



**EVALUATION OF TECHNICAL SKILLS IN SOFTWARE ENGINEERING
COURSE PROJECTS**

Pınar CİHAN*, Oya KALIPSIZ

Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Bilgisayar Müh. Bölümü, Esenler-İSTANBUL

Received/Geliş: 18.06.2013 Revised/Düzelme: 11.08.2013 Accepted/Kabul: 25.09.2013

ABSTRACT

Today, software engineering is one of the toughest challenges in the software industry and software engineering demand is increasing. However, software engineering industry indicates that, software engineering graduates does not meet the needs of. When new graduates start to job, industry spends time for their education. This waste of time and the low success rate of software projects in industry, causes a very serious cost. In this study, the questionnaire was performed on the projects to students of *Systems Analysis and Design, Software Engineering* and *Agile Software Development* courses at two different university. This questionnaire consists of questions designed to measure students' technical and soft skills. In this paper, Technical skills evaluated for *Systems Analysis and Design, Software Engineering* and *Agile Software Development* courses and these results compared with social skills results. Aim of this study is to determine the capabilities of students to be insufficient skills in software development courses project for improve the quality of software engineering graduates. Thus, more experienced student trained and higher-quality projects will be developed.

Keywords: Software engineering, software engineering education, technical skills, soft skills.

YAZILIM MÜHENDİSLİĞİ DERS PROJELERİNİN TEKNİK YETKİNLİKLERİNİN DEĞERLENDİRİLMESİ

ÖZET

Günümüzde yazılım mühendisliği yazılım sektöründe en zorlu işlerden biridir ve yazılım mühendisliğine olan talep artmaktadır. Ancak Yazılım Mühendisliği Endüstrisi, Yazılım Mühendisliği mezunlarının ihtiyaçlarını karşılamadıklarını belirtmektedir. Yeni mezunlar işe başladıkları zaman endüstriler bunların eğitimleri için zaman harcamaktadır. Bu zaman kaybı ve endüstrilerdeki düşük başarı oranlı yazılım projeleri çok ciddi maliyete sebep olmaktadır. Bu çalışma, iki farklı üniversitede, *Sistem Analizi ve Tasarımı, Yazılım Mühendisliği* ve *Çevik Yazılım Geliştirme* derslerinde grup olarak gerçekleştirilen projeler ile ilgili öğrencilere anket yapılmıştır. Anketler öğrencilerin teknik ve sosyal yeteneklerini ölçmeye yönelik sorulardan oluşmaktadır. Bu çalışmada *Sistem Analizi ve Tasarımı, Yazılım Mühendisliği* ve *Çevik Yazılım Geliştirme* derslerinde uygulanan anketlerdeki teknik yetenek soruları değerlendirilerek elde edilen sonuçlar sosyal yetenek sonuçları ile karşılaştırılmıştır. Bu çalışmadaki amaç, daha verimli yazılım mühendisliği mezunları yetiştirmek için yazılım geliştirme derslerinde gerçekleştirilen projelerde öğrencilerin yetersiz oldukları yetenekleri tespit edip bunları iyileştirmektir. Böylelikle daha tecrübeli öğrenciler yetişmiş olacak ve daha kaliteli projeler gerçekleştirilecektir.

Anahtar Sözcükler: Yazılım mühendisliği, yazılım mühendisliği eğitimi, teknik yetenekler, sosyal yetenekler.

*Corresponding Author/Sorumlu Yazar: e-mail/e-ileti: pkaya@yildiz.edu.tr, tel: (212) 383 57 30

1. GİRİŞ

Yazılımın hayatımızdaki artan önemi ve ülke ekonomilerine etkileri, yazılım mühendisliğine olan ilgiyi hem akademik çevrelerde hem de endüstride artırmıştır. Dünyada birçok üniversitede, Yazılım Mühendisliği Lisans Programları oluşturulmuştur [1]. Günümüzde yazılımlarda karşılaşılan problemler, yazılım mühendisliği eğitiminin önemini vurgulamaktadır. Bu nedenle, yazılım problemlerinin çözümü ve iyileştirmeler için, gelecekteki yazılım mühendislerinin eğitimine odaklanılması gerektiği vurgulanmaktadır [2,3]. Yazılım mühendisliği eğitimi, öğrenciler mezun olmadan ve önemli tasarımı ve uygulama sorumluluklara almadan önce mutlaka bazı pratik deneyimleri kazandırmalıdır [4].

