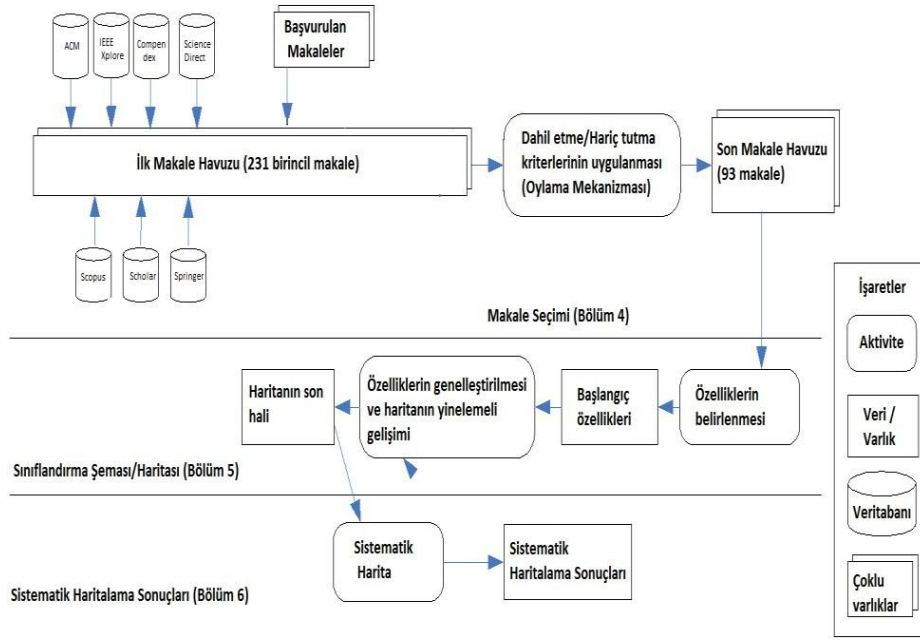
- **HS1:** Çalışmaların araştırmaya katkı tipleri: Hata atama konusundaki makalelerden kaç tanesi metot/yöntem, teknik, araç, metrik vb. sunmaktadır? Peterson'un yazılım mühendisliği sistematik haritalama araştırmalarına göre, katkı tipi yaygın olarak kullanılan bir uygulamadır. Bu sorunun cevabını vermek, hata tayininde mevcut araştırmaların odaklandığı alanın eğilimini anlamamıza yardımcı olacaktır. (Yeni teknik, yeni araç, yeni metrik vb. üzerine odaklanılmıştır.)
- **HS2:** Araştırma yöntemi tipleri: Makalelerde hangi araştırma tipleri kullanılmıştır? Peterson ve ark. [2] ayrıca çalışmaların araştırma yaklaşımlarını sınıflandırmak için kılavuz ilkeler getirmiş ve bu ilkeler bu soruya cevap bulmak için kullanılmıştır. HS 2'yi cevaplamak için birincil çalışmalar 7 farklı araştırma yöntemine göre, her çalışma sadece bir araştırma yöntemi türüne dâhil olacak şekilde sınıflandırılmıştır.
- **HS3:** Makine öğrenme yöntemi tipleri: Hata atama konusundaki makalelerde hangi öğrenme yöntemleri kullanılmıştır? Bu sorunun cevabı, hata tayininde kullanılan mevcut öğrenme ve sınıflandırma algoritmalarının eğilimini anlamaya yardımcı olacaktır.
- **HS4:** Teste tabi tutulan projelerin isim ve sayıları: Test edilen sistemde kullanılan projelerin isim ve sayıları nelerdir? Bu sorunun cevabı, hata tayininde hedeflenen katkıyı gerçekleştirebilmek için kullanılan projelerin isim ve sayıları ile ilgili istatistik sunacaktır.

