

LAPIS Çevik Yazılım Geliştirme Sürecinin Ölçülmesi ve İyileştirilmesi

Tuğrul Tekbulut¹, Nurdan Canbaz¹, Ersin Gülaçtı¹, Tuğba Öztürk Kaya¹

¹ Logo Yazılım, Kocaeli, Türkiye
tugrul.tekbulut@logo.com.tr, nurdan.canbaz@logo.com.tr,
ersin.gulacti@logo.com.tr, tugba.ozturkkaya@logo.com.tr

Özet. Ana faaliyet alanı yazılım geliştirme olan şirket/organizasyonların, zamanında, kaliteli ve düşük maliyetli bir yazılım ürünü geliştirmesinin en önemli adımlardan biri şirketin yaptığı işe en uygun yazılım sürecinin belirlenmesidir. Belirlenen süreçteki her adım sürekli ölçülerek iyileşme sağlanmalıdır. Yazılım geliştirme kavramının soyut olması, ölçülmesini de zorlaştırmaktadır. Bu yüzden ölçülecek kriterlerin çok iyi belirlenmesi ve amaca yönelik olması gerekmektedir.

Bu çalışmada, Logo Yazılımın yalın üretim felsefesinden esinlenerek geliştirdiği, iyileşme odaklı çevik ürün geliştirme süreci olan LAPIS'in (Logo Agile Process Improvement System) şirket beklentileri ve çevik yazılım geliştirme eğitimleri ışığında iyileşme faaliyetlerinin Hedef Soru Metrik (Goal Question Metric) yaklaşımı ile belirlenmesi, ölçümlerin sürecin akışına uygun olarak otomatize edilmesi, her role uygun olarak kişilere ve ekiplere oyunlaştırma teknikleri kullanılarak aktarılması anlatılmaktadır. Bunlara ek olarak, ölçümlerin takip edilerek oluşabilecek aksaklıkların ilerlemeden önlenmesine odaklanan iyileştirme faaliyetlerine değinilmektedir.

Anahtar Kelimeler: Çevik, Yazılım Süreci, Sürekli İyileşme,

Measurement and Improvement of LAPIS Agile Software Development Process

Tuğrul Tekbulut¹, Nurdan Canbaz¹, Ersin Gülaçtı¹, Tuğba Öztürk Kaya¹

¹ Logo Yazılım, Kocaeli, Tukey
tugrul.tekbulut@logo.com.tr, nurdan.canbaz@logo.com.tr,
ersin.gulacti@logo.com.tr, tugba.ozturkkaya@logo.com.tr

Abstract. One of the most important steps for the companies/organizations engaging mainly in software development in developing a timely, high-quality, low-cost software product is to identify the most appropriate software process for the company's business. Every step in the specified process should be continuously measured in order to ensure improvement. The abstract nature of the software development concept makes it difficult to be measured. Therefore, the criteria to be measured must be defined very well and they must be goal oriented.

In this study, it is explained how to determine improvement activities for the Logo Agile Process Improvement System (LAPIS), an agile product development process developed by Logo Software and inspired by the lean production philosophy, by using the Goal Question Metric approach based on the company's expectations and agile software development trends, to automate the measurements according to the flow of the process and to transfer them to the people and the teams using the playing techniques suitable for each role. In addition to these, improvement activities focusing on the prevention of the inconveniences that may occur by following the measurements are addressed.

Keywords: Agile, Software Process, Continuous Improvement, LAPIS

1 Giriş

Modern dünyada bilgisayar yazılımlarının insan hayatına her geçen gün daha çok girmesi ile birlikte yüksek kaliteli yazılım geliştirmek firmalar açısından önemli hale gelmiştir. Herhangi bir metodolojiye dayanmadan geliştirilen yazılımlar, zamanında teslim edilememekte hem yazılım şirketlerine hem de müşterilere vakit kaybı yaşatmaktadır. Bu nedenle yazılım kalitesi farklı açılardan değerlendirilmelidir. Bunlar, geliştirici ekip, ürün yönetimi ve yazılım ürünü satın alacak müşteri vb. paydaşlardır.

Müşteri açısından incelendiğinde, kaliteli bir yazılımın temel özellikleri işlevsellik, gereksinimlere uygunluk, kullanım kolaylığı, güvenilirlik (hata sıklığının az olması) ve güncelleme kolaylığıdır. Geliştirici ve ürün yönetimi açısından bakıldığında ise kalite kriterleri, çıkabilecek hataları veya ilerde sorun yaratabilecek bileşenleri önceden tespit edebilmek, maliyeti azaltmak, daha başarılı bir projeye imza atabilmek vb. olarak sıralanabilir [1].

Bakış açısı ve role göre değişen bu kalite bileşenlerine sahip ürünler geliştirebilmek için arkada sağlam bir süreç disiplini olmalıdır, ancak bu disiplin oluşturulurken yazılım sektörünün insan temelli olduğu gerçeğinin getirdiği zorluklar unutulmamalıdır. Logo Yazılımın yalın üretim felsefesinden esinlenerek geliştirdiği, iyileşme odaklı çevik ürün geliştirme süreci olan LAPIS (Logo Agile Process Improvement System) [2] müşteri ihtiyaçlarına hızlı yanıt verebilmeye, farklı bakış açılarına göre değişen kalite niteliklerini sağlamaya ve ekiplerin sürece katkı sağlaması üzerine odaklanmıştır.

