

Yalın Yazılım Geliştirme: Sistemik Eşleme Çalışması

Gözde KARATAŞ, Çağatay ÇATAL

Bilgisayar Mühendisliği
İstanbul Kültür Üniversitesi
34156, İstanbul, TÜRKİYE
{g.karatas, c.catal}@iku.edu.tr

Özet. Bu çalışmada; Yalın Yazılım Geliştirme konusundaki yayınları sistemik olarak değerlendirebilmek üzere, Kanıta Dayalı Yazılım Mühendisliği alanında kullanılan bir yöntem olan Sistemik Eşleme Çalışması gerçekleştirildi. Çalışmanın asıl hedefi; Yalın Yazılım Geliştirme’de kullanılan araştırma yöntemlerini incelemek, konu ile ilgili yayınların ağırlıklı olarak indekslendiği veritabanlarını belirlemek, ortaya konulan araştırma sorularına yanıt bulmak ve gelecek çalışmalar ile ilgili öneriler sunmaktır. IEEE Explorer, ACM Digital Library, Science Direct ve Wiley veritabanlarında, son 10 yıl içerisinde yayımlanmış bilimsel çalışmalar incelenmiştir. Bu çalışma sonucunda, Yalın Yazılım Geliştirme topluluğu için aşağıdaki önerilerde bulunmaktadır: (1) Yalın Yazılım Geliştirme konusunda deneysel çalışmaların sayısı artırılmalı; (2) Yalın Yazılım Geliştirme prensiplerinin Çevik Süreçler ile birlikte kullanımı ile ilgili yapılacak araştırmaların sayısı artırılmalı; (3) Yalın Yazılım Geliştirme’nin uygulanabileceği farklı çalışma alanları saptanmalı ve daha rahat kullanılabilmesi için çeşitli altyapılar hazırlanmalıdır.

Anahtar Kelimeler: Kanıta dayalı yazılım mühendisliği, sistemik eşleme çalışması, çevik prensipler, yalın yazılım geliştirme.

1 Giriş

Yalın Üretim (Lean Manufacturing), Toyota Ürün Geliştirme grubu tarafından ortaya konulmuş bir yaklaşımdır. Yalın Üretim Sistemi, müşterilerin talebinin en az kaynakla, en kısa zamanda, en ucuza ve eksiksiz olarak karşılanmasını amaçlayan bir sistemdir [1]. Tüm bu olumlu durumlar nedeniyle Yalın Üretim, otomotiv sektörünün dışındaki diğer alanların da dikkatini çekmiş ve farklı disiplinlere uygulanmaya başlanmıştır [2]. Yalın Üretim aşağıda verilen 7 ilkeye dayanmaktadır:

1. İsrafi giderme (Eliminate waste)
2. Bütünü optimize etme (Optimize the whole)
3. Kalite ile geliştirme (Build quality in)
4. Devamlı öğrenme (Learn constantly)
5. Hızlı teslim (Deliver fast)

6. Takımı güçlendirme (Empower the team)
7. Mümkün olduğunca geç karar verme (Decide as late as possible)

Bu ilkeler aşağıda açıklanmaktadır:

1. *İsrafi giderme*: Sistemin en önemli ilkesidir. Müşterinin talebini öncelikli tutar ve bu talebi daha etkili kullanabilmeyi hedefler [3]. Örneğin yazılım geliştirmede; müşterinin talep etmediği, bir diğer ifadeyle zorunlu olmayan gereksinimlerin gerçekleşmesi israfa sebep olur.
2. *Bütünü optimize etme*: Bu ilke, geliştiricilere sistemi bir bütün olarak araştırmalarını ve anlamalarını önerir. Örneğin, yazılım geliştirmede sadece test fazını iyileştirmeye odaklanılmamalı, bu iyileştirmenin tüm yazılım geliştirmede etkisinin ne olacağı değerlendirilmelidir. Bu ilke için her seviyede ölçüm kritik bir unsur olarak karşımıza çıkmaktadır [7].
3. *Kalite ile geliştirme*: Yazılım geliştirmenin farklı aşamalarında kullanılan gözden geçirme (inspection) aktiviteleri, kalite ile geliştirme ilkesine fayda sağlamaktadır, israf olarak değerlendirilmemelidir. Çevik geliştirmede kullanılan Test Güdümlü Geliştirme ve Eşli Programlama pratikleri, bu ilkeye katkı sağlamaktadır [6].
4. *Devamlı öğrenme*: Yazılım geliştirme aktivitesi, teknolojinin devamlı olarak gelişmesi nedeniyle sürekli şekilde yeni bilgiler öğrenmeyi gerektirmektedir. Bu ilke, bilginin önemini vurgulamaktadır.
5. *Hızlı teslim*: Mümkün olduğunca hızlı bir şekilde yazılım geliştirmeyi ve müşteriye teslimi amaçlar. Bu
6. *Takımı güçlendirme*: Bu ilke, geliştiricilerin birbirlerine güvenmeleri, saygı duymaları ve birlik içerisinde çalışmalarını gerektiğini vurgular.
7. *Mümkün olduğunca geç karar verme*: Bu ilke, geliştiricilere çok hızlı şekilde karar almak yerine uzun uzun düşünerek karar vermelerini önermektedir. Çok erken verilen kararlar, çoğu durumda hataya neden olabilmektedir.

