

Türkiye'deki Üniversitelerdeki Yazılım Mühendisliği Öğretim Programlarının SWEBOK Kılavuzu Kullanılarak Değerlendirilmesi

Görkem Giray¹, Eray Tüzün², Vahid Garousi^{3,4}

² Teknoloji ve Akademi Direktörlüğü, HAVELSAN, Ankara, Türkiye

³ Yazılım Mühendisliği Araştırma Grubu, Bilgisayar Mühendisliği Bölümü,
Hacettepe Üniversitesi, Ankara, Türkiye

⁴ Maral Yazılım Danışmanlık ve Ar-Ge Corporation, Calgary, Kanada

¹ gorkemgiray@gmail.com, ² etuzun@havelsan.com.tr, ^{3,4} vahid.garousi@hacettepe.edu.tr

Özetçe. Yazılımın hayatımızdaki önemi arttıkça yazılım mühendisliği öğretiminin önemi de artmaktadır. Bu bağlamda Türkiye'de 2000'li yıllarda başlayarak yazılım mühendisliği eğitimi veren bölümler kurulmaya başlanmıştır. Bu bildiriye, yazılım mühendisliği eğitimi veren 13 bölümün öğretim programları yazılım mühendisliği bilgi birikimini derleyen ve sınıflandıran SWEBOK kılavuzu kullanılarak değerlendirilmiştir. Yazılım mühendisliği öğretim programlarının alan derslerindeki yazılım mühendisliği konularının hafta bazında nicel analizleri ve SWEBOK kılavuzunun 15 bilgi alanına uyumunu incelemek üzere çapraz denetimleri yapılmıştır. Elde edilen sonuçlara göre, öğretim programlarının iyileştirilmesi konusunda önerilerde bulunulmuştur.

Anahtar Kelimeler: Yazılım mühendisliği eğitimi, Yazılım mühendisliği öğretim programı, SWEBOK, Öğretim programı değerlendirme, Öğretim programı iyileştirme

Assessment of the Software Engineering Programs in Turkish Universities using the SWEBOK Guide

Abstract. As the importance of software in our lives increases, the importance of software engineering education is also increasing. In this regard, software engineering departments have been established in Turkey starting from the beginning of 2000. In this paper, software engineering curricula of 13 software engineering departments are assessed using SWEBOK guide which compiles and classifies software engineering body of knowledge. Weekly topics covered in software engineering courses have been analyzed quantitatively and assessed their compliance with SWEBOK guide's 15 knowledge areas. According to the results obtained, suggestions were made for improving the curricula.

Keywords: Software engineering education, Software engineering curriculum, SWEBOK, Curriculum assessment, Curriculum improvement

1 Giriş

Bilgi ve iletişim teknolojilerindeki gelişmeler hayatımızı derinden etkilemektedir. Bu teknolojiler insanların yaşam biçimlerini, üretim süreçlerini, şirketlerin iş yapma biçimlerini, eğitim kurumlarının kullandıkları yöntemleri, kısaca birçok hayati ögeyi dönüştürmektedir.

Bu gelişmeler devletlerin de bilgi ve iletişim teknolojilerindeki gelişmeleri izlemesine ve buna yönelik politikalar geliştirmesine neden olmuştur. T.C. Kalkınma Bakanlığının yayımladığı 2014-2018 dönemini kapsayan, Onuncu Kalkınma Planı'ndaki [1] yazılım ile ilgili maddeler şu şekildedir: (1) adli süreç ve hizmetlerin elektronik ortama taşıyan donanım ve yazılım altyapısının geliştirilmesine devam edilmesi [1]; (2) açık kaynak kodlu yazılımlar, büyük veri, bulut bilişim, yeşil bilişim, mobil platform, nesnelerin interneti gibi alanlar değerlendirilerek kamu için uygun olabilecek çözümlerin hayata geçirilmesi [1]; (3) imalat sanayi ile yazılım sektörü arasındaki işbirliğinin geliştirilerek katma değer artırılması, çevreye duyarlı yeni teknolojilerin geliştirilmesi [1]. Ayrıca kalkınma planında bilgi ve iletişim teknolojileri sektöründe nitelikli insan kaynağı açığı bulunduğu belirtilmektedir [1].

Yazılımın dünyamızda ve ülkemizdeki bu artan önemi göz önüne alınarak, yazılımın geliştirilmesinde ve işletilmesinde rol alan yazılım mühendislerini yetiştiren Türkiye'deki üniversitelerdeki yazılım mühendisliği bölümlerinin öğretim programları incelenmiştir. Bu öğretim programlarında yapılacak güncellemelerle bu artan gereksinimleri karşılayacak insan kaynağının oluşturulması mümkün olacaktır. Bu güncellemelerin hızla artan ve değişen gereksinimler doğrultusunda sürekli yapılması gerektiği unutulmamalıdır.

Bildirinin ikinci bölümünde konunun bağlamı ve ilgili çalışmalar özetlenmiştir. Üçüncü bölümde, yazılım mühendisliği öğretim programları, belirlenen yöntemlere göre değerlendirilmiştir ve değerlendirme sonuçlarına göre iyileştirme önerileri paylaşılmıştır. Dördüncü bölümde ise, sonuçlar ve gelecek çalışmalar sunulmuştur.

2 Bağlam ve İlgili Çalışmalar

Bu bölümde, ilgili çalışmalar, Türkiye'deki yazılım mühendisliği eğitimi ve yazılım mühendisliği bilgi birikimini sınıflandıran kılavuzlar hakkında kısaca bilgi verilecektir.

