

Üniversite Yazılım Mühendisliği programları müfredatının SWEBOK kılavuzu kullanılarak değerlendirme ve iyileştirilmesi: Türkiye’de bir vaka çalışması

Vahid GAROUSİ
Yazılım Mühendisliği Araştırma Grubu
Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi, Ankara
vahid.garousi@hacettepe.edu.tr

Alok MISHRA, Ali YAZICI
Yazılım Mühendisliği Bölümü
Atılım Üniversitesi, Ankara
{alok.mishra, ali.yazici}@atilim.edu.tr

Özet: Bu makalede SWEBOK 3.0 (Yazılım Mühendisliği (YM) Bilgi Tabanı Kılavuzu 3.0) sürümü kullanarak, Türkiye’deki bir üniversitenin YM müfredatının bir değerlendirme ve iyileştirme çalışması raporlanmaktadır. Sistematik vaka çalışması yaklaşımı ile alan derslerindeki YM konularının saat bazında nicel analizleri ve SWEBOK’un 15 bilgi alanına uyumunu incelemek üzere çapraz denetimleri yapılmıştır. Uygulanan yaklaşım ile çalışma kapsamında olan YM programına yararlı gözlemler, öneriler ve somut faydalar sağlanmıştır, örneğin: (1) Yazılım Bakımı kavramının YM derslerinde yeterli derecede işlenmediği gözlemlenmiş ve bununla ilgili olarak uygulamaların müfredata katılması önerilmiştir; (2) yapılan analize dayanarak, üst düzey YM derslerinde (3. ve 4. yılında verilen) dönem projelerinde GitHub gibi bir kod sürüm sistemi kullanımı önerilmiştir ve (3) Ekonominin yazılım projelerindeki önemi nedeni ile çeşitli YM derslerinde ekonomi-tabanlı-YM, örneğin: kullanım-vakaların fayda maliyet analizi, yazılım kalite faaliyetlerinin yatırım getirisi ve ayrıca YM ekonomisi hakkında derslerde pratik çalışmalar yapılması gerektiği raporlanmıştır. YM müfredatın değerlendirme ve iyileştirilmesi ile ilgili olarak SWEBOK bağlamında birçok çalışma yapılmıştır, ancak, bu yeni yaklaşımda öncekilerden farklı olarak, SWEBOK’un en son 2014 sürümü kullanılmış ve daha kapsamlı analizler yapılmıştır. Ayrıca, bu çalışmada önerilen yaklaşımın geliştirilmesi ve diğer üniversitelerdeki YM programlarına uygulanması mümkündür.

Anahtar sözcükler: Yazılım Mühendisliği Eğitimi, Yazılım Mühendisliği müfredatı, müfredat değerlendirme, müfredat iyileştirme, SWEBOK, vaka çalışması, Türkiye’deki üniversiteler, Türkiye’de eğitim

Abstract: The paper reports an assessment and improvement of the Software Engineering (SE) curriculum in the context of a SE university program in Turkey, taking the latest version 3.0 of the Software Engineering Body of

Knowledge (SWEBOK) as the baseline. Using the systematic case-study approach and quantitative analysis of hourly topics covered in each SE course, we conducted cross checking of the hourly topics with the 15 knowledge areas of the SWEBOK version 3.0. Our approach and study has provided useful observations and recommendations and tangible benefits that have improved the SE program under study. Among our empirical findings in the context of the case SE program under study are the followings: (1) We noticed that the concept of software maintenance has barely only mentioned in the SE courses, and there is a need for practical exercises on this very important SE topic; (2) Based on our analysis, we recommended that there is a need to require from students in their senior SE courses (3rd and 4th year) to use code-versioning system, e.g., GitHub; and (3) We are aware that economics is a major driver of many software projects and the relevant decision making scenarios in the industry. We thus recommended that there is a need to talk more about SE economics in various courses, e.g., cost benefit analysis of use-cases, ROI of software quality activities and also the need for more practical exercises about SE economics. While a number of previous studies have conducted assessment and improvement of SE curriculum using SWEBOK, our approach uses the SWEBOK's latest 2014 version and is based on more in-depth and quantitative analysis. Furthermore, our approach is generalizable and can be applied to other SE programs in other universities elsewhere.

Keywords: Software engineering education; software engineering curriculum; assessment of curriculum; improvement of curriculum; SWEBOK; case study; universities in Turkey; education in Turkey

1 Giriş

Yazılım Mühendisliği (YM) en hızlı gelişen mühendislik disiplinlerinden birisi olup, buna bağlı olarak, yazılım geliştiren organizasyonlarda yapılan aktiviteler de doğal olarak çeşitlilik göstermektedir. Çeşitli araştırmalar, örneğin [1-4], yazılım endüstrisinin gereksinimleri ile geleceğin yazılım mühendislerine yönelik verilen eğitim arasında bir uçurum olduğunu göstermektedir. YM müfredatı endüstrinin gereksinimlerine yanıt verebilmelidir. Bu sağlandığında, üniversiteler endüstrinin gereksinimlerine uygun yetenekli profesyoneller yetiştirebilecektir [5]. Öte yandan, yazılım firmalarının yeni mezunlardaki, yazılım geliştirme pratiğine yönelik eksiklikleri ile ilgili şikâyetlerine sıkça rastlanmaktadır [6].