1995 yılında ABD’de yapılan çalışmada yazılım projelerinin ortalama %16.2’sinin başarılı, %31.1’inin tamamlanmadan iptal edildiğini, %52.7’si ise değiştirilmiştir. Tamamlanan projelerin ise yarısından fazlasının tahmin edilen bütçenin %189 üzerinde bir maliyet ile tamamlandığını göstermektedir [5]. 1999 yılında yapılan araştırmada bu projelerin yaklaşık %75’inin ya zamanında tamamlanmadığını ya da iptal edildiğini göstermektedir [6]. 2009 yılında aynı grup tarafından yapılan çalışmada, başarılı projelerin ortalaması %32’ye çıkmasına rağmen halen çok düşük seviyededir. Başarısız projelerin oranı %24, değiştirilmiş projelerin oranı ise %44’tür [7]. 2013 yılında yapılan çalışmada öğrenciler yaptıkları projeleri değerlendirilmiş ve sadece %31’i projesini başarılı bulmuştur [8]. Projelerin iyi planlanamaması kuruma yüklü bir maliyet getirmekte ve iş kaybına yol açmaktadır. Yazılımda yaşanan başarısızlık endüstriler için büyük bir kayıptır.

Yazılım endüstrisi, yazılım mühendisleri mezunlarının çeşitli konularda hazır olmadıkları görülmektedir [9, 10]. Yazılım mühendisliği eğitimi söz konusu olduğunda, endüstrinin ihtiyaçları ve üniversitenin sundukları arasında bir boşluk vardır [11]. Bu boşluğun sebebi endüstrinin, gereksinimleri karşılamayan mezunlar yetiştirildiğini iddia etmesidir [12]. Birçok çalışmada bu boşluğun kapatılması için endüstriler ile üniversitelerin işbirliği yaparak ortak proje yürütmeleri önermiştir [11, 13, 14]. Bu projeler ya sadece endüstrilerde gerçekleştirilen projelerin taklidi olarak gerçekleştirilmekte [15, 16], ya da [13, 14][17-21] çalışmalarda olduğu gibi gerçek dünyadan projeler olabilir. Bu çalışmada değerlendirmeye aldığımız derslerde projeler, endüstrilerde gerçekleştirilen projelerin taklidi olarak gerçekleştirilmektedir. Georgetown Üniversitesinde gerçekleştirilen iki farklı yazılım mühendisliği dersindeki projelerden birinde bireysel çalışma diğeri ise ekip çalışması yapılarak öğrenciler gözlemlenmiştir. Yazılım mühendisliği derslerindeki bu pratik yaklaşım ile yazılım mühendisliği lisans öğrencilerine teknik yazılım mühendisliği prensiplerini öğretmek kadar işbirliğine dayalı becerileri de öğretmek hedeflenmiştir [22]. Diğer bir çalışmada; bilgi teknolojisi şirketlerinden birinin yazılım geliştirme laboratuvarında üniversite çalışanları, öğrenciler ve laboratuvar çalışanları bir araya gelerek birlikte gerçek proje üzerinde çalışmışlardır. Gözlemler sonucunda işbirliği zayıf gruplar belirlendikten sonra zayıf olanlar iyileştirilebilmek için seçilmiştir [13]. Başka bir çalışmada ise Microsoft firmasında yeni işe başlayan 8 geliştirici seçilerek bunlara programlama, hata üzerinde çalışma, test, proje yönetimi, dokümantasyon ve iletişim gibi görevler verilerek gözlemlenmişlerdir. Sonuç olarak yeni üniversite mezunlarının ilk yazılım geliştirme işlerine başladıklarında temeli, zayıf iletişim becerisi ve sosyal naiflikten kaynaklanan birçok problem yaşadıkları gözlemlenmiştir [23].

