

Endüstriyel Bağlamda Yazılım Olay Kaydı Yönetim Sürecinin İyileştirilmesi

Ethem Utku Aktaş^{1,2}, Haluk Altunel¹, Halil İbrahim Elalmış¹, Songül Nişancı¹, Cemal Yılmaz²

¹ Arge Merkezi, Softtech A.Ş., İstanbul, Türkiye
{utku.aktas, haluk.altunel, halil.elalmis,
songul.nisanci}@softtech.com.tr

² Mühendislik ve Doğa Bilimleri Fakültesi, Sabancı Üniversitesi, İstanbul, Türkiye
cyilmaz@sabanciuniv.edu

Özet. Yazılım sorunlarının takibi amacıyla olay kaydı yönetim sistemleri kullanılmaktadır. Bu sistemler aracılığıyla sorunların iletimi kolaylaşmakta, dolayısıyla daha fazla sorun tespit edilerek çözüm üretilmektedir. Bu durum da yazılım ürünlerinin kalitesinin artmasını ve çabuk geri dönüşlerle müşteri memnuniyetinin artırılmasını sağlamaktadır. Ancak sisteme girilen kayıt sayısı arttıkça kayıtların geçerliliğinin kontrolü, önceliklendirilmesi ve ilgili yazılım ekibine atanması süreçlerinin yönetilmesi zorlaşmaktadır. Softtech A.Ş. Türkiye’de bankacılık sektöründe hizmet vermekte olan bir yazılım şirketi olup günde açılan 300’ün üzerinde olay kaydıyla bu süreci takip etme, yönetme ve sürekli iyileştirme zorunluluğundadır. Bu amaçla geçtiğimiz dönemde olay kayıtlarını çözmek ve diğer operasyonel işleri yazılım geliştiricilerden devralmak üzere “uygulama destek takımı” kurulmuştur. Bu çalışmayla olay kaydı çözüm süreci “değer akışı haritalama” yöntemiyle incelenmiş, ölçerek iyileştirme hedefiyle sistemde oluşan veri analiz edilmiş, gerçekleştirilen iyileştirmeler, önümüzdeki dönemde yapılması planlanan diğer araştırma ve çalışmalar aktarılmıştır.

Anahtar kelimeler: Yazılım Bakımı, Olay Kaydı Yönetimi, Değer Akışı Haritalama.

Improvement of Software Issue Record Management Process in an Industrial Context

Abstract. Issue record management systems are used for tracking software problems. Through these systems, reporting these problems becomes easier and more problems can be detected and solved, which in turn increases the quality of the software products and customer satisfaction with faster solutions. However, as the number of bug reports in the system increases, it becomes more difficult to validate the records, prioritize them and assign them to the relevant software

teams. Softtech Inc. is a software company that serves in the banking sector in Turkey and with more than 300 issue records submitted daily, has to manage this process and improve continuously. For this purpose, "application support team" has been set up to examine and solve issue records and take over other operational tasks from software developers. In this study, the issue record solution process is examined with "value stream mapping" methodology, with the aim of improving by measuring the related data is analyzed, improvements achieved and other research studies planned to be carried out in the forthcoming period are shared.

Keywords: Software Maintenance, Issue Record Management, Value Stream Mapping.

1 Giriş

Yazılım uygulamalarının kullanıcılarının, yaşadıkları sorunları kolayca raporlayabilmesi ilgili uygulamaların kalitesinin artmasını sağlamaktadır. Bu amaçla çeşitli olay kaydı takip sistemleri bulunmaktadır. Bu sistemler olay kaydı yaşam döngüsünü tanımlamakta ve ilgili yaşam döngüsünün uygulanmasında, dolayısıyla olay kayıtlarının çözümünde kullanılmaktadır [1].

İlgili sistemler şüphesiz ki yazılım kalitesinin artırılmasına olanak sağlamaktadır. Ancak kayıt sayısının artması ile birlikte kayıtların gözden geçirilmesi, onaylanması, ilgili ekibe atanması ve çözüm üretilmesi karmaşıklaşabilmektedir. Softtech A.Ş.'nin hizmet verdiği ana sektör bankacılık sektörü olup sisteminde tanımlı yazılımla ilişkili 400'ün üzerinde farklı ürün bulunmaktadır. Olay kaydı yönetimi için Jira uygulaması kullanılmaktadır. Sisteme girilen günlük 300-350 arası olay kaydı ile bu süreci sürekli bir şekilde takip etme, yönetme ve iyileştirme zorunluluğundadır.

Bu amaçla 2017 yılı başında "uygulama destek takımı" kurulmuş ve kayıtların yazılım geliştiricilerden önce incelenmesi ve çözümlenmesi sorumluluğunu üzerine almıştır. Bu çalışmayla ilgili takımın sürece dahil olmasıyla birlikte oluşan olay kaydı yaşam döngüsü ve sistemde oluşan veriler analiz edilmiş, çözüm önerileri, sonuçlar ve olası araştırma konuları aktarılmıştır.

Çalışmanın aşağıdaki konularda katkısı bulunmaktadır:

1. Yazılım olay kayıtları ile ilgili endüstriyel çalışmalara nadiren rastlanmakta olup büyük endüstriyel bir yazılım sisteminde olay kaydı çözüm süreci akışı ve sistemde oluşan kayıtlarla ilgili istatistiksel verilere yer verilmiştir.
2. Süreçle ilgili yapılan tespitler çerçevesinde yazılımsal iyileştirme önerileri, bu kapsamda devam eden çalışmalar ve olası araştırma konuları ortaya konmuştur.

