

Yazılım Mühendisliği Araştırmalarına Öz Çerçeve Yaklaşımı

Murat Paşa Uysal¹, Görkem Giray²

¹ Yönetim Bilişim Sistemleri Bölümü, Başkent Üniversitesi, Ankara, Türkiye
muysal@baskent.edu.tr

² Bağımsız Araştırmacı, İzmir, Türkiye
gorkemgiray@gmail.com

Özet. Yazılım Mühendisliği (YM) alanında tasarlanan ya da gerçekleştirilen bilimsel çalışmaların çeşitli boyutlar ve değişkenler açısından ortak bir temelde karşılaştırılmalarının, bilimsel alana olduğu kadar endüstriye de önemli katkıları olabileceği düşünülmektedir. Çalışmamızda, Öz Çerçeve (ÖÇ) (Essence Framework) Tabanlı Araştırma Tasarımı ve Değerlendirme Modeli (ÖÇTATDM) geliştirilmiş ve bu model YM'deki deneysel bir araştırmanın çözümlemesinde kullanılmıştır. Araştırmamız, Sistematik Tarama (ST) ve Nitel Araştırma Yöntemleri kullanılarak aşamalı olarak yürütülmüştür. Birinci aşamada, ST ile YM alanındaki bilimsel çalışmalara ulaşılmış ve ikinci aşamada ÖÇTATDM geliştirilmiştir. Üçüncü aşamada, geliştirilen model kullanılarak çevik yazılım geliştirmeyle ilgili deneysel bir YM araştırması modellenmiştir. Yer ve kapsam sınırlılığından dolayı bazı çözümler çalışma dışında tutulmuş ve gelecekteki çalışmalara bırakılmıştır. İlk izlenimlerimiz, ÖÇ'nin YM araştırma alanı ve endüstri arasında bir köprü görevini görebileceği gibi aynı zamanda öteden beri dile getirilen kuram ve uygulama arasındaki temel sorunlara da ışık tutabileceği yönündedir.

Anahtar Kelimeler: Yazılım mühendisliği, Araştırma yöntemi, Öz Çerçeve (Essence Framework)

An Essence Framework Approach to Software Engineering Research

Abstract. It is thought that the comparison of studies designed and conducted in the research field of Software Engineering (SE) on a common basis in terms of various dimensions and variables may bring important contributions to the industry as much as to the scientific world. In this study, therefore, we develop an Essence Framework-based Research Design and Evaluation Model (EFBRDEM), and then apply this model to the evaluation of an empirical research in SE. Our study is conducted according to the guidelines of Systematic Review and Qualitative Research. At first, we systematically review the related

literature, and then develop the EFBRDEM. Finally, an empirical research on agile SE methods is modeled using the EFBRDEM. The first impressions are such that the proposed model can serve as a bridge between the research field and the industry as well as it can shed light on long-standing fundamental problems between the theory and SE practices.

Keywords: Software Engineering, Research Method, Essence Framework

1 Giriş

Yapılan arařtırmalar, yeni ve özgün bir yazılım ürününün yaygın olarak kullanımı ve kabulü için kavramsal tasarım, daha sonra temel arařtırma, ürün geliştirme vb. bir dizi ařamalardan geçirilmesi gerektiğini göstermekte ve süreç uzun yıllara yayılabilmektedir [1, 2]. Bu kapsamda, yazılım mühendisliđi (YM) arařtırmaları ile endüstrideki uygulamalar arasındaki bađın güçlendirilmesi amacıyla çeřitli çalışmalar ortaya konulmuřtur [3]. Çalışmalarda, YM alanındaki arařtırmacılar ve uygulayıcılar ortak bir çerçevede bir araya getirilmeye çalışılarak yazılım geliştirme süreçlerindeki verimliliđin ve etkinliđin artırılması hedeflenmektedir [4]. Bu sürecin önemli bir boyutunu dođal olarak YM arařtırma alanı oluřturmaktadır. Konuyla ilgili yapılan literatür arařtırmaları ve çeřitli çalışmalar [5, 6], YM endüstri uygulamaları ile bilimsel arařtırma projelerinin bütünleřtirilmesi ve iřbirliđinin sađlanabilmesi için her iki alanda önemli mesafelerin kat edilmesi gerektiğini göstermektedir [7].

Yazılım, hem süreç hem de ürün kapsamında ele alındığında insan, teknoloji ve süreç bileřenlerinden oluřan sosyo-teknik bir yapıdır. Diđer mühendislik disiplinlerinin aksine aynı takım, aynı yazılım geliştirme yöntemi, araç ve teknikler kullanarak aynı sonucu almak her zaman mümkün olmayabilmektedir. Çeřitli nedenlerden dolayı (projenin iptali, gecikmesi, bütçe ařımı vb.) her yıl çok sayıda yazılım projesi başarısızlıkla sonuçlanırken büyük kaynaklar israf edilmekte ve bu konuya çeřitli raporlarda değinilmektedir [14]. Ancak, söz konusu duruma kalıcı çözümler getirebilmek, endüstri ve arařtırma alanını bütünleřtirebilmek yine kaliteli bilimsel çalışmalarla mümkün olabilecektir. Ayrıca, YM alanındaki bilimsel çalışmalara ait sonuçların mümkün olduđu kadar genellemesi ve arařtırma problemlerinin de gerçek hayat yazılım problemleriyle örtüřmesi beklenmektedir. Bunun ise YM'deki bilimsel çalışmaların, (a) birbirlerini tamamlaması ve (b) bu arařtırmaların içerdikleri yazılım geliştirme yöntemleri, uygulamalar, araçlar, olgunluk düzeyi vb. deđiřkenler açısından karşılařtırılmasıyla mümkün olacađı düşünölmektedir. Bu durum, özellikle yazılım geliştirme problemlerini içeren bilimsel arařtırmaların, ortak bir temelde formal yöntemler kullanılarak gösterimini, modellenmesini ve çözümlenmesini sađlayacak yeni yaklařımları gerektirmektedir.