Yazılım Mühendisliği disiplinindeki hızlı değişim, üniversitelerin YM müfredatlarını sürekli olarak değiştirmeye zorlamaktadır. YM eğitimi alanında öncü olan ve 1987 yılından başlayan IEEE Conference on Software Engineering Education and Training (CSEE&T) konferans serisinin yirmi-sekizincisi 2015 yılında yapılmış ve bu konferans kapsamında konu ile ilgili birçok deneyim ve yaklaşım raporlanmıştır.

YM'nin, örneğin YM Proje Yönetimi gibi, teknik konularındaki sürekli iyileştirme gereksinimine benzer biçimde, dünyadaki üniversitelerdeki çoğu YM müfredatlarında da sürekli iyileştirme çalışmalarına rastlanmaktadır. Bu iyileştirmeler için, YM Bilgi Tabanı Kılavuzu (SWEBOK) [7-9] ve Software Engineering (SE) 2004 [1] gibi farklı

referans modelleri kullanılmaktadır. SWEBOK kılavuzu, YM bilgi alanı için genel kabul gören 43444 rehberine yönelik ISO/IEC TR 19759:2005 uluslararası standardını belirlemektedir. Bu kılavuz birçok profesyonel kuruluş ve sektör temsilcilerinin katkıları ile geliştirilmiş ve ayrıca IEEE-CS tarafından yayımlanmıştır. Bu süregelen ve gelişen bir çalışmadır. İlk deneme sürümü [7] 2001, ikincisi 2004 ve son sürümü olan SWEBOK 3.0 [9] 2014 yılında çıkmıştır. Diğer taraftan, SE 2004 projesi IEEE-CS ve ACM'in ortak bir ürünüdür. Bu çalışma özellikle YM lisans programları için müfredat önerilerini içermekte ise de 2004 yılından beri güncellenmesi yapılmamıştır.

Bu çalışma, Atılım Üniversitesi (1996'da Atılım Vakfı tarafından kurulmuştur), YM programındaki Yazılım Mühendisliği alan derslerindeki örtüşmenin tespit edilmesi ve SWEBOK kılavuzunun son sürümüne göre ders içeriklerinin çözümlenmesi ve kılavuza uyarlanması gereksinimi üzerine başlatılmıştır. Bu makale her ne kadar Türkiye'deki belli bir YM müfredatına yönelik iyileştirme ve geliştirme çalışmasını kapsasa da, aynı yöntemin diğer YM programlarına uygulanmasında bir engel bulunmamaktadır. YM müfredatlarının değerlendirilmesi ve geliştirilmesine yönelik başka çalışmalar da vardır (bkz.[11-15]). Bu çalışmanın öncekilerine göre farkı ve yeniliği, çözümlenen müfredatın 2014 yılındaki en son SWEBOK 3.0 sürümüne göre yapılmış olması ve daha derin ve nicel bir karşılaştırma içermesidir.

Makalenin bundan sonraki bölümleri şu şekilde düzenlenmiştir. Bölüm 2'de literatürdeki ilgili çalışmalar özetlenmiştir. Bölüm 3'de SWEBOK kapsamında Atılım Üniversitesi YM müfredatının değerlendirilmesi ve geliştirilmesi için yaklaşımlar anlatılmıştır. Son olarak da, Bölüm 4'de genel değerlendirmeler ve ileriye yönelik çalışmalarla ilgili yönlendirmeler verilmiştir.

2 İlgili çalışmalar ve arka plan

Bu bölümde SWEBOK özetlenecek ve kısa bir tarihsel gelişimi anlatılacaktır. Daha sonra, ilgili çalışmalar ve Türkiye'deki YM eğitimi gözden geçirilecektir.

2.1 SWEBOK

SWEBOK kılavuzunun ilk deneme sürümü [7] 2001 yılında yayımlanmıştır. Daha durağan olan ikinci sürüm [8] ise 2008 yılında çıkmıştır. Hem ilk hem de ikinci sürümlerde YM'nin tüm alanlarının içeren toplam 10 adet bilgi alanı kümesi bulunmaktadır. Bu makalede 2014 yılında yayımlanan ve toplam 15 bilgi alanını kapsayan SWEBOK 3.0 [9] sürümü kullanılmıştır. Tablo 1.'de her üç sürümdeki bilgi alanları toplu ve karşılaştırmalı olarak görülmektedir.