Bildirinin organizasyonu, ilgili çalışmalar ikinci bölümde, olay kaydı yaşam döngüsü üçüncü bölümde, veri analizi dördüncü bölümde, öneriler, devam eden çalışmalar ve araştırma konuları beşinci bölümde, sonuç ve gelecek çalışmalar altıncı bölümde aktarılacak şekilde yapılmıştır.

2 İlgili çalışmalar

“Yalın düşünme” (Lean thinking), Toyota üretim sistemini analizinden sonra Womack tarafından ortaya atılmıştır ve bir değer akışında (value stream) atıkların (waste) tanımlanması ve kaldırılması sürecini tanımlar [2]. Değer akışı haritalama yöntemi ile üç süreç ayırılır: Değer yaratan aktiviteler, müşteri için değer yaratmayan ama ürünün ortaya konması için şart olan aktiviteler ve müşteri için değer yaratmayan, gereksiz ve süreçten çıkarılması gereken aktiviteler [3]. Mevcut durum haritası ortaya konduktan sonra atıklar tanımlanarak gelecek durum haritası ortaya konulur. Yalın düşünme ve Toyota üretim sisteminin yazılım geliştirme sürecine uyarlanması Poppendieck’lerin çalışmaları ile olmuş, “Yalın yazılım geliştirme” tanımlanmış ve yazılım geliştirme sürecindeki atıklar tanımlanmıştır [4].

Değer akışı haritalama yönteminin yazılım geliştirme sürecinde uygulanması ile ilgili çeşitli çalışmalar mevcuttur [5], [6], [7]. Bu çalışmalar, organizasyonu şelale modelinden iteratif yazılım geliştirmeye dönüştürerek veya iteratif yazılım geliştirmede üretim miktarını azaltarak atık miktarını azaltmıştır. Sedano ve arkadaşları yazılım geliştirme sürecinde dokuz atık türü tanımlamışlardır: Yanlış özellik veya ürün ortaya konması, biriken işlerin (backlog) yanlış yönetilmesi, yeniden işleme (rework), gereksiz karmaşık çözümler, konu ile ilgisi olmayan bilişsel yük, psikolojik sıkıntı, bekleme/çoklu görev, bilgi kaybı ve etkisiz iletişim [3].

Olay kayıtları yazılım projelerinin kalitesini artırmakla birlikte iletilen alakasız kayıtlar nedeniyle konuyla ilgilenen yazılımcıların verimini de düşürebilmektedir [1]. Softtech’te kurulmuş olan uygulama destek ekibi yazılımcıların üzerindeki bu yükü kaldırmayı hedeflemiştir. Bu çalışmayla olay kaydı sürecinin mevcut durumu değer akışı haritalama yöntemiyle incelenmiş ve bu haritadan yola çıkılarak iyileştirme önerileri sunulmuştur.

Olay kaydında iletilen bilginin kalitesi, bildirilen sorunu zamanında çözmek için önemlidir. Kaliteli bir olay kaydının içermesi gereken bilgileri bulmak için, yazılım geliştiriciler ve olay kayıtlarını ileten kişilerin dahil olduğu bir anket yapılmış ve sonuç olarak yazılım geliştiricilere ihtiyaç duyduğu bilgileri toplayarak olay kaydı iletilmesini sağlayan bir araç ortaya konmuştur [8].

İletilmiş olan kayıtlar, yazılım hata kaydı değil de, iyileştirme talepleri veya bilgi talebi olabilir. Makine öğrenmesi algoritmaları ile bu tip kayıtları tespit etmek sistemdeki kayıtların sayısının azaltılmasını ve iş yükünün hafiflemesini sağlayacaktır [9].

Bazı olay kayıtları daha önceden iletilmiş olabilir, bu durumda ilgili kayıt çift (duplicate) olarak kabul edilir, bu da iletilen yeni kaydın “çift” olarak işaretlenmesine ve incelenmemesine neden olmaktadır. Ancak, bu tür kayıtlar, hatanın tespiti ve düzeltilmesi için bazı ek bilgiler içerebilmektedir. Dolayısıyla hata ile ilgili daha fazla bilgi toplanması için çift kayıtların tespiti önem arz etmektedir [10].