Bu çalışmanın amacı; derslerde gerçekleştirilen projelerde öğrencilerin yetersiz oldukları yetenekleri tespit edip bunları iyileştirmektir. Çalışmanın 2. bölümünde malzeme ve yöntem, 3. bölümünde analiz ve bulgular, 4. bölümünde ise sonuç ve gelecek çalışmalar sunulmaktadır.

2. MALZEME VE YÖNTEM

2.1. Anketler

Yapılan anketlere Yıldız Teknik Üniversitesi, Bilgisayar mühendisliği bölümü, *Sistem Analizi ve Tasarımı* dersinden 86 öğrenci, *Yazılım Mühendisliği* dersinden 70 öğrenci ve Namık Kemal Üniversitesi, Bilgisayar mühendisliği bölümü, *Çevik Yazılım Geliştirme* dersinden 33 öğrenci yani toplamda 189 öğrenci katılmıştır. Projedeki gruplar derslere göre değişmekte olup 2 ile 5 arasındadır. Önerilen proje konuları ve grup arkadaşları öğrenciler tarafından belirlenmektedir. Öğrencilere yapılan bu anketler, ders projeleriyle ilgili olup 20'şer sorudan oluşmaktadır. Bu anket sorularının bir kısmı öğrencilerin sosyal yeteneklerini (ekip iletişimi, dokümantasyon, süre kısıtlılığı vb.) ölçen sorulardan oluşurken bir kısmı da teknik yeteneklerini (diyagramların kullanımları) ölçen sorulardan oluşmaktadır. Derslere ait örnek anket soruları Çizelge 1'de gösterilmektedir.

Çizelge 1. Örnek anket soruları

SİSTEM ANALİZİ VE TASARIMI SORULARI
Varlık İlişki, Veri Akış ve Yapı Diyagramlarından uygulama süresince yararlandı mı?
Veri tabanı oluşturulurken varlık ilişki diyagramından yararlandınız mı?
Veri akış diyagramında tasarladığımız proje ile geliştirdiğiniz projenin varlık, modül ilişkileri ve veri akışları ne ölçüde paraleldir?
YAZILIM MÜHENDİSLİĞİ SORULARI
Aşağıdaki diyagramlardan hangisi yazılım süreçlerinde ilk oluşturduğunuz diyagramdır?
Ardışıl ve sınıf diyagramını projenin hangi aşamasında kullandınız?
Aşağıdaki diyagramlardan hangisi modelleme ve tasarımı ifade etmek için daha faydalıdır?
ÇEVİK YAZILIM GELİŞTİRME SORULARI
Daha önceki projelerinizi çevik yöntemler ile kıyasladığında aşağıdakilerden hangisi sizin için daha baskın bir farktır?
Test güdümlü yaklaşımı projenizde ne oranda kullandınız?
Kullanıcı senaryosu ve sınıf diyagramlarını projenin hangi aşamasında kullandınız?
ORTAK ANKET SORULARI
Proje sonunda elde ettiğiniz ürünü nasıl değerlendiriyorsunuz?
Proje içi ekip uyumunuz ne ölçüdeydi?
Proje sonunda elde ettiğiniz ürüne müşteri gözünden bakarsanız nasıl değerlendirirsiniz?

