

Yazılım Mühendisliği Dersi İçin Proje Ağırlıklı ve Problem Çözmeye Dayanan Yeni Bir Yaklaşım

Işık AYBAY

Bilgisayar Mühendisliği Bölümü Doğu Akdeniz Üniversitesi Gazimagosa – KKTC

e-posta : isik.aybay@emu.edu.tr

Özet

Bu çalışmada yazılım mühendisliği dersleri için proje temelli ve problem çözme ağırlıklı bir ders verme yaklaşımı anlatılmaktadır. Kuzey Kıbrıs Türk Cumhuriyeti Doğu Akdeniz Üniversitesi Bilgisayar Mühendisliği Bölümü'nde 4 yıldır bu yaklaşımla verilen Yazılım Mühendisliği dersinden örneklerle konu pekiştirilmektedir.

Abstract

In this study, a project-based software engineering course teaching approach using problem solving-type questions in exams is introduced. With examples from a software engineering course which has been taught at the Computer Engineering Department of Eastern Mediterranean University of the Turkish Republic of Northern Cyprus, the concepts mentioned are explained and illustrated.

1. Giriş

Yazılım Mühendisliği, 20 yılı aşan geçmişine karşın gene de diğer bir çok mühendislik dalına göre oldukça yeni bir dal olarak kabul edilmektedir. Bu alan içinde ele alınması gereken konularla ilgili çeşitli önemli çalışmalar yapılmıştır. [1, 2, 7, 8]. Son yıllarda yayınlanan makalelerde özellikle yazılım mühendisliği eğitiminde iş ortamının öğrenilmesinin önemi vurgulanmaktadır. [1, 2, 3]. Kısacası, gerçek hayatta karşılaşılabilecek konulara yakın bir konunun işlendiği yazılım projelerinin, ayrıca problem çözme, takım çalışması ve iletişim becerilerinin önemi ortaya çıkmaktadır.

Kuzey Kıbrıs Türk Cumhuriyeti Doğu Akdeniz Üniversitesi (DAÜ) Bilgisayar Mühendisliği Bölümü CMPE-412 Yazılım Mühendisliği dersi, dört yıldır proje ağırlıklı olarak verilmektedir. Dersin toplam dönem notu içinde grup halinde yapılan dönem projesinin ağırlığı %50 olarak belirlenmiştir. Böylece, öğrenciler dersin dönem projesinin önemini açıkça görmekte ve ders için toplam çalışma süresinin yarısından fazlasını proje çalışmalarına ayırmaktadırlar. Sınavların (bir arasınava ve bir sonsınava)

toplam ağırlığı %50 olarak belirlenmiştir. Sınavlarda problem çözme yeteneklerini geliştirmeye yönelik tasarım sorularına en az %50 ağırlıkla yer verilmiştir.

Bu makalede, yukarıda belirtilen dersin işlenişi ile ilgili bilgiler sunulacak, verilen proje konuları, öğrencilerin grup oluşturması, ara sunuşlar ve son sunuşlar, dersin işlenişi süresince verilen raporlar, sınavlarda sorulan tasarım ile ilgili sorulara örnekler ve öğrencilerin başarı düzeylerinin belirlenmesi ile ilgili olarak izlenen politikalar aktarılacaktır. Tartışılacak konular ile verilecek örneklerin, yazılım mühendisliği konusunda Türk üniversitelerinde verilecek benzer dersler için yararlı bir örnek olacağını düşünmekteyiz.

2. Proje Ağırlıklı Ders Tasarımı

Pressman'ın yüksek satış sayılarına ulaşan ilk kitabının [9] 1982 yılında yayınlanmasından bu yana Yazılım Mühendisliği dersleri için yazılmış başarılı ders kitaplarının sayısı giderek çoğalmaktadır. Ancak bu tür bir dersin işlenmesinde tek bir ders kitabına bağlı kalınamayacağı da açıktır. İnternet üzerinden ulaşılabilecek bazı kaynaklar, örneğin işlev noktaları (function points) yaklaşımının açıklandığı web sayfaları dersi alan öğrencilere okuma çalışması olarak verilmeli, izlenen ders kitabı dışındaki kitapları da okumaları sağlanmalıdır.