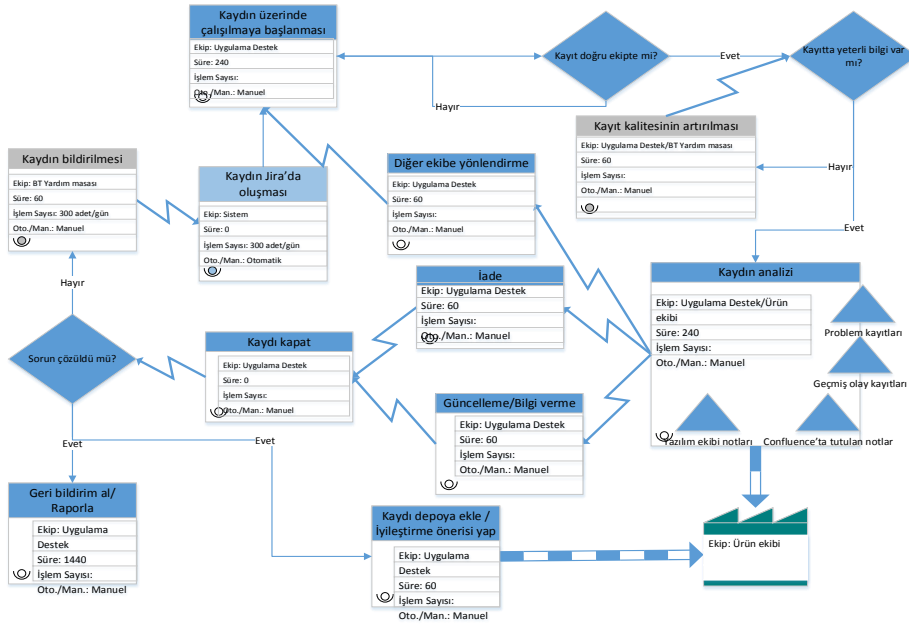
Olay kaydının önceliği iş perspektifinden aciliyeti tanımlamaktayken, şiddet hatanın yazılım sistemindeki etkisidir. İdeal olarak, hatayı bildiren farklı kişilerin aynı şiddet değerini belirlemesi beklenir. Bu değer kaydın hangi aciliyette çözümlenmesi gerektiğine karar verirken çok önemlidir [11].

Olay kayıtlarının doğru yazılımcıya veya yazılım ekibine atanması zaman alıcı bir iştir [12]. Makine öğrenmesi algoritmaları kullanılarak olay kayıtlarının atanması için çeşitli çalışmalar yapılmıştır [13], [14]. Softtech'te de kayıtların yazılım ekiplerine otomatik olarak atanması için çalışmalar yapılmış [15], bu konuda gelinen aşamadan beşinci bölümde bahsedilmiştir.

3 Olay Kaydı Yaşam Döngüsünün Analizi

Uygulama destek ekibi olay kaydı çözüm sürecinde devreye girmeden önce kayıtlar yazılım geliştiriciler tarafından incelenmekte ve çözüm üretilmekteydi. Yazılım geliştiricilerse o günkü müsaitlik durumlarına göre kayıtlarla ilgilenmekteydiler. İlgililer, kayıtları geçici çözümlerle kapatabilmekte veya yazılımsal değişikliklerle kalıcı çözümler üretebilmekteydiler. Olay kaydı olmadığı düşünülen kayıtlar içinse iade seçeneği mevcuttu. Uygulama destek ekibinin devreye girmesi bu süreçte temel bir değişikliğe neden olmamıştır. İlgili çalışanların sorumlu oldukları ekibe ait kayıtları görebilmesi ve güncelleyebilmesi sağlanmıştır. Ancak yazılım geliştirilerek kalıcı bir şekilde çözülmesi istenen kayıtlar için, yazılım ekibine olay kaydından ayrı problem kaydı da açılmaya başlanmıştır.

Sürecin analizi için uygulama destek ekibi çalışanları ile birebir görüşmeler yapılmış ve gün boyunca olay kaydı çözümü için yaptıkları işlemler gözlenmiştir. Mevcut duruma ait değer akış haritası Şekil 1'de ortaya konmuş, ayrıca uygulama destek ekibi yöneticisinden çizilen sürecin doğruluğu konusunda teyit alınmıştır.



Şekil 1. Olay kaydı yaşam döngüsünün mevcut durum haritası.

Kayıtlar öncelikle şubeler veya iş birimleri aracılığı ile BT yardım masası çalışanlarına aktarılmakta ve BT yardım masası çalışanları, gelen çağruların %70'ini Softtech'e ilemeden karşılamaktadır. Yardım masası tarafından çözüm üretilmeyen kayıtlar Jira üzerinden sisteme girilmekte ve Softtech'te uygulama destek ekibi kayıt üzerinde çalışmaya başlamaktadır. Bu süreçte ilk yapılan işlem kayıta çözüm üretmeye yetecek yeterli bilginin olup olmadığının ve doğru ekipte olduğunun teyit edilmesidir. Kayıta yeterli bilgi olmaması veya hatalı ekibe aktarılmış olması, yeterli bilgi sağlanana kadar veya doğru ekip bulunana kadar bir döngü içinde farklı kişiler tarafından tekrar edilebilen bir akışa neden olmaktadır. Geçmiş bir çok çalışma ile bu sürecin makine öğrenmesi yöntemleri ile otomasyonunun sağlanabileceği önerilmiştir [12] , [13] , [14].

Kayıta yeterli bilginin olduğunun ve doğru ekipte olduğunun teyit edilmesi ile birlikte kayıt üzerinde çalışmaya başlanmaktadır. Burada kaydın çözümüne adanmış bir ekibin olmaması, ilgili ekibin olay kaydı çözümünün dışında başka işlerinin olması süreci uzatmaktadır. Ana sorumluluğu olay kaydı çözümü olan uygulama destek ekibinin devreye girmesi ile kayıtların çözüm süreçlerinin hızlanmış olduğu öngörülmüştür. Uygulama destek ekibi ilgili kaydı incelerken geçmiş benzer olay kayıtlarından, konuyla ilgili olabilecek problem kayıtlarından (sık tekrar eden sorunlar için açılmış kayıtlar) veya çeşitli depolarda tutulan bir takım notlardan faydalanmaktadır. Yeni gelen kayda benzeyen olay, problem veya notlara hızlı bir şekilde ulaşılabilmesi analiz sürecini hızlandırmaktadır.