Söz konusu probleme çözüm olarak önerilen çalışmalardan birisi de SEMAT (Software Engineering Method and Theory) girişimi ve organizasyonu tarafından geliştirilen Öz Çerçeve'dir (ÖÇ) (Essence Framework) [3]. YM literatürü incelendiğinde, ÖÇ ile YM'deki arařtırma yaklařımlarıyla iliřkili ve yukarıda bahsedilen problemlere çözüm arayan çalışmaların sınırlı olduđu gözlenmektedir [15]. Bu amaçla çalışmamızda, Öz Çerçeve Tabanlı Arařtırma Tasarımı ve Deđerlendirme Modeli

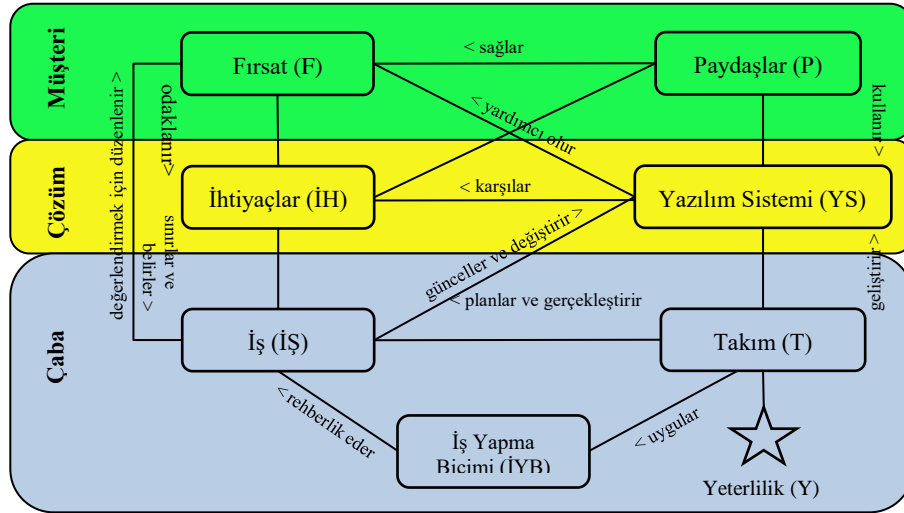
(ÖÇTATDM) geliştirilmiş ve bu model YM'deki deneysel bir araştırmanın modellenmesinde kullanılmıştır. Çalışmamızın YM araştırma alanına olan katkılarını aşağıdaki gibi sıralamak mümkündür:

- (a) YM alanındaki bilimsel çalışmaların ortak bir çerçevede modellenmesine, tasarımına ve değerlendirilmesine olanak tanıyacak bir modelin geliştirilmesi,
- (b) Söz konusu modelin, deneysel ve özellikle çevik yazılım geliştirmeyi içeren araştırmalara uygulanması sonucunda, gelecekteki YM bilimsel araştırmalarının çok boyutlu çözümlemesini sağlayacak bir girişimi başlatmasıdır.

Bildirinin sonraki bölümlerini, çalışmanın kuramsal temeli, araştırma yöntemi ve geliştirilen modele ilişkin uygulama oluşturmaktadır.

2 Öz Çerçeve (Essence Framework)

ÖÇ ile yazılım geliştirmede kullanılan yöntemler ve uygulamalar, kavramsal olarak bir çekirdek içerisinde formal biçimde ele alınmakta, kendine özgü alan bağımlı görsel ve metin tipi bir dil ve kavramsal yapılarla ortak bir temelde bütünleştirilmektedir [6]. Böylece, (a) YM çalışma alanındaki araştırmalar ve endüstri uygulamaları arasındaki uyumsuzluğun giderilmesi, (b) çevik yaklaşım doğrultusunda yazılım geliştirme yöntemlerinin ve uygulamalarının dinamik yapıda oluşturulması, (c) yazılım süreçlerinin sağlıklı bir şekilde izlenmesi ve (d) YM yöntemlerinin ve uygulamalarının ortak bir temelde bütünleştirilmesine olanak tanınması hedeflenmektedir [18].