Yukarıda bahsedilen ilkelerin en temel amacı, üretimin ve kalitenin başarılı bir şekilde arttırılmasıdır [5]. Yalın Yazılım Geliştirme kavramı, yazılım mühendisliği alanına bu Yalın İlkelerin uygulanması olarak açıklanabilir. Bu yaklaşım; benzetim oyunları, modelleme, haberleşme yazılımı ve sağlık sistemleri gibi birçok alanda uygulanmaktadır. Bu çalışmada, Yalın Yazılım Geliştirme konusunda yayımlanmış çalışmaları sistematik olarak sınıflandırmak için Kanıta Dayalı Yazılım Mühendisliği alanı içerisindeki Sistematik Eşleme Çalışması gerçekleştirilmiştir. Literatürde Yalın Yazılım Geliştirme ile ilgili çok fazla makale ve bildiri bulunmasına rağmen Sistematik Eşleme Çalışması yapan bir yayına ulaşılmamıştır. Dolayısıyla bu alandaki açığı kapatmanın faydalı olacağı değerlendirilmektedir. Bu amaçla, farklı araştırma soruları ortaya konulmuş ve bu sorulara yanıtlar aranmıştır. Sonraki aşamada, bu konuda kapsamlı bir Sistematik Literatür Taraması gerçekleştirilerek Yalın Yazılım Geliştirmenin faydalarının deneysel olarak gösterildiği yayınlara ulaşılmaya çalışılacak ve son olarak, farklı yazılım firmaları temsilcilerinden odak gruplar oluşturularak birebir görüşmeler gerçekleştirilip sonrasında ilgili anketler ile pratikte

yaşanan sorunlar tespit edilmeye çalışılacaktır. Bölüm 2’de araştırma soruları, araştırma yöntemi ve kullanılan dahil etme ve dışlama kriterleri açıklanmaktadır. Bölüm 3’de sonuçlar ayrıntılı olarak sunulmaktadır. Bölüm 4’de sonuç verilmektedir.

2 Araştırma Yöntemi

1990’lü yılların başında, tıp alanında birçok araştırmacı Kanıta Dayalı Tıp (evidence-based) yaklaşımını kullanmaya başlamıştır [1]. Bu yaklaşımın başarısı, bu yaklaşımı kullanma konusunda birçok araştırmacıya cesaret vermiş ve farklı alanlarda kısa zamanda uygulama imkanı doğmuştur. Kanıta Dayalı Yazılım Mühendisliği (KDYM) ilk olarak 2004 yılında ortaya konulmuştur. KDYM, birincil çalışmalarından deneysel kanıtlar toplar ve Sistematik Literatür Taraması (Systematic Literature Review-SLR) veya Sistematik Eşleme Çalışması (Systematic Mapping Study) ile araştırmayı gerçekleştirir. Bu çalışmada, Yalın Yazılım Geliştirme konusunda Sistematik Eşleme Çalışması gerçekleştirmek üzere, konu ile ilgili son 10 yılda yayımlanmış makaleler, konferans bildileri ve kitap bölümleri incelenmiştir. Araştırma yönteminin adımları sırasıyla; Planlama, Yürütme ve Raporlama olarak belirlenmiştir.

Planlama

Yalın Yazılım Geliştirmedeki birincil çalışmaları belirleyebilmek için aşağıdaki araştırma soruları belirlenmiştir:

- **AS1:** En çok kullanılan araştırma teknikleri nelerdir? (teori, anket, deney, deneyim, inceleme kategorilerinden birisi)
- **AS2:** Elektronik veritabanlarından hangisi yalın yazılım geliştirme konusunda daha fazla ilişkili yayın içermektedir?
- **AS3:** Hangi dergi ve konferanslarda Yalın Yazılım Geliştirme konusundaki yayınlar daha fazla yer bulmuştur?
- **AS4:** Yalın ve Çevik tekniklerin birleşiminden oluşan çalışmaların dağılımı nasıldır?
- **AS5:** Yayınların yıllara göre dağılımı nasıl gerçekleşmektedir?
- **AS6:** Yayın tiplerine göre çalışmaların dağılımı nasıldır?