Sonrasında ilgili olay kaydı aşağıdaki akışlardan birisine girebilmektedir:

- Kayıt bilgi eksikliğinden oluşturulmuşsa ilgili bilgiler verilerek, olay kaydı olarak iletilmesi uygun değilse veya Softtech dışındaki bir ekibin çözmesi gerekiyorsa bu durum belirtilerek veya yetersiz bilgi olmasından iade,
- Farklı bir ekibe gitmesi gerekiyorsa ilgili ekibe aktarım,
- Çözüm üretilmesi.

Kapatılan kayıt, sorunu yaşayan kullanıcı tarafından teyit edilmekte, sorunu çözülmüşse geri bildirim alınmakta, çözülmemişse tekrar açılmakta ve süreç tekrar başlatılmaktadır.

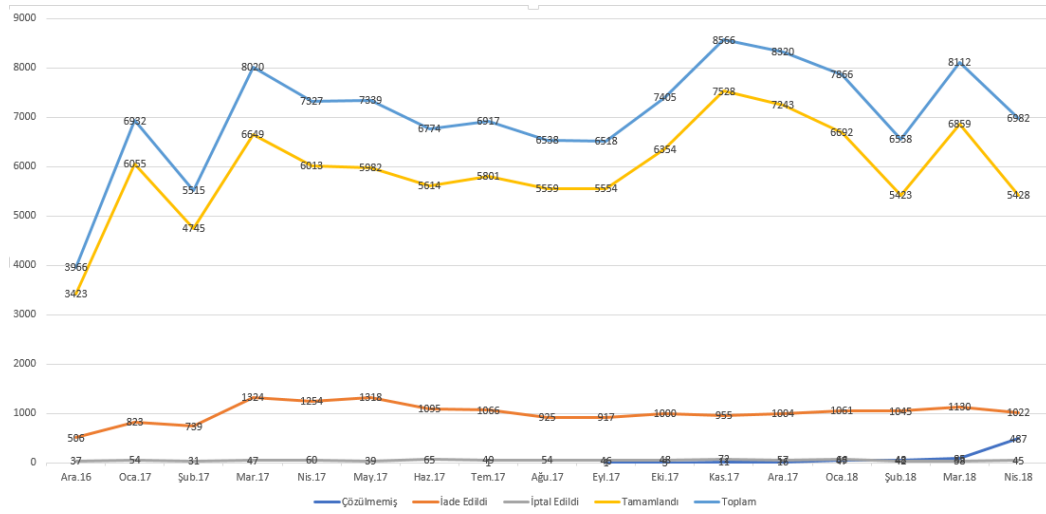
İyileştirme önerilerinin yapılabilceği noktalar olarak aşağıdaki maddeler belirlenmiş ve nedenleriyle birlikte verilmiştir:

- Kaydın doğru ekibe aktarılmaması farklı kişilerin aynı analizleri yapmasına neden olmaktadır. Kaydın tek seferde doğru ekibe aktarımı süreci hızlandırabilecektir.
- Benzer kayıtları sürekli olarak çözen kişiler belli bir uzmanlık kazanmaktadır. Ancak konuyla yeni ilgilenmeye başlayan bir kişinin kaydı çözebilmesi için geçmişte kapatılmış benzer olay kayıtlarına ve konuyla ilgili açılmış problem kayıtlarına hızlı bir şekilde ulaşabilmesi gerekmektedir.

4 Veri Analizi

4.1 Olay Kaydı Adetleri

Sistemde açılan olay kayıtlarının Jira sistemine geçilen Aralık 2016 tarihinden itibaren adetleri incelenmiş ve şekil 2’de verilmiştir.



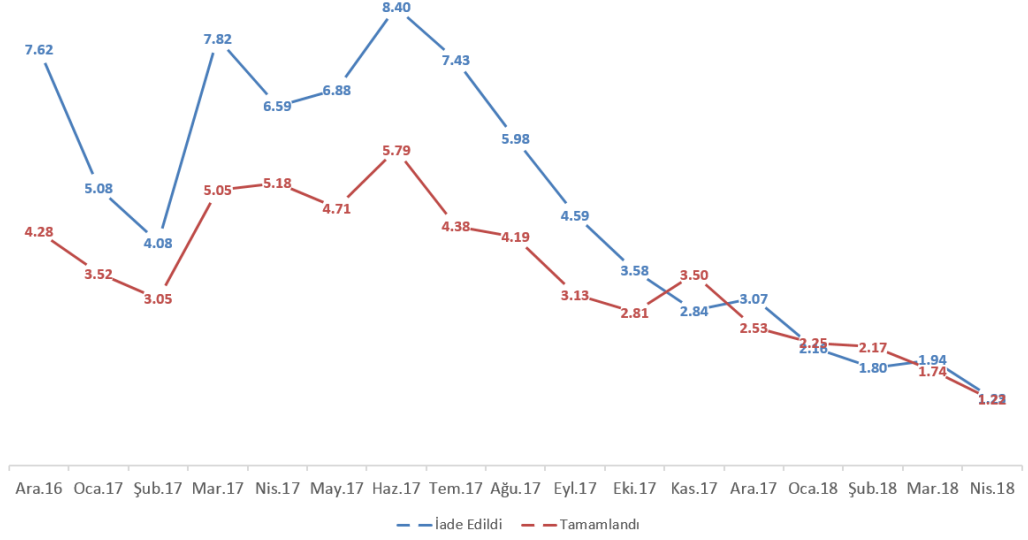
Şekil 2. Olay kaydı adet değişimi.