Şekil 1. Öz Çerçeve Çekirdeğindeki Alfa'lar ([3]'ten uyarlanmıştır)

Şekil 1'de görüldüğü gibi YM bilgi alanı, Çekirdek (Kernel) olarak ifade edilen, Müşteri (Customer), Çözüm (Solution) ve Çaba'dan (Endeavor) oluşan üç ayrı ilgi alanı biçiminde organize edilmiştir. Çekirdek'in ana bileşenleri olan Alfa'lar (Alphas), Et-

kinlik Uzayları (Activity Spaces) ve Yetkinlikler (Competencies) bu ilgi alanları içerisinde yapılandırılmaktadır. Alfa'lar, her yazılım geliştirme projesinde ele alınması gereken önemli boyutları temsil etmektedir (fırsat, paydaşlar, ihtiyaçlar, yazılım sistemi, takım, iş yapma biçimi, iş). Alfa'lar aynı zamanda yazılım geliştirme süresince projenin ilerleme ve gelişimini gösteren farklı seviyedeki durumları (state) içermektedir. Etkinlik Uzayları ise YM'nin faaliyet tabanlı bir görünümünü sağlayarak yazılım geliştirme etkinliklerini ilgi alanlarına göre gruplayarak kümelere ayırmaktadır. Yeterlilikler, YM etkinliklerinin yerine getirilebilmesi için edinilmesi gereken yetenekleri göstermektedir. Çekirdeği oluşturan Alfa'lar, Etkinlik Uzayları ve Yetkinlikler üç ayrı ilgi alanına göre sınıflandırılmıştır. ÖÇ ile ilgili detaylı açıklamalar yer kısıtlamasından dolayı bu bildiride verilememiştir (Bakınız [3]).

3 Yöntem

Bu çalışmanın amacı, YM araştırma alanına yönelik bir Öz Çerçeve Tabanlı Araştırma Tasarımı ve Değerlendirme Modeli (ÖÇTATDM) geliştirmektir. Bu amaç doğrultusunda araştırmamız, Sistematik Tarama (ST) ve Nitel Araştırma Yöntemleri kullanılarak aşamalı olarak yürütülmüştür. Birinci aşamada, ST ile YM alanındaki bilimsel çalışmalara ulaşılmış, ikinci aşamada ÖÇTATDM geliştirilmiştir. Üçüncü aşamada geliştirilen model kullanılarak çevik yazılım geliştirmeyle ilgili deneysel bir çalışma modellenmiştir. Araştırma soruları ve yöntemini içeren araştırmamızın deseni tasarlanırken Shaw'un [6] YM araştırmaları için önerdiği ölçütler dikkate alınmıştır. Çalışmamızın; (a) SCI'de taranan makalelere ait çözümleme bilgisini sunması ile (b) ÖÇTATDM olmak üzere iki ana çıktısını bulunmaktadır. Bunlara yönelik araştırma soruları aşağıdaki gibidir:

AS-1: YM alanında yazılım geliştirme ve değerlendirmesiyle ilgili en çok atıf alan çalışmalar nelerdir?

AS-2: YM alanındaki bilimsel çalışmalar ortak bir çerçevede nasıl modellenebilir ve değerlendirilebilir?

Bildirinin bundan sonraki bölümlerini oluşturan birinci ve ikinci aşama AS-1'i, üçüncü aşama ise AS-2'i cevaplamaya yöneliktir.

3.1 Birinci Aşama: Sistematik Tarama

Veri Toplama. Araştırmaya başlamadan önce çalışmanın sistematikliğini, yansızlığını, geçerlik ve güvenilirliğini sağlamak amacıyla bir tarama protokolü geliştirilmiştir. Bu protokol, Sistematik Literatür Taramasında (SLT) olduğu gibi detaylı olmasa da araştırma sorularına yönelik arama deyimlerini, yayınların seçim ölçütleri ile bunların ulaşılan kaynaklara nasıl uygulanacağını, veri depolama ve tasnif yöntemini, kaynaklardan hangi veri ve bulguların alınacağı ve çalışma takvimini içermiştir [16]. YM ile ilgili birincil kaynaklara ulaşılmadan önce daha önce gerçekleştirilen benzer tarama çalışmalarındaki araştırma konusu ve sorularıyla ilgili konular incelenmiştir.

Veri Sorgulama ve Kaynaklar. Taranacak veritabanları belirlenirken çalışmanın amacı, araştırma soruları ve sınırlılıklar dikkate alınmıştır. Buna göre ana kaynak olarak

elektronik veritabanları ve arama motoru olarak Scopus'un kullanılması kararlaştırılmıştır. Araştırmacıların daha önceki deneyimleri, diğer arama motorlarıyla uyumlu sonuçlar vermesi ve makalelerdeki atıf sayısını güncel olarak sunması bu arama motorunun seçilmesinde dikkate alınan önemli ölçütlerdir. Tarama deyimleri ve operatörleri kullanılarak oluşturulan tarama cümleleri Tablo 1'de gösterilmiştir.

Tablo 1. Tarama cümleleri

Yazılım Çalışma Alanı	Tarama Deyimleri ve Operatörleri
Cümle-1 (Yazılım Analizi)	TITLE ("SOFTWARE ANALYSIS") AND SRCITITLE ("SOFTWARE") AND (LIMIT-TO (SUBJAREA,"COMP ")) AND (LIMIT-TO (LANGUAGE, "English "))
Cümle-2 (Yazılım Geliştirme)	TITLE ("SOFTWARE DEVELOPMENT") AND SRCITITLE ("SOFTWARE") AND (LIMIT-TO (SUBJAREA,"COMP ")) AND (LIMIT-TO (LANGUAGE, "English "))
Cümle-3 (Yazılım Analizi veya Yazılım Geliştirme)	TITLE-ABS-KEY ("SOFTWARE DEVELOPMENT") OR TITLE-ABS-KEY ("SOFTWARE ANALYSIS") AND SRCITITLE ("SOFTWARE") AND (LIMIT-TO (SUBJAREA,"COMP")) AND (LIMIT-TO (LANGUAGE, "English"))