#### 3.2 Sürece genel bir bakış

Önceki bölümlerde bahsedildiği üzere, bu sistematik haritalandırma çalışması Peterson ve arkadaşları [2, 3] tarafından sağlanan kılavuzlara dayalı olarak yürütülmektedir. Bu SH için metodolojiyi tasarlarken, [4] gibi çeşitli SH'lardan yöntemler de dâhil edilmiştir. Bu SH in temelinde yatan süreç Şekil 1'de özetlenmekte olup, bu süreç üç aşamadan oluşmaktadır:

- Makale seçimi (Bölüm 4)

- Sistematik haritalamanın gelişimi (Bölüm 5)
- Sistematik haritalamanın sonuçları (Bölüm 6)



Şekil 1. Sistematik haritalama çalışmasında kullanılan protokol

#### 4 Makale seçimi

Sistematik haritalama çalışmamızın ilk aşaması makale seçimi aşamasıdır. Bu aşamada aşağıdaki adımlar sırasıyla uygulanmıştır:

- Kaynak seçimi ve anahtar kelimeleri arama (Bölüm 4.1)
- Dâhil etme/Hariç tutma kriterleri (Bölüm 4.2)
- Makale havuzunun tamamlanması (Bölüm 4.3)

##### 4.1 Kaynak seçimi ve arama anahtarları

Peterson'un sistematik haritalama kurallarına [2, 3] dayanarak, ilgili çalışmaları bulmak için aşağıdaki yedi büyük dijital kütüphane araştırılmıştır: (1) IEEE Xplore, (2) ACM Digital Library, (3) Compendex, (4) Science Direct, (5) Scopus, and (6) Springer (7) Google Scholar. Aramalara “bug assignment” arama anahtarı ile başlandı. Araştırmacılardan biri 1., 2. ve 3. veritabanlarında aramayı gerçekleştirirken, diğer araştırmacı 4., 5., 6. Ve 7. veritabanlarında arama gerçekleştirdi. Bu arama sonucunda başlangıç havuzumuzda toplamda 172 makale elde ettik. Bu işlemden sonra iki araştırmacı elde ettikleri makaleleri birleştirildi ve benzer makaleler havuzdan çıkarıldı. Bu işlem sonucunda 116 birincil makale elde edildi. Bu anahtar kelimeyle elde edilen makalelerin araştırılıp ilgili makalelerin özet ve giriş bölümleri incelendikten

sonra yeni bir arama anahtarı olarak “bug triaging” arama anahtarı çıkarıldı. Yinelemeli ve tekrarlı bir işlemde sonra bu yeni arama anahtarı ile 115 farklı birincil çalışma elde edildi. Bu 115 makale sadece Google Scholar veritabanında yapılan arama ile elde edilmiştir. Final havuzumuzda son olarak 231 birincil çalışma elde edildi. Olası makalelerin gözden kaçmasına engel olabilmek amacıyla son olarak bug and {(assign or assignment) or (triage or triaging)} arama anahtarı ile yeni bir arama gerçekleştirildi. Buna ek olarak, ilgili çalışmaların gözden kaçma riskini azaltabilmek amacıyla incelenen makalelerin referans verdiği bazı makaleler manüel olarak aratıldı. Makale havuzunda olmayıp da konuyla ilgili olmaya aday makaleler de havuza dâhil edildi.