Son sürümdeki en önemli değişiklik, bilgi alanları arasına, YM Uzmanlık Uygulamaları, YM Ekonomisi ile Hesaplama, Matematik ve Mühendisliğin Temellerinin eklenmiş olmasıdır.

Tablo 1- SWEBOK'un evrimi ve sürümlerdeki bilgi alanları

Deneme Sürümü (#1), 2001 [2]	İkinci Sürüm, 2004 [3]	Sürüm 3.0, 2014 [4]
1. Yazılım gereksinimleri 2. Yazılım tasarımı 3. Yazılım inşası 4. Yazılım sınama 5. Yazılım bakımı 6. Yazılım konfigürasyon yönetimi 7. YM yönetimi (mühendislik Yönetimi) 8. YM süreçleri 9. YM araçları ve yöntemleri 10. Yazılım kalitesi	Deneme sürümündeki bilgi alanları ile aynıdır (bazı iyileştirmeler ve düzeltmeler yapılmıştır)	1. Yazılım gereksinimleri 2. Yazılım tasarımı 3. Yazılım inşası 4. Yazılım sınama 5. Yazılım bakımı 6. Yazılım konfigürasyon yönetimi 7. YM yönetimi 8. YM süreçleri 9. <u>YM modelleri ve yöntemleri</u> (YM araçları ve yöntemleri yerine) 10. Yazılım kalitesi 11. <u>YM uzmanlık uygulaması</u> (profesyonellik) (yeni) 12. <u>YM ekonomisi</u> (yeni) 13. <u>Hesaplama temelleri</u> (yeni) 14. <u>Matematik temelleri</u> (yeni) 15. <u>Mühendislik temelleri</u> (yeni)

2.2 İlgili Çalışmalar

SWEBOK kılavuzunun YM alanları için, soru-tabanlı anket düzenleme [16], YM profesyonellerinin eğitimi [17] ve YM eğitimi [11-15] gibi farklı amaçlara yönelik kullanımına literatürde rastlanmaktadır. Son kategorideki yani YM eğitimi ile ilgili çalışmalar bu makalenin kapsamına yakın olup, aşağıda kısaca özetlenecektir.

SWEBOK ve yine eğitim amaçlı bir girişim olan YM Eğitim Bilgisi (SEEK) arasındaki bir ön eşleştirme çalışması [11]'de verilmektedir. SEEK [18] 2003 yıllarında IEEE-CS'in gerçekleştirdiği YM eğitimi ile ilgili bir girişimdir. Bu ön eşleştirmede, her iki bilgi alanı arasında temelde herhangi bir "düşünce ekolu" farkı olmasa da, sözlük, ayrıntı seviyesi, alanı parçalama yaklaşımı ve alana eklenen konularda bazı farklılıklar olduğu tespit edilmiştir.

Gerçek projelerin gereksinimleri ile YM derslerindeki projelerde kazanılan bilgi ve beceriler arasındaki farklılıklarının bir çözümlemesi [12]'de verilmektedir. Bu çalışmada, yazarlar SWEBOK kılavuzunu Bloom'un sınıflandırmasından yararlanarak "Yazılım Mühendisliğine Giriş" dersine uygulamışlar ve derste geliştirilmesi gerekli yerleri belirlemişlerdir.

Hindistan'daki *Siemens Corporate Development* Merkezindeki uygulayıcılar, endüstrinin gereksinimlerine uygun olarak, SWEBOK-tabanlı "*FOundation CUrriculum for Software Engineers*" (FOCUS) isimli bir YM müfredatı geliştirmişlerdir [13].

Bir başka çalışmada [14], SWEBOK, ABD'deki *Securities Industry Automation Corporation (SIAC)* gibi bir yazılım geliştirme evinde, YM kapasite geliştirme programının uyarlamasını hızlandırmak amacı ile kullanılmıştır. Bu çalışmada ayrıca, SWEBOK yardımı ile, yazılım süreci iyileştirme girişimine yardımcı olmak üzere, iş gücü gelişiminin düzenlenmesi; Southern Methodist University (SMU)'daki bir

Yüksek Lisans programının geliştirilmesi verilmiş ve çalışma sonucunda yenilikçi bir eğitim programı önerilmiştir.

Son olarak, SWEBOK kılavuzunun YM eğitimindeki denemeleri [15] çalışmasında verilmiştir. Bu çalışma aynı zamanda, SWEBOK projesi ile alandaki diğer girişimler arasındaki bağlantıları özellikle bilişim müfredatları kapsamında betimlemektedir.