Birinci tarama cümlesi kullanılarak en çok atıf alan "Yazılım Analizi" ile ilgili 35 çalışma, ikinci tarama cümlesiyle "Yazılım Geliştirme" ile ilgili 200 çalışma belirlenmiştir. Her iki sorgu sonucunda bazı çalışmaların örtüşmesinin görülmesi üzerine üçüncü tarama cümlesi kullanılmış, ulaşılan 9835 makale, Şekil 1'de gösterildiği gibi atıf sayısına göre en yüksekte en düşüğe göre sıralanmıştır.

The screenshot shows the Scopus search results page for the query: TITLE-ABS-KEY ("SOFTWARE DEVELOPMENT") OR TITLE-ABS-KEY ("SOFTWARE ANALYSIS") AND SRCITITLE ("SOFTWARE") AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO (LANGUAGE, "English")). The page displays 9,835 document results. The search results are sorted by 'Cited by (highest)'. The first two results are:

Document title	Authors	Year	Source	Cited by
1 A Metrics Suite for Object Oriented Design	Chidamber, S.R., Kemerer, C.F.	1994	IEEE Transactions on Software Engineering 20(6), pp. 476-493	2434
2 A validation of object-oriented design metrics as quality indicators	Basili, V.R., Briand, L.C., Melo, W.L.	1996	IEEE Transactions on Software Engineering	846

Şekil 2. Tarama çıktısı ekran görüntüsü

Çalışmaların Seçim Ölçütleri ve Sınıflandırılması. Tarama sonucunda elde edilen çalışmalardan ilgili olanlarının araştırmaya dâhil edilmesi veya dışarıda bırakılmasına yönelik ölçütler belirlenmiş, sınıflamalar bunlara göre gerçekleştirilerek yazılım geliş-

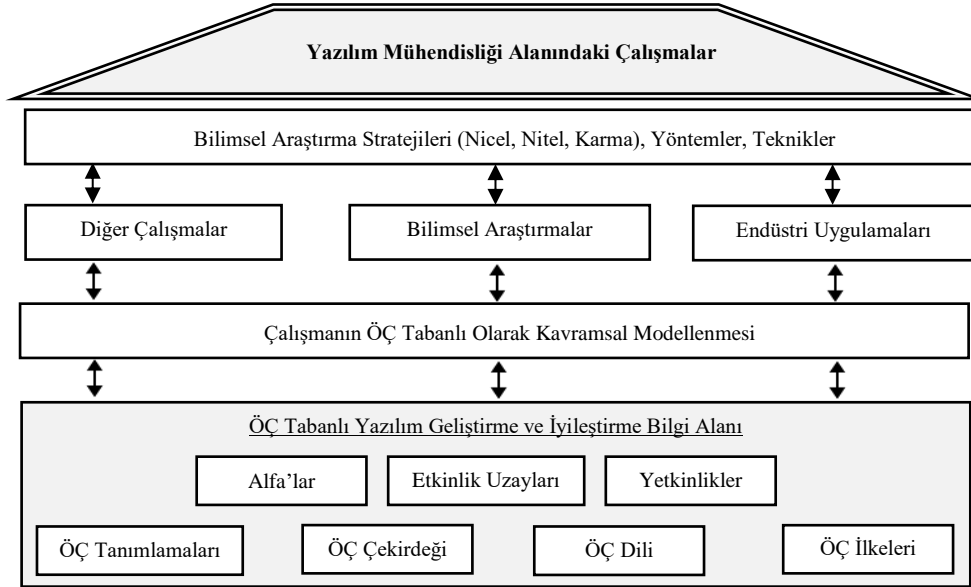
tirme ile ilgili çalışmalar iki ana başlık altında toplanmıştır. Birinci gruptakiler, çalışmaların nitelikleri, kullanılan araştırma yöntemleri ve yaklaşımlara göre oluşturulmuştur. İkinci gruptaki çalışmalar ise anahtar kelimeler, dayandıkları kuramlar, yazılım yöntemi, teknik, teknoloji, model vb. yapıları içermiştir. İncelenen çalışmaların, araştırma sorularıyla ilgi düzeyi, çalışma çıktıları, alana katkısı vb. faktörler birincil çalışmaların seçimlerine yönelik kararları desteklemiştir. İki araştırmacı toplanan çalışmaları incelemiş, araştırmaya dâhil edilip edilmeyeceklerini belirtilen bu ölçütlere göre karar vermiştir. Buna göre atıf sayısına göre sıralanan ilk 200 çalışma içerisinde, en yüksek atıf alan (a) “Deneysel Yazılım Geliştirme” ile ilgili 10, (b) “Yazılım Geliştirme Yöntemi” ile ilgili 5, (c) “Çevik Yazılım Geliştirme “ ile ilgili olarak 25 makale belirlenmiştir. Son olarak bu üç ölçütü aynı anda içeren ve en çok atıf alan bir makale [17] belirlenmiş ve ÖÇTATDM’nin uygulanmasında kullanılmıştır (Tablo 2).

