

Bimar Yazılım Kalite Portalı: Deneyimler ve Kazanımlar

Bimar Software Quality Portal: Experience and Lessons Learned

Serkan Apul
Yazılım Geliştirme
Bimar Bilgi İşlem Hiz. A.Ş.
Depark, İzmir, Türkiye
serkan.apul@arkas.com

Özet—Bir yazılımda çıkan hata adedi, yazılımın işlem performansı, kesintisizliği, verimli çalışması ve yazılımı kullanan müşterinin memnuniyeti o yazılımın kalitesini belirleyen öğelerdir. Yazılım ekipleri geliştirdikleri yazılımların kalitesini artırmak için test otomasyonlarını işletmek, performans izlemek ve iyileştirmek gibi yazılım kalitesine hizmet eden birçok faaliyette bulunurlar. Gerçekleştirilen bu faaliyetlerin sayısı arttıkça, bunların kendine özgü uyarı mesajlarını ve diğer çıktılarını takip etmek kendi başına bir iş olmaya başlar. Bunun yanı sıra yolları pek kesişmeyen yazılım ekipleri, diğer bir ekibin hayata geçirdiği yeni bir kalite faaliyetinden habersiz olabilmektedir. Bu şirket içi potansiyelin değerlendirilmemesi anlamına gelmektedir. Gözlemlediğimiz bu iki sorunu çözmek amacıyla Bimar Kalite Portalı (KAP) yazılımı ortaya çıkmıştır. Bu deneyim bildirisinde KAP yazılımını ortaya çıkaran ihtiyaçlar, KAP uygulamasının bu ihtiyaçları nasıl karşıladığı ve elde edilen kazanımlar ortaya konacaktır.

Anahtar Kelimeler—yazılım kalitesi, yazılım geliştirme yönetimi, süreç iyileştirme

Abstract—The number of failures, operation performance, reliability, operation efficiency, and customer satisfaction are the elements that determine the quality of that software. Software development teams engage in many activities that serve software quality, such as operating test automations, monitoring and improving performance, to increase the quality of the software (applications) they develop. As the number of these activities carried out grows, keeping track of their distinctive warning messages and other outputs becomes a serious task. In addition, software teams in a company that cannot find the opportunity to collaborate may be unaware of a new quality activity implemented by another team. This means that the potential within the software company cannot be efficiently utilized. Bimar Quality Portal (KAP) software has emerged to solve these problems that we observed in the field. In this experience paper, the needs that reveal the KAP software, how the KAP application meets these needs, and the achievements will be revealed.

Keywords—software quality, software development management, process improvement

I. GİRİŞ

Yazılım kalitesinin izlenmesi yüksek kalite hedefi güden yazılım firmaları için önemli bir etkinliktir. Bu **deneyim bildirisinde** Bimar yazılım firmasında kullanılan Bimar Kalite Portalı (KAP) ile edinilmiş deneyimler ve kazanımlar paylaşılmıştır.

Yazılım ekiplerinin geliştirdiği her bir yazılım projesi KAP içinde bir proje ismi ile tekil bir kayıt olarak temsil edilir. Otomatik veya elle gerçekleştirilen operasyon ve kalite faaliyetlerinin çıktıları proje ismi birlikte KAP sistemine çeşitli kanallardan gönderilebilir. Hali hazırda KAP 'a dahil edilen bir projenin aşağıdaki bilgileri takip edilebilmektedir:

- Projenin teknik ve idari bilgileri,
- Statik kod taraması başarı durumu (Quality Gate),
- Birim testleri kapsama oranı,
- Otomatik ara yüz testleri sonuçları ve test adetleri,
- Entegrasyon testleri sonuçları ve test adetleri,
- Uygulama devreye alım hattı derleme sonuçları,
- Hazırlanan veri tabanı performans raporları.

Yazılım ekibi, uygulamalarındaki kalite öğelerinde bir uygunsuzluk tespiti varsa veya uygulama devreye alım hattında bir sorun yaşamışsa bundan sadece KAP ekranlarını takip ederek haberdar olabilirler. Kalite öğelerinin birinde bile bir sorun varsa (örneğin, statik kod tarama sonucu kodun istenen kalitede yazılmadığı ortaya çıktı veya bir tane başarısız olan entegrasyon testi var) böyle bir durumda Şekil 1’de görüldüğü gibi ilgili proje satırında “Genel Durum” simgesi sorun var anlamına gelen kırmızı bir ünlem (!) olacaktır. Ekiplerin ana ekranı açtıklarında ilk olarak sadece “Genel Durum” simgesini kontrol etmesi oldukça pratik olmaktadır. Kırmızı ünlem var ise “Detaylar” kısmından sorunlu unsurlar kontrol edilebilir.