2.2. Yöntem

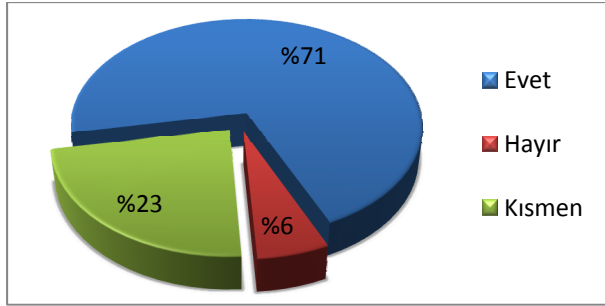
Öncelikle anket kağıtları elektronik ortama aktarılmıştır. Anket sorularına cevap veren öğrencilerin bir kısmı sorulara cevap vermemiş veya bir soruya birden fazla cevap vermiş olabilir bu nedenle sorulardan çıkarım yapılabilmesi için anket verilerine ön işlem adımlarının uygulaması şarttır. Öğrencilerin anket sorularına cevap vermemesi durumunda ilgili sorulara cevap atanması kayıp değer yerleştirme (Missing Value Replacement) işlemine göre yapılmıştır. Kayıp değer yerleştirme yaklaşımına göre cevabı boş olan soruya cevap olarak ilgili soru için en çok cevaplanan şıkkın atanması gerçekleştirilmiştir. Aynı işlem bir soruya birden fazla cevap verilmiş sorular için de gerçekleştirilmiştir ve kayıp değer yerleştirme işlemiyle ön işlem adımı tamamlanmıştır. Daha sonra verilerin frekans analizi yapılarak sonuçlar bölüm 4'de verilmiştir.

3. ANALİZ VE BULGULAR

Sistem Analizi ve Tasarımı, Yazılım Mühendisliği ve Çevik Yazılım Geliştirme ders anketlerindeki teknik yetenek soruları değerlendirilmiştir. Her derste kullanılan diyagramlar, metotlar ve yöntemler farklı olduğu için teknik yetenek soruları ders bazında analiz edilmiştir.

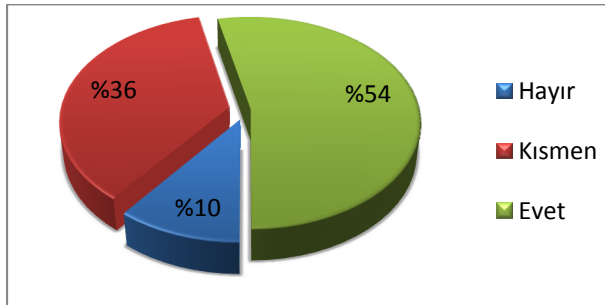
3.1. Sistem Analizi ve Tasarımı Ders Projelerinin Değerlendirilmesi

Sistem Analizi ve Tasarımı ders projesinde öğrencilerin Varlık ilişki, Veri akış ve Yapı diyagramlarından uygulama süresince yararlanılma yüzdeleri Şekil 1’de verilmiştir. Veri akış diyagramları ve Varlık ilişki diyagramları yazılım mühendisliğinin ana gösterimleridir. Yazılım mühendisleri bu diyagramları bilgi sistemini anlamak, geliştirmek ve değerlendirmek için kullanırlar [24]. Bu nedenle de bu diyagramlar projenin büyük bir kısmında uygulanmaktadır. Anket sorularına verilen cevaplar değerlendirildiğinde öğrencilerin %71’i projesinde Veri akış, Varlık ilişki ve Yapı diyagramlarından yararlanmış, %23’ ü kısmen yararlanmış ve %6’sı bu diyagramlardan yararlanmamıştır. Yani derste uygulanması istenilen diyagramlardan yararlanmayan öğrenciler sadece %6 oranındadır.



Şekil 1. Varlık ilişki, veri akış ve yapı diyagramlarından uygulama süresince yararlanma grafiği

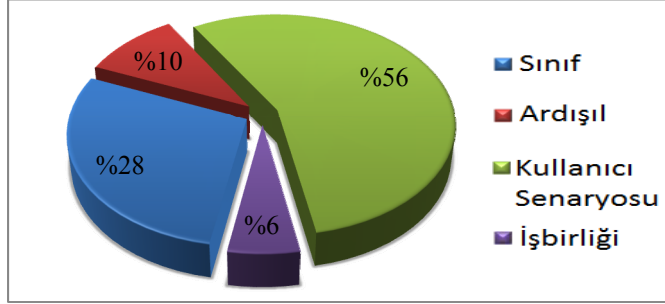
Sistem Analizi ve Tasarımı ders projesinde öğrencilerin veri tabanı oluştururken Varlık ilişki diyagramından yararlanma yüzdeleri Şekil 2’de verilmiştir. Öğrencilerin verdikleri cevaplar değerlendirildiğinde öğrencilerin sadece %10’u projesinde veri tabanı oluştururken varlık ilişki diyagramından yararlanmamıştır. Nitekim veri tabanı oluşturulurken çoğunlukla kullanılan diyagram varlık ilişki diyagramıdır [25].