Tablo 2. En çok atıf alan makale

Kaynakça Numarası	Makale
17.	Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J, “The impact of agile practices on communication in software development”, Empirical Software Engineering, 13 (3), 303-337, 2008.

3.2 İkinci Aşama: ÖÇ Tabanlı Bilimsel Araştırma Değerlendirme Çerçevesinin Geliştirilmesi

ÖÇTATDM geliştirilmeden önce yazarlar, ilk olarak YM araştırma alanındaki yazılım geliştirme yöntemleri ile bunları incelemede kullanılan araştırma yöntemleri, teknikleri ve araçlarıyla ilgili konuları belirlemeye çalışmışlardır [4]. Bu çalışmaların bir kısmını deneysel araştırmalar [8, 9], bir bölümünü saha/alan çalışmaları [10], önemli bir grubunu örnek olay yönteminin kullanıldığı araştırmalar [11] ve geri kalanını YM ile ilgili kuramsal çalışmalar oluşturmaktadır [12]. Bunlar arasında YM araştırmaları ile ilgili önemli problem sahalarına değinen araştırmalar da ön plana çıkmaktadır. Örneğin Glass vd., YM araştırma alanının araştırma stratejileri ve yöntemleri açısından dar kapsamlı olduğunu belirtmekte [13], Stol vd. YM araştırma yöntemleriyle ilgili terminoloji birliğinin olmadığını ve bu konuda görüş ayrılıklarının bulunduğunu ifade etmektedir [4]. Örnek Olay Yöntemi, bazı araştırmacılarca araştırma stratejisi olarak ele alınırken bazıları ise onu araştırma yöntemi olarak nitelemektedir [4]. Kimi çalışmalarda nitel araştırma yöntemi olarak ele alınırken deneysel yöntem olarak kullanıldığı çalışmaların da olduğu gözlenmektedir [9]. Dolayısıyla araştırma stratejileri, yöntemleri ve teknikleri ile ilgili kavramlar, YM alanında birbirlerinin yerine yanlış kullanılabilen kavramlar olarak karşımıza çıkmaktadır.



Şekil 3. Öz Çerçeve Tabanlı Araştırma Tasarımı ve Değerlendirme Modeli

Çalışmamızın bu aşamasında ilk olarak [4], [6], [8] ve [9] kaynakları incelenmiştir. Şekil 3'te gösterildiği gibi YM ile ilgili stratejiler, yöntem ve teknikler, bilimsel araştırmalar ve çalışmalar, ÖÇ'de yer alan kavramlar ve bileşenler kullanılarak modellen-

mektedir. Bundaki ana amaç, yazılım endüstrisindeki uygulamaları, akademik ve bilimsel çalışmaları ortak bir temel üstünde bütünleştirmektir. İlk olarak YM çalışmaları, Bilimsel Araştırmalar, Endüstri Uygulamaları ve Diğer Çalışmalar olmak üzere üç ana grupta toplanmaktadır. Daha sonraki katmanda söz konusu yazılım geliştirme problemlerini inceleyen çalışmalar ÖÇ doğrultusunda modellenmektedir. En alt katmanda ise ÖÇ kendisi ile onu oluşturan temel yapılar ve kavramlar bulunmaktadır. Ayrıca, literatürdeki çalışmalardan hareket edilerek bir Deneysel YM Araştırmasında bulunması gereken temel etkinlikler şu şekilde belirlenmiştir: (a) Araştırmanın Tanımlanması, (b) Araştırmanın Tasarımı, (c) Araştırmanın Yürütülmesi ve Veri Toplama, (d) Analiz, (e) Bulgular ve Yorumlamadır. Sadece (a), (b) ve (c) etkinlikleri ile bunların içerdiği kavramlar bildirinin sonraki bölümünde incelenmiş, araştırmanın sınırlılıkları dolayısıyla diğer araştırma etkinlikleri ise sonraki çalışmalara bırakılmıştır.

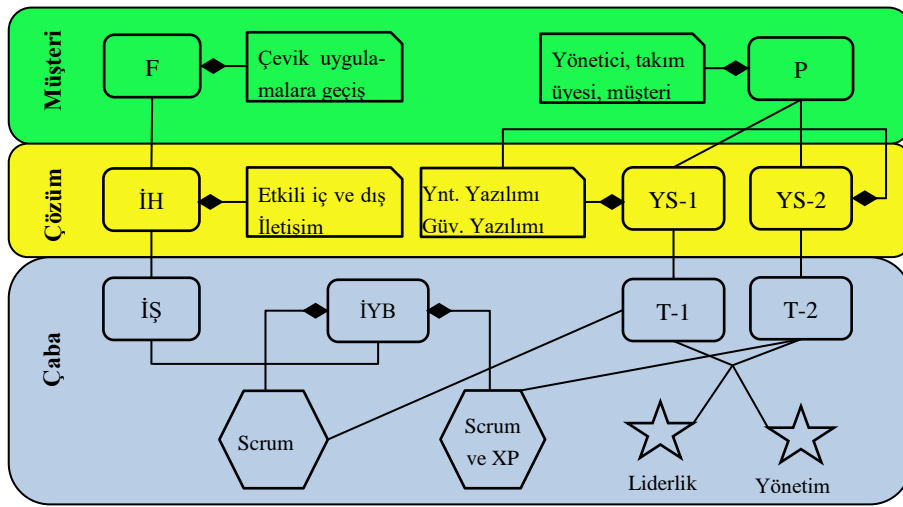
3.3 Üçüncü Aşama: YM Alanında Deneysel Bir Çalışmanın Değerlendirilmesi

[17]'deki çalışmaya ait araştırmanın ÖÇ tabanlı olarak tanımlanması, tasarımı ve yürütülmesine ait modelleme aşağıdaki bölümlerde açıklanmıştır.