2.3 Türkiye’de YM eğitimi

Türkiye’deki YM endüstrisindeki büyüme ve gelişmeye paralel olarak üniversitelerinden mezun yüksek nitelikli işgücüne talep hızla artmıştır. Buna rağmen, ülkedeki YM lisans programları ancak 2000’li yılların başında açılmıştır. YM talebi bu yıllara kadar ve halen Bilgisayar Mühendisliği ve diğer ilgili alan mezunlarınca karşılanmaya çalışılmaktadır. Türkiye’de 185 üniversitenin yalnızca on-üçünde YM bölüm ya da programları yürütülmektedir. Bunların onu vakıf üniversitelerinin mühendislik, üçü ise devlet üniversitelerinin teknoloji fakülteleri altında yapılmıştır. 2014-2015 Akademik yılında, bu programların 853 olan kontenjanının ancak %80’ine yerleştirme sağlanmıştır [6].

Yazılım endüstrisinin nitelikli mühendise olan gereksinimi bir hayli fazla iken, YM programlarına rağbet şaşırtıcı derecede düşüktür. Bunun nedenlerinin başında, Yazılım Mühendisliğinin, Bilgisayar mühendisliğine göre toplumda daha az bilinir olması ve vakıf üniversitelerindeki yüksek eğitim ücreti gösterilebilir.

Atılım Üniversitesi dâhil, Türkiye’deki üniversitelerin çoğu, ortak bir Avrupa girişimi olan ve Yüksek Öğrenim kurumlarındaki standartlaşma ve kalite güvencesini destekleyen Bologna sürecini tamamlamışlardır [19].

3 YM müfredatının SWEBOK yardımı ile değerlendirilmesi ve iyileştirilmesi

Bu bölümde öncelikle, Atılım Üniversitesi YM müfredatı özetlenecek ve daha sonra değerlendirme yöntemi ve hedef programa uygulaması ve sonuçları tartışılacaktır.

3.1 Bir Durum Çalışması: Atılım Üniversitesi YM Müfredatı

2005 yılında açılan Atılım Üniversitesi, YM programı bu alanda eğitim veren 13 program arasında olup faaliyetlerini hem lisans hem de lisansüstü düzeyde sürdürmektedir. Program, kalite güvencesi kapsamında, 2014 yılında MÜDEK tarafından 5 yıl süre ile akredite edilmiştir. Her ne kadar bu akreditasyon ile Atılım Üniversitesi, YM müfredatının kalite güvencesi sağlanmışsa da, müfredatın SWEBOK, IEEE, ACM ve diğer profesyonel kuruluşların evrensel önerileri kapsamında iyileştirilmesine her zaman gereksinim olacaktır. Atılım Üniversitesi, YM müfredatı 2012 ve 2014 yıllarında gözden geçirilmiş ve SWEBOK kılavuzunun eski sürümleri bağlamında geliştirilmiştir. Programda SE221 (Yazılım Gereksinim Mühendisliği), SE222 (Yazılım Kalite Güvencesi), SE322 (Yazılım mimarisi) ve

SE344 (Sistem Yazılımlarının Sınanması ve Onaylanması) gibi YM alanlarında dersler bulunmaktadır.

Buna ek olarak, özellikle son yıllarda, bazı YM derslerindeki bazı konularda büyük ölçüde örtüşmeler olduğu şeklinde geri bildirimler alınmaktadır. Yazarlar buradan yola çıkarak, kapsamlı bir çözümlenme ile örtüşen ve gereksiz olarak yinelenen konuları belirlemeye ve aynı zamanda YM derslerini SWEBOK 3.0 kılavuzu önerileri çerçevesinde değerlendirmeye ve sonuçlarını YM derslerini veren öğretim elemanları ile paylaşmaya karar vermişlerdir.

3.2 Değerlendirme Yöntemi

Bu çalışmada yöntem olarak sistematik bir çözümlenme yapabilmek adına, YM araştırma literatüründe, “deşeyleyici” (exploratory) ve “iyileştirme” (improving) durum çalışmaları olarak bilinen “durum-çalışması” yaklaşımı [20] kullanılacaktır. Runeson ve Höst’ün çalışmalarında tanımladığı gibi [5],deşeyleyici ve durumu iyileştirmeye yönelik çalışmalar şu şekilde tanımlanmaktadır: (1)deşeyleyici çalışmalar: ne olduğunun bulunması, yeni anlayışların ve fikirlerin aranması ve yeni araştırma için varsayımların yaratılması, (2) durum çalışmalarını iyileştirme: çalışılan olayın belli bir yönünü geliştirmeye çalışmak.

Keşfe dayalı çalışma aşamasında, bizim yaklaşımımız, SWEBOK kılavuzundaki 15 bilgi alanını ele almak ve nicelik olarak durum çalışmasına konu müfredattaki YM derslerinin bu bilgi alanı ile ne ölçüde örtüştüğünü tespit etmektir. Bu aşamadan sonra, müfredattaki derslerde, hangi YM konularının yeterince kapsamadığı değerlendirilmiş ve yerleri ile yinelenen/örtüşen konular belirlenmiştir. Bu sonuçlar derslere ve ilgili öğretim elemanlarına geri bildirim olarak sunulmuştur.