Kanımızca dersin başarılı olabilmesi için anahtar rol, verilen dönem projesi ve bunun için yapılan takım çalışmasıdır. Yazılım Mühendisliği derslerinde proje ağırlıklı öğretimin önemi hakkında son yıllarda çeşitli makaleler yayınlanmıştır. [3, 4, 5, 6].

Dönem projesi için uygun olacak konunun seçilmesinde dikkat edilecek iki nokta vardır. İlk olarak, projede öğrencilere mutlaka takım çalışması yaptırılmalıdır. Günümüzde yazılım mühendislerinin karşılaştığı en önemli sorunların başında, proje çalışması için oluşturulan takımlar içindeki uyum ve iletişim sorunu gösterilmektedir. Bu nedenle öğrencilerin :

- a) Bir proje için uyumlu çalışabilecekleri kişilerle takım oluşturmayı,
- b) Takım içinde uygun bir işbölümü gerçekleştirmeyi,
- c) Bir yazılım projesi için takım çalışmasıyla yazılım gereksinimleri belgesi hazırlamayı,
- d) Yazılım projesi süresince teknik gözden geçirme toplantıları düzenlemeyi ve projenin gidişini takım içinde tartışmayı,
- e) Sunuşlar için takım içinde işbirliği yapmayı,
- f) Takım içinde yaşanabilecek kişisel sorunları çözmeyi – tartışmalar, görüş ayrılıkları, diğer dersler nedeniyle uygun toplantı saati belirlemede güçlükler, vb.,
- g) Krizleri aşabilmek için takım dayanışmasının gerekliliğini öğrenmeleri gerekir.

Bunları sağlayabilmek için, projelerin en az 3-4 kişilik öğrenci takımları oluşturularak yapılması önerilmektedir. Bu sayı, yazılım projelerinde gereken kullanıcı arayüzü, veritabanı ve programlama parçaları düşünüldüğünde üçün altında olmamalıdır. Öğrencilerden birinin takım lideri olarak çalışması isteniyorsa, sayı en az dört olmalıdır.

İkinci olarak, proje konusu gerçek is yaşamında o sırada popüler olan konulardan birisi olarak seçilmelidir. Böylece öğrenciler gerçek bir sorunla uğraştıklarını bilecekler, dersin sonunda başarılı bir proje teslim ettiklerinde de yazılım projesi deneyimi yanında önemli bir özgüven kazanacaklardır.

DAÜ Bilgisayar Mühendisliği Bölümü CMPE-412 Yazılım Mühendisliği dersinin dönem projesi için güncel ve öğrenciler mezun olduktan sonra yazılım sektöründe çalışırken gerçek hayatta karşılarına çıkabilecek konular seçilmektedir. Son dört yıl içinde derste verilen proje konuları Tablo 1’de yer almaktadır.

YIL	PROJE KONUSU
2002	Lefkoşa Dr. Burhan Nalbantoğlu Hastahanesi için yazılım otomasyonu
2003	Turizm ve tanıtma için kullanılacak kiosk makinelerine program hazırlama
2004	KKTC Milletvekili seçimleri için hazırlık ve seçim sonuçları programı
2005	Lisansüstü programlar öğrencileri için kayıt ve izleme programı

Tablo 1. DAÜ CMPE-412 Dersinde Son Dört Yılda Verilen Proje Konuları

Proje konularını belirlerken, sınıftaki öğrenci sayısına da bakarak, iki ayrı yaklaşım söz konusu olabilir. Eğer öğrenci sayısı 20 civarındaysa, her takımın belirlenen genel bir çerçeve içinde ayrı bir konuyu ele alması düşünülebilir. Ancak öğrenci sayısı 30’un üzerindeyse, önerimiz oluşturulacak tüm takımlara aynı konunun verilmesidir.