Şekil 2. Veri tabanı oluştururken varlık ilişki diyagramından yararlanma grafiği

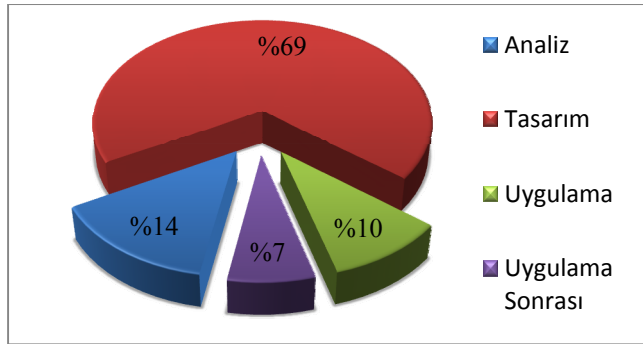
3.2. Yazılım Mühendisliği Ders Projelerinin Değerlendirilmesi

Yazılım Mühendisliği ders projesinde öğrencilerin yazılım süreçlerinde ilk oluşturdukları diyagramların yüzdeleri Şekil 3’de verilmiştir. Bir yazılım mühendisinin yaptığı ilk şey kullanıcı senaryosu diyagramlarını çizmektir [26]. Ayrıca kitabın yazarı işinde ilk olarak kullanıcı senaryosu diyagramlarını oluşturduğunu belirterek, kullanıcı senaryosu diyagramlarının kullanıcılara yardımcı olduğunu ortaya çıkarmıştır [27]. Veriler değerlendirildiğinde öğrencilerin çoğunluğu yazılım süreçlerinde kullanıcı senaryosu diyagramları kullanmıştır.



Şekil 3. Yazılım süreçlerinde ilk oluşturulan diyagramlar

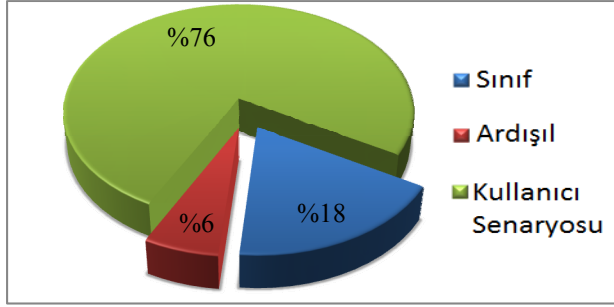
Şekil 4 Yazılım Mühendisliği ders projesinde öğrencilerin ardışıl ve sınıf diyagramlarını projenin hangi sürecinde kullandıklarını göstermektedir. Ardışıl ve sınıf diyagramları tasarım aşamasında kullanılır [28]. Nitekim anket soruları değerlendirildiğinde öğrencilerin %69’u projesinde ardışıl ve sınıf diyagramını tasarım aşamasında kullanmıştır.



Şekil 4. Ardışıl ve sınıf diyagramının kullanım aşaması grafiği

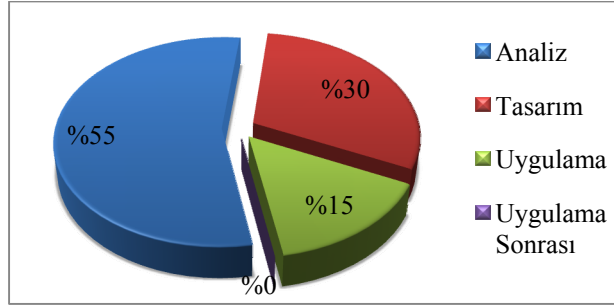
3.3. Çevik Yazılım Geliştirme Ders Projelerinin Değerlendirilmesi

Çevik Yazılım Geliştirme ders projesinde öğrencilerin yazılım süreçlerinde ilk oluşturdukları diyagramların yüzdeleri Şekil 5’de verilmiştir. Daha öncede belirtildiği gibi bir yazılım mühendisinin yaptığı ilk şey kullanıcı senaryosu diyagramlarını çizmektir ve öğrencilerin çok büyük bir kısmı ilk olarak kullanıcı senaryosu diyagramlarını çizmişlerdir.