Araştırmanın Tanımlanması ve Tasarımının ÖÇ ile Gösterilmesi. Bu çalışma kapsamında incelenen araştırma [17], Finlandiya'da güvenlik yazılımları geliştiren F-Secure şirketindeki bir örnek olay uygulamasını içermektedir. Şirket, geleneksel yazılım geliştirme yöntemlerini bırakarak çevik yöntemlere geçmek, performans ve verimliliği artırmak ve rekabet avantajını sağlamayı amaçlamaktadır. Söz konusu deneysel çalışmada [17], SCRUM ve XP (eXtreme Programming) çevik yazılım geliştirme uygulamaları kullanılmış farklı iki yazılım geliştirme takımının kendi içindeki iletişimi ile şirketin diğer paydaşları (müşteri, test takımı vd.) arasındaki iletişimi nasıl etkilediği araştırılmıştır. İç iletişim ve dış iletişim çalışmanın bağımsız değişkenlerini oluşturmuştur. Çevik uygulamaların bilgi transferini olumlu yönde etkilediği ve bunun da yazılım geliştirme süreçlerine katkıda bulunacağı araştırmanın temel kabulünü oluşturmaktadır. Çok popüler olmalarının yanında, XP'nin yazılım geliştirici, SCRUM'ın ise proje odaklı olması, yazılım geliştirme sürecine farklı yaklaşımlar benimsemeleri bu yöntemlerin araştırma için seçimindeki önemli ölçütlerdir. Yazarlar ayrıca, araştırmanın tasarımına temel teşkil eden Koordinasyon Kuramı, Çevik Uygulama vb. konularla ilgili kuramsal yapılara ait kavramları sunmuş ve bunların araştırma ile olan bağlantılarını ortaya koymuşlardır.

Şekil 4'te söz konusu çalışmada yer alan yazılım geliştirme süreciyle ilgili Alfa bileşenleri, ÖÇ Dilinin grafik söz dizim bileşenleri ile gösterilmiştir. Kullanılan grafik sembollerin çeşitli dil bilimsel ve kavramsal anlamları olup açıklamalar bu çalışmanın kapsamı dışında bırakılmıştır. Şekil 4'ü şu şekilde yorumlamak mümkündür: F: Teknolojik gelişmeler ve rekabet ortamı, güvenlik yazılımları geliştiren şirketin çevik yöntemlere geçişi için gereken durumları oluşturmuş ve fırsatları yaratmıştır. P: Şirket yöneticileri ve diğer aktörler bu fırsatı yaratan paydaşlar arasındadır. İH: Yazılım ürünlerinin yanında ihtiyaç duyulan diğer bir konu ise seçilecek çevik yöntemlerin takım içi iletişim ve diğer paydaşlar arasındaki dış iletişimi nasıl etkilediğinin belirlenmesidir.

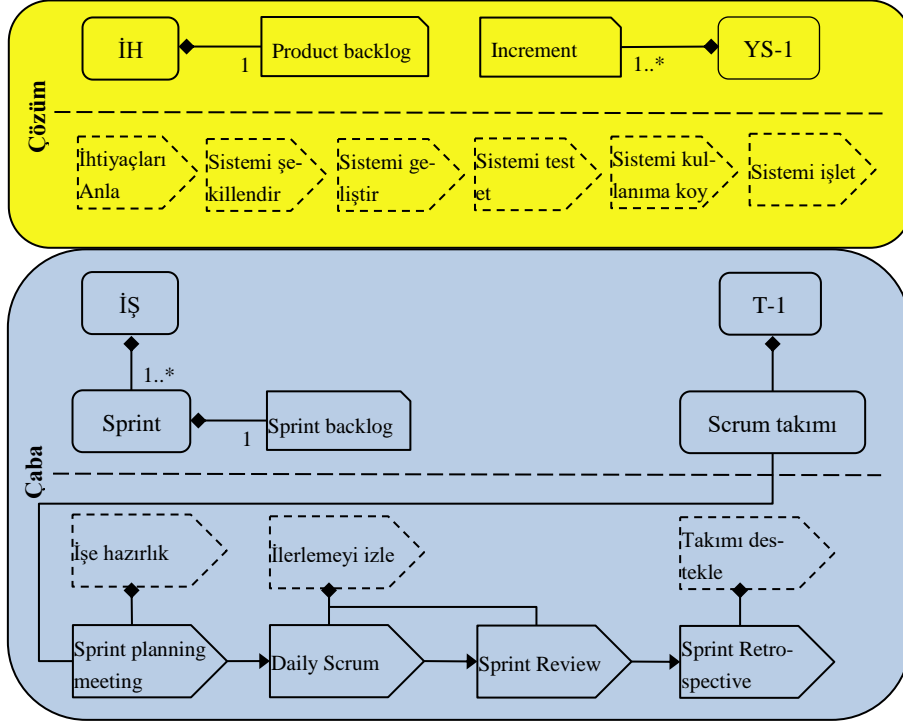
YS-1: Güvenlik Sistemleri Yönetim Yazılımı ile YS-2: Mobil Güvenlik Yazılımı araştırma kapsamında geliştirilecek iki yazılım sistemini oluşturmaktadır. T-1: Birinci yazılım takımını ve T-2: İkinci yazılım takımını temsil etmektedir. Çalışmada iş yapma biçimi (İYB) olarak farklı iki yazılım geliştirme yöntemi benimsenmiştir. Birinci takım yazılım geliştirirken Scrum'ı, ikinci takım ise Scrum ve XP'nin kendilerine uyarlanmış biçimini kullanmışlardır. Y: Liderlik ve Yönetim konularında her iki yazılım takımının sahip olmaları gereken yeterlilikler bulunmaktadır. Ancak, bu durum ilgili makalede [17] belirtilmemiş olup bu bildirinin yazarları tarafından modele eklenmiştir. İŞ: Her iki yazılım takımının kullanmış oldukları yönteme göre yapacakları işlerin detaylarıdır.