İlk yaklaşıma, yani takımların genel bir çerçeve içinde ayrı ayrı konuları ele almasına bir örnek verelim. Doğu Akdeniz Üniversitesi’nde verilen CMPE-412 Yazılım Mühendisliği dersinde 2002 yılında dönem projesi konusu olarak Lefkoşa Dr. Burhan Nalbantoğlu Hastahanesi için yazılım otomasyonu programları hazırlanması belirlenmiştir. İlk olarak öğrencilerle hastahane ziyaret edilmiş, değişik birimlerde işlerin nasıl yürütüldüğü izlenmiş, yetkililerden alınan bilgiler değerlendirilmiştir. Bunun ardından, önceden oluşturulan üçer kişilik proje takımları aralarında görev paylaşımı yaparak her biri hastahane birer birimi ile ilgili yazılım otomasyonu işlemlerini üstlenmiştir. Bundan sonra takımlar kendi yazılım gereksinimleri belgelerini hazırlayarak dersin öğretim üyesi ve asistanlarına onaylatmışlar, daha sonra hastahaneyi gerektiğinde tekrar ziyaret ederek projelerini geliştirmişlerdir.

Daha sonraki yıllarda dersi alan öğrenci sayısının artması nedeniyle takımların dörder öğrenciden oluşturulması ve tüm takımlara aynı proje konusunun verilmesi kararlaştırılmıştır.

Dönem projelerinin yürütülmesinde her beş takımdan ayrı bir asistan sorumlu olmaktadır, Asistanlar dönem boyunca takımların çalışmasını izlemekte, iki haftada bir ders için belirlenen özel laboratuvar-tasarım saatlerinde teknik gözden geçirme toplantılarına katılmakta, öğrencilerin sunuşlarını dinlemektedir.

Projeler ve buna bağılı olarak derste işlenen konular için 14 haftalık bir dönem için şöyle bir takvim izlenmektedir:

Hafta	Proje ile ilgili olarak yapılan işler	Derste işlenen konular
1	Proje konusunun ve dönem takviminin açıklanması	Dersin amaçlarının ve dönem projesinin önemini anlatılması
2	Proje takımlarının oluşturulması ve sorumlu asistanların belirlenmesi	Yazılım geliştirme süreci, proje yönetiminin temelleri
3	Proje konusu ile ilgili öğretim üyesi ve uzman sunuşları	Yazılım gereksinimleri, proje boyutlarının belirlenmesi
4	Yazılım gereksinimleri belgesinin yazılması	Farklı sistem modelleri, boyut ve maliyet tahmin yöntemleri
5	Proje için ara hedef tarihlerinin belirlenmesi	Yazılım tasarımı, modüler çalışmanın önemi
6	Tasarım -1	İşlevsel çözümlenme, kullanıcı ara yüzü tasarımı
7	Tasarım - 2	Arasınava
8	Ara sunuşlar	Yazılım geliştirme yöntemleri, prototip geliştirme
9	Alınan geri besleme ile düzeltme/değişikliklerin yapılması	Yazılım süreç denetimi
10	Yazılım geliştirme - 1	Yazılımın kalitesinin belirlenmesi-ölçülmesi
11	Yazılım geliştirme - 2	Test teknikleri -1
12	Test	Test teknikleri - 2
13	Son sunuşlar ve proramın çalıştırılması	Tekrar
14	Son sunuşlar ve proramın çalıştırılması (devam)	-

Tablo 2. Ondört Haftalık bir dönem için yazılım mühendisliği ders takvimi.

Kuşkusuz verilen proje konularına bakıldığında, çoğu henüz deneyimsiz birer programcı olan öğrencilerin yaklaşık üç aylık bir sürede dört dörtlük profesyonel bir iş ortaya koyamayacağı açıktır. Buna rağmen her yıl sunulan projelerin çoğu oldukça başarılı olmakta, son sunuşlarda yapılan denemelerde önceden düşünülmeyen ve test edilmeyen bazı bir kaç aykırı veri dışında genellikle istenenlerin yapıldığı gözlenmektedir.