#### 4.2 Dâhil etme/hariç tutma kriterleri

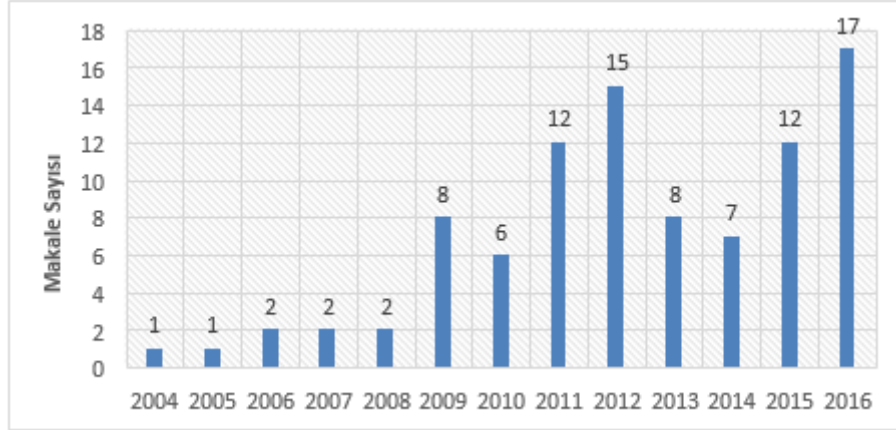
Bu çalışmada göz önünde bulundurulmuş dâhil etme kriterleri şunlardır: (1) Her çalışmanın konusunun hata atama bağlamıyla ilgisi, (2) Çalışmada takip edilen kapsamlılık ve değerlendirme ve geçerlilik düzeyi. Yalnızca İngilizce olarak yazılan ve yalnızca elektronik olarak erişilebilen çalışmalar dâhil edilmiştir. Kapsamla ilgili ancak geçerli kanıtı sahip olmayan makaleler hariç tutuldu. Kapsamla ilişkili ancak içeriğine ücretsiz olarak erişilemeyen makaleler de hariç tutulmuştur.

İlk havuza dâhil etme / dışlama kriterlerini uygulamak için, makalenin yazarları ilk havuzdaki çalışmaları değerlendirerek her çalışmaya “1” ve “0” olarak oy vermiştir; “1”, bir çalışmanın dâhil edilmesi yönünde bir görüş belirttiğini, “0” ise çalışmanın hariç tutulması yönünde bir görüş belirttiğini gösterir. İncelenen makalenin dışlanmasına ilişkin karar için 3 puanlık bir eşik kullanmaya karar verilmiştir, i.e. toplamdaki oylarla 3 puanın altındaki araştırmalar hariç tutulmuştur. Her bir çalışmayı oylamak için makalenin başlığı, özeti ve anahtar kelimeleri gözden geçirilmiştir. Bu kaynaklarla yeterli bilgi bulunamazsa, daha derinlemesine bir değerlendirme gerçekleştirilmiştir. İki araştırmacının yapmış olduğu ortak oylama sonucunda, son havuz 231’den 93’e düşürülmüştür.

#### 4.3 Son makale havuzu ve online depo

Okuyucular 93 birincil çalışmanın tam referans listesi için e-tabloya (<https://goo.gl/AgrL6m>) başvurabilirler. Seçilen çalışmaların son havuzu, Google Dokümanlar sistemini kullanarak çevrimiçi bir depoda yayınlanmıştır. Seçilen her yayının Bölüm 5’de tanımlanan sınıflandırma şemasına göre sınıflandırılması çevrimiçi depoda da mevcuttur.

Hata atamasının yıllık yayın hacmi Şek.2’de gösterilmektedir. Yayın aralığının başlama yılı açısından, hata atama makalelerinin 2004 yılında ortaya çıkmaya başladığı ve 2016 yılına kadar bu kapsamda artan bir çalışma yoğunluğu olduğu görülmektedir. 2016 yılında hata atamayla ilgili 17 makale yayınlanmıştır.



Şekil 2. Yıllara göre makalelerin dağılımı

## 5 Sistematik haritanın geliştirilmesi (sınıflandırma şeması)

Tablo 1, yukarıda açıklanan işlemlerin uygulandıktan sonra geliştirilen nihai sınıflandırma şemasını göstermektedir. Tabloda, sütun 1, HS listesidir, sütun 2, ilgili öznelik / özelliktir. Sütun 3, özellik için olası tüm değerler kümesidir. Son olarak, sütun 4, birden fazla seçimin uygulanıp uygulanamayacağı konusunda bir öznelik belirtir. Örneğin, RQ 1 (Katkı tipi) için, son sütundaki ilgili değer 'Ç' (Çoklu) 'dur. Bir çalışmanın birden fazla seçenek türüne (ör. Yöntem, araç vb.) katkıda bulunabileceğini belirtir.