İlgili şekilde o ayda açılan olay kayıtları statülerine göre ve toplam adedine göre listelenmiştir. Statüsü “Tamamlandı”, “İade Edildi”, “İptal Edildi” ve “Çözülmemiş” olan kayıtlar ayrı ayrı verilmiştir. Örnek vermek gerekirse Nisan 2018’de açılan 6982 adet kayıttan 5428 adedinin durumu “Tamamlandı”, 1022 adedinin “İade Edildi”, 487 adedinin “Çözülmemiş”, 45 adedininse “İptal Edildi” durumunda olduğu görülmektedir. Son aylarda “Çözülmemiş” durumunda olan kayıtların adedinin daha çok olması ilgili raporun Mayıs ayı içinde çekilmiş olmasından kaynaklanmaktadır. Dolayısıyla yakın tarihli olan ayların grafiğinde değişim olması mümkündür.

İlgili şekil incelendiğinde, iade edilen kayıtların 1000 adet civarında seyreden gidişatı dikkat çekmiştir.

4.2 Olay Kaydı Ortalama Çözüm Süreleri

Olay kayıtlarının aylık ortalama çözüm süreleri “Tamamlandı” ve “İade Edildi” durumundaki kayıtlar için ayrı ayrı hesaplanmış ve şekil 3’te verilmiştir.



Şekil 3. Aylık ortalama olay kaydı çözüm süreleri.

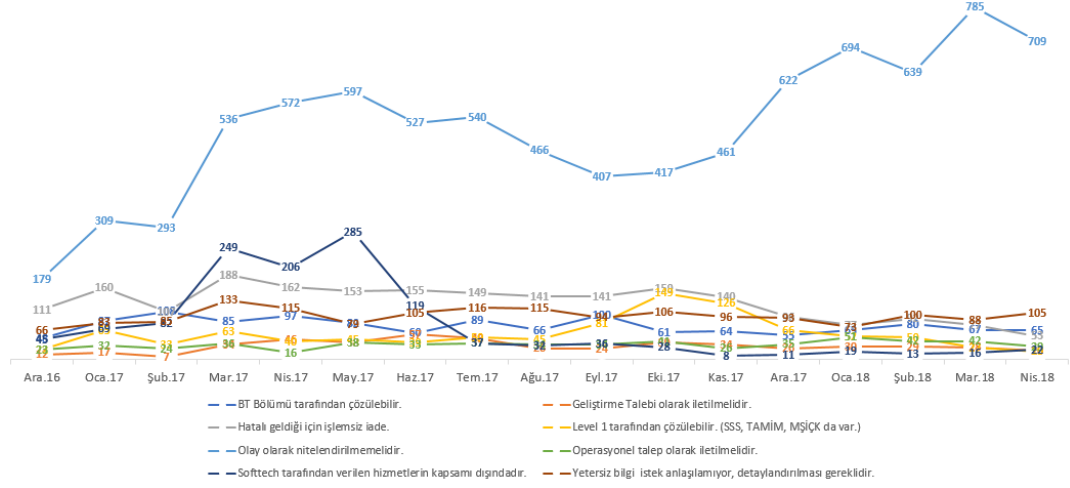
İlgili şekilde çözüm süreleri gün olarak ve aylık bazda verilmiştir. Örnek vermek gerekirse Haziran 2017’de açılmış olup “İade Edildi” durumunda olan kayıtların ortalama çözüm süresi 8.40 günken, “Tamamlandı” durumunda olan kayıtların ortalama çözüm süresi 5.79 gün olarak hesaplanmıştır.

Çözüm sürelerinde Haziran 2017’den itibaren çarpıcı bir düşüş olduğu gözlemlenmektedir. Bu düşüşün nedeni olay kaydı çözümüne adanmış uygulama destek ekibinin kurulmuş olması ve ilgili ekibin yetkinliğinin zaman içinde artmış olması ile ilişkilendirilmiştir. Bununla birlikte, şekil 2’de ise Haziran 2017’den sonra olay kaydı adetlerinde bir artış olduğu da gözlenmektedir. Bu durumun, artık bu kanaldan kullanıcıların daha hızlı yanıt almaları ile ilgili olduğu tahmin edilmektedir.

İlgili şekilde Nisan 2017 itibarıyla iade edilen ve tamamlanan kayıtların ortalama çözüm sürelerinin 1 günün biraz üzerinde olduğu gözlemlenmektedir. Ancak bir olay kaydının iade edilmesine karar verilmesi durumunda kaydın hızlı bir şekilde iade edilebileceği öngörülmektedir. Dolayısıyla iade edilen kayıtların ortalama çözüm sürelerinin neden 1 günün üzerinde seyrettiği araştırmaya değer bir konu olarak değerlendirilmiştir.

4.3 Olay Kayıtlarının İade Nedenleri

İade edilen kayıtlar için iade nedeninin seçilmesi zorunlu olup ilgili adetler alınmış ve şekil 4’te verilmiştir.