2.1 İlgili Çalışmalar

Mishra ve Yazıcı Türkiye'deki üniversitelerdeki yazılım mühendisliği lisans öğretim programlarını 2004 yılında yayımlanmış olan SWEBOOK kılavuzuna göre değerlendirmiştir [2]. Toros Üniversitesi Bilgisayar ve Yazılım Mühendisliği bölümü dahil 11 bölümün öğretim programı değerlendirilmiştir. Bu çalışmada ise SWEBOOK'un son sürümündeki 15 bilgi alanı (bir önceki sürümündeki 10 bilgi alanının yerine) ve 13 yazılım mühendisliği bölümünün (Toros Üniversitesi Bilgisayar ve Yazılım Mühendisliği bölümü hariç; Celal Bayar, İstanbul Sabahattin Zaim ve Karadeniz Teknik Üniversitesi yazılım mühendisliği dahil) öğretim programları değerlendirilmiştir.

Garousi ve arkadaşları, SWEBOK kılavuzunun 3.0 sürümünü kullanarak Atılım Üniversitesi'nin yazılım mühendisliği bölümünün öğretim programını değerlendirmişler ve bazı iyileştirme önerileri sunmuşlardır [3]. Bu çalışmada benzer bir yöntemle, Türkiye'deki 13 yazılım mühendisliğinin öğretim programları değerlendirilmiştir.

Bir anket çalışmasında [4], Garousi ve arkadaşları Türkiye'deki yazılım mühendisliği uygulamalarını ve pratikleri araştırıp raporlamıştır. Bu çalışmada yazılım mühendislerinin en çok kaynak ayırdıkları etkinliklerin yazılım inşası (%31), yazılım testi (%14), yazılım gereksinimleri (%12), yazılım tasarımı (%12) ve yazılım bakımı (%11) bilgi alanları kapsamında olduğu bulunmuştur.

Jaakkola ve arkadaşları, yazılım mühendisliği öğretim programının değişik boyutlarını tartışmışlardır [5]. Yazılım mühendisliğinin bilgi birikimini sınıflayarak kapsamaya çalışan farklı kaynakları kullanarak yinelemeli bir öğretim programı geliştirme süreci deneyimlemişlerdir. Bu süreç sonunda herhangi bir öğretim içeriğini ya da programını yayınlarında paylaşmamışlardır.

2.2 Türkiye'de Yazılım Mühendisliği Eğitimi

Dünyadaki ve Türkiye'deki teknolojik ve yazılım sektöründeki gelişmeler nedeniyle yazılım mühendisliği eğitimi almış nitelikli insan kaynağına olan talep artmıştır. Türkiye'deki üniversitelerde 2000'li yılların başlarından itibaren yazılım mühendisliği bölümleri kurulmaya başlanmıştır. Türkiye'de bulunan 193 üniversitenin 13'ünde lisans eğitimi veren yazılım mühendisliği bölümü bulunmaktadır [6]. Bu 13 bölüme ek olarak Mersin ilindeki Toros Üniversitesinde Bilgisayar ve Yazılım Mühendisliği Bölümü bulunmaktadır.

T.C. Ölçme, Seçme ve Yerleştirme Merkezinin (ÖSYM), 2015 yılı Öğrenci Seçme ve Yerleştirme Sistemi (ÖSYS) yerleştirme sonuçlarına göre 13 yazılım mühendisliği bölümündeki 948 kişilik kontenjanın 926'sına yerleştirme yapılmıştır [6].

2.3 Yazılım Mühendisliği Bilgi Birikiminin Sınıflandırılması

Yazılım mühendisliği disiplinde oluşan bilgi ve deneyim birikimi birçok kişi tarafından farklı şekillerde ve detay seviyelerinde sınıflandırılmıştır. Örneğin, Carnegie Mellon Üniversitesi yazılım mühendisliğini üç temel alana ayırmaktadır: (1) Bilgisayar bilimlerinin temelleri, (2) Mühendislik temelleri, (3) Sosyal ve ekonomik temeller [7]. Bu tür bir sınıflandırma, daha detaylı taksonomiler oluşturmak için bir başlangıç noktası olabilirken bir öğretim programını tasarlamak ya da değerlendirmek için gerekli olan detaydan yoksundur.

Software Engineering Body of Knowledge (SWEBOK), yazılım mühendisliği alanındaki bilgi birikiminin kapsamını ve içeriğini düzenlemek amacıyla oluşturulmuştur. 2001 yılında ilk deneme sürümü, 2004 yılında IEEE Computer Society tarafından resmi olarak onaylanan ilk sürümü yayımlanmıştır. 2014 yılında ise şu andaki son sürümü olan üçüncü sürümü [8] yayımlanmıştır. SWEBOK kılavuzunun beş temel amacından biri öğretim programı geliştirmek için bir temel oluşturmaktır [8].

ACM Education Board ve IEEE Computer Society Educational Activities Board tarafından yazılım mühendisliği lisans eğitimi için bir öğretim programı kılavuzu "Software Engineering SE2014" adıyla yayımlanmıştır [9]. SE2014'ün bilgi alanları

için SWEBOK kılavuzundaki bilgi alanları başlangıç noktası olarak alınmıştır [9]. SE2014 bilgi alanları belirlenirken yazılım mühendisliği disiplininin evrimi ve mevcut lisans programlarından elde edilen deneyimler göz önüne alınmıştır [9]. SWEBOK 3.0 ve SE2014 bilgi alanlarının karşılaştırmalı listesi Tablo 1’de gösterilmektedir.