Tablo 1. Çalışmada geliştirip kullanılan sistematik harita

HS	Öznelik / Özellik	Kategoriler	(Ç)oklu/ (T)ekil
1	Katkı tipi	{Metot (teknik), Tool, Metrik, Model, Süreç, Deneysel çalışma, diğer}	Ç
2	Araştırma türü	{Çözüm önerisi, Doğrulama Araştırması, Değerlendirme Araştırması, Deneyim çalışması, Felsefe Araştırmaları, Görüş Çalışmaları, diğer}	T
3	Makine öğrenme metodu türleri	{Sinir Ağları, Kümeleme (kNN etc.), Bayes (NaiveBayes vb.), Karar Ağaçları, Genetik Algoritmalar, Destek vektör makinaları}	Ç

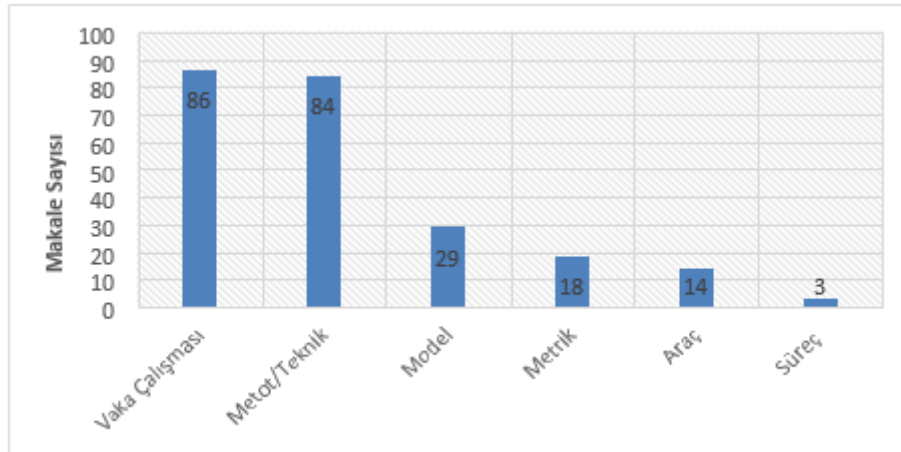
## 6 Sonuçlar

Bu bölümde sistematik haritalama çalışmasında sorulan haritalama soruları kapsamında elde edilen sonuçlardan bahsedilmektedir.

### 6.1 HS1-Çalışmaların araştırmaya katkı tipleri

Birinci araştırma sorusunun amacı, kaç tane makalenin hata atama yöntemi / teknikleri, araçları, modelleri, metrikleri ve süreçleri ile literatüre katkı sunduğunu bulmaktır. Bu katkı türleri belirlenirken Petersen ve arkadaşlarının önerdiği katkı türlerinden yararlanılmıştır. Çalışmaya dâhil olan tüm 93 makale için Şekil 3, araştırmaların katkı türünün dağılımını göstermektedir.

Şek.3 çok sayıda makalenin deneysel (vaka) çalışma önerdiğini göstermektedir. (86 makale). Ayrıca 84 makalenin yeni metot / tekniklerle katkıda bulunduğu veya var olan bir tekniği / metodu geliştirdiği tespit edilmiştir. Yeni bir model sunan 29 makale, metrik ile katkıda bulunan 18 makale, yeni araç ile katkıda bulunan 14 makale ve yeni bir sürece odaklanan 3 makale tespit edilmiştir. Katkılarına göre bir makale birden fazla sınıflandırmaya dâhil olabilmektedir. Örneğin birincil çalışmaların bulunduğu çevrimiçi referans listesinden erişim sağlanabilen makalelerden “A Hybrid Bug Triage Algorithm for Developer Recommendation” makalesinde ‘activity factor’ adında yeni bir metrik ortaya çıkarılırken bu metrik kullanılarak makalede yeni bir modele katkıda bulunulmuştur. “Automated Change Request Triage using Alpha Frequency Matrix” makalesinde yeni bir hata raporu için uygun geliştiriciyi önermek üzere ‘alphabet frequency matrix’ (AFM) adında yeni bir teknik geliştirilmiştir. “Cost-aware triage ranking algorithms for bug reporting systems” makalesinde ‘costriage’ adında yeni bir araç (tool) önerilmiştir. “Automated Bug Triage in an Industrial Context” makalesinde hata izleme verilerden tahmini sonuçlara doğru yeni bir veri analiz süreci (process) geliştirmişlerdir.