Şekil 5. Yazılım süreçlerinde ilk oluşturulan diyagramlar

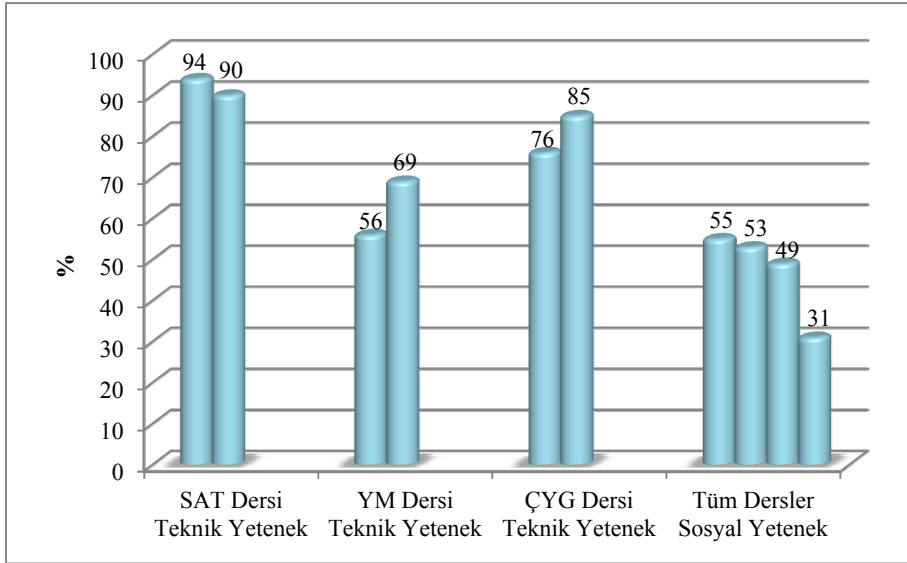
Çevik Yazılım Geliştirme ders projesinde öğrencilerin kullanıcı senaryosu ve sınıf diyagramlarını projenin hangi sürecinde kullandıkları Şekil 6'da gösterilmektedir. Kullanıcı senaryosu diyagramları ve kavramsal sınıf diyagramları projenin analiz aşamasında, tasarım sınıf diyagramları ise projenin tasarım aşamasında kullanılır [29]. Nitekim anket soruları değerlendirildiğinde öğrencilerin %55 analiz aşamasında, %30'u tasarım aşamasında kullanmıştır. Buda diyagramların projenin hangi aşamasında kullandıklarının bilincinde olduklarını göstermektedir.



Şekil 6. Kullanıcı senaryosu ve sınıf diyagramının kullanım aşaması

3.4. Sonuçların Değerlendirilmesi

Anket sorularından bazıları değerlendirildiğinde; *Sistem Analizi ve Tasarımı(SAT)* dersinde öğrencilerin sadece %94'ü projesinde kullanılması gereken diyagramları kullanmamış ve öğrencilerin %90'ı veri tabanı oluşturulurken yararlanılması gereken veri akış diyagramından yararlanmamıştır. *Yazılım Mühendisliği(YM)* dersinde öğrencilerin %56'sı projenin yazılım sürecinde ilk oluşturduğu diyagram kullanıcı senaryosu diyagramıdır ve öğrencilerin %69'u ardışıl ve sınıf diyagramlarını tasarım aşamasında kullanmıştır. *Çevik Yazılım Geliştirme(ÇYG)* dersinde ise öğrencilerin %76'sı projenin yazılım sürecinde ilk oluşturduğu diyagram kullanıcı senaryosu diyagramıdır ve %85'i kullanıcı senaryosu ve sınıf diyagramlarını, analiz ve tasarım aşamasında kullanmıştır. Cihan ve Kalpsız çalışmasında [8], sosyal yetenekleri değerlendirmiş ve elde edilen sonuçlara göre; öğrencilerin sadece %55 dokümantasyonu büyük ölçüde kullanmakta, %53'ü proje içi ekip uyumunu iyi bulmakta, %49'u projedeki süre kısıtlılığı ve çalışanların çalışmamalarından dolayı zorlanmakta ve %31'i süreç sonunda elde edilen ürünü iyi bulmaktadır. Bölüm 3'de değerlendirilmesi yapılan anket sorularının yüzdeleri ve sosyal yetenek sorularının değerlendirilmesi sonucu elde edilen yüzdelerin karşılaştırılmalı biçimde grafiksel gösterimi Şekil 7'de verilmiştir.