Tablo 1. SWEBOK 3.0 ve SE2014 kılavuzlarının bilgi alanlarının karşılaştırmalı listesi

SWEBOK 3.0	SE2014
Yazılım Gereksinimleri	Gereksinim Analizi ve Belirtimi
Yazılım Tasarımı	Yazılım Tasarımı
Yazılım İnşası	
Yazılım Testi	Yazılım Doğrulama ve Geçerleme
Yazılım Bakımı	
Yazılım Konfigürasyon Yönetimi	
Yazılım Mühendisliği Yönetimi	
Yazılım Mühendisliği Süreci	Yazılım Süreci
Yazılım Mühendisliği Modelleri ve Yöntemleri	Yazılım Modelleme ve Analiz
Yazılım Kalitesi	Yazılım Kalitesi
Yazılım Mühendisliğinin Mesleki Uygulaması	Mesleki Uygulama
Yazılım Mühendisliğinin Ekonomisi	
Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri	Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri
Matematiksel Temeller	Matematik ve Mühendislik Temelleri
Mühendislik Temelleri	
	Güvenlik

ACM ve IEEE Computer Society’nin desteklediği bilgisayar bilimleri öğretim kılavuzunda yazılım mühendisliği bir bilgi alanı olarak yer almaktadır [10]. Bu bilgi alanı altında şu alt başlıklar yer almaktadır: (1) Yazılım süreçleri, (2) Yazılım proje yönetimi, (3) Araçlar ve ortamlar, (4) Gereksinim mühendisliği, (5) Yazılım tasarımı, (6) Yazılım inşası, (7) Yazılım doğrulama ve geçerleme, (8) Yazılım evrimi, (9) Yazılım güvenilirliği, (10) Biçimsel yöntemler [10].

Yazılım mühendisliği lisansüstü eğitimi için de 2009 yılında ilk sürümü yayımlanmış olan bir kılavuz bulunmaktadır [11].

3 Yazılım Mühendisliği Öğretim Programlarının Değerlendirilmesi

Bu bölümde değerlendirme yöntemi ve değerlendirilen öğretim programları hakkında bilgi verilmiştir. Sonrasında keşfe dayalı çalışma aşamasında nicel verilerin nasıl oluşturulduğu anlatılmıştır. Son olarak elde edilen sonuçlar ışığında öğretim programlarının iyileştirilmesi için öneriler sunulmuştur.

3.1 Değerlendirme Yöntemi

Bu çalışmada sistematik bir değerlendirme yapabilmek için “keşfe dayalı” ve “iyileştirici” durum çalışması yaklaşımı kullanılmıştır. Keşfe dayalı çalışma aşaması, mevcut durumun anlaşılmasını, yeni anlayışların aranmasını, yeni araştırmalar için

fikirlerin ve hipotezlerin oluşturulmasını kapsamaktadır [12]. İyileştirme aşamasında ise üzerinde çalışılan durumun bir yönünün iyileştirilmesi hedeflenmektedir [12].

Bu çalışmada, keşfe dayalı çalışma aşamasında mevcut öğretim programlarının değerlendirilmesi için SWEBOK kılavuzundaki bilgi alanları kullanılmıştır. Bunun nedeni Tablo 1’de görüldüğü gibi SWEBOK kılavuzunun daha fazla bilgi alanını kapsamasıdır. İyileştirme aşamasında ise bu bilgi alanları içinde yeterince kapsanmadığı düşünülenler için önerilerde bulunulmuştur.

3.2 Değerlendirilen Öğretim Programları

Türkiye’deki üniversitelerdeki yazılım mühendisliği bölümlerinin listesi, ÖSYM’nin 2015 yılına ait lisans programları listesinden [6] alınmıştır. Bu bölümlerin web sitelerindeki öğretim programları listesi alınmıştır. Karadeniz Teknik Üniversitesinin yazılım mühendisliği bölümü için 184 AKTS’lik (Avrupa Kredi Transfer Sistemi), diğer tüm bölümler için 240 AKTS’lik tam ders listesi elde edilmiştir. Bu listedeki seçmeli dersler ve Türk Dili, Atatürk İlkeleri ve İnkılap Tarihi dersleri gibi yazılım mühendisliği ile bağlantısı olmayan dersler listeden çıkartılmıştır. Sonuç olarak 13 bölümdeki toplam 357 ders değerlendirme kapsamına alınmıştır. 13 bölümün yer aldığı üniversitelerin listesi Tablo 2’de gösterilmektedir.

Tablo 2. Öğretim programları değerlendirilen 13 yazılım mühendisliği bölümünün bulunduğu üniversiteler

Üniversite	Bulunduğu İl	Kontenjan
Atılım Üni.	Ankara	40
Bahçeşehir Üni.	İstanbul	78
Beykent Üni.	İstanbul	60
Celal Bayar Üni.	Manisa	127
Fırat Üni.	Elazığ	290
Işık Üni.	İstanbul	11
İstanbul Aydın Üni.	İstanbul	50
İstanbul Sabahattin Zaim Üni.	İstanbul	40
İzmir Ekonomi Üni.	İzmir	55
İzmir Üni.	İzmir	50
Karadeniz Teknik Üni.	Trabzon	64
Maltepe Üni.	İstanbul	33
Yaşar Üni.	İzmir	50

3.3 Keşfe Dayalı Çalışma Aşaması: Bilgi Alanlarına Göre Değerlendirme

Kapsama alınan 13 bölümdeki toplam 357 dersin içeriklerinin SWEBOK kılavuzundaki bilgi alanlarına göre dağılımı hafta bazında yapılmıştır. Çalışmamızın tekrarlanabilmesi ve diğer araştırmacılar tarafından daha derin incelenebilmesi için değerlendirmeye ait tüm ham veriler [13]’te sunulmuştur. Dağılım yapılırken aşağıdaki kurallar uygulanmıştır:

- Bir dersin birden fazla bilgi alanını ilgilendirdiği durumlarda dersin içeriğine ilgili bölümün web sitesinden ulaşılmaya çalışılmıştır. Dersin hafta bazında içeriğine ulaşıldığı durumlarda dağılım hafta bazında yapılmıştır. Örnek olarak

bir dersin içeriğinin SWEBOK kılavuzundaki bilgi alanlarına göre dağılımı Tablo 3'te gösterilmektedir.