| Genel Durum | Uygulama | Sonarqube Link | Quality Gate | Code Coverage | UI-Test | Kalite Öğelerinin Durumu | Detaylar |
|-------------|----------------|----------------|--------------|---------------|---------|--------------------------|----------|
| ! | Uygulama 1.0.0 | >Sonar2PB | Failed | 16 / 31 | X | Bugün | Detaylar |
| | .NET FR, 4.6.1 | | Bugün | Bugün | Bugün | Bugün | |
| ✓ | Uygulama 2.0.0 | >Sonar2PB | OK | 41 / 38 | ✓ | Bugün | Detaylar |
| | MVC | | Dün | Dün | Bugün | Bugün | |

Şekil 1. KAP Ana Ekranı

Şekil 2’de sorun olan projenin detayını göstermektedir. Bu örneğe göre “SSRS-Test-Otomasyonunda” tespit edilen 1 adet başarısız test vardır.

| | | |
|---------------------------------------|--|----------------------------|
| SSRS- TESTOTOMASYON- MTST | 1 adet sorun var. Settings- MTST-.xml | 04.08.21 09:42 Bugün |
| UI-Test- Otomasyon-A | Sorun Yok | 04.08.21 10:03 Bugün |
| UI-Test- Otomasyon- Arkasonline | Sorun Yok | 04.08.21 10:11 Bugün |

Şekil 2. Projenin Kalite Öğelerinin Durumu

KAP uygulamasının başka bir işlevi ekipler arasındaki bilgi akışını sağlayarak toplam kalite bilincini yaygınlaştırmasıdır. Ofiste bir araya gelme şansı olmayan ekipler, birbirlerinin projelerini KAP üzerinden izleyerek, “bu otomasyon bize niye yok, bunu biz de kendi uygulamamızda devreye alalım” diyebilmektedirler.

II. BENZER ÇALIŞMALAR

Yazılım kalite izleme araçlarında bir yaklaşım çeşitli kaynaklardan verileri toplayıp bir kullanıcı arayüzü ile bunları yazılım ekiplerine sunmaktır. Alitheia Core, yazılım kalitesi izleme amacıyla geliştirilmiş, çeşitli veri kaynaklarını birleştirip bunların çeşitli kullanıcı arayüzleri aracılığıyla erişilebilir olmasını sağlayan, takılabilir, genişletilebilir bir mimari etrafında modellenmiş bir platformdur [1]. KAP ile Alitheia Core platformu bu açılardan birbirine benzemektedir.

Bir başka yaklaşım ise yazılım haritalarını kullanmaktır. Bir yazılım haritası, modüler bir hiyerarşide düzenlenen kaynak kod dosyaları gibi yazılım uygulama yapılarının uyarlanabilir, hiyerarşik bir temsiliyi temsil eder. Haritalar, yazılım geliştirme, yazılım kalitesi ve sistem dinamikleri hakkındaki bilgileri ifade edebilir ve birleştirebilir; şablonlar tarafından sistematik olarak belirlenebilir, otomatik olarak oluşturulabilir ve organize edilebilirler. Yazılım haritaları, karar verme süreçlerini desteklemeyi amaçlar [2]. Yazılım haritaları yaklaşımı şu anda KAP içinde bulunmamakla beraber gelecek çalışmalar kapsamında değerlendirilmektedir.

III. PORTAL ANLAYIŞI

KAP tüm kalite faaliyetlerine giriş kapısı olma işlevini görmektedir. Yazılım ekiplerine bu şekilde kolay erişebilecekleri ve kullanabilecekleri merkezi bir iletişim kanalı açtığımızda benzer faaliyetler ve bilgi alışverişi için bu ortamı kullanma şansınız olmaktadır. Şekil 3’te görüldüğü gibi yazılım ekiplerine ortak hizmet veren servis ve eğitim belgelerine erişim linkleri yine KAP üzerinde yer almaktadır.