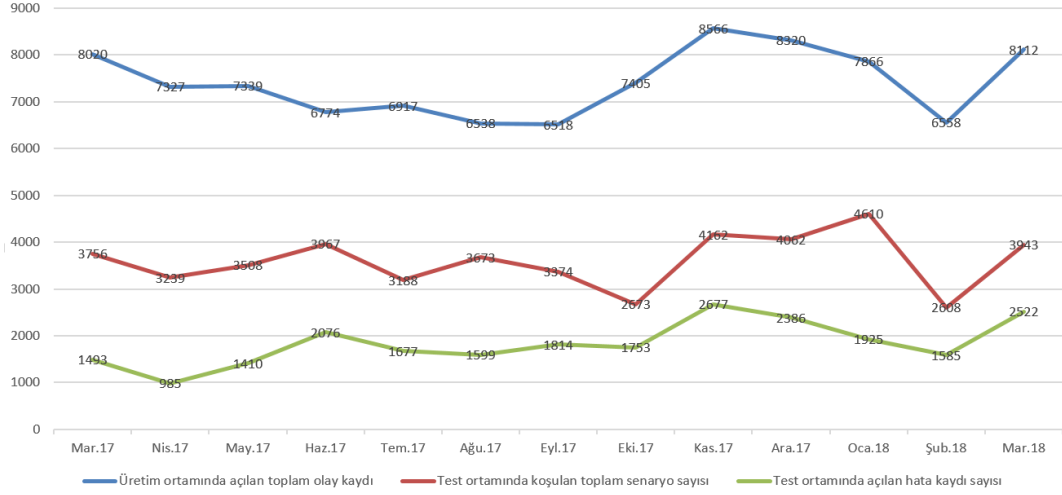
Şekil 4. Olay kaydı iade nedenleri.

Olay kayıtları iade edilirken seçimlik alandaki ilgili seçenekler, kaydın sık sorulan sorular, tamim veya ilgili dokümanlarda yanıtı bulunan durumlarla ilişkili olabileceği, Softtech tarafından verilen bir hizmet olmayıp farklı bir BT ekibine iletilmesi gerektiği, yetersiz bilgi içerdiği, geliştirme talebi veya operasyonel talep olarak iletilmesi gerektiği veya olay kaydı olarak nitelendirilmemesi gerektiğini ifade eden durumları içermektedir.

İlgili seçeneklerle iade edilen kayıtların listesi alınmış, özellikle “Olay olarak nitelendirilmemelidir” seçeneği ile iade edilen kayıtların aylık 700 adedin üzerine çıkarak artış eğiliminde olması dikkat çekici bulunmuştur. Diğer seçeneklerle iade edilen kayıtların Nisan 2018 itibarıyla 100 adet ve altında seyrettiği görülmüştür. Sonuç olarak, iade edilen kayıtlardan “Olay kaydı olarak nitelendirilmemelidir” seçeneği ile iade edilenlerin adetlerinin oldukça yüksek olması araştırmaya değer bir konu olarak değerlendirilmiştir.

4.4 Üretim Ortamı ve Test Ortamında Açılan Kayıtların İlişkisi

Üretim ortamında ilgili ayda açılan kayıtlar ile test ortamında aynı ayda koşulan senaryo sayısı ve açılan kayıtların adetleri alınmış ve şekil 5’te verilmiştir. Çalışmada aktarılan tüm grafiklerde zamanın başlangıç ve bitiş tarihleri Aralık 2016 – Nisan 2018 olmasına rağmen şekil 5’te Mart 2017 – Mart 2018 olması, test sisteminden bu tarih aralığı için veri çekilebilmiş olmasından kaynaklanmaktadır.



Şekil 5. Üretim ortamı olay kaydı ve test ortamı senaryo ve hata kaydı adetleri.

Mart 2017 ile Mart 2018 arasında üretim ortamında o ay içinde açılan olay kaydı adetleri ile test ortamında aynı ayda test ortamında koşulan senaryo sayısı ve açılan hata kaydı sayısı adetleri listelenmiş ve aynı aylarda benzer iniş ve çıkışların olduğu gözlemlenmiştir. Örnek vermek gerekirse, Ocak 2018’de 7866 olan üretim ortamı olay kaydı sayısı Şubat 2018’de 6558’e düşmüş, Mart 2018’de ise 8112’ye çıkmış. Test senaryo adetleri ise Ocak 2018’de 4610’ken Şubat 2018’de 2608’e düşmüş, Mart 2018’de ise 3943’e çıkmış. Test ortamında açılan hata kaydı sayısı Ocak 2018’de 1925’ken Şubat 2018’de 1585’e düşmüş, Mart 2018’de 2522’ye çıkmıştır.

Sonuç olarak, ilgili aylarda üretim ortamında açılan olay kaydı adedi ve test ortamında koşulan senaryo sayısı ve açılan hata kaydı arasında bir ilişki olabileceği görülmüş ve bu ilişki araştırmaya değer bir konu olarak değerlendirilmiştir. Bu ilişkinin olası nedenleri yazılım geliştirme sürecinde yapılması gereken iyileştirmeleri işaret ediyor olabilir: Test sürecinin etkinliği ve yazılımların kalitesi olası iki neden olarak öngörülmüştür.

5 Öneriler, Devam Eden Çalışmalar ve Araştırma Konuları

5.1 İyileştirme Önerileri

Olay kaydı yaşam döngüsünün mevcut durum haritasının çıkarılmasıyla aşağıdaki tespitler yapılmıştır. Bu tespitler değer akış haritasında müşteri için değer yaratmayan ancak yapılması gereken aktiviteler olarak ortaya konmuştur:

- Kaydın sahibi olan yazılım ekibinin tespiti deneyim gerektirmekte ve hatalı ekibe iletilmesi kaydın farklı ekipler arasında dolaşmasına neden olabilmektedir.