Tablo 3. Yazılım mühendisliğinin temelleri dersinin (İzmir Üniversitesi) içeriğinin SWEBOK kılavuzunun bilgi alanlarına göre dağılımı

Ders Adı	Konu	Yazılım Gereksinimleri	Yazılım Tasarımı	Yazılım İnşası	Yazılım Testi	Yazılım Bakımı	Yazılım Konfigürasyon Yönetimi	Yazılım Mühendisliği Yönetimi	Yazılım Mühendisliği Süreci	Yazılım Mühendisliği Modelleri ve Yöntemleri	Yazılım Kalitesel Uygulamaları	Yazılım Mühendisliği Mesleki Uygulamaları	Yazılım Mühendisliği Ekonomisi	Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri	Matematiksel Temeller	Mühendislik Temelleri
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Yazılım mühendisliğine giriş								1							
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Yazılım geliştirme süreç modelleri								1							
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Yazılım proje yönetimi									1						
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Gereksinim mühendisliğine giriş, İş gereksinimleri	1														
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Kullanım hikayeleri, Kullanım senaryoları	1														
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Unified Modeling Language (UML), Nesneye yönelik analiz			1												
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Yazılım tasarımına giriş, Nesneye yönelik tasarım			1												
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Yazılım inşasına giriş, Bütünleşik geliştirme ortamı			1												
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Sürüm kontrol sistemleri, Yeniden yapılandırma			0,5			0,5									
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Doğrulamaya ve geçermeye giriş, Doğrulama etkinlikleri (statik test etme)				1											
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Kara kutu testi, Beyaz kutu testi				1											
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Yazılım bakımına giriş					1										
YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ	Yazılım konfigürasyon yönetimi						1									

- Kapsama alınan 13 bölümdeki toplam 357 dersin içeriğinden oluşan tablonun boş hali yazarlardan biri tarafından oluşturulduktan sonra, iki yazar birbirinden bağımsız olarak bu tabloları doldurmuş, daha sonra iki yazarın tabloları arasında karşılaştırma yapılmıştır. Farklı sonuçlara ulaşılan dersler için anlaşma sağlanamadığı durumda, üçüncü yazarın görüşü alınıp son değerler elde edilmiştir.
- Dersin SWEBOK kılavuzundaki bir bilgi alanına karşılık geldiği durumlarda ilgili bilgi alanı için 14 haftalık ders yapıldığı varsayımı yapılmıştır. Örneğin, Ayrık Yapılar dersi için Matematiksel Temeller bilgi alanına 14 hafta değeri verilmiştir.
- Haftalık konular bazında incelenen derslerin sadece ara sınav ve final sınavı yapılan haftaları değerlendirmede göz önüne alınmamıştır. İçeriği hafta bazında incelenmeyen derslerin süresi 14 hafta olarak alınmıştır.
- Yazılım mühendisliğine giriş, yazılım mühendisliğinin temelleri gibi birden fazla bilgi alanını kapsadığı kesin olan ve ders içerikleri hafta bazında elde edilemeyen dersler için 14 haftalık içerik, SWEBOK kılavuzundaki ilk 12 bilgi alanı için 1,17 (14/12) hafta olarak verilmiştir.
- Stajlar için, Yazılım Mühendisliği Mesleki Uygulamaları bilgi alanına 14 haftalık bir değer verilmiştir.
- Bitirme projesi için, haftalık olarak yapılan etkinliklerin listesi olduğu durumlarda yine SWEBOK kılavuzunun bilgi alanlarına göre dağılım yapılmıştır; olmadığı durumlarda ise, Garousi ve arkadaşlarının Atılım Üniversitesi'ndeki bitirme projesi için oluşturdukları dağılım değerlendirmeye alınmıştır [3]. Bu

değerlendirme, bitirme projesinin yazılım mühendisliğindeki çoğu alanına ve özellikle de tasarım, inşa ve yönetime biraz daha çok ağırlık vermesi deneyimine dayanmaktadır. Dağılım hafta bazında şu şekilde yapılmıştır: Yazılım Gereksinimleri (1 hafta), Yazılım Tasarımı (2 hafta), Yazılım İnşası (4 hafta), Yazılım Testi (1 hafta), Yazılım Mühendisliği Yönetimi (2 hafta), Yazılım Mühendisliği Süreci (1 hafta), Yazılım Mühendisliği Modelleri ve Yöntemleri (1 hafta), Yazılım Kalitesi (1 hafta), Yazılım Mühendisliği Ekonomisi (1 hafta).

Yukarıdaki kurallar çerçevesinde 13 bölümdeki yazılım mühendisliği ile bağlantılı tüm zorunlu derslerin içeriklerinin SWEBOK kılavuzundaki bilgi alanlarına göre dağılımı yapılmıştır. Sonuç olarak bu veriden Tablo 4'teki gibi bir özet tablo elde edilmiştir.