Şekil 3. Her yazılım geliştiricinin bilmesi gereken hizmet ve belgelere erişimi KAP üzerinden sağlayabiliriz

IV. OYUNLAŞTIRMA

Oyunlaştırma; oyun felsefesinin, oyun mekaniklerinin oyun-dışı alanlarda, motivasyonu arttırmak ve kullanıcıları problem çözmeye teşvik etmek için kullanılmasıdır [3]. Kalite faaliyetlerine dikkat çekmek ve özendirmek adına oyunlaştırma unsuru kullanılmıştır. Yazılım ekipleri gerçekleştirdiği kalite faaliyetleri çeşitliği ve durumu

üzerinden hesaplanan bir yöntem ile kalite olgunluk seviyelerini temsil eden “Kap Yıldızı” 🌟, “Nazar Boncuğu” 🎯 gibi rozetler kazanmaktadır. Bu rozetler KAP ana ekranında tüm ekipler tarafından görülebilmektedir. Kendi projelerinde böyle bir rozetin olmadığını gören ekipler, “bunu almak için ne yapabiliriz” diye motive olup rozetin gerektirdiği kalite faaliyetini devreye almak istemektedirler. Benzer şekilde sadece bir projesinde “Kap Yıldızı” almaya hak kazanan bir ekip, sorumlu oldukları tüm projelerde “Kap Yıldızını” alma hedefi verip, kendi standartlarını yükseltebilmektedir.

V. UYGULAMA KALİTE ÇERÇEVESİ BELGESİ

Şirket içindeki kalite faaliyetlerinin sürekliliğini ve yaygınlaşmasını sağlayan KAP uygulaması fikrini olgunlaştıran önemli gelişmelerden biri “uygulama kalite çerçevesi” belgesidir [4]. Şirketin o ana kadar edindiği operasyon ve kalite kapsamındaki edinilmiş tecrübe ve pratikler bir kontrol listesi olarak bu belgede sunulmaktadır. Bu belge KAP içinde gösterimi yapılan kalite faaliyetleri ile ilgili maddeleri de içermektedir. Yeni bir yazılıma başlayan ekip, kodlarını barındırdığı ortamda ilgili proje dosyasını açtıktan sonra bu kontrol listesinden bir kopya alarak yeni yazılımlarında izlemeleri gereken operasyon ve kalite adımlarını kontrol ederler. Uygulamalarda, test sistemi açmak gibi çok temel bazı adımların atıldığı bu belge üzerinden yapılan kontrollerde ortaya çıkabilmektedir. Sürekli güncel tutulan bu belge KAP içinde kullanıcının dikkatini çekecek şekilde en kolay erişeceği konumda bulunmaktadır. Şekil 4’te bu belgedeki kontrol adımlarından birkaç tanesi yer almaktadır.

Temel İşlevler

1. Uygulama Azure Devops reposuna eklendi mi?
2. Uygulamanın test sistemi var mı?
3. Development için belirlenen branch’in “Pull Request” politikasında, Gate keeper derlemesi kriteri eklendi mi?
4. Development için belirlenen branch’in “Pull Request” politikasında en az x adet “reviewer” olmalıdır kriteri eklendi mi?
5. Devreye alım için BUILD ve RELEASE tanımları Azure Devops’ta tanımlandı mı?

İzleme

6. Application Insights bağlantısı yapıldı mı?
7. Dynatrace izlemesi bir kez aktif edildi mi?
8. Dynatrace izlemesini ilgili ekipten kişi veya kişiler yapabildiklerini gördüler mi? (Dynatrace şifrelerini aldılar mı, temel işlevleri kullandılar mı?)

Şekil 4. Uygulama Kalite Çerçeve Dokümanından bir kesit

VI. TEKNİK ÖZELLİKLER

Portala veri akışı Azure Devops™, Azure Logic Apps™, SonarQube™ araçları ve web kancası teknolojisi kullanarak otomatik olarak sağlanmaktadır. Elle bir veri girişi yoktur.

VII. GÖZLEMLER VE RİSKLER

Yeni geliştirilen bir test otomasyonunu devreye almadan önce, yeterli olgunluğa eriştiğinden emin olunmalıdır. Farz edelim ki aslında başarılı olması gereken bir test, otomasyonumuzun kendisinden kaynaklı bir problemden dolayı hata veriyor. Test hatalı uyarısını alan yazılımcı, testin kendisinde sorun olmadığını görerek, sorunun otomasyondan kaynaklı olduğu anlayacaktır. Bu şekilde zamanı heba olan yazılım geliştiricinin, yanlış konuda alarm veren bu otomasyona güveni azalacaktır. Bu durumun yaşanmaması için otomasyon sistemlerinizin uyarı mesajlarını öncelikle bu sistemi kuran kişilere (otomasyon sorumlularına) yönlendirmek gerekir. Otomasyonun sorumluları bir müddet sistemin verdiği uyarıları proje ekipleri ile paylaşır ve doğruluğunu teyit eder. Gerekirse otomasyonda düzeltmeler yapar. Bir otomasyon ancak bu şekilde olgunlaştırarak hem KAP’a dahil edilebilir hem de uyarı mesajları herkesi kapsayacak hale getirilebilir.