Yayınlarla ulaşmak için aşağıdaki anahtar sözcükler kullanılarak yarı-otomatik bir arama yapılmıştır: *Lean Principles, Principles of Lean, Lean, Lean Software Development, Lean Development, Lean Software Analysis, Lean Software Practices, Lean Practices.*

Yayın aramaları için kullanılan veritabanları, ACM Digital Library, IEEE Explorer, Science Direct ve Wiley’dir. Bazı yayınlara birden fazla veritabanında karşılaşıldığı için, bu yayınlardan sadece bir tanesi seçilerek kullanılmıştır. Bu çalışma için 2003 ile 2013 yılları arasında yayımlanmış yayınlar incelenmiştir. Yalın Yazılım ile ilgili olmayan yayınlar inceleme kapsamından çıkarılmıştır. İçerme (İK) ve dışlama kriterleri (DK) aşağıda belirtilmektedir:

İK1: Yalın Yazılım Geliştirme konusundaki birincil çalışmalar.

İK2: Yalın Yazılım Geliştirme konusundaki ikincil çalışmalar.

İK3: Yalın Yazılım ve Çevik Yazılımın birlikte ele alındığı çalışmalar.

DK1: Yalın Yazılım Geliştirme ile ilgili olmayan çalışmalar.

DK2: İngilizce yazılmamış yayınlar.

DK3: Sadece özet bilgisi olan yayınlar.

DK4: Kısa notlar, paneller ve poster oturum özetleri.

Yürütme

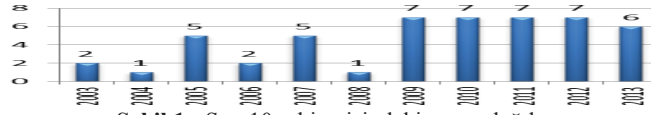
Elektronik veritabanları üzerinde çeşitli aramalar gerçekleştirilmiş ve ilgili yayınları tespit edebilmek için İK ve DK kriterleri kullanılmıştır. Dışarma kriterleri uygulandıktan sonra tespit edilen 50 yayın bu çalışma kapsamında kullanılmıştır.

Raporlama

Araştırma sorularından yola çıkarak, her bir soruya yanıt oluşturabilmek üzere gerekli raporlama gerçekleştirilmiş ve sonuçlar değerlendirilmiştir.

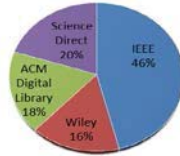
3 Sonuçlar

Şekil 1 yayınların yıllara göre dağılımını göstermektedir. İncelenen çalışmaların çoğunluğunun 2009 ve 2013 yılları arasında basılmış olduğu gözlemlenmiştir. Bu sayısal veriye göre, Yalın Yazılım Geliştirme ile ilgili araştırmaların 2009 yılından itibaren artmaya başladığı ifade edilebilir.



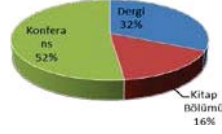
Şekil 1: Son 10 yıl içerisindeki yayın dağılımı

Elektronik veritabanlarında yapılan aramalarda, 65 yayın bulunmuş ve 15 tanesi çalışma konusuyla doğrudan ilgili olmadığından incelemeye dahil edilmemiştir. Şekil 2, inceleme için kullanılan yayınların veritabanlarına göre dağılımını göstermektedir. Bu sayısal verilere göre, konuyla ilişkili en çok yayının bulunduğu veritabanı IEEE Explorer'dır.



Şekil 2: Veritabanlarından kullanılan yayınların dağılımı

İncelemede kullanılan yayınların büyük bir kısmı konferans bildirisidir. Şekil 3, dergi yayınları, konferans bildirileri ve kitap bölümü türlerinin dağılımını göstermektedir. Bir simülasyon konferansı olmasına rağmen “Winter Simulation Conference” en çok yayın içeren konferans ve “The Journal of System and Software” en çok yayının bulunduğu dergidir. Tablo 1 dergi dağılımını; Tablo 2, konferans bildiri dağılımını göstermektedir.