Yazılım geliştirme süreçlerinin temel sorunlarından biri, soyut kavramlara dayandığı için beklenen faydaların ölçülmesinin zor olmasıdır. Tom de Marco'nun "Ölçülemeyen bir şeyin kontrol edilebilmesi oldukça zordur" [3] sözüyle belirttiği gibi süreci de ölçemediğimizde aslında eksik bırakmış oluruz. Tanımlanan süreçlerde birçok değer ölçmek mümkündür, ancak arka planda kurgulanmış bir ölçüm süreci olmadığında, ölçüm yapmak belirli noktalarda sadece metrikler üretmenin ötesine geçemez. Bu nedenle oluşturduğumuz ölçüm metodunda, yazılım endüstrisinde ölçüm sürecine yönelik etkinlik ve işleri tanımlayan uluslararası bir standart olan ISO 15939 Yazılım Ölçme Standardı [4] temel alınmıştır. Süreçte takip edilebilecek ürün hedeflerine uygun metriklerin belirlenebilmesi için Hedef-Soru-Metrik (İng. Goal-Question-Metric) yaklaşımı [5] benimsenmiş ve bu yaklaşım doğrultusunda ilerlenmiştir. Bu çalışmada, uygulanan yazılım geliştirme süreci için kurgulanan ölçme ve iyileştirme çalışmalarından bahsedilecektir.

2 İlgili Çalışmalar

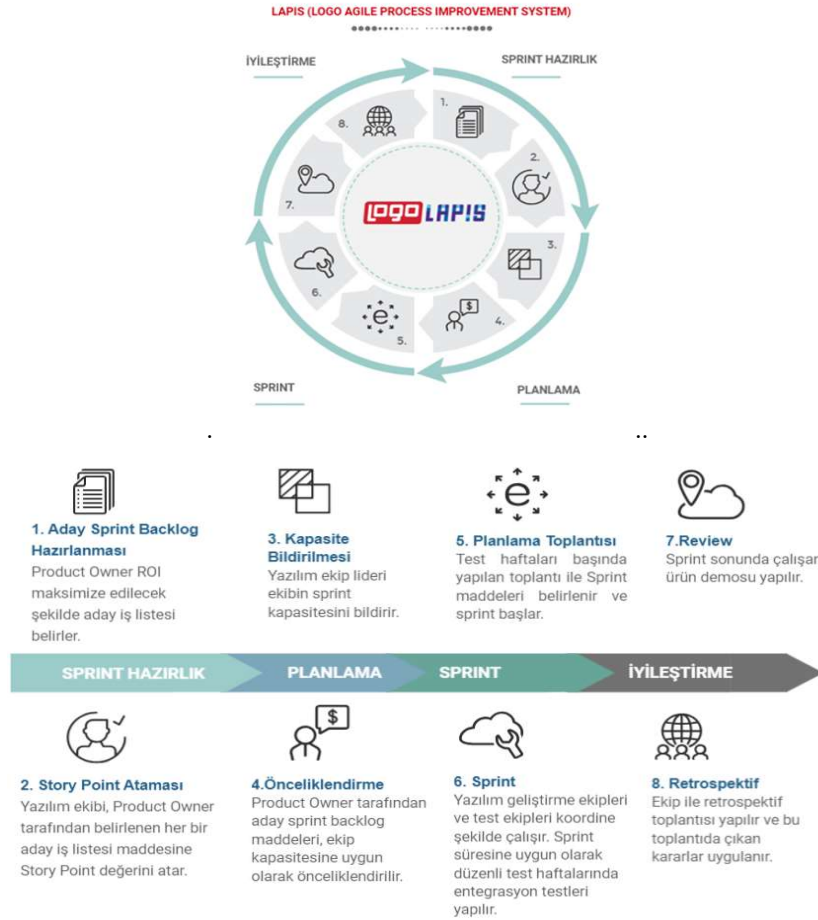
Çevik metodolojiler, günümüzde oldukça tercih edilen yazılım pratikleri olarak karşımıza çıkmaktadır. Büyük ve orta ölçekli firmalarda değişen müşteri gereksinimlerine hızlı yanıt verebilmek ve değişen önceliklere hızlı adapte olabilmek için çevik yazılım geliştirme yöntemlerinin kullanımı dünyada ve ülkemizde her geçen yıl yaygınlaşmaktadır [6]. Çevik süreçlerin etkilerinin firmalar tarafından takip edilebilmesi de bu paralelde önem kazanmaktadır.

Çevik yazılım süreçlerinin dayandığı çevik manifesto [7], yazılım geliştirme için gerekli pratikleri sunar ancak çevikliği ölçmek için araç seti ya da standart bir yöntem sunmaz. Çevik metodolojilerin değerlendirilmesi ile ilgili çeşitli çalışmalar bulunmaktadır. Gandomani ve Nafchi yaptıkları çalışmada 44 farklı çevik pratiği değerlendirerek bir çevik değerlendirme modeli sunmuşlar ve bu pratikleri kullanan firmaların çeviklik seviyesini ölçmeyi amaçlamışlardır [8]. Qumer ve Henderson yaptıkları çalışmada uygun çevik yöntemi seçmek için 4 boyutlu bir model sunmuştur. Bu model ile firmaların uygun çevik süreci seçmelerini amaçlamışlardır [9]. Çevik metodolojilerin ölçülmesi ile ilgili olarak, Christopher W. H farklı bakış açılarına göre sürecin detayda nasıl ölçülmesi gerektiğine ışık tutmuştur ve yazılımcının takip etmesi gereken metrikler ile yönetici seviyesinde takip edilmesi gereken metriklerin farklılığından bahsetmiştir. Aynı zamanda ölçmenin sürekli iyileştirme için temel teşkil ettiğine değinerek sürecin belirli bir aşamasının ölçülüp değerlendirilmesinden çok, ölçümün trendinin takip edilmesinin önemini vurgulamıştır [10]. Hartmann ve Dymond çevik süreçlerin

ölçülmesi ile ilgili yaptıkları çalışmada müşteriye sunulan değere odaklanmışlar ve ölçümlerde trende ağırlık verilmesi gerektiğinden bahsetmişlerdir [11]. Ayrıca çevik dönüşümün, kaliteye yönelik etkilerinin araştırıldığı çalışmalar mevcuttur [12]. Furtato ve arkadaşları yaptıkları çalışmada geleneksel ve çevik süreçler arasındaki izlenebilirliği karşılaştırarak izlenebilirliğin takip edilmesi ve ölçülmesi için bir yöntem sunmuşlardır [13].