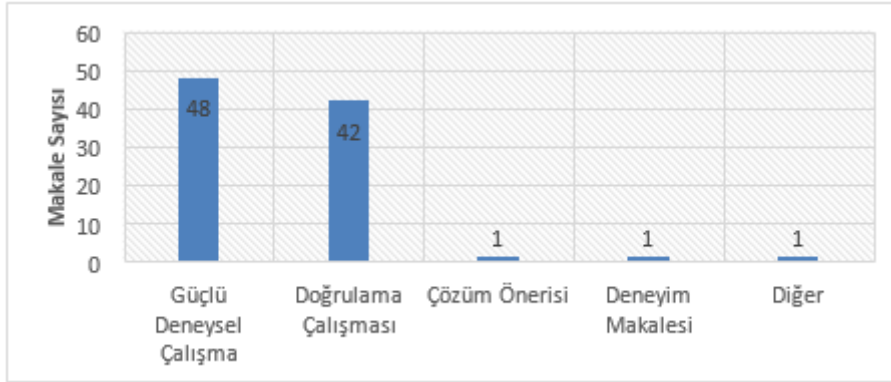
Şekil 3. Katkı Tipleri

### 6.2 HS2-Araştırma yöntemi tipleri

Bu araştırma sorusu ile makalelerde ne tür araştırma yöntemleri kullanıldığının tespit edilmesi amaçlanmıştır. Şek.4, araştırma tipi yönünden makalelerin dağılımını göstermektedir.



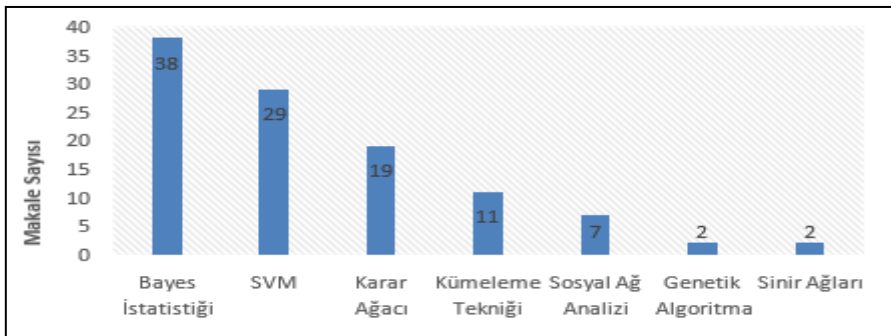
Haritalama çalışmasına dâhil edilen 93 bildirinin bulunduğu havuzda, Şek. 4, en fazla kullanılan araştırma yönteminin güçlü deneysel (vaka) çalışma olduğunu göstermektedir. Sonuçlar, 46 makalenin araştırma soruları veya hipotez kullandığını ve ayrıca bu bildirimlerde geçerliliğe tehdit oluşturan unsurların belirtildiğini göstermektedir. Ancak, hiçbir araştırma sorusu olmamasına rağmen, geniş bir vaka çalışması yapan ve geçerliliğe yönelik tehdit unsurları belirtilmiş makaleleri güçlü deneysel (vaka) çalışması olarak kategorize edilmiştir. Hiçbir hipotez veya araştırma sorusu bulunmayan makaleler zayıf deneysel çalışma (42 makale) olarak sınıflandırılmıştır. Sadece bir makale çözüm önerisi araştırma yöntemine dâhil olurken, bir makale deneysel araştırma yöntemine dâhil edilmiştir. 'Diğer' kategorisinde, herhangi bir araştırma metodu içermeyen makaleler sınıflandırılmış ve bu kategoriye yalnızca bir adet araştırma anketi dâhil edilmiştir.