3. Problem Çözmeye Dayalı Sorular

Yazılım mühendisliği derslerinin sınavları için soru hazırlamak, yakın mühendislik disiplinlerine göre daha zordur. Özellikle proje ağırlıklı olarak verilen bir derste, sınav sorularının da proje yönetimi ağırlıklı olması, ve öğrencilerin problem çözmeye yeteneklerini ölçen soruların sorulması gerekir. Aşağıda, Doğu Akdeniz Üniversitesi Bilgisayar Mühendisliği Bölümü'nde verilen CMPE-412 Yazılım

Mühendisliği dersinde 2002-2005 yılları arasında sorulan çeşitli arasınava ve sonsınava sorularından örnekler yer almaktadır. Bu soruların ortak özelliği, matematiksel olarak hesaplanabilecek belli tek bir doğru çözümleri olmamasıdır. Ayrıca problemin tanımında çoğu kez belirsizlikler de yer almaktadır. Sorular, öğrencilerin problemi anladıktan sonra, verilen kısıtlamaları da dikkate alarak, belli bir tasarım ortaya koymalarını istemektedir. Öğrencilerin sorunun çözümüne nasıl yaklaştıkları, kuramsal bilgileri ile problem çözme yeteneklerini nasıl birleştirebildikleri, ve kısıtlı sürede ortaya koydukları tasarımın gücü alacakları notu belirlemektedir.

Örnek soru 1 : (Kasım 2002, arasınava)

Büyük bir üniversite yerleşkesinde üç ayrı binada yer alan bir kütüphane sistemi bulunmaktadır. Kütüphane binalarında çeşitli bilimsel kitap ve süreli yayınlar tutulmaktadır. Üniversitenin öğrenci ve öğretim elemanları, kütüphane yönetimince belirlenen kurullarla kitap ya da dergi ödünç alabilmektedirler. Üniversite yönetimi, kütüphane sisteminde otomasyona gitmek arzusundadır.

Yeni otomatik kütüphane yazılım-yönetim sistemi şu bileşenlerden oluşacaktır:

- Server bilgisayarları : Kütüphanede yer alan dergi ve kitapların veri tabanını tutacak, kitap arama için bir indeksleme ve sorgulama alt sistemini barındıracak, öğrenci ve akademik personelin bilgilerini tutacak, ödünç alınan dergi ve kitaplarla ilgili gerekli bilgileri tutacaktır.
- Ödünç verme noktaları : Server bilgisayarlarına bağlantılı olacak, ayrıca yerel yazıcı ve bar kod okuyucuları olacaktır. Öğrenciler ve akademik personel bar kod bilgisi olan özel kartlarını kullanarak bu noktalardan kitap ya da dergi ödünç alabilecektir.
- Arama bilgisayarları : Her üç binada uygun noktalarda yer alacak, öğrenciler ve akademik personel tarafından kitap ya da süreli yayın bilgisi aramak için kullanılacaktır.

Sizden bu yeni sistemle ilgili olarak istenenler :

- Yeni sistemin ana öğelerini ve bunların yerleşke içine nasıl dağıldıklarını gösteren bir çizim sununuz,
- Bu sistem için hangi yazılım modüllerinin gerektiğini belirleyiniz ve her modül için hangi ana yazılım işlemlerinin hangi sıra ile gerçekleştirileceğini (akış şeması benzeri bir gösterimle) anlatınız,
- Sistem gereksinimlerinin ana hatları ile kısa bir tanımını veriniz, donanım öğelerini ihmal etmeyiniz.

Örnek soru 2 : (Ocak 2005, sonsınava)

Özel bir şirket bir otomatik araç servis hizmetleri sistemi geliştirmek istemektedir. Sistem beş ayrı hizmet noktasında kullanılacaktır. Her hizmet noktası günde 15 araca servis verebilmektedir. Müşteriler hizmet noktalarını normal bakımlar için veya acil durumlar için kullanabilmektedir. Her aracın 5,000 km. Veya 6 aylık süre sonunda normal bakıma gireceğini varsayınız.

Servis isteyen bir müşteri, önce beş hizmet noktasından birine başvurmakta, normal bakım için mi, yoksa acil onarım için mi başvurduğunu belirtmektedir. İlgili hizmet noktasının iş yükü uygunsa, başvuru o gün için kabul edilmekte, günlük yük dolmuşsa, daha sonraki günler için randevu verilmektedir.