Şekil 4. Araştırma bileşenlerinin tanımlanması ve tasarımının modellenmesi

Araştırmanın Yürütülmesinin ÖÇ ile Gösterilmesi. ÖÇ Tanımlama Dokümanından [18] faydalanılarak birinci proje takımının kullandığı Scrum etkinlikleri ve bileşenleri Şekil 5'te örnek olarak gösterilmiştir. Kesikli çizgilerle gösterimi yapılan simgeler bildirinin önceki bölümlerinde bahsedilen ÖÇ Çekirdeğinde bulunan Etkilik Uzaylarını temsil etmektedir. Yazılım sistemleri bu uzaylarda (müşteri, çözüm ve çaba) yer alan ÖÇ etkinlikleri ile geliştirilmektedir. Çözüm ilgi alanında sistem geliştirme etkinlikleri bulunurken, Çaba ilgi alanında ise Scrum'a ait etkinlikler yer almaktadır. Birinci yazılım sistemini YS-1'in sahip olması gereken özellik ve işlemlere ait bilgiler, Alfa İH'nin bir türü olan "Product Backlog"larda tutulmaktadır. YS-1'in gelişimi her bir "Sprint" sonunda "Increment" olarak gerçekleştirilmektedir. Çaba ilgi alanındaki Alfa İŞ ise birden çok "sprint"ten oluşmakta ve bunlara ait bilgiler "Sprint Backlog"da bulunmaktadır. Alfa T-1 takımının gerçekleştirdiği etkinlikler ("Sprint Planning Meeting", "Daily Scrum", "Sprint Review", "Sprint Retrospective"), "İşe Hazırlık", "İlerlemeyi İzle" Takımı Destekle" Alfalarından türetilmiştir ("extend" edilmiştir). Öte yandan, ÖÇ'nin gösterim gücü hakkında fikir vermek amacıyla da Şekil 5'te yer alan alfa T-1 ile bir

Scrum etkinliđi, ÖÇ Dilinin Metin Tipi Söz Dizim Kuralları kullanılarak Şekil 6'da gösterilmiştir.



Şekil 5. Araştırmada kullanılan Scruma ait etkinliklerin modellenmesi

Şekil 5 ve Şekil 6'da görüldüğü gibi önceden gerçekleştirilen deneysel bir araştırma [17], ÖÇ Yaklaşımının içerdiği kavramsal yapı, bileşenler, grafik dili ve söz dizim dili araçları kullanılarak modellenebilmiştir. Buradan hareketle, ÖÇ'nin sadece YM endüstrisi uygulamaları için değil aynı zamanda YM araştırma alanına da katkı sağlayacak önemli bir potansiyeli olduğunu söylemek mümkündür.

```

// Birinci proje takımının alfa olarak tanımlanması -----
alpha T1_ScrumTakimi:
    "İnci yazılım takımı olan Scrum Takımı 6 kişiden oluşmaktadır. 1 Scrum master, 4 yazılımcı
ve 1 kalite mühendisi bulunmaktadır. Diğer paydaşları yöneticiler, ürünün sahibi, müşteriler, test
ekibi, dökümantasyon ekibi vb. aktörler oluşturmuştur."
    with states {
        state Olusturuldu {
            "Yazılım takımı oluşturulmuştur."
            checks {
                checkpoint c1 {"Ürünün sahibi belirlendi"}
                checkpoint c2 {"Yazılımcılar T-1 takımına atandı"}
                checkpoint c3 {"Scrum master belirlendi"}
                .....{.....}
            }
        }
    }

// Bir Scrum etkinliğinin tanımlanması -----
activity SprintPlanningMeeting:
    "Sprint'de yapılacak işler bu toplantıda belirlenmiştir. Her bir sprint 1 ay
olarak planlanmış, proje toplam 18 ayda bitirilmesi hedeflenmiştir. Bu toplantı araştır-
mada formal iletişim yöntemlerinden birisi olarak kabul edilmiştir".
    targets Sprint.Planned
    SprintPlanningMeeting -- part of --> ESSENCE_kernel.IseHazirlik