3 LAPIS (Logo Agile Process Improvement System)

LAPIS, farklı metodolojilere dayanan ürün geliştirme yönetim modellerinden ve yalın üretim felsefesinden etkilenmiş bir ürün geliştirme modelidir [1]. Sürecin genel akışı Şekil 1'de verilmiştir



Şekil 1. LAPIS (Logo Agile Process Improvement System)

Ana felsefesini yalnız üretim felsefesinden alan LAPIS, çevik yöntemlerden olan Scrum'dan [14] esinlenmiştir. Scrum'dan farklı olarak yazılım geliştirme ekipleri çapraz işlevli değil, farklı alanlarda özelleştirilmiştir. LAPIS sürecinde Ürün Sahibi (Product Owner), iş analisti, sistem analisti, yazılım geliştiriciler, test uzmanları gibi roller kendi uzmanlıkları doğrultusunda ekiplerde farklı görevlerde çalışmaktadır. Özellikle ürün geliştiren şirketlerde uzmanlık alanları büyük önem taşımaktadır.

Her ürün için ürünün piyasadaki konumu ve ihtiyaca göre sprint süreleri sene başında belirlenir. Sprint sürelerinin minimum %20'lik zamanı entegrasyon testlerinin yapıldığı test haftalarına ayrılır. Sprint başlamadan önce Ürün Sahibi, ürün iş listesinde (Product Backlog) bulunan maddeleri önceliklendirir. Mutabakat toplantısında (Sprint Planlama Toplantısı) Ürün Sahibi ve geliştirme ekibi önceliklendirilmiş maddelerden hangilerinin izleyen sürüm içeriğine dahil edileceğini, geliştirme ve test ekibinin kapasitesine uygun olarak, belirler.

Yapılmasına karar verilen maddeler Sürüm İş Listesi (Sprint Backlog) olarak ilan edilir. Bu maddeler Logo'da kullanılan destek sistemi üzerinden otomatik olarak iş ortaklarına da duyurulur. Entegrasyon testlerinin yapıldığı hafta geliştirme takımları test ekibinden gelen hata düzeltme (bug-fix) taleplerini yapmakla beraber bir sonraki sürümün geliştirmesine başlar. Aynı hafta içinde, tamamlanan sürüm içeriğini Ürün Sahibi, destek ve proje ekipleri ile paylaşır. (Sprint Review) Sprint sonunda sprintte yapılanların değerlendirilmesi ve iyileşme noktalarının tespit edilmesi için süreç yöneticisi tarafından retrospektif toplantısı düzenlenir.

Key	Summary	Status	Fix Version/s
LP-2146	TIGER - ERP 2019.R1 için Sürüm Planlaması (2.56-2019.R1 01/28)	OPEN	2.56-2019.R1 01/28
LP-2145	TIGER - ERP 2018.R7 için Sürüm Planlaması (2.55-2018.R7 12/10)	OPEN	2.55-2018.R7 12/10
LP-2144	TIGER - ERP 2018.R6 için Sürüm Planlaması (2.54-2018.R6 10/22)	OPEN	2.54-2018.R6 10/22
LP-2143	TIGER - ERP 2018.R5 için Sürüm Planlaması (2.53-2018.R5 09/03)	OPEN	2.53-2018.R5 09/03
LP-2142	TIGER - ERP 2018.R4 için Sürüm Planlaması (2.52-2018.R4 06/25)	IN PROGRESS	2.52-2018.R4 06/25
LP-2141	TIGER - ERP 2018.R3 Sürüm Planlaması (2.51-2018.R3 05/07)	RESOLVED	2.51-2018.R3 05/07

Details		Dates	
Type:	Epic	Status:	OPEN
Priority:	Normal	Resolution:	Unresolved
Lapis Product:	WIN - ERP	Fix Version/s:	2.56-2019.R1 01/28
Sprint:	TIGER 2.56-2019.R1 01/28	Created:	06/Dec/17 08:06
		Updated:	16/Jun/18 13:45
		Sprint Start Date:	09/Dec/18
		Sprint End Date:	27/Jan/19
		Test Start Date:	13/Jan/19
		Test End Date:	27/Jan/19

Issues in Epic		
LP-2207	TIGER - ERP 2019.R1 için Sprint Backlog Hazırlanması (2.56-2019.R1 01/28)	OPEN Unassigned
LP-2209	TIGER - ERP 2019.R1 için Story Point Atanması (2.56-2019.R1 01/28) - Ekip I	OPEN Unassigned
LP-2210	TIGER - ERP 2019.R1 için Story Point Atanması (2.56-2019.R1 01/28) - Ekip II	OPEN Unassigned
LP-2211	TIGER - ERP 2019.R1 için Kapasite Bildirilmesi (2.56-2019.R1 01/28) - Ekip I	OPEN Unassigned
LP-2212	TIGER - ERP 2019.R1 için Kapasite Bildirilmesi (2.56-2019.R1 01/28) - Ekip II	OPEN Unassigned
LP-2213	TIGER - ERP 2019.R1 için Mutabakat Toplantısı (2.56-2019.R1 01/28)	OPEN Unassigned
LP-2214	TIGER - ERP 2019.R1 için Önceliklendirme Yapılması (2.56-2019.R1 01/28)	OPEN Unassigned
LP-2216	TIGER - ERP 2019.R1 için Sürüm Onayı (2.56-2019.R1 01/28)	OPEN Unassigned
LP-4173	TIGER - ERP 2019.R1 için Sürüm Duyurusu (2.56-2019.R1 01/28)	OPEN Unassigned