3.3 Keşfe Dayalı Çalışma Aşaması: Bilgi Alanlarına Göre Değerlendirme

Bölüm 2.3’de tartışıldığı üzere, Bologna süreci Atılım Üniversitesinin tüm bölümlerinde işlemeye başlatılmış ve müfredattaki tüm dersler için Bologna standartlarında ders izlenceleri hazırlanmıştır. Derslerin haftalık konuları bu izlencelerden çıkartılmış ve SWEBOK kılavuzunun bilgi alanları ile eşleştirilmiştir. Şekil 1’de bu işlem gösterilmektedir. Örnekte, SE112 (Yazılım Mühendisliğine Giriş) dersinin içerikleri verilmekte ve matris içerisinde her bir bilgi alanı için, toplam 14 haftalık eğitim süresinde, kaç hafta ayrıldığı verilmektedir.

Weekly Course Outline (SE 112)

Weeks	Topics
1. Week	Software Engineering and Socio-technical Systems
2. Week	Software Engineering and Socio-technical Systems
3. Week	Software Process Models
4. Week	Project Management
5. Week	Project Management
6. Week	Software Requirements
7. Week	Requirements Engineering Concepts
8. Week	Requirements Engineering Concepts
9. Week	System Models
10. Week	Design Concepts: Architectural Design Object-Oriented Design, User Interface Design
11. Week	Testing, Verification and Validation
12. Week	Testing, Verification and Validation
13. Week	Quality Management
14. Week	Configuration Management
15. Week	Final Exam
16. Week	Final Exam

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	total (14 weeks)	14														
SE 112	Introduction to SE	3	2		2		1	2	2		1					

Şekil 1- YM müfredatının 15 SWEBOK bilgi alanına göre bir değerlendirmesi

Tüm dersler için yapılan benzer çalışma Tablo 2'de görülmektedir. Tablodan da anlaşılacağı üzere, YM ders sayısı 13'dür.

Tablo 2- Atılım Üniversitesi YM müfredatının 15 bilgi alanına göre bir değerlendirmesi. (Tablodaki sayılar her bir bilgi alanının ders içerisinde kaç hafta işlendiğini göstermektedir)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	total (14 weeks)	14														
SE 100	Fundamentals of SE			2												2
SE 112	Introduction to SE	3	2		2		1	2	2	1	1					
SE 212	Human Computer Interaction	1	1		1									11		
SE 221	SW Requirements Engineering	14	13		0,5			0,5								
SE 222	SW Quality Assurance	14			2		1				11					
SE 320	SW Project Management	14							12		1		1			
SE 321	Object-Oriented Analysis and Design	14	2	11						1						
SE 322	SW Architecture	13		12							1					
SE 344	Systems SW Validation and Testing	14			12	0,5	0,5			1						
SE 394	Project Orientation	14						3						1		10
SE 399	Summer Practice I	14										14				
SE 494	Senior Project	14	1	2	4	1			2	1	1	1	1			
SE 499	Summer Practice II	14											14			
	Totals	20	28	6	18,5	0,5	2,5	19,5	4	4	15	28	2	21	0	12

Tablo 2 kullanılarak deşeylici deęerlendirmelerin özeti ve bir tartışması aşağıda verilmektedir:

- Çözümleme öncesinde gözlemlendięi gibi, programda YM konuları yeter derecede kapsamakta ve programda her temel YM konusunda, örneęin, yazılım gereksinimleri, ayrı ders bulunmaktadır.
- Gereksinim, tasarım, inşa, sınama, YM Yönetimi, YM Süreç, YM model ve yöntemler, YM kalitesi ve YM profesyonel uygulama konuları iyi bir şekilde kapsamıştır.
- Birkaç bilgi alanının kapsamı yetersiz bulunmuştur. Örneęin, YM bakım, YM konfigürasyon Yönetimi ve YM ekonomisi bilgi alanlarının yeterince kapsamadığı görülmüştür.
- SE494 bitirme projesi dersi dikkat çekmektedir. Yazarların genel kanaati, bu derste hemen hemen tüm YM konuları kapsamakta ve uygulanmaktadır. Ancak,

bu derste öğrencilerden YM bakımı ve konfigürasyon yönetimi konularını proje kapsamına almaları talep edilmemektedir.

- Yazılım ekonomisi bilgi alanında, öğrencilerin, Endüstri Mühendisliği Bölümünce verilen IE305 (Mühendislik Ekonomisi) dersini aldıkları belirlenmiştir. Ancak, bu dersin yazılım ekonomisi alanı için yetersiz kaldığı ve yalnızca genel mühendislik ekonomisi konularını kapsadığı anlaşılmaktadır. Bitirme projesi dersinde, öğrencilerden, yalnızca geliştirecekleri yazılım ile ilgili maliyet kısıtlarını listelemeleri istenmektedir. Bu ise Yazılım ekonomisi alanı için yetersiz bulunmuştur.