```

Şekil 6. Metin tipi söz dizim kuralları ile modelleme

4 Sonuç ve Öneriler

Günümüzde YM araştırmaları ile yazılım endüstrisindeki uygulamalar arasındaki kopukluğun giderilmesi yönündeki çalışmalar artarak devam etmektedir. Öte yandan YM'nin, araştırma stratejileri ve yöntemleri açısından diğer disiplinlere kıyasla göreceli olarak daha dar kapsamlı olması, araştırma yöntemleriyle ilgili terminoloji birliğinin bulunmaması ve görüş ayrılıkları göze çarpan önemli problemler arasındadır. YM alanındaki bilimsel çalışmaların çeşitli ölçütler açısından ortak bir temelde karşılaştırılabilmeleri ise yine farklı yaklaşım ve yöntemleri gerektirmektedir. ÖÇ yaklaşımının, yazılım geliştirme araştırmaları ve endüstri uygulamalarını bütünleşik olarak ele alabilecek yapıda ve nitelikte olduğu düşünülmektedir.

Bu bildiriye, YM'ne yönelik ÖÇ tabanlı tasarım ve değerlendirme modeli geliştirilmiş, ÖÇ'nin kavramsal yapısı ile alan bağımlı grafik ve metin dilleri kullanılarak YM'de deneysel bir araştırma modellenmiştir. Yer ve kapsam sınırlılığından dolayı kimi konular çalışma dışında tutulurken bir bölümü ise gelecek çalışmalara bırakılmıştır. Örneğin bunlardan birisi, literatürdeki önemli kabul edilen YM araştırmalarının ÖÇ kullanılarak yapısal, nicel ve nitel açıdan çözümlenmeleri, formal yöntemlerle karşılaştırılarak YM alanına olan katkılarının somut olarak ortaya konulması konusudur. Bu çalışmadaki ilk izlenimlerimiz, ÖÇ'nin YM araştırma alanı ve endüstri arasında bir

köprü görevini görebileceği ve öteden beri dile getirilen kuram ve uygulama arasındaki sorunlara da ışık tutabileceği yönündedir. Bildirimiz, ortaya konulan bulguları ve görüşleri dikkat alan yeni araştırmaların yapılması önerisiyle son bulmaktadır.

Kaynakça

1. Redwine .S. ve Riddle W. “Software technology maturation”. Proceedings of the Eighth International Conference on Software Engineering, 189-200, May 1985.
2. Balzer B., Litoiu M., Müller H., Smith D., Storey M.A., Tilley S., Wong K. “4th International Workshop on Adoption-Centric Software Engineering,” in Proceedings of 26th International Conference Software Engineering (ICSE 2004), 748-749, 2004.
3. SEMAT (Software Engineering Method and Theory),”[http:// www.semat.org](http://www.semat.org)”, Son erişim tarihi: 16.04.2017.
4. Stol K. ve Fitzgerald B. 2015. “A holistic overview of software engineering research strategies”. Proceedings of the 3rd International Workshop on Conducting Empirical Studies in Industry, co-located with ICSE '15, Florence, Italy, 2015.
5. Hannay J.E., Sjöberg D.I.K., Dybå T., “A systematic review of theory use in software engineering experiments”, IEEE Transactions on Software Engineering, vol. 33(2), 87-107, 2007.
6. Shaw M. “What makes good research in software engineering?” International Journal of Software Tools for Technology Transfer, 4(1), 1-7, 2002
7. Jacobson I., Meyer B., and Soley R., “The SEMAT initiative: A Call for Action”, Dr Dobb's Journal, 9, 2009.
8. Kitchenham B.A., Pfleeger, S.L., Pickard L.M.P., Jones P.W., Hoaglin D.C., Emam K. El, Rosenberg J., “Preliminary guidelines for empirical research in software engineering,” IEEE Trans. Softw. Eng., vol. 28, no. 2, 721-734, 2002.
9. Malhotra R., “Empirical research in software engineering: concepts, analysis, and applications”, CRC Press, Taylor & Francis Group, NW; 2016.
10. Klein H, Myers M, “A set of principles for conducting and evaluating interpretive field studies in information systems”, MIS Quarterly 23(3):67-94, 1999.
11. Runeson ,P., Höst M., Rainer A., Regnell, B., “Case study research in software engineering: guidelines and examples”, Wiley, 2012.
12. Uysal M.P. “In search of software engineering foundations: a theoretical and trans-disciplinary perspective”. International Journal of Computer Theory and Engineering, 8(4), 328-332, 2016.
13. Glass R.L., Vessey I., Ramesh V., “Research in software engineering: an analysis of the literature,” Inform. Software Tech., vol. 44, 2002.
14. SGR, “Chaos report on software projects”. Project Smart, The Standish Group, USA, 2014.
15. Ng P.W., Huang S., Wu Y. On the value of Essence to software engineering research: a preliminary study”, “Proceedings of GTSE, San Francisco, CA, USA, 2013.
16. Kitchenham, B., Charters, S., “Guidelines for performing systematic literature reviews in software engineering”, Keele University and Durham University Technical Report EBSE 2007-001, 2007.
17. Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J, “The impact of agile practices on communication in software development”, Empirical Software Engineering, 13 (3), 303-337, 2008.
18. OMG (Object Management Group), “Essence-Kernel and language for software engineering methods, *Document ID: SMSC/15-12-02*, 2015.