Tablo 4. 13 yazılım mühendisliği bölümündeki derslerin içeriğinin SWEBOK kılavuzunun bilgi alanlarına göre dağılımı

	Yazılım Gereksinimleri	Yazılım Tasarımı	Yazılım İnşası	Yazılım Testi	Yazılım Bakımı	Yazılım Konfigürasyon Yönetimi	Yazılım Mühendisliği Yönetimi	Yazılım Mühendisliği Süreci	Yazılım Mühendisliği Modelleri ve Yöntemleri	Yazılım Kalitesi	Yazılım Mühendisliği Mesleki Uygulamaları	Yazılım Mühendisliği Ekonomisi	Bilgisayar Mühendisliği/Bilimlerinin Temelleri	Matematiksel Temeller	Mühendislik Temelleri	TOPLAM
ATILIM ÜNİ.	17	18	19	10	2	1	18	6	4	9	29	2	158	84	0	379
BAHÇEŞEHİR ÜNİ.	15	9	4	11	1	0	17	11	3	1	14	0	112	84	11	292
BEYKENT ÜNİ.	16	59	7	8	0	0	31	4	2	22	28	1	183	84	14	459
CELÂL BAYAR ÜNİ.	19	18	6	17	2	1	24	4	3	16	15	1	126	70	0	323
FIRAT ÜNİ.	4	19	5	2	1	0	18	5	1	15	28	1	224	98	0	421
IŞIK ÜNİ.	3	4	6	3	2	2	18	3	3	3	30	3	182	70	14	350
İSTANBUL AYDIN ÜNİ.	19	27	22	21	0	1	35	11	5	20	43	3	182	98	2	488
İSTANBUL SABAHATTİN ZAİM ÜNİ.	17	34	6	17	2	2	18	5	3	17	15	2	224	84	14	462
İZMİR EKONOMİ ÜNİ.	7	25	9	17	2	0	20	6	6	2	14	2	84	70	0	263
İZMİR ÜNİ.	22	34	17	19	1	3	20	5	4	15	28	5	104	60	21	356
KARADENİZ TEKNİK ÜNİ.	17	24	10	16	1	3	5	11	4	16	28	2	157	66	0	359
MALTEPE ÜNİ.	16	31	5	16	1	1	17	2	2	16	43	2	196	84	0	434
YAŞAR ÜNİ.	17	19	23	17	1	1	19	3	3	17	15	3	126	70	0	336
Ortalama	14,59	24,67	10,67	13,40	1,24	1,17	20,05	5,94	3,40	13,09	25,47	2,17	158,31	78,62	5,85	378,62
Standart Sapma	5,90	13,59	7,00	6,01	0,85	0,97	7,19	3,09	1,30	6,92	10,27	1,28	45,17	11,90	7,69	69,36

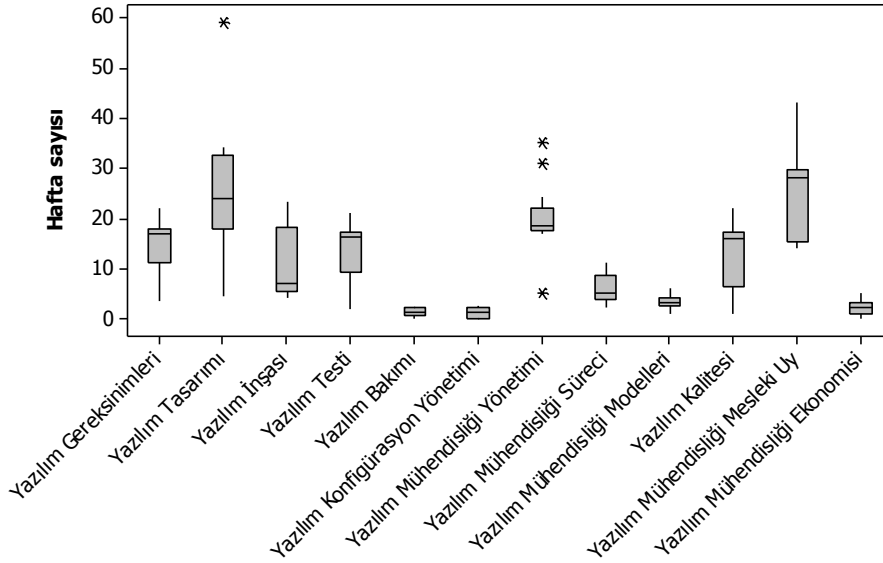
Tablo 4'teki toplam kolonlarına bakıldığında İzmir Ekonomi Üniversitesinin zorunlu derslerinin toplam 263 hafta, İstanbul Aydın Üniversitesinin ise 488 hafta olduğu görülmektedir. Tüm bölümlerin aynı ya da benzer ders yükü ile öğrenci mezun edeceği düşünüldüğünde İzmir Ekonomi Üniversitesinin seçmeli derslere daha fazla ağırlık verdiği görülmektedir.

Tablo 4'teki ortalamalardan görüleceği gibi sırasıyla Yazılım Konfigürasyon Yönetimi, Yazılım Bakımı ve Yazılım Mühendisliği Ekonomisi bilgi alanları zorunlu derslerle en az kapsanan bilgi alanları olarak karşımıza çıkmaktadır. Bu sonuç, Garousi ve arkadaşlarının Atılım Üniversitesi Yazılım Mühendisliği öğretim programını değerlendirdikleri çalışmanın [3] sonuçlarıyla birebir örtüşmektedir. Bu üç bilgi alanından sonra en az kapsanan diğer bilgi alanları ise sırasıyla Yazılım Mühendisliği Modelleri ve Yöntemleri, Mühendislik Temelleri ve Yazılım Mühendisliği Süreci bilgi alanlarıdır.

Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri ve Matematiksel Temeller bilgi alanları tüm bölümler tarafından diğer bilgi alanlarına göre oldukça fazla

kapsanmaktadır. Bu bilgi alanları için hem ortalama değerler diğerlerine göre yüksektir hem de bölümler arasındaki farklar görece azdır. Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri bilgi alanı oldukça geniş bir alanı kapsamaktadır. SWEBOK kılavuzuna göre bu bilgi alanı algoritma, programlama, veri yapıları, veri yönetimi ve veritabanı yönetim sistemleri, işletim sistemleri, insan bilgisayar etkileşimi, bilgisayar ağları, donanım ile ilgili dersleri kapsamaktadır. Bu kapsamın oldukça geniş olması ve bu derslerin çoğu bölümde zorunlu olması nedeniyle bu bilgi alanına ayrılan pay tüm bölümlerde yüksektir. Matematiksel Temeller bilgi alanı genel matematik, kalkülüs, diferansiyel denklemler, olasılık ve istatistik, analiz, cebir, ayık yapılar gibi standart dersleri kapsamaktadır ve bu dersler çoğu bölüm tarafından zorunlu olarak verilmektedir.