Şekil 3: Kullanılan yayınların dağılımı

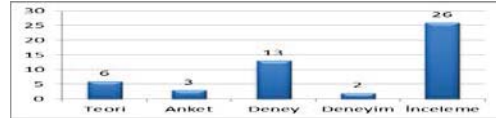
Tablo 1: Dergi yayınları dağılımı

Dergi İsmi	Yayın Sayısı
Journal of Systems and Software	3 (Sıra 1)
Journal of Operations Management	2 (Sıra 2)
Computers in Industry	2 (Sıra 2)

Tablo 2: Konferans yayınları dağılımı

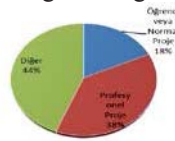
Konferans İsmi	Yayın Sayısı
Winter Simulation Conference	5 (Sıra 1)
Portland International Center for Management of Engineering and Technology	2 (Sıra 2)
Int'l Conference on Information and Communication Technologies and Development	2 (Sıra 2)
International Conference on Software and System Process	2 (Sıra 2)
Euromicro Conf. on Software Eng. and Adv. Applications	2 (Sıra 2)

Araştırma tekniklerine göre yayınlar sınıflandırılmıştır. Yayınların çoğunun araştırma tekniğinin, inceleme olduğu gözlemlenmiştir. Şekil 4 araştırma teknikleri dağılımını göstermektedir.

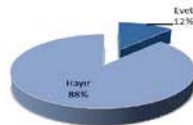


Şekil 4: Araştırma tekniklerinin dağılımı

Şekil 5 çalışma alanlarında yayınların dağılımını göstermektedir.



Şekil 5: Çalışma alanlarının dağılımı



Şekil 6: Çevik ve Yalın tekniklerinin birlikte kullanımı

Son yıllarda, Çevik Yazılım Geliştiriciler, Yalın Yazılım Geliştirme yaklaşımlarını araştırmaya ve uygulamaya başlamışlardır [4]. Bu nedenle, Çevik Yazılımın Yalın İlkeleri ile birlikte kullanıldığı yayınlar incelenmiştir. Şekil 6, iki yaklaşımın birlikte kullanıldığı çalışma yüzdesinin sadece %12 düzeyinde olduğunu göstermektedir.

4 Sonuç

Çalışmada dört elektronik veritabanı kullanılmış, 65 yayın bulunmuş ancak 50 tanesi çalışmada kullanılmıştır [8]. Araştırma sonucundaki öneriler aşağıda sunulmaktadır:

- (1) Yalın Yazılım Geliştirme konusunda deneysel çalışmaların sayısı artırılmalı,
- (2) Yalın Yazılım Geliştirme prensiplerinin Çevik Süreçler ile birlikte kullanımı ile ilgili yapılacak araştırmaların sayısı artırılmalı,
- (3) Yalın Yazılım Geliştirme'nin uygulanabileceği farklı çalışma alanları saptanmalı ve daha rahat kullanılabilmesi için çeşitli altyapılar hazırlanmalıdır.

Şekil 4 incelendiğinde, araştırma tekniklerinden deneysel çalışmaların yetersiz olduğu görülmektedir. Bu kapsamda Yalın Yazılım geliştirme ile deneysel çalışmaların artırılması madde 1'de önerilmiştir. Şekil 6'da görüldüğü gibi Yalın Yazılım Geliştirme ile Çevik Geliştirme çok az çalışmada birlikte kullanılmıştır. Son olarak yaptığımız incelemelerde, sınırlı sayıda alanda şu anda uygulandığı ve bunun geliştirilmesi gerektiği, uygulamada kolaylık sağlanması için bazı altyapı yazılımlarının da hazırlanması gerektiği sonucuna varılmıştır. Bu çalışma bir başlangıç niteliğindedir. Devamında Sistematik Literatür Taraması ile deneysel kanıtlara ulaşmaya çalışılacak ve son aşamada yapılacak olan anketlerle uygulamadaki sorunlar tespit edilmeye çalışılacaktır.

Referanslar

- [1] Kitchenham, B. A., Dyba, T., and Jorgensen, M., "Evidence-Based Software Engineering", Proc. of the 26th International Conference on Software Engineering (ICSE '04), Scotland, UK, pp. 273-281.
- [2] Poppendieck, M., Lean Software Development, Addison Wesley, 2003.
- [3] Ali, A., Wong, Y. C., Wong, K. Y. "Key Practice Areas of Lean Manufacturing", International Association of Computer Science and Information Technology - Spring Conference, 267-271, 2009.
- [4] Wang, X. Conboy, K. Cawley, O. "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. Journal of Systems and Software. 85(6), pp.1287-1299.
- [5] Bocock, L., Martin, A, There's Something About Lean. Agile Conference, pp.10-19, 2011.
- [6] https://www.ibm.com/developerworks/community/blogs/ambler/entry/principles_lean_software_development?lang=en
- [7] <http://agile.dzone.com/news/optimize-whole-measure-all>
- [8] <https://dl.dropboxusercontent.com/u/80943636/YalinYazilimGelistirme-incelenen-yayinlar.docx>