3.4 İyileştirme Aşaması: YM Müfredatı için Öneriler

Bölüm 3.3'deki bahsedilen deşeyici araştırma ile yapılan değerlendirmeleri takiben, ilgili üniversite bölümün sürekli iyileştirme komitesi için bir dizi öneri hazırlanmıştır. Bu öneriler YM müfredatının iyileştirilmesine yönelik bulgulardır:

A. YM bakımı için

- Derslerde “Yazılım bakımı” başlığı ile bir haftalık bir yer ayrıldığı görülmektedir. Bunun açılımını yapılması, ve Yazılım bakımı uygulamalı alıştırmalarla desteklenmesi gerekmektedir. Gerçek hayattaki yazılım projelerinde geliştirme zamanının çoğunun Yazılım bakımına ayrıldığı bilinmektedir [21]. Bu alıştırmalar aşağıdaki konuları içermelidir:
 - Adaptif bakım – sistemin deęişen yazılım ortamlarına (VTYS, İşletim Sistemleri) göre deęiştirilmesi
 - Mükemmelleştirici bakım – yazılımın fonksiyonel iyileştirmeleri kapsamında deęişen ya da yeni kullanıcı gereksinimlerini gerçekleştirme
 - Düzeltici bakım – kullanıcının tarafından belirlenen hataları bulma ve giderme
 - Önleyici bakım – olası problemleri önlemek için yazılımın sürdürülebilirliğini ve güvenirliliğini artırma
 - Tersine mühendislik ve deęişim mühendisliği – örneğin, yazılım kodunu tersine mühendislikle UML'e dönüştürme, tersine mühendislik için yazılım görselleştirme araçlarını kullanmak ve programı kavrama

B. YM konfigürasyon yönetimi için:

- 3-4 sınıf öğrencilerinin projelerinde GitHub, SVN gibi kod versiyonlama sistemlerini kullanmaları önerilmektedir. Bu özellikle SE494 Bitirme Projesi dersi için önemlidir. İlk yazarın Atılım Üniversitesi, YM programında verdiği proje bitirme projesi dersi deneyimlerine göre, öğrenciler, herhangi bir kod versiyonlama sistemi kullanmadıklarından, kod senkronizasyonunda önemli problemler yaşamışlardır. Kod versiyonlama sistem kullanımının

artık sıradan ve yapılması gereken bir işlem olduđu ve yazılım endüstrisi için standart bir uygulama olduđu bilinmektedir.

C. YM ekonomisi bilgi alanı için:

- Ekonominin endüstride, Yazılım projeleri için ana faktör olduđu bilinci ile, çeşitli derslerde Yazılım ekonomisinin daha kapsamlı olarak işlenmesi önerilmektedir. Örnek konular olarak, kullanım senaryolarının maliyet-fayda analizi, yazılım kalite faaliyetlerinin yatırım getirisi ve konu ile ilgili daha çok uygulamalı alıştırmalar verilebilir.
- Yazılım süreç geliştirme ve yazılım süreç modelleri daha uygulamalı olarak işlenmeli ve bu sayede öğrencilerin gerçek hayatta karşılaşacakları durumlarda hangi stratejilerin kullanılacağı ile ilgili fikir sahibi olmaları sağlanmalıdır.
- Yaz Stajı dersleri kapsamında ya da yazılım teknoparklarına/firmalarına gerçekleştirilecek eğitim gezileri ile öğrencilerin YM profesyonel uygulamalarını görmeleri sağlanmalıdır. Bu sayede, kuramsal derslerde yaşanması mümkün olmayan, yazılım pratiklerini görme ve yazılım kültürünü öğrenmeleri fırsatı yaratılacaktır.

Bu öneriler bağlamında, MÜDEK akreditasyon sürecinde bölümde kurulmuş olan Sürekli İyileştirme Komitesi'nden bu sonuçların irdelenmesi ve ders içeriklerinde/işlenmesinde revizyonlar yapılması talep edilmiştir.