Elde edilen niceliksel verileri görsel şekilde özetlemek amacıyla Şekil 1’de gösterilen kutu grafiği çizilmiştir. SWEBOK kılavuzundaki yazılım mühendisliği bilgi alanlarına (ilk 12 bilgi alanı) odaklanmak amacıyla, grafikte Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri, Matematiksel Temeller ve Mühendislik Temelleri bilgi alanları gösterilmemiştir. Böylece y ekseninin uzaması ve ilk 12 bilgi alanı arasındaki farkların küçülerek daha zor gözlenebilir olması engellenmiştir.



Şekil 1. Tablo 4’teki verilerin kutu grafiği

Şekil 1’de görüldüğü gibi sadece Yazılım Tasarımı, Yazılım Mühendisliği Yönetimi ve Yazılım Mühendisliği Ekonomisi bilgi alanlarında aykırı değerler (yıldız ile gösterilmiştir) bulunmaktadır. Aykırı değerlerin çok olmaması değerlendirilen verinin homojene yakın olduğunu göstermektedir.

Yazılım Bakımı, Yazılım Konfigürasyon Yönetimi, Yazılım Mühendisliği Modelleri ve Yöntemleri ve Yazılım Mühendisliği Ekonomisi bilgi alanları tüm bölümlerde hem en az kapsamıştır hem de benzer bir dağılıma sahiptir. Ortalamaları

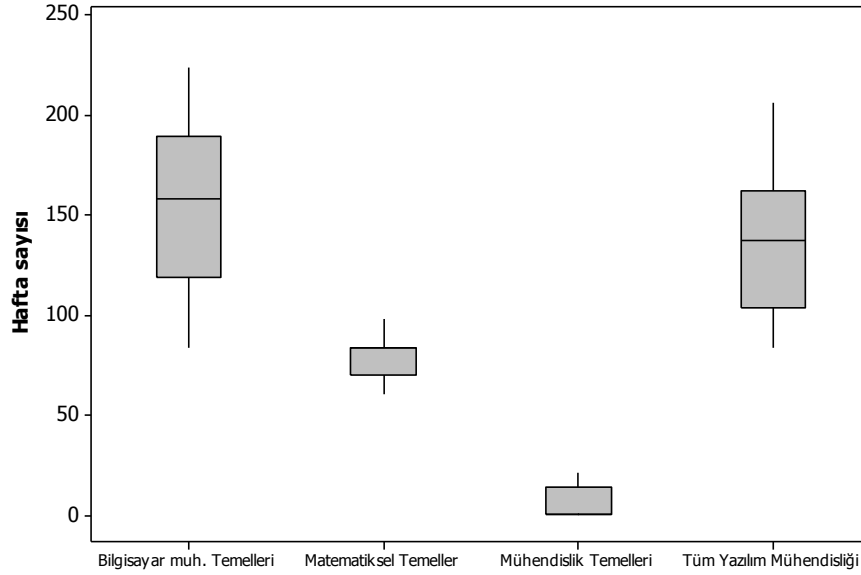
1,17 ile 3,40 arasında değişmektedir. Sadece Yazılım Mühendisliği Ekonomisi bilgi alanı için bir aykırı değer bulunmaktadır; bu aykırı değer de 5 haftadır.

Yazılım Mühendisliği Yönetimi ve Yazılım Mühendisliği Süreci bilgi alanları için de tüm bölümlerdeki dağılım görece homojen görünmektedir. Yazılım Mühendisliği Yönetimi bilgi alanı için elde edilen 13 değerden 10 tanesi ortalamaya yakındır.

Yazılım Gereksinimleri, Yazılım İnşası, Yazılım Testi ve Yazılım Kalitesi bilgi alanları çok homojen bir dağılım göstermese de aykırı değerlere de sahip değildir.

Yazılım Mühendisliği Mesleki Uygulamaları bilgi alanında dağılım daha heterojendir. Ortalama 25,47 hafta iken bu bilgi alanına İzmir Ekonomi Üniversitesi 14 hafta, İstanbul Aydın ve Maltepe Üniversitesi ise 43 hafta ayırmıştır. Dağılım konusunda bölümler arasında en farklı olan Yazılım Tasarımı bilgi alanıdır. Ortalaması 24,67 hafta iken bu bilgi alanına Işık Üniversitesi 4 hafta, Beykent Üniversitesi 59 hafta ayırmıştır.

Şekil 2’de SWEBOK kılavuzunun ilk 12 bilgi alanının toplamının (Tüm Yazılım Mühendisliği) ve diğer 3 temel bilgi alanının kutu grafiği gösterilmektedir.



Şekil 2. Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri, Matematiksel Temeller, Mühendislik Temelleri ve geri kalan 12 bilgi alanının toplamının kutu grafiği

Şekil 2’de görüldüğü gibi en az zaman (Ortalama 5,85 hafta) Mühendislik Temelleri bilgi alanına ayrılmaktadır. Bu bilgi alanına ayrılan zaman homojene yakındır. Matematiksel Temeller bilgi alanına Mühendislik Temellerine göre daha fazla zaman (Ortalama 78,62 hafta) ayrılmaktadır ve bu bilgi alanı için de dağılım homojene yakındır. Bilgisayar Mühendisliğinin/Bilimlerinin Temelleri bilgi alanı için ortalama 158,31 hafta, Tüm Yazılım Mühendisliği bilgi alanları için ise 135,85 hafta zaman ayrılmaktadır. Bu iki alan için de üniversiteler arasındaki dağılım heterojendir.