Servis için kabul edilen her arabanın bakım-onarım işi aynı gün içinde bitirilmektedir. İşlem sonunda, müşteri cep telefonu veya normal telefonla aranarak aracını teslim almak üzere hizmet noktasına çağrılmaktadır. Müşteriye, araç için yapılan işlemleri ve değiştirilen parçaları içeren bir liste verilmekte, belirlenen toplam bedeli ödemesi istenmektedir. Ödemeler nakit olarak ya da kredi kartıyla YTL cinsinden yapılmaktadır.

Geliştirilecek sistemin ülke çapında hizmet verilen tüm araçlarla ilgili bilgileri tutması ve normal bakımı yaklaşan araç sahiplerini otomatik olarak uarması istenmektedir. Bu sistem için :

- a) Sistemin ana öğelerini belirleyiniz ve genel bir sistem çizimi sununuz.
- b) Bu sistem için gereken ana modülleri ve bunların arasındaki etkileşimi gösteren bir şekil çizin.
- c) Bu sistem için gereken veritabanını ana hatları ile tanımlayınız.
- d) Sistemin sizce en önemli kullanıcı ara yüzlerini kısaca çizimlerle anlatınız.

4. Sonuç

Yazılım mühendisliği eğitiminde derslerin artık takım çalışması deneyimi kazandıracak, güncel konularla ilgili bir problemin dönem projesi olarak ele alındığı yeni bir yaklaşımla verilmesi gerektiği kanısındayız. Buna ek olarak, sınavlarda da öğrencilerin problem çözme yeteneklerini ölçecek ve geliştirecek, gene güncel konulardan alınma soruların sorulması gerektiğini düşünüyoruz.

Bu iki yaklaşımı, yani proje temelli ve problem çözmeye dayalı soruların ağırlık taşıdığı sınavlardan oluşan bir Yazılım Mühendisliği dersini, KKTC'de Doğu Akdeniz Üniversitesi Bilgisayar Mühendisliği Bölümü'nde 4 yıl süreyle verdik. Dersi alan öğrencilerden ve özellikle mezunların çalıştığı işyerlerinden aldığımız olumlu geri beslemeler, bu yaklaşımın başarılı olduğunu göstermektedir.

Kaynakça

- [1]. Yeh R. T., 'Educating Future Software Engineers', IEEE Transactions on Education, Cilt 45, Sayı 1, Şubat 2002, sayfa 2-3.
- [2]. Mayer B., 'Software Engineering in the Academy', IEEE Computer, Mayıs 2001, sayfa 28-35.
- [3]. Gnatz M. vd., 'A Practical Approach to Teaching Software Engineering', 16th Conference on SE Education and Teaching (CSEET) 2003, bildiriler kitabı (Internet üzerinden erişildi).
- [4]. Peslak A. R., 'Teaching Software Engineering Through Collaborative Methods', Issues in Information Systems, Cilt 5, No. 1, 2004, sayfa 247-253.
- [5]. Navarro E. O. vd., 'Teaching Software Engineering Using Simulation Games', International Conference on Software Engineering 2004, bildiriler kitabı (Internet üzerinden erişildi).
- [6]. Stirewalt, R. E. K., 'Teaching Software Engineering Bottom Up', American Society for Engineering Education Annual Conference 2004, bildiriler kitabı (Internet üzerinden erişildi).
- [7]. Sobel A. E. K., ed., 'Computing Curricula – Software Engineering Volume, First Draft', IEEE ve ACM ortak komitesi raporu, Ağustos 2002.
- [8]. Bagert D. J. vd., 'Guidelines for Software Engineering Education, Version 1.0', Ekim 1999, Carnegie-Mellon Univ. Teknik Rapor : CMU/SEI-99-TR032, (Internet üzerinden erişildi).
- [9]. Pressman, R. S., Software Engineering: A Practitioner's Approach, McGraw-Hill, 1982. (Beşinci baskısı yayımlanmıştır)