



ULUBORLU MESLEKİ BİLİMLER DERGİSİ (UMBD)

Uluborlu Journal of Vocational Sciences

<http://dergipark.gov.tr/umbd>

YAZILIM MÜHENDİSLİĞİ DERSİNİN UYGULAMASI İLE İLGİLİ YENİLİKÇİ BİR YÖNTEM

Mustafa Batar^{1*}, Kökten Ulaş Birant²

¹Burdur Mehmet Akif Ersoy Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Burdur, Türkiye.

²Dokuz Eylül Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İzmir, Türkiye.

*Sorumlu Yazar: mbatar@mehmetakif.edu.tr

(Geliş/Received: 10.08.2021; Kabul/Accepted: 25.09.2021)

ÖZET: Yazılım geliştirme projeleri maddi olarak büyük bir götürüye sahiptir ve yüksek miktarlarda yatırıma ihtiyaç duymaktadır. Bilgisayar yazılımı ile ilgili uluslararası somut verilere dayanan maliyetlere bakıldığında zaman; 1985 yılında 150 milyar dolar, 2010 yılında 2 trilyon dolar, 2015 yılında 5 trilyon dolar ve 2020 yılından sonra 7 trilyon doları geçmektedir. Her yıl katlanarak artan maliyetlere rağmen, yazılım projelerinin başarılı bir şekilde geliştirilme oranı çok da yüksek değildir. 2015 yılında hazırlanan “CHAOS” raporuna göre, geliştirilen yazılım projelerinin sadece %17’si tam zamanında, ayrılan bütçede ve verilen isteklere uygun bir şekilde tamamlanmıştır. Projelerin %53’ü ise süre ve bütçe aşımıyla ve ayrıca gereksinimleri tam doğru biçimde karşılayamadan bitirilmiştir. Yazılım projelerinin %30’u ise, geliştirilme evresinde tamamlanamayıp iptal edilmiştir. Bununla birlikte 2020 yılında ise, geliştirilen 3 yazılım projesinden sadece 1 tanesi tam anlamıyla başarılı (zaman, bütçe ve gereksinim bağlamında) olmuştur. Bu derece yüksek giderlere ve ona keza düşük başarı oranına sahip yazılım geliştirme projelerinin daha kaliteli bir yapıya sahip olabilmesi için, geliştiricilere teknik özelliğe sahip bilimsel verilere dayanan bir eğitimin verilmesi gerekmektedir. Bu bağlamda, “Yazılım Mühendisliği” dersleri, bu can sıkıcı durumlarla baş edebilmenin ve üstesinden gelebilmenin yol ve yöntemlerini gösterebilmek ve anlatabilmek için lisans düzeyinde öğrencilere verilmektedir. Bu dersin etkinliğini ve işlevselliğini artırabilmek amacıyla, Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği bölümünde (2014-2015, 2015-2016 ve 2016-2017 eğitim öğretim dönemlerinde) gerçek yazılım projelerine bağlı bir uygulama denenmiş ve bu şekilde ders işlenmiştir. Bu işleyiş içerisinde, bu projeler gerçek müşteriler (yazılım firmaları veyahut yazılıma ihtiyaç duyan şirketler, şahıslar, vb.) ile beraber yapılmıştır. Tasarlanan ve gerçekleştirilen bu projeler ışığında, kâğıt üstünde teorik ders gibi görünen “Yazılım Mühendisliği” dersi, öne sürülen bu yenilikçi uygulama ile daha pratik ve öğrencilerin daha aktif bir şekilde katılım sağlayabildiği, gerçek yaşam sorunlarına ve gerçek yazılım problemlerine cevap verebilen daha fonksiyonel bir ders hâline getirilmiştir. Bu çalışmada, “Yazılım Mühendisliği” dersi kapsamında ortaya konan bu yenilikçi yöntemin amaç, kapsam, işleyiş, yaklaşım ve hedefleri kademe kademe somut ve elle tutulur gerçek vaka analizlerinden yararlanılarak detaylı bir şekilde anlatılmıştır.

Anahtar Kelimeler: Yazılım Mühendisliği, Yazılım Geliştirme Projeleri, Bilgisayar Yazılımı.

AN INNOVATIVE METHOD FOR PRACTICE OF SOFTWARE ENGINEERING COURSE

ABSTRACT: Software development projects have a large financial burden and require large amounts of investment. Considering the costs based on international concrete data about computer software; 150 billion dollars in 1985, 2 trillion dollars in 2010, 5 trillion dollars in 2015 and it exceeds 7 trillion dollars after 2020. Despite the costs that increase exponentially in each year, the successful development rate of software projects is not very high. According to the “CHAOS” report prepared in 2015, only 17% of the developed software projects were completed on time, within the allocated budget and in accordance with the requests. On the other hand, 53% of the projects were completed over time and budget, and also without meeting the requirements properly. 30% of the software projects could not be completed during the development phase and were cancelled. In addition to this,

in 2020, only 1 of the 3 software projects, which have been developed, was fully successful (based on time, budget and requirements). For that software development projects with such high expenses and low success rate can have a better quality structure, an education with technical requirements and specifications is needed to provide for software developers based on the scientific data. In this context, “Software Engineering” courses are given to undergraduate students in order to show and explain the ways and methods of dealing with and overcoming these annoying situations. In order to increase the effectiveness and functionality of this course, in the department of the computer engineering at Dokuz Eylül University (2014-2015, 2015-2016 and 2016-2017 academic semesters), a real-software-projects-based application was tried/tested and the course was taught in this way. In this process and progress, these projects were carried out and applied with real customers (software companies or individual/companies that need software/software projects, etc.). In the light of these designed and implemented projects, the “Software Engineering” course, which seems like a theoretical course on the paper, gets to be a more practical and more functional course that allows students to participate more actively, and that respond to real life troubles and real software problems, with the contribution of this suggested innovative application. In this study, the aim, scope, operation, approach and objectives of the innovative method, introduced within the scope of the “Software Engineering” course, have been explained in detail by using concrete and tangible real case analyzes step by step.