3.4 İyileştirme Aşaması: Yazılım Mühendisliği Öğretim Programı için Öneriler

Bir önceki bölümde yapılan nicel değerlendirmenin sonucunda yazılım mühendisliği öğretim programlarının iyileştirilmesi için aşağıdaki öneriler sunulmuştur.

- **Yazılım Bakımı**
Yazılım, doğası gereği, yaşam döngüsü içinde çeşitli nedenlerle değişime uğrar ve yazılım projelerinde kaynakların önemli bir bölümü yazılım bakımına ayrılır [14]. Yazılım, birlikte çalıştığı diğer bileşenlerdeki değişikliklerden etkilenir. Örneğin işletim sistemindeki değişikliklere yazılımı uyarlamak amacıyla bakım yapılabilir. Yazılım kullanılırken farklı kaynaklardan farklı nedenlerle değişiklik istekleri gelebilir. Yazılımı daha iyileştirmek için bakım yapılabilir; yani yazılıma bazı işlevler eklenebilir. Hatasız bir yazılımın olmayacağı genel olarak kabul görmüştür. Bu nedenle yazılımdaki hataları düzeltmek için düzeltici bakım etkinliklerinin gerçekleştirilmesi kaçınılmazdır. Bazı durumlarda olası problemleri önceden önlemek için önleyici bakım etkinlikleri gerçekleştirilebilir. Bir yazılımın yaşam döngüsü içinde bu bakım etkinlikleri oldukça sık görülür. Bundan dolayı Yazılım Bakımı bilgi alanı hakkında öğrencilere daha fazla bilgi verilmelidir.
- **Yazılım Konfigürasyon Yönetimi**
Yazılım yoğun sistemler gittikçe daha karmaşık hale gelmekte ve ölçekleri büyümektedir. Dolayısıyla daha büyük takımlar yazılım geliştirmek için birlikte çalışmakta ve çok sayıda paydaş yazılım sürecinde üretilen iş ürünleri (kod, gereksinim belgeleri, vs.) üzerinde sık sık değişiklik yapmaktadır. Bu değişikliklerin takip edilmesi yazılımın kalitesi için büyük önem taşımaktadır. Dolayısıyla Yazılım Konfigürasyon Yönetimi bilgi alanında öğrencilere daha fazla bilgi verilmeli ve GitHub, SVN gibi versiyon kontrol araçlarının kullanımı teşvik edilmelidir.
- **Yazılım Mühendisliği Ekonomisi**
Mühendisliğin önemli amaçlarından birisi, bir fayda elde ederken bunun maliyetini de göz önüne almak ve uygun sınırlar içinde tutmaktır. Dolayısıyla yazılım mühendisliğinin ekonomik boyutu hakkında öğrencilere bilgi verilmesi gerekmektedir.
- **Yazılım Mühendisliği Modelleri ve Yöntemleri**
Modelleme etkinliği, yazılım mühendisliği alanındaki merkezi etkinliklerden biridir. Problemi ve problemin bulunduğu alanı anlamak, probleme çözümler üretmek için soyutlama yaparak modeller üretmek oldukça faydalıdır. Mevcut öğretim programlarında modellemenin veri modelleme ve UML üzerine yoğunlaştığı görülmektedir. Modellemeyi mevcut derslerin içinde daha geniş bir yelpazede ele almanın öğrenciler için faydalı olacağı düşünülmektedir.
- **Yazılım Mühendisliği Süreci**
Yazılım mühendisliği süreci, yazılım mühendisliği etkinlikleri için bir temel oluşturur. Dolayısıyla öğrencilerin yazılım gereksinimleri, yazılım tasarımı, yazılım inşası gibi bilgi alanlarını anlayabilmeleri için tüm bu etkinliklerin büyük resmini gösteren yazılım mühendisliği süreci bilgisine ihtiyaçları vardır. Dolayısıyla öğretim programlarında plan güdümlü ve çevik yazılım geliştirme süreçleri hakkında kapsayıcı bilgiler vermek öğrencilerin hem büyük resmi

görmeleri açısından hem de diğer derslerde anlatılanları büyük resim içinde anlamlandırabilmeleri için önemlidir.

- Zorunlu dersler dışında destekleyici eğitim
Her bölümün kapsadığı bilgi alanlarına bakılarak öğretim programının yeteri kadar kapsamadığı alanlarda öğrenciler bölüm içindeki seçmeli derslere ve üniversite dışındaki kaynaklara (endüstride açılan yaz okulları, sertifika programları, Massive Open Online Courses - MOOC) yönlendirilebilir.