- Olay kaydı geldiğinde kaydı inceleyen kişide bir bilgi birikimi oluşmakta, kayıt üzerinde, kendi bilgisayarında veya ortak ulaşımı mümkün olan ortamlarda kaydın nasıl çözüldüğü ile ilgili notlar alabilmektedir.
- İlgili olay kaydına benzer kayıtlar sık sık gelmekteyse kalıcı çözüm üretilmesi için problem kaydı açılmaktadır.

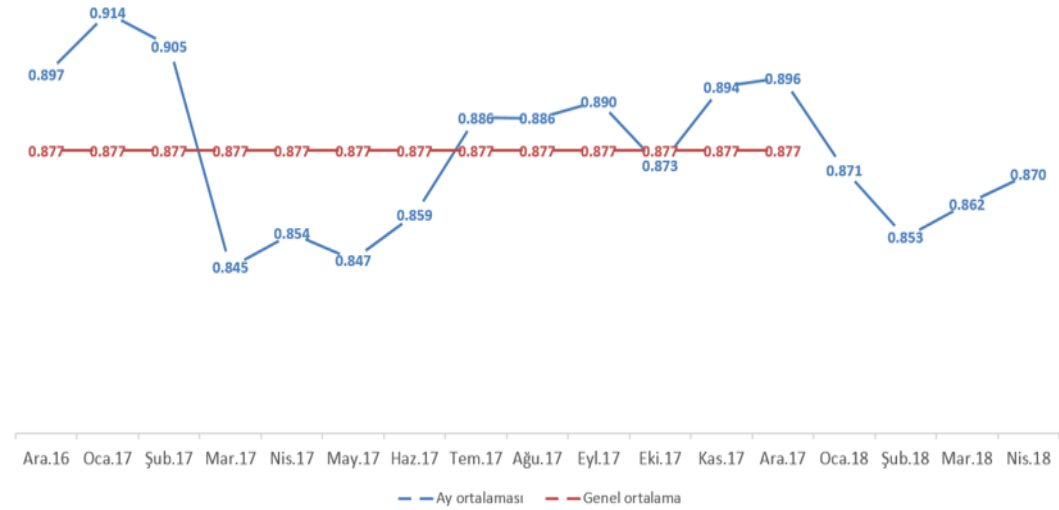
Bu tespitlerden yola çıkılarak aşağıdaki iyileştirme önerileri ortaya konmuştur:

- Kaydın sahibi olan ekibin kayıttaki metin bilgilerinden yola çıkılarak makine öğrenmesi yöntemleriyle otomatik olarak tespit edilmesi ve atanmanın yapılması,
- Kaydın çözümü ile ilgilenen kişilere benzer geçmiş olay kayıtlarının ve problem kayıtlarının Jira üzerinde listelenmesi.

5.2 Devam Eden Çalışmalar

Kaydın sahibi ekibin belirlenmesi konusu bu çalışma öncesinde de bir problem olarak tespit edilmiş ve bu konuda makine öğrenmesi yöntemleri ile ilgili yazılım ekibine otomatik olarak atanması çalışmalarına başlanmıştır [15]. Geçmiş olay kayıtlarının özet ve açıklama alanlarında geçen metin bilgisi önce metin madenciliği teknikleri ile işlenmekte ve sayısal değerlere çevrilmektedir. Sonrasında ilgili sayısal değerler üzerinde makine öğrenmesi algoritmaları çalıştırılmış ve en başarılı sonuca Destek Vektör Makinesi (Support Vector Machine) algoritması ile ulaşılmıştır. Yeni gelen bir kaydın metin bilgisi de benzer şekilde sayısal değerlere çevrilmekte ve Destek Vektör Makinesi ile oluşturulan model kullanılarak uygulama kodu tahmin edilmektedir.

İlgili sistem Ocak 2018’de devreye alınmıştır. Aylık olarak uygulama kodu değiştirilmeden, dolayısıyla farklı bir ekibe atanmadan kapatılan kayıtların yüzdesi alınmış ve şekil 6’da aktarılan sonuçlara ulaşılmıştır.



Şekil 6. Farklı bir ekibe atanmadan çözülen kayıtların yüzdesi.

Aralık 2017 sonuna kadar ilgili arayüzden kullanıcılar tarafından girilen olay kayıtlarına ait uygulama kodlarının ortalama %87.7 oranında ekibinin değişmeden “Tamamlandı” durumuna getirildiği tespit edilmiştir. Uygulamanın otomatik olarak atadığı, Nisan 2018’de açılan ve “Tamamlandı” durumundaki kayıtların %87’sinin uygulama kodunun, dolayısıyla ilgili yazılım ekibinin değiştirilmeden kapatıldığı tespit edilmiştir.

Aylık olarak Jira’dan çekilen kayıtlarla tekrar öğrenme algoritması çalıştırılmakta (Destek Vektör Makinesi) ve ilgili model tekrar oluşturulmaktadır. Doğru ekibin tespiti bu çalışmada ortaya konduğu gibi sürecin iyileşmesini sağlayacak bir adım olarak görünmekte ve sistemin doğru ekibi tespiti için iyileştirme çalışmalarına devam edilmektedir.