Şekil 4. Araştırma Yöntemi Tipleri

### 6.3 HS3- Makine öğrenme yöntemi tipleri

Hata atama makalelerinde hangi öğrenme yöntemlerinin kullanıldığını incelemek için, makaleler öğrenme tekniklerine göre sınıflandırılmıştır. Şek 5, incelenen 93 makalenin tamamında kullanılan öğrenme tiplerinin dağılımını göstermektedir.

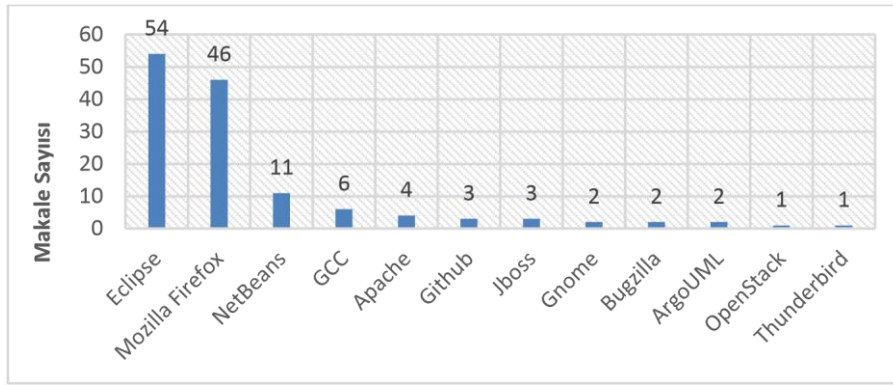


Şekil 5. Öğrenme Yöntemi Tipleri

38 makalede öğrenme yöntemi olarak Bayes istatistikleri kullanılmıştır. En çok kullanılan ikinci öğrenme metodu ise 29 makale ile destek vektör makinesi (SVM) 'dir. Ayrıca, 19 makalenin C4.5, J48, vb. gibi karar ağacı yöntemlerini kullandığı gözlemlenmiştir. Makalelerin çoğunda karşılaştırma amaçlı olarak birden fazla öğrenme yöntemi kullanılmıştır. Örneğin, “A Decision Support Platform for Guiding a Bug Triager for Resolver Recommendation Using Textual and Non-Textual Features” makalesinde iki öğrenme yöntemi kullanılmıştır. Sonuç olarak otomatik hata atama görevi için karar ağacı ve Naive Bayes tekniklerinin en çok kullanılan yöntemler olduğu tespit edilmiştir.

#### 6.4 HS4- Teste tabi tutulan projelerin isim ve sayıları

Şek. 6, test edilen sistemlerde kullanılan her projenin sayısını göstermektedir. Yapılan analizlere göre, deneylerde en çok kullanılan projenin Eclipse olduğu görülmüştür. 54 makale Eclipse'i kullanırken, çalışmalarında Mozilla Firefox kullanan 46 makale vardır. Diğer projeler aşağıda gösterilmiştir: NetBeans, Apache, GCC, GitHub vb.



Şekil 6. Test edilen projelerin dağılımı

## 7 Geçerliliği tehdit eden unsurlar

Çalışmayı yapan araştırmacılar, seçilen veri tabanları, oluşturulan arama anahtar kelimeleri, seçilen zaman kısıtlamaları ve birincil çalışmaların havuzu gibi farklı faktörler, sistematik haritalama araştırmasının sonuçlarını etkileyebilmektedir.