4 Sonuç ve Gelecek Çalışmalar

Bu bildiri Türkiye’de yazılım mühendisliği öğretimi veren 13 bölümün öğretim programları SWEBOK kılavuzu kullanılarak değerlendirilmiştir. SWEBOK kılavuzunun bilgi alanlarına göre her bir öğretim programının hafta bazında dağılımı yapılarak bir analiz yapılmıştır. Bu analiz sonucunda öğretim programlarını iyileştirici öneriler sunulmuştur. Yapılan değerlendirme sonucunda yapılan öneriler şunlardır: (1) Yazılım yaşam döngüsü içinde önemli bir yeri olan yazılım bakımı hakkında daha fazla teorik ve pratik bilgi verilmelidir; (2) Yazılım Konfigürasyon Yönetimi bilgi alanı kapsamında özellikle versiyon kontrol sistemleri hakkında pratik bilgiler verilmeli ve öğrencilerin çeşitli araçları kullanmaları teşvik edilmelidir; (3) Yazılım Mühendisliği Ekonomisi bilgi alanında fayda maliyet analizi gibi yazılım mühendisliğinin ekonomik boyutları hakkında bilgi verilmelidir; (4) Yazılım Mühendisliği Modelleri ve Yöntemleri bilgi alanı kapsamında problem ve çözüm alanlarında modellemeye daha fazla değinilmelidir; (5) Yazılım mühendisliği eğitimi için kapsayıcı bir temel oluşturan Yazılım Mühendisliği Süreci bilgi alanında plan güdümlü ve çevik süreçler hakkında karşılaştırmalı olarak temel bilgiler ve endüstriden örnekler verilmelidir; (6) Bu ve buna benzer bir değerlendirme sonucunda öğrenciler üniversite dışındaki kaynaklara da (MOOC gibi) yöneltilerek kendilerini geliştirmeleri desteklenmelidir.

Öğretim programlarının değerlendirilmesi için güncel, doğru, tam ve yeterli seviyede detaylı bilginin olması büyük önem taşımaktadır. Bu çalışmada sadece web sitesindeki bilgiler değerlendirme için kullanılmıştır. Bundan sonraki aşamada 13 bölümdeki dersler için ilgili öğretim üyelerinden daha detaylı bilgiler alınarak daha sağlıklı bir değerlendirme yapılması planlanmaktadır. Seçmeli derslerin de değerlendirmeye katılması öğretim programlarının daha kapsamlı ele alınması için gerekli olduğu değerlendirilmiştir. İleriki dönemlerde, Türkiye’deki yazılım mühendisliği bölümleriyle yurt dışındaki yazılım mühendisliği bölümlerinin derslerinin içerik dağılımları karşılaştırılabileceği değerlendirilmiştir.

SWEBOK kılavuzundaki 15 bilgi alanına ideal olarak nasıl bir ağırlık verilmesi gerektiğinden söz edilmemiştir. İlerleyen dönemlerde SWEBOK kılavuzundaki her bilgi alanına ideal olarak ne kadar ağırlık verilmesi gerektiği konusunda bir çalışma yapılması değerlendirilebilir. İdeal bir dağılım elde edildiğinde öğretim programları için daha net öneriler verebilmek mümkün olacaktır. Çalışmada kullandığımız SWEBOK kılavuzunun içeriğine getirilen eleştiriler de vardır [5]. SWEBOK kılavuzu, SE2014 kılavuzu ve yazılım mühendisliği disipliniyle bağlantılı diğer disiplinler için oluşturulmuş diğer kılavuzlar ve endüstrideki gereksinimler de incelenerek yeni bir değerlendirme çerçevesi oluşturulabileceği değerlendirilmiştir.

Kaynakça

1. T.C. Kalkınma Bakanlığı, Onuncu Kalkınma Planı 2014 - 2018, Ankara, 2013.
2. Mishra, A., Yazıcı, A.: An Assessment of the Software Engineering Curriculum in Turkish Universities: IEEE / ACM Guidelines Perspective, 188–219 (2011)
3. Garousi, V., Mishra, A., Yazıcı, A.: Üniversite Yazılım Mühendisliği Programları Müfredatının SWEBOK Kılavuzu Kullanılarak Değerlendirme ve İyileştirilmesi: Türkiye'de bir Vaka Çalışması, UYMS (2015)
4. Garousi, V., Coşkunçay, A., Can, A.B., Demirörs, O.: A Survey of Software Engineering Practices in Turkey, Journal of Systems and Software, cilt 108, 148-177 (2015)
5. Jaakkola, H., Henno, J., Rudas, I.J.: IT curriculum as a complex emerging process. 2006 IEEE International Conference on Computational Cybernetics, ICCCYB (2006). <http://doi.org/10.1109/ICCCYB.2006.305731>
6. ÖSYM, 2015-ÖSYS Yerleştirme Sonuçlarına İlişkin Sayısal Bilgiler: Tablo-4 Merkezi Yerleştirme İle Öğrenci Alan Yükseköğretim Lisans Programları, <http://dokuman.osym.gov.tr/pdfdokuman/2015/OSYS/OSYS2015YerlestirmeMinMaxTablo-423072015.pdf>, Erişim tarihi: 07 Haziran 2015.
7. Shaw, M. (Editör): Software Engineering for the 21st Century: A basis for rethinking the curriculum – CMU-ISRI-05-108, <http://www.cs.cmu.edu/~Compose/SEprinciples-pub-rev2.pdf> (2005). Erişim tarihi: 09 Haziran 2016.
8. Bourque, P., Fairley, R.E. (Editörler): Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society (2014) www.swebok.org.
9. Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery. (2014). Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. The Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery.
10. ACM/IEEE-CS Joint Task Force on Computing Curricula. 2013. Computer Science Curricula 2013. ACM Press and IEEE Computer Society Press. DOI: <http://dx.doi.org/10.1145/2534860>
11. Graduate Software Engineering 2009 (GSWE2009) Curriculum Guidelines for Graduate Degree Programs in Software Engineering, Version 1.0 (2009)
12. Runeson, P., Höst, M.: Guidelines for conducting and reporting case-study research in software engineering, Empirical Software Engineering, cilt 14, 131-164 (2009)
13. Giray, G., Tüzün, E., Garousi, V.: Ham veriler: Türkiye'deki Üniversitelerdeki Yazılım Mühendisliği Öğretim Programlarının SWEBOK Kılavuzu Kullanılarak Değerlendirilmesi, <http://goo.gl/pVXfjr>, Erişim Tarihi: 17 Haziran 2016.
14. Grubb, P., Takang, A.A.: Software Maintenance: Concepts and Practice: World Scientific (2003)