Şekil 7. Değerlendirilmesi yapılan anket sorularının karşılaştırmalı biçimde gösterimi

Şekil 7’de görüldüğü gibi öğrencilerin teknik yetenek yüzdeleri sosyal yetenek yüzlerinden daha iyidir.

4. SONUÇ VE GELECEK ÇALIŞMALAR

İki farklı üniversitenin bilgisayar mühendisliği bölümündeki yazılım geliştirme derslerinden olan *Sistem Analizi ve Tasarımı*, *Yazılım Mühendisliği* ve *Çevik Yazılım Geliştirme* derslerinden toplamda 189 öğrenciye ders projeleri ile ilgili anket yapılmıştır. Bu çalışmada öğrencilerin projelerdeki teknik yeteneklerini ölçen sorular değerlendirilmiştir. Elde edilen teknik yetenek sonuçları ile sosyal yetenek sonuçları karşılaştırıldığında, öğrencilerin teknik yeteneklerinin sosyal yeteneklerinden daha iyi olduğu sonucu elde edilmiştir. Gerçekleştirilen projelerdeki teknik yetenek sorularının değerlendirmesi sonucunda oranların yüksek çıkması, öğrencilerin akademide aldıkları eğitimi projelerinde başarılı bir şekilde uyguladıklarını göstermektedir. Elde edilen bu sonuçlar doğrultusunda akademinin sosyal yeteneklere önem verip bu yetenekleri iyileştirilmesi durumunda hem başarılı projelerin oranı artabilir hem de daha kaliteli yazılım mühendisliği mezunları yetiştirilebilir. Gelecek çalışmalarda anketler endüstrilere uygulanarak sonuçlar değerlendirilebilir ve öğrenci profillerinden elde edilen sonuçlar ile karşılaştırılıp çıkarımlar yapılabilir.

REFERENCES / KAYNAKLAR

- [1] Mishra, A., Cagiltay, N. E. and Kılıc, O., “Software engineering education: some important dimensions”, *European Journal of Engineering Education*, 32, 3, 349-361, 2007.
- [2] Saiedian, H., Bagert, D. and Mead, N., “Software Engineering Programs: Dispelling the Myths and Misconceptions”, *IEEE Software*, 19, 5, 35-41, 2002.
- [3] Hilburn, T. and Humphrey, W., “The Impending Changes in Software Education”, *IEEE Software*, 19, 5, 22-24, 2002.