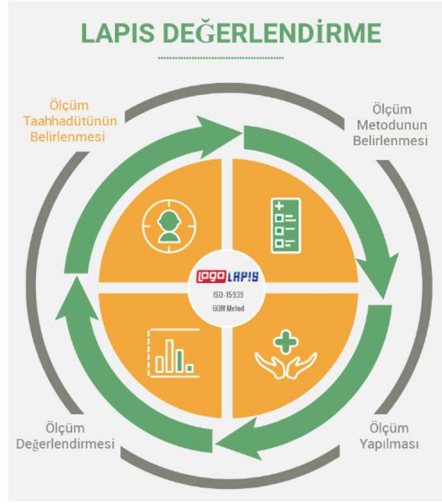
Şekil 2. LAPIS İş Akışı

LAPIS sürecindeki her bir adım Atlassian firmasına ait olan Jira [15] proje ve talep yönetim sisteminde takip edilir. Şekil 2'de gösterildiği gibi sene başında hazırlanan

takvimde her bir sürüm için önemli tarihler (sprint başlangıç/bitiş tarihleri, test haftaları başlangıç/bitiş tarihleri) yapılacak işlerin akışı ve bitiş süreleri tanımlanır ve ilgili kişilere otomatik olarak atanır. Bu işlemlerin takibi ve kontrolü süreç yöneticileri tarafından sağlanır.

4 Ölçüm Modeli

Yapılan araştırmalar, yazılım süreçlerinin tanımlanması ve ölçülmesi konusunda çalışmaların halen devam ettiğini göstermektedir. Bu nedenle, şirket stratejisine, geliştirilen ürünlere ve ekip dağılımlarına göre tasarlanan LAPIS sürecinin, ölçülmesinin de benzer şekilde olması için uygun ölçüm metodu üzerine çalışmalar yapılmıştır.



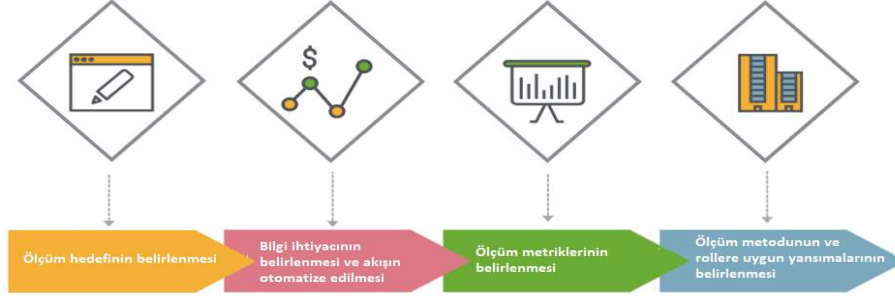
Şekil 3. LAPIS Değerlendirme Yöntemi

Ölçüm metodu oluşturulurken ISO 15939 Yazılım Ölçme Standardından yararlanılmıştır [4]. Oluşturulan ölçüm metodunun genel döngüsü Şekil 3'te verilmiştir. LAPIS sürecine uyumun değerlendirilebilmesi için, ilk adım ölçüm taahhüdü yönetim ile birlikte belirlenmesidir. Yönetim ile ihtiyaçları karşılayacak şekilde bir metod oluşturulması ve ekiplerin buna destek sağlaması konusunda ölçüm taahhüdü yapılmıştır. Ölçüm metodu belirlenirken Şekil 4'te verilen adımlar izlenmiştir. Ölçümler yapıldıktan sonra süreçteki tüm paydaşlarla birlikte değerlendirilerek yeni iyileştirme noktaları belirlenmesi amaçlanmıştır.

4.1 Ölçüm Metodunun Belirlenmesi

Ölçme sürecinin oluşturulması ile ilgili adımlar Şekil 4'te verilmiştir. Detaylar Bölüm 5 ve 6'da anlatılmaktadır.

LAPIS Ölçüm Metodunun Belirlenmesi



Şekil 4. LAPIS Ölçüm Metodu Belirlenmesi

Ölçüm Hedefinin Belirlenmesi . Hedef olmadan süreçteki adımlar ölçülmeye başlandığında sadece amaçsız sayı kümeleri oluşmaktadır. Bu sebeple model hazırlanırken Hedef-Soru-Metrik yaklaşımı benimsenerek hedefe uygun ölçüm metodu belirlenmesi amaçlanmıştır. Ölçüm hedefinin belirlenmesinin detayları Bölüm 5’te, hedefler ve belirlenen metrikler Tablo 1’de verilmiştir.

Bilgi İhtiyacının Belirlenmesi ve Akışın Otomatize Edilmesi. Hedef-Soru-Metrik yaklaşımı ile belirlenen soruların cevaplarını alabilmek için süreç akışının otomatize edilmesi ihtiyacı ortaya çıkmıştır. Bu ihtiyacın giderilmesi için süreçteki tüm adımların bir araç ile takip edilmesine karar verildi ve LAPIS süreci Jira’da gerekli uyarlamalar yapılarak tamamen bu sistem üzerinden takip edilmeye başlandı. Böylelikle hem süreçteki tanımlamaların daha eksiksiz yapılması hem de insan müdahalesi ile oluşabilecek aksaklıkların önüne geçilmesi sağlandı.