**İçsel Geçerlilik:** Bu çalışma kapsamında olabildiğince eksiksiz bir birincil çalışma havuzu oluşturulmaya çalışılmış, farklı anahtar sözcükler ve bunların birleşimi ile bir araştırma sözcük kümesi belirlenip ilgili çalışmalar toplanmıştır. Dahil etme ve hariç tutma kriterleri belirlenmiştir. Araştırmacıların kişisel değerlendirmelerinin etkilerini minimize etmek amacıyla çalışmayı yürüten araştırmacılar ile oylama mekanizması kullanılmıştır; bağımsız çalışan ve belirli günlerde incelemelerin üzerinden ortak gözden geçirme yapan iki araştırmacı tarafından ilgili makalelerin seçimi gerçekleştirilmiştir. Bazı ilgili makalelerin araştırmacılar tarafından farklı oylanması sonucunda dahil edilmemeleri mümkün olabilmektedir. Bununla birlikte,

incelemelerin sonuçları, her iki arařtırmanın da yüksek düzeyde bir mutabakata sahip olduklarını göstermektedir.

Dıřsal Geerlilik: Bir alıřmanın sonuçlarının ne lüde genelleřtirilebileceęi ile ilgilidir. Sistematik haritalama alıřmasının sonuçları, hata atama alanı kapsamında ve yazılım mühendislięi perspektifine göre deęerlendirilmiřtir. Bu nedenle, sunulan hari tutma yöntemlerinin, verilerin ve izelgelerin sonuçlarının yalnızca verilen baęlamda (hata ataması) geerli olduęu düşünölmelidir. Bu sonuçlar, bu alandaki arařtırmacılar ve uygulayıcılar için bir bařlangı noktası oluşturabilir.

## 8 Sonular

Bu makalede hata atama alanını karakterize etmek için sistematik bir haritalama alıřması yapılmıřtır. Toplam 231 birincil alıřma analiz edilmiř ve dâhil etme ve hari tutma kriterleriyle filtrelendikten sonra son havuzda 93 makale kalmıřtır. Ardından, alıřma kapsamı çerevesinde arařtırma soruları oluşturulmuřtur. Arařtırma sorularına cevap olarak, sorularla iliřkili izelgeler analiz edilmiřtir. Makalelerin çoęunun vaka analizlerinde akademiden katkı saęlayan kiřiler olduęu ve bu makalelerde genelde aık kaynak projelerin kullanıldıęı görölmüřtür. Ayrıca, Eclipse ve Mozilla Firefox'un, test edilen sistemde en ok kullanılan projeler olduęu gözlenmiřtir. Hata atama yaklařımlarını analiz etmek için birok farklı teknik kullanılmıřtır. Sonu olarak makine öęrenme tekniklerinden SVM ve Naive bayes in en ok kullanılan öęrenme metotları olduęu görölmüřtür. Elde edilen bulgulara göre, 2004 yılından 2016 yılına kadar, hata ataması konusunda kayda deęer bir eęilim olduęu görölmüřtür.

Gelecekteki alıřmalar olarak, bu alıřmaya dayanarak, bu alıřmayı farklı aılardan genişleterek hata ataması alanında bir sistematik literatür deęerlendirme (Systematic Literature Review, SLR) alıřması yapmak planlanmaktadır.

## 9 Referanslar

1. Laplante, P., Ahmad, N.: Pavlov's Bugs: Matching Repair Policies with Rewards. IT Professional. 11, 45-51 (2009).
2. Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M.: Systematic Mapping Studies in Software Engineering. EASE . 8, 68-77 (2008).
3. B. Kitchenham, S. Charters.: Guidelines for performing Systematic Literature Reviews in Software Engineering . EBSE. (2007).
4. Amannejad, Y., Garousi, V., et al.: A Search-Based Approach for Cost-Effective Software Test Automation Decision Support and an Industrial Case Study. 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops. 302-311 (2014).