5.3 Araştırma Konuları

Veri analizi çalışmaları sonucunda aşağıdaki araştırma soruları ortaya konulmuş olup ilgili çalışmaların önümüzdeki dönemde başlatılması planlanmaktadır:

- İade edilen kayıtların adedinin aylık olarak 1000 civarında seyrettiği, bunlardan önemli bir kısmının “Olay kaydı olarak nitelendirilmemelidir” seçeneği ile kapatıldığı ve ortalama çözüm sürelerinin “Tamamlandı” durumundaki kayıtlara benzer şekilde 1 günün üzerinde seyrettiği göz önüne alındığında, bu kayıtlar için kullanıcılara otomatik olarak yanıt dönülebilir mi veya öneri sunulabilir mi?
- Test ortamında koşulan senaryo sayıları ve açılan hata kaydı adetleri ile üretim ortamında açılan kayıtlar arasında ne tür bir ilişki bulunmaktadır ve varsa bu ilişkiden nasıl bir anlam çıkarabiliriz?

6 Sonuç ve Gelecek Çalışmalar

Bu çalışmayla olay kaydı yaşam döngüsü değer akışı haritalama yöntemi ile incelenmiş ve kayıtların otomatik olarak ilgili ekibe atanması ile Jira üzerinde geçmiş olay kayıtlarının ve problem kayıtlarının kullanıcılara listelenmesi süreçte yapılabilecek iyileştirme önerileri olarak sunulmuştur. Ayrıca üretim sisteminde oluşan olay kaydı verileri incelenmiş, “Olay kaydı olarak nitelendirilmemelidir” seçeneği ile iade edilen kayıtların çokluğu ve bu kayıtların ortalama çözüm sürelerinin “Tamamlandı” durumundaki kayıtlara benzer olması dikkat çekici bulunmuştur. Üretim sisteminde açılan olay kaydı adetleri ile test sisteminde koşulan senaryo sayısı ve açılan hata kaydı sayısı karşılaştırılmış, aralarında bir ilişki bulunabileceği sonucuna varılmıştır. Her iki madde de olası araştırma konusu olarak ortaya konmuştur.

Sunulan iyileştirme önerileri arasında olan olay kayıtlarının yazılım ekiplerine makine öğrenmesi yöntemleri ile otomatik olarak atanması için hazırlanan uygulama Ocak 2018’de üretim ortamında devreye alınmış olup sistem düzenli olarak izlenmektedir. İlgili uygulamanın ürüne dönüştürülmesi için önümüzdeki dönemde kalan çalışmaların tamamlanması planlanmıştır.

Kullanıcıların herhangi bir olay kaydını incelerken Jira üzerinde geçmiş dönemde açılmış benzer olay ve problem kayıtlarını görebilmelerinin sürecin hızlanmasında faydalı olacağı öngörülmüş, yapılacak çalışmalar arasında listelenmiştir.

Kaynaklar

1. Anvik, J., Hiew, L., Murphy, G. C.: Coping with an open bug repository. Proceedings of the 2005 OOPSLA workshop on Eclipse technology eXchange (pp. 35-39). ACM, (2005).
2. Womack, J. P., Jones, D. T.: Lean Thinking: banish waste and create wealth in your corporation. Simon and Schuster, (1996).
3. Sedano, T., Ralph, P., Péraire, C.: Software development waste. Proceedings of the 39th International Conference on Software Engineering (pp. 130-140). IEEE, (2017).
4. Poppendieck, M., Poppendieck, T.: Lean software development: an agile toolkit. Addison-Wesley Professional, (2003).
5. Ali, N. B., Petersen, K., Schneider, K.: Flow-assisted value stream mapping in the early phases of large-scale software development. Journal of systems and software, (2016).
6. Khurum, M., Petersen, K., Gorschek, T.: Extending value stream mapping through waste definition beyond customer perspective. Journal of software: evolution and process, vol. 26, no. 12, (2014).
7. Mujtaba, S., Feldt, R., Petersen, K.: Waste and lead time reduction in a software product customization process with value stream maps. Proceedings of the 21st Australian Software Engineering Conference, ASWEC (2010).
8. Bettenburg, N., Just, S., Schröter, A., Weiss, C., Premraj, R., Zimmermann, T.: What makes a good bug report?. Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering (pp. 308-318). ACM, (2008).
9. Antoniol, G., Ayari, K., Di Penta, M., Khomh, F., Guéhéneuc, Y. G.: Is it a bug or an enhancement?: a text-based approach to classify change requests. In Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds (p. 23). ACM, (2008).
10. Bettenburg, N., Premraj, R., Zimmermann, T., Kim, S.: Duplicate bug reports considered harmful... really?. International conference on Software maintenance (pp. 337-345). IEEE, (2008).
11. Lamkanfi, A., Demeyer, S., Giger, E., Goethals, B.: Predicting the severity of a reported bug. Working Conference on Mining Software Repositories (pp. 1-10). IEEE, (2010).
12. Baysal, O., Godfrey, M. W., Cohen, R.: A bug you like: A framework for automated assignment of bugs. International Conference on Program Comprehension (pp. 297-298). IEEE, (2009).
13. Murphy, G., Cubranic, D.: Automatic bug triage using text categorization. International Conference on Software Engineering & Knowledge Engineering, (2004).
14. Anvik, J., Hiew, L., Murphy, G. C.: Who should fix this bug?. International Conference on Software engineering (pp. 361-370). ACM, (2006).
15. Aktaş, E. U., Göktepe, A., Pehlivan, G., Yıldırım, Ü. Ü., Yılmaz, C.: Olay Kayıtları Ürün ve Platform Kodu Tespit Süreci Otomasyonu. CEUR Workshop Proceedings, UYMS, (2017).