Ölçüm Metriklerinin Belirlenmesi. Hedef-Soru-Metrik yaklaşımı ile oluşturulan metrikler ve bu metriklerin etki katsayıları belirlendi. Metriklerin otomatik şekilde süreçten beslenmesi ve oluşturulan veri havuzunda toplanması sağlandı.

Ölçüm Metodunun Oluşturulması ve Rollere Dağıtılması. Ölçüm metodunu oluşturulup, bu ölçümlerin takip edileceği ortama karar verilmiştir. Farklı departmanların da kullandığı Confluence [16] aracı üzerinde gerekli uyarlamalar yapılarak, ölçümler için kurumsal hafızayı oluşturacak platform hazırlanmıştır. Burada amaçlanan hem maliyeti düşürmek hem de ölçüm sonuçlarının şirketteki herkesin erişebildiği bir ortamda yayınlanması ile kişiler tarafından sahiplenilmesini sağlamaktır. Ölçüm sonuçlarının rollere göre detaylandırılması Bölüm 6’da verilmiştir.

5 Ölçme Metodu

ISO 15939 [4] temel alınarak ölçüm modeli oluştururken, üst düzey hedefler ile alt düzeydeki metrikleri uygun amaçla bağlamayı hedefleyen Hedef-Soru-Metrik yaklaşımı baz alınmıştır. Çünkü süreç içinde rastgele veri toplayıp bunlardan anlam çıkarmak yerine uygun hedeflere, uygun metriklere ulaşmak ve uçtan uca izlenebilirlik sağlamak amaçlanmıştır. Tablo 1'de ölçüm hedefleri, hedeflere ulaşmak için sorulan sorular ve belirlenen metrikler verilmiştir.

Hedefler ve metrikler belirlenirken hem yönetimin belirlediği genel yaklaşım hem de şirket içinde farklı bölümlerdeki kişilerle yapılan çalışmalar sonucunda belirlenen ihtiyaçlar göz önünde bulundurulmuştur.

Tablo 1. Ölçüm Hedefleri ve Metrikleri

HEDEF	SORU	METRİK
Müşteri Odaklılık	Sürümler zamanında çıkıyor mu?	Sürüm Uyum Oranı Sene başında belirlenen sürüm tarihlerine uyum oranı
	Müşterilere verilen sözler zamanında yerine getiriliyor mu?	Sürüm Tamamlanma Oranı Sürüm kapsamına alanın maddelerin tamamlanma oranı (Madde Sayısı) Sürüm kapsamına alanın maddelerin tamamlanma oranı (Story Point)
	Müşteri isteklerine yazılım ekiplerimizin cevap süresi nedir?	Ek Özellik Yaşı Ürün İş Listesinde (Product Backlog) bulunan ek özelliklerin ortalama yaşı
	Hatalara yazılım ekiplerimizin cevap süresi nedir?	Hata Yaşı Ürün İş Listesinde (Product Backlog) bulunan hataların ortalama yaşı
Kaliteli Ürün Geliştirmek	Müşterilerden önce hataları tespit edebiliyor mu?	Hata Tespit Oranı Yıl içinde test ekibi tarafından tespit edilen hataların tüm hatalara oranı
	Düzenli test haftalarına uyum sağlayabiliyor mu?	Test Haftasına Uyum Test haftasında hata düzelme dışında teste gönderilen maddelerin planlanan maddelere oranı
Verimlilik	Kapasite planlamasına uyum sağlanabiliyor mu?	OTEQ (Overall Team Efficiency, Quality) [2] Uygunluk* Performans*Kalite
	Kapasite planlaması verimli yapılabiliyor mu?	Efektif Planlama Oranı Sürümde toplam yapılan madde sayısının planlı madde sayısına oranı (İç test hataları hariç)
	Ürünlerdeki stratejik hedeflere uyum sağlayabiliyor muyuz?	Stratejik Hedeflere Uyum Yıl başında belirlenen stratejik hedeflere uyum oranı

Her metrik için kartlar oluşturulmuş, hangi rollerin hangi zamanlarda bu metrikleri nereden takip edeceğine karar verilmiştir. Hata tespit oranı ve Logo'ya özgü OTEQ raporuna ait örnek metrik kartları Tablo 2 ve Tablo 3'te verilmiştir.