4 Tartışmalar ve Sonuçlar

Bu makalede SWEBOK 3.0 kılavuzu çerçevesinde, bir YM müfredatının değerlendirilmiş ve iyileştirilmesine yönelik öneriler raporlanmıştır. Sistemik durum çalışması yaklaşımı ve nicel çözümleme yaklaşımı ile müfredattaki dersler SWEBOK bilgi alanları ile saat bazında eşlendirilmiştir. Bu yaklaşım ve çalışma birçok faydalı gözlem ve öneri ile sonuçlanmış ve ilgili YM müfredatının iyileştirilmesine yönelik elle tutulur yararlar elde edilmiştir. Değerlendirilen müfredatla ilgili deneysel bulgular şunlardır: (1) Yazılım bakımı bilgi alanı pratik uygulamalarla desteklenerek kapsamalıdır; (2) 3-4. sınıf derslerindeki projelerde GitHub, SVS gibi bir kod versiyonlama sistemi kullanılmalıdır; (3) Yazılım Mühendisliğine Giriş ve Yazılım Proje Yönetimi derslerinde kısmen değinilen Yazılım ekonomisi bilgi alanı konuları için ayrı bir ders geliştirilmelidir; (4) Yazılım geliştirme araçları ve metotları derslerde daha fazla kullanılmalıdır ve (5) Bitirme Projesi dersindeki projeler endüstriden temin edilmeli ve bu sayede öğrencilerin gerçek yazılım geliştirme ortamına alışmaları desteklenmelidir.

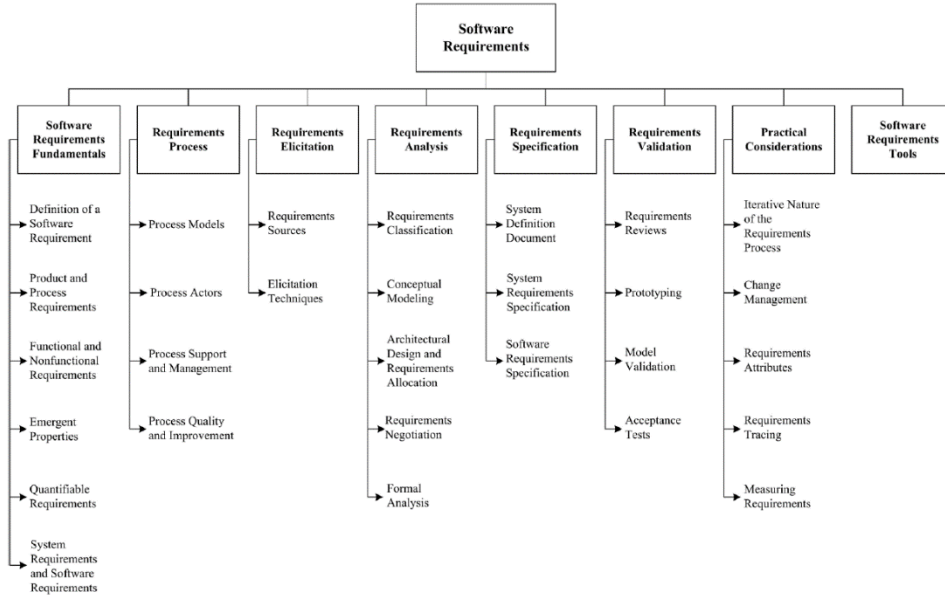
Bölüm 3.4'deki önerilerden ayrı olarak, bir YM programının iyileştirilerek daha çekici ve etkin hale getirilmesi için aşağıdaki noktalar tespit edilmiştir [6]:

- Ankara'da bulunan çok sayıda teknopark ile müfredat geliştirme, staj konularında daha yakın işbirliği kurulmalıdır.

- Müfredat geliştirme çalışmalarına yazılım geliştiriciler ve yazılım firmalarının icra kurulu üyeleri dâhil edilmelidir.
- Önde gelen yazılım firmaları ile ortaklıklar tesis edilip, eğitimde, firmanın yazılım geliştirme ve sınav araçları ve ortamlarından yararlanılmalıdır.
- Müfredat uygulanırken, ders ön-koşullarına titizlikle uyulmalıdır. Örneğin, Yazılım mimarisinden önce, Yazılım Mühendisliğinin temelleri, nesneye-yönelik kavramlar, gereksinim mühendisliği ve yazılım kalitesi konuları işlenmelidir.

Öğrencilerin IBM, Microsoft, Google ve diğer önde gelen firmaların organize ettiği seminerlere katılımları özendirilmelidir. Ayrıca, öğrencilerin ACM ve IEEE gibi kuruluşların düzenlediği yarışmalara etkin katılımları desteklenmelidir.

Bu çalışmanın devamında, SWEBOK 3.0 kılavuzundaki alt bilgi alanları çözümlenmesi yapılacak, ve müfredatın bu alt-alanlara uyumları aynı yaklaşımla tespit edilecektir. Örneğin, “yazılım gereksinimleri” bilgi alanında 8 alt-alan vardır ve bu alt-alanlar Şekil 2.’de görüldüğü gibi, kendi içinde 28 alt-alt-alan bulundurmaktadır. Son olarak da, ileride yapılacak araştırma kapsamında, müfredattaki YM derslerinin hem kuramsal temelleri hem de uygulamaları (örneğin, güncel YM araçları) belli bir dengede bulundurması sağlanacaktır.