- [4] Saiedian, H., Bagert, D. J. and Mead, N. R., "Software Engineering Programs: Dispelling the Myths and Misconceptions", IEEE Software, 19, 5, 35-41, 2002.
- [5] Standish Group, "Chaos Report", The Standish Group International, 1995.
- [6] Standish Group, "Chaos: Recipe for Success", The Standish Group International, 1999.
- [7] Standish Group, "Chaos Summary 2009: The 10 Laws of Chaos", The Standish Group International, 2009.
- [8] Cihan, P. and Kalipsız, O., "Assessing the Human Factors in Software Development Courses Students Project", International Conference on Education and Educational Technologies, 2013.
- [9] Rusu, A., Rusu, A. and Elloitt R., "Work in Progress: Smoothing the Border between Academic and Professional Software Engineering Environment through Entrepreneurship", Proceeding of 36th ASEE/IEEE Frontiers in Education Conference, 17-18, 2006.
- [10] Garg, K. and Varma, V., "Software Engineering Education in India: Issues and Challenges", IEEE 21st Conference on Software Engineering Education and Training 2008 (CSEET'08), IEEE Computer Society, 110-117, 2008.
- [11] Beckman, K. and Khajenoori, S., "Collaboration Closing the Industry-Academia Gap", IEEE Software, 14, 6, 49-57, 1997.
- [12] Almi, N.E.A.M., Rahman, N. A., Purusothaman, D. et. al., "Software Engineering Education: The Gap Between Industry's Requirements and Graduates' Readiness", IEEE Symposium on Computers & Informatics (ISCI), 542-547, 2011.
- [13] Shami, N.S., "Together apart: An ethnographic study of industry academia collaboration", Proceedings of the CSCW Workshop 2006 on Supporting the Social Side of Large Scale Software Development, 2006.
- [14] Rusu, A., Rusu, A. and Santiago C., "Academia-Academia-Industry Collaborations on Software Engineering Projects Using Local-Remote Teams", ACM Special Interest Group on Computer Science Education(SIGCSE'09), 2009.
- [15] Aybay, H.I., "Yazılım Mühendisliği Dersi için Proje Ağırlıklı ve Problem Çözmeye Dayanan Yeni Bir Yaklaşım", Ulusal Yazılım Mühendisliği Sempozyumu(UYMS'05), 2005.
- [16] Teles, V.M. and Oliveira, C.E.T., "Reviewing the Curriculum of Software Engineering Undergraduate Courses to Incorporate Communication and Interpersonal Skills Teaching", Proceedings of the 16th Conference on Software Engineering Education and Training, 2003.
- [17] Kumar, V., Kinzel, G., Wei, S. et. al., "Multi-University Design Projects". Journal of Engineering Education(JEE), 89, 3, 353-59, 2000.
- [18] Liu, C., "Enriching software engineering courses with service-learning projects and the open-source approach", Proceedings of the 27th international Conference on Software Engineering(ICSE'05), 613-614, 2005.
- [19] Reichlmayr, T.J., "Collaborating With Industry—Strategies for an Undergraduate Software Engineering Program", Proceedings of the international workshop 2006 on Summit on Software Engineering Education(SSEE'06), 13-16, 2006.
- [20] Rusu A., Paglione, M., Snyder, F. et. al., "Overcoming Limited Resources: An Academia-Government Partnership on Software Engineering and Capstone Projects". Proceedings of 37th ASEE/IEEE Frontiers in Education Conference, IEEE Computer Society, 2007.
- [21] Rusu, A., Webb, R., Shanline, D. et. al., "A Multiple-Projects-Multiple-Winners Approach for Teaching Software Engineering", Proceedings of 9th IASTED International Conference on Software Engineering and Applications, 2005.
- [22] Blake, M.B., "A Student-Enacted Simulation Approach to Software Engineering Education", IEEE Transactions on Education, 46, 3, 2003.

- [23] Begel, A. ve Simon, B., “Novice Software Developers, All Over Again”, Proceedings of the Fourth international Workshop on Computing Education Research(ICER’08), 3-14, 2008.
- [24] Kılıç, Ö., Çağıltay, N.E. ve Tokdemir, G., “Yazılım mühendisliği diyagramlarının kullanımındaki bilişsel ve davranışsal özellikler”, II. Ulusal Mühendislik Kongresi, 349-355, 2006.
- [25] Song, I.Y. and Froehlich,K.,“Entity-relationship modeling”,IEEE Potentials, 13, 5, 29-34,1994.
- [26] “Introduction to Software Engineering”, Available from: http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Print_version, [accessed February 15, 2013].
- [27] Fowler, M. and Scott, K., UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison Wesley, 55-56, 1999.
- [28] “Nesneye yönelik analiz ve tasarıma giriş”, Available from: yilmazklicaslan.trakya.edu.tr/teaching/uml_giris.ppt, [accessed February 15, 2013].
- [29] Kalıpsız, O., Buharalı, A. ve Biricik, G., Sistem Analizi ve Tasarımı, Papatya Yayıncılık, İstanbul, 2011.

Industrial Engineering Article
/
Endüstri Mühendisliđi Makalesi