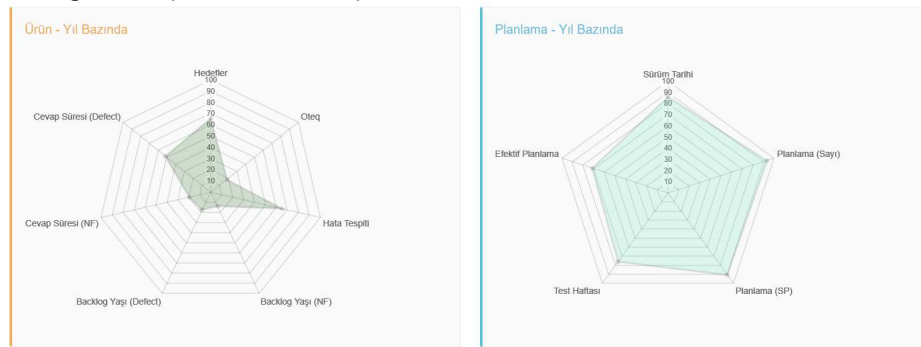
Tablo 2. Hata Tespit Oranı

METRİK	HATA TESPİT ORANI																
Amacı	Kaliteli Ürün Geliştirmek																
Hesaplama Yöntemi	Yıl içinde test ekibi tarafından tespit edilen hataların tüm hatalara oranı																
Ne zaman kontrol edilmeli	Her Çeyrekte																
Nereden kontrol edilmeli	Confluence Sistemi																
Takip eden paydaşlar	LAPIS Paydaşları (Product Owner, Test Ekibi, Yazılım Ekibi)																
Hedef	%95																
Rapor	<p>Hataların Müşteriden Önce Tespit Edilme Oranı</p> <table border="1"> <thead> <tr> <th>Yıl</th> <th>ÜRÜN I</th> <th>ÜRÜN II</th> <th>ÜRÜN III</th> </tr> </thead> <tbody> <tr> <td>2015</td> <td>~65</td> <td>~50</td> <td>~55</td> </tr> <tr> <td>2016</td> <td>~60</td> <td>~65</td> <td>~70</td> </tr> <tr> <td>2017</td> <td>~65</td> <td>~70</td> <td>~75</td> </tr> </tbody> </table>	Yıl	ÜRÜN I	ÜRÜN II	ÜRÜN III	2015	~65	~50	~55	2016	~60	~65	~70	2017	~65	~70	~75
Yıl	ÜRÜN I	ÜRÜN II	ÜRÜN III														
2015	~65	~50	~55														
2016	~60	~65	~70														
2017	~65	~70	~75														

Tablo 3. OTEQ

METRİK	OTEQ (Overall Team Efficiency, Quality) [2]														
Amacı	Verimlilik														
Hesaplama Yöntemi	<p>Yalın üretimde kullanılan OEE formülünden esinlenilmiştir. Uygunluk Durumu x Performans x Kalite (Uygunluk durumu, yönetim kararı ile 1 alınmıştır)</p> <p>Performans: Toplam tüketilen Story Point Kalite: $1 / \sum (\text{Müşteriden gelen hata} * \text{Zarar Puanı})$ Zarar Puanı: Gelen hatanın önemine uygun olarak (1-13 arası katsayı) OTEQ = (1) $(\sum \text{Story Point} / (\sum \# \text{Yazılımcı}) (\# \text{Gün})) (1 / \sum \text{Hata Zarar Puanı})$</p>														
Ne zaman kontrol edilmeli	Her Çeyrekte														
Nereden kontrol edilmeli	Confluence Sistemi														
Takip eden paydaşlar	LAPIS Paydaşları (Product Owner, Test Ekibi, Yazılım Ekibi)														
Hedef	Artan eğilimde olması														
Rapor	<p>Sürüm bazlı örnek OTEQ raporu</p> <table border="1"> <thead> <tr> <th>Sürüm</th> <th>OTEQ</th> </tr> </thead> <tbody> <tr> <td>2.41-2016 R6</td> <td>~0.15</td> </tr> <tr> <td>2.42-2017 R1</td> <td>~0.18</td> </tr> <tr> <td>2.43-2017 R2</td> <td>~0.18</td> </tr> <tr> <td>2.44-2017 R3</td> <td>~0.22</td> </tr> <tr> <td>2.45-2017 R4</td> <td>~0.22</td> </tr> <tr> <td>2.46-2017 R5</td> <td>~0.15</td> </tr> </tbody> </table>	Sürüm	OTEQ	2.41-2016 R6	~0.15	2.42-2017 R1	~0.18	2.43-2017 R2	~0.18	2.44-2017 R3	~0.22	2.45-2017 R4	~0.22	2.46-2017 R5	~0.15
Sürüm	OTEQ														
2.41-2016 R6	~0.15														
2.42-2017 R1	~0.18														
2.43-2017 R2	~0.18														
2.44-2017 R3	~0.22														
2.45-2017 R4	~0.22														
2.46-2017 R5	~0.15														

Belirlenen metriklerin detaylarının oluşturulmasının ardından tüm metriklerin bir arada görüntülenebileceği bir çalışma yapıldı. Burada amaçlanan hem metriklerin daha okunabilir olması hem de oyunlaştırılarak ekipler tarafından benimsenmesinin sağlanmasıdır. Rol bazında görüntülenebilecek raporlara ait örnekler Şekil 6 ve Şekil 7’de verilmiştir. Çalışmada öncelikle takip edilen metrikler iki gruba ayrıldı. Bunlardan ilki planlamaya ve cevap süresine hizmet ederken, diğeri yazılım süreçleri tarafından takip edilen ürün iş listesi durumuna yöneliktir. Bu nedenle, iki grup için ayrı ayrı radar grafikler hazırlanmıştır. Radar grafikler arabaların tekerlerine benzetilebilir, eğer bir kriterdeki değer, diğerlerine oranla farklı ise teker daire şeklini kaybedecek ve aynı arabanın ilerlemesinde olduğu gibi sürecin gidişatında da aksaklıklar olacaktır. Örnek radar grafikler Şekil 5’te verilmiştir.



Şekil 5. Ürün – Planlama Radar Grafiği

LAPIS ölçüm metodunu oyunlaştırmak için her sürüm sonunda alınabilecek rozetler oluşturulmuştur. Tablo 1’de verilen metrikler için bir katsayı belirlendikten sonra her metrik değeri kendi katsayısı ile çarpılır. Oluşan değerler toplanarak her sürüm için Sürüm Puanı elde edilir. Sürüm puanına göre Tablo 4’te belirtilen aralıklara uygun olarak ekibe rozet verilir. Oyunlaştırma tekniklerinin kullanılmasının ve Sürüm Puanını ekipler ile paylaşmak yerine rozet verilmesinin amacı, olumsuz motivasyon yaratılmadan sürece dahil olduklarını hissettirmek ve daha yüksek rozet hedeflerine yönelik çalışılması yönünde teşvik etmektir. Yapılan çalışma kapsamında, seçilen bir ürüne ait sürüm bazlı planlama puanları ve rozetleri Tablo 5’te verilmiştir. Ürüne ait yıl bazında Planlama ve Ürün kapsamındaki değerlendirme sonuçları Tablo 6’da verilmiştir.