Şekil 2- Yazılım gereksinimleri bilgi alanı için konuların bölünmesi [9]

Kaynaklar

- [1] V. Garousi, "Incorporating Real-World Industrial Testing Projects in Software Testing Courses: Opportunities, Challenges, and Lessons Learned," in *Proceedings of the IEEE Conference on Software Engineering Education and Training (CSEE&T)*, 2011, pp. 396-400.

- [2] V. Garousi and A. Mathur, "Current State of the Software Testing Education in North American Academia and Some Recommendations for the New Educators " in *Proceedings of the 23rd IEEE Conference on Software Engineering Education and Training*, 2010, pp. 89-96.
- [3] V. Garousi, "An Open Modern Software Testing Laboratory Courseware: An Experience Report " in *Proceedings of the 23rd IEEE Conference on Software Engineering Education and Training*, 2010, pp. 177-184.
- [4] Organizers: C. Kaner, V. Garousi, and D. Marinov, "The first Software Testing Education Workshop (STEW), co-located with the ICST 2009, Denver, Colorado, USA," <http://www.softqual.ucalgary.ca/events/STEW2009/>, 2009.
- [5] A. Mishra and D. Mishra, "Industry Oriented Advanced Software Engineering Education," *Croatian Journal of Education*, vol. 14, pp. 595-624, 2012.
- [6] A. Mishra and A. Yazici, "An Assessment of the Software Engineering Curriculum in Turkish Universities: IEEE/ACM Guidelines Perspective," *Croatian Journal of Education*, vol. 13, pp. 188 - 219, 2011.
- [7] A. Alain, B. Pierre, D. Robert, and W. M. James, Eds., *Guide to the Software Engineering Body of Knowledge (SWEBOK), version 1*. IEEE Press, 2001, p.^pp. Pages.
- [8] A. Alain, B. Pierre, D. Robert, and W. M. James, Eds., *Guide to the Software Engineering Body of Knowledge (SWEBOK), version 2*. IEEE Press, 2004, p.^pp. Pages.
- [9] P. Bourque and R. E. Fairley, Eds., *Guide to the Software Engineering Body of Knowledge (SWEBOK), version 3.0*. IEEE Press, 2014, p.^pp. Pages.
- [10] Institute for Electrical and Electronic Engineers (IEEE-CS) and Association for Computing Machinery (ACM), "Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," in *A Volume of the Computing Curricula Series*, <http://sites.computer.org/ccse/>, 2004.
- [11] P. Bourque, F. Robert, J. M. Lavoie, A. Lee, S. Trudel, and T. C. Lethbridge, "Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Knowledge (SEEK) - a preliminary mapping," in *Software Technology and Engineering Practice, 2002. STEP 2002. Proceedings. 10th International Workshop on*, 2002, pp. 8-23.
- [12] S. Ludi and J. Collofello, "An analysis of the gap between the knowledge and skills learned in academic software engineering course projects and those required in real: projects," in *Frontiers in Education Conference, 2001. 31st Annual*, 2001, pp. T2D-8-T2D-11 vol.1.
- [13] G. Samarthyam, G. Suryanarayana, A. K. Gupta, and R. Nambiar, "FOCUS: an adaptation of a SWEBOK-based curriculum for industry requirements," *Proceedings of the 34th International Conference on Software Engineering*, Zurich, Switzerland, 2012.
- [14] D. J. Frailey and J. Mason, "Using SWEBOK for education programs in industry and academia," in *Software Engineering Education and Training, 2002. (CSEE&T 2002). Proceedings. 15th Conference on*, 2002, pp. 6-10.
- [15] R. Dupuis, P. Bourque, and A. Abran, "SWEBOK guide an overview of trial usages in the field of education," in *Frontiers in Education, 2003. FIE 2003 33rd Annual*, 2003, pp. S3C-19-23 vol.3.
- [16] V. Garousi and J. Zhi, "A Survey of Software Testing Practices in Canada," *Journal of Systems and Software*, vol. 86, pp. 1354-1376, 2013.
- [17] R. Colomo-Palacios, E. Tovar-Caro, #193, n. Garc, #237, a-Crespo, *et al.*, "Identifying Technical Competences of IT Professionals: The Case of Software Engineers," *Int. J. Hum. Cap. Inf. Technol. Prof.*, vol. 1, pp. 31-43, 2010.
- [18] A. E. K. Sobel, Ed., *Software Engineering Education Knowledge (SEEK), Final Draft* (<http://sites.computer.org/ccse/known/FinalDraft.pdf>. 2003, p.^pp. Pages.
- [19] V. G. Furuzan, "Adaptation to the Bologna Process: The Case of Turkey," *Excellence in Higher Education*, vol. 3, pp. 104-110, 2012.
- [20] P. Runeson and M. Höst, "Guidelines for conducting and reporting case-study research in software engineering," *Empirical Software Engineering*, vol. 14, pp. 131-164, 2009.
- [21] P. Grubb and A. A. Takang, *Software Maintenance: Concepts and Practice*: World Scientific, 2003.