The screenshot shows the 'Bimar Kalite Portalı' interface. At the top, there is a navigation bar with several links: 'Uygulama Kalite Çerçevesi', 'Güvenlik Portalı', 'UHS Performans Raporları', 'Dynatrace Logiri', 'Sonuçları', 'Kalite Etkileri', and 'Geri Bildirim'. Below the navigation bar, the main content area displays a line chart titled 'Code-Coverage'. The chart's y-axis represents the percentage of code coverage, ranging from 0 to 80. The x-axis shows dates from 14.06.2021 to 07.09.2021. The data is represented by a blue line with circular markers at each data point. The coverage starts at approximately 69%, dips to around 67% in mid-June, then rises and stabilizes between 69% and 72% through July and August. A tooltip for the date 02.07.2021 indicates a coverage of 67.

| Date | Code-Coverage (%) |
|------------|-------------------|
| 14.06.2021 | 69 |
| 15.06.2021 | 69 |
| 17.06.2021 | 68 |
| 21.06.2021 | 67 |
| 23.06.2021 | 67 |
| 25.06.2021 | 67 |
| 29.06.2021 | 68 |
| 01.07.2021 | 68 |
| 05.07.2021 | 68 |
| 06.07.2021 | 67 |
| 08.07.2021 | 69 |
| 09.07.2021 | 69 |
| 13.07.2021 | 69 |
| 27.07.2021 | 68 |
| 29.07.2021 | 70 |
| 02.08.2021 | 70 |
| 04.08.2021 | 70 |
| 07.08.2021 | 71 |
| 07.09.2021 | 72 |

KAP 'ı kullanan yönetici ve yazılım geliştiricilere “bu uygulamanın çalışmalarınıza kattığı en önemli değer nedir?” sorusunu yönelterek bir anket yaptık. Gözlemlediğimiz kazanımlar ile gelen yorumların örtüşüğünü fark ettik. KAP uygulamasının yazılım geliştirme ekibinin hayatına girmesi ile elde ettiğimiz kazanımlar şunlardır:

- Şekil 5. KAP uygulamasında sayısal değerle ifade edilebilen kalite öğelerinin zamanla değişimini takip edebilirsiniz.

Kalite Portalı (KAP) uygulamasının şirket içinde kullanılmasıyla ekipler arası iletişim artmış, farklı projelerde uygulanan doğru ve iyi pratikler diğer projelere sıçramıştır. Örneğin, uygulamanın devreye alındığı dönemde birim test alt yapısı olmayan eski projelerimizde bu konu sıcak gündem olmuş ve birim testler yazılmaya başlanmış, yeni geliştirimlerde birim testlerin faydasından mahrum kalmamıştır.

Proje ekiplerinin üzerinde, şirket seviyesinde yazılım kalite faaliyetlerinin durumunun sayısal veriler ile şeffaf bir şekilde izleme olanağı sağlanmıştır. Şirket içindeki her yazılım geliştirici ve yöneticisi tüm ekiplerin kalite faaliyetleri ve bu faaliyetlerin zaman içindeki gelişimi ilgili bilgi sahibi olabilmektedir. Bu şeffaf iletişim şekli, kod taraması gibi temel kalite öğelerinde nispeten geride olan ekiplerin kendileri belli düzeye getirmeleri için teşvik edici bir ortam oluşturmuştur. Bu ekipler eksik faaliyetlerini tamamlayarak şirket içi bir standardın oluşmasını sağlamıştır.

- [1] G. Gousios and D. Spinellis, "Alitheia core: An extensible software quality monitoring platform." 2009 IEEE 31st International Conference on Software Engineering. IEEE, 2009.
- [2] J. Bohnet and J. Döllner, "Monitoring code quality and development activity by software maps." Proceedings of the 2nd Workshop on Managing Technical Debt. 2011.
- [3] A. Bozkurt and E. Genç-Kumtepe, "Oyunlaştırma, Oyun Felsefesi ve Eğitim: Gamification." Akademik Bilişim. 2014. Erişim Adresi: https://www.academia.edu/6169090/Oyunlastirma_Oyun_Felsefesi_ve_Egitim_Gamification.
- [4] S. Apul, "Kalite Çerçeve Dokümanı." 2021. Erişim Adresi: <https://github.com/BimarBilgiIslem/Kalite/blob/master/Bimar-Uygulama-Kalite-Cercevesi.pdf>.