Tablo 4. LAPIS Ölçüm Tablosu

Kategori	Aralık	Rozet
A	90-100	LAPIS GURU
B	80-89	LAPIS EXPERT
C	60-79	LAPIS MASTER
D	30-59	LAPIS EXPLORER
E	0-29	LAPIS STARTER

Tablo 5. LAPIS Sürüm Bazında Değerlendirme (Planlama)

Sürüm	Puan	Kategori	Rozet
R1	66.75	C	LAPIS MASTER
R2	90.93	A	LAPIS GURU
R3	87.93	B	LAPIS EXPERT
R4	87.21	B	LAPIS EXPERT
R5	89.40	B	LAPIS EXPERT

Tablo 6. LAPIS Değerlendirme Sonucu

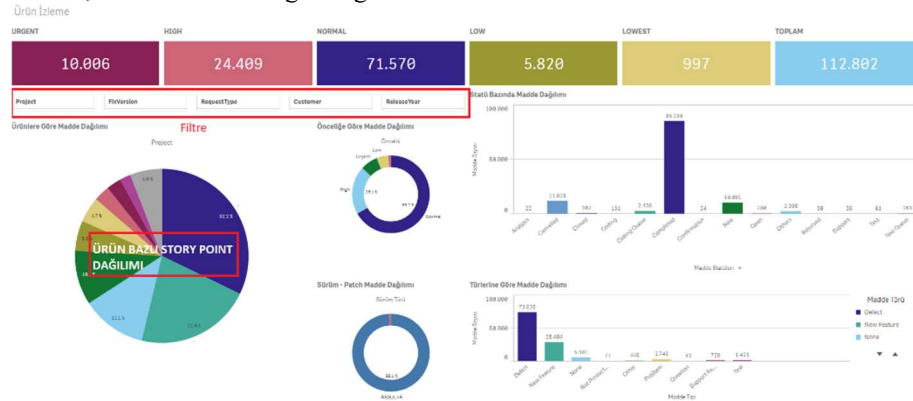
Tüm Yıl Sonucu	Puan	Kategori	Rozet
Planlama	84.45	B	LAPIS EXPERT
Ürün	33.16	D	LAPIS EXPLORER

6 Ölçümlerin Rollere Dağılımı

LAPIS ölçümlerinin genel sonuçları Bölüm 5’te verilmiştir. Tablo 1’de belirlenen hedef ve metriklerin rol bazlı olarak detaylı analiz edilebilmesi, belirlenen hedeflere daha kısa çevrimler ile ulaşılabilmesi, sürüm devam ederken fark edilen sorunlara proaktif olarak çözüm üretilebilmesi için çeşitli detay ölçümler hazırlanmıştır. Bu ölçümlere ait örnek raporlar aşağıda anlatılmıştır.

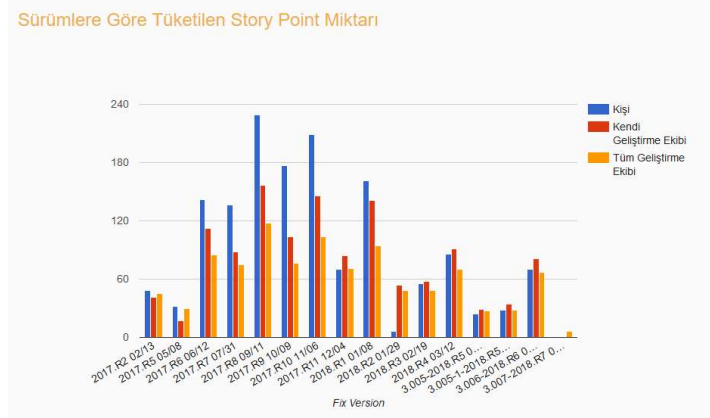
6.1 Kişisel Raporlar

Ekiplerin çevik yazılım geliştirme metodlarına uyum sağlayabilmesi için süreçte kendilerine özgü takip edebileceği [17] ve ekipçe iyileştirme fırsatı bulacağı raporlar oluşturulmuştur. Bu raporlar ile ekip yöneticileri sürümleri ve ekibi detaylı olarak analiz edebilmektedir. Şekil 6’da verilen raporda ürün bazlı olarak sürümlerdeki iş listesinin durumları, öncelikleri vb. değerler gösterilmektedir.



Şekil 6. Ekip Bazlı Raporlar

Yetkisi dahilinde ekip üyelerinin kendine ait bilgilerini görebildiği raporlar Şekil 7’de verilmiştir. (Örnek: Story Point Tüketimi, Ekip Ortalamasına göre kendi durumu gibi)

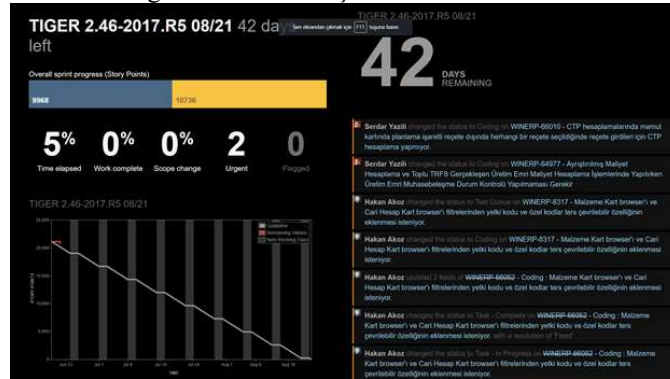


Şekil 7. Kişi Bazlı Raporlar

6.2 Ürün ve Süreç Raporları

Şirket içinde çevik yazılım süreçlerinin farkındalığını arttırmak ve ekipleri sürece dahil etmek için sürümün ilerleme durumunu gösteren raporlar tüm ekip tarafından izleyebileceği alanlarda bulunan ekranlarda paylaşılmaktadır.

Şekil 8'de örnek ekran görüntüsü verilmiştir.



Şekil 8. Sürüm Takip Raporu

7 Sonuçlar ve Gelecek Çalışmalar

Ölçümler süreçteki tüm paydaşlara dikkat edilmesi gereken noktaları gösterir. Burada asıl önemli olan şirkete ve ürüne uygun doğru sürecin tanımlanması, ölçümlerin de bunun paralelinde yapılıp, iyileştirme için hızlı aksiyon alınmasıdır.

LAPIS ile birlikte, ölçülebilir ve geliştirilebilir bir sistem kurularak, kaynak yönetiminin kolaylaştığı, ihtiyaçların ve toplam faydanın hızlı bir şekilde belirlenebildiği bir süreç oluşturulmuştur. Bu sürecin değerlendirilmesi için ISO 15939 standardı temel

alınarak, Hedef Soru Metrik yaklaşımı ile oluşturulan değerlendirme yöntemi kullanılmış ve uçtan uca izlenebilirlik sağlanmıştır. Bu yaklaşımda hem görselleştirmenin hem de oyunlaştırmanın olumlu etkileri gözlenmeye başlanmıştır. Planlaması iyi giden bir ürünün iş listesinde (Product Backlog) bekleyen maddelerin ortalama bekleme süresinin hedef süreden farklı olduğu tespit edilerek, ekip tarafından önlem alınmıştır. Kapasite planlamasının daha verimli yapılabileceği tespit edilmiş ve bu konuda iyileştirme çalışmaları yapılmaya başlanmıştır. Örneğin, bu kapsama alınan ürünlerde verimli planlama oranı için 2017 ve 2018 Mart-Mayıs dönemleri karşılaştırıldığında %32 oranında artış gösterdiği gözlenmiştir.

Şeffaflığın ve öngörülebilirliğin artırılması sayesinde, paydaşlar arasındaki uyum ve buna bağlı verimlilik artışı sağlanmış, tüm bunların sonucunda da farklı bakış açılara sahip paydaşların ortak kalite amacına ulaşması için adım atılmıştır.

İlerleyen aşamalarda, yazılım iç kalite metriklerinin de ölçüm metoduna dahil edilmesi planlamaktadır. Sürecin kod kalitesi ile birleşmesi sayesinde hedeflerde oluşabilecek sapmaların daha erken belirlenebilmesi ve iyileşme noktalarının daha net şekilde tespit edilmesi amaçlanmaktadır.

Referans

1. Nurdan Canbaz, Feza.Buzluca, "Yazılım Kalitesi İçin Yinelemeli Ölçme Yöntemi", 4.Ulusal Yazılım Mimarisi Konferansı (UYMK'12), 27-29 Eylül, İzmir (2012).
2. Tuğrul Tekbulut, Ayhan İnal, Betül Doğanay, LAPIS – (LOGO Agile Process Improvement System,UYMS, 2015
3. DeMarco, T., "Controlling Software Projects: Management, Measurement, and Estimation,Book, Prentice Hall PTR (1986).
4. Standardization, I.O.f. ISO/IEC 15939:2002 Software engineering -- Software measurement process. Available from: <https://www.iso.org/standard/29572.html>.
5. Van Solingen, R., et al., Goal question metric (gqm) approach. Encyclopedia of software engineering (2002).
6. VersionOne, 12th Annual State of Agile Report (2018).
7. Beck, et al., "Agile Manifesto," 2001.<http://www.agilemanifesto.org/>
8. Taghi Javdani Gandomani and Mina Ziaei Nafchi, Agility Assessment Model to Measure Agility Degree of Agile Software Companies, Indian Journal of Science and Technology, Vol 7(7), 955–959 (2014).
9. Asif Qumer, Brian Henderson, Measuring Agility and Adoptability of Agile Methods: A 4-Dimensional Analytical Tool, International Conference Applied Computing (2006).
10. Agile Metrics in Action: Measuring and Enhancing the Performance of Agile Teams 1st Edition by Christopher W. H. Davis (Author)
11. Deborah Hartmann, Robin Dymond, Appropriate Agile Measurement: Using Metrics and Diagnostics to Deliver Business Value, (AGILE'06), (2006).
12. Jeanette Heidenberg, Marta Olszewska, Quantitatively measuring a large-scale agile transformation. Journal of Systems and Software, 117: p. 258-273 (2016).
13. Felipe Furtado, Andrea Zisman, IEEE 24th International Requirements Engineering Conference (RE), Beijing, China, (2016).
14. K. Schwaber and J. Sutherland, "The Scrum Guide," (2014).
15. JIRA. The leading product for Project and Issue Tracking, [https:// www.atlassian.com/software/jira/](https://www.atlassian.com/software/jira/) (2002)

16. CONFLUENCE. The leading product for Document Collaboration, [https:// www.atlassian.com/software/confluence/](https://www.atlassian.com/software/confluence/)(2003)
17. Software Development Metrics, 1st Edition by Dave Nicolette (Author)