Keywords: Software Engineering, Software Development Projects, Computer Software.

1. YAZILIM MÜHENDİSLİĞİ

Yazılım mühendisliği, yazılımın geliştirilmesi için gerekli mühendislik işlerini yerine getiren bir anabilim dalıdır [1]. Bilgisayar yazılımı da, bilgisayar ya da yazılım mühendisleri tarafından gereksinimleri ortaya çıkarılan, ihtiyaç analizi yapıp belirtim ve teknik dokümantasyonu hazırlanan, iç ve dış mimarisi tasarlanan, ilgili kaynak kodları, değişkenleri, metotları, sınıfları, modülleri ve bunlar arasındaki ilişkileri yazılan, gerekli sınama, ayrıştırma, birleştirme ve test işlemleri yapılan, gerektiği zaman bakım, onarım ve güncelleştirme operasyonları gerçekleştirilen, kendine ait çeşitli dokümantasyon işleri ve kullanım kılavuzu olan ve ayrıca, içerisinde rakamsal ve metinsel veriler ve de resim, fotoğraf, ses ve video gibi multimedya araçları bulunan bir ürün, bir hizmet ya da bir projedir [1].

Yazılım mühendisliği, tasarımı ve sistem uygulamasını içermek zorundadır. Bu tasarım ve uygulama işlemi, içinde bulunduğumuz toplumdan başlayarak, binalardan biyoteknolojiye, oyunlardan hükümete ve sporlardan denizaltılara kadar birçok parametreye (değişkene, bilinmeyene) bağlıdır [2]. Yazılım mühendisliği, çevremizde her yerde ve her taraftadır. Hem yazılımın hem de sistemin bulunduğu Google, Facebook, Twitter ve YouTube gibi siteler güçlü birer yazılım mühendisliğine gereksinim duymaktadırlar. Çünkü, bu siteler çok sayıdaki ziyaretçiye etkin bir şekilde cevap verebilmeli, ciddi boyutlarda bulunan verileri elinde etkin bir şekilde bulundurabilmeli, kullanabilmeli ve 7/24 çalışarak aksaklıklara karşı bakım/onarım işlemlerini anında yapabilmelidirler [2].

“Halo” gibi oyunlar yazılım mühendisliği çalışmalarına gereksinim duymaktadırlar. Yazılım mühendisleri; oyun tasarımı için sofistike grafikler, hızlı animasyon, modelleme, görüntü işleme, yorumlama ve internet etkileşimi için gerekli mühendislik işlerini yerine getirmektedirler. Wii, PlayStation ve Xbox gibi oyun imkânı sunan konsollar da yazılım mühendisliği işlerine ihtiyaç duymaktadırlar [3].

Yazılım mühendisliği, kritik sistemlerin (sağlık, savunma sanayi, havayolları, vb.) geliştirilebilmesi ve kontrol altına alınabilmesi için bilim dünyasında büyük bir rol oynamaktadır. Bu kritik sistemlerin arasında, havacılık ya da savunma için bir uydu yazılımı veya endüstri sanayide, madencilikte ya da su ıslahında kullanılan robotlar için bir yazılım olabilmektedir. Bu yazılımlar insan sağlığı açısından da bizlere fayda sağlamaktadırlar. iPhone’lar gibi mobil cihazların içindeki yazılımların ve uygulamaların geliştirilmesi ve

gerektiği zaman güncellenebilmesi için de yazılım mühendisliği işlerine ihtiyaç duyulmakta ve yazılım mühendisleri bu amaçla çalışmaktadırlar [4].

Yazılım mühendisliği, mühendislik kavramlarının yazılım geliştirmeye uygulanmasıdır. Temel amacı, yazılımın oluşturulması, iyileştirilmesi ve bakımınıdır. Yazılım mühendisliği, bir program üzerinde çalışırken donanım ve yazılım ortamı gibi mühendislik yönlerini dikkate alır. Yazılım mühendislerinin iş tanımları genellikle yazılım geliştiricilerle büyük ölçüde örtüşürken, yazılım mühendisleri ve yazılım geliştiricileri aynı değildir. Temel fark, yazılım mühendislerinin mühendislik kavramlarını ve ilkelerini yazılım geliştirmeye uygulamalarıdır. Mühendisler, yazdıkları kodun ötesini ve programın bulunacağı ortamı her daim düşünürler [5].

En önemli bileşenlerinden biri olan bilgi teknolojisi ve yazılım sistemleri, birçok ülkenin ekonomik kalkınmasında büyük rol oynamaktadır. Günümüzde yazılım sistemleri, finansal sistemler, ticari sistemler, sağlık bilgi sistemleri, havayolu taşımacılığı, telekomünikasyon sistemleri, otomotiv mühendisliği vb. günlük hayatın hemen hemen her alanında kullanılan bilgisayar sistemlerinin çok önemli ve kritik bir bileşenini oluşturmaktadır. Yazılım mühendisliği bilim dalı, 1968’den beri var olan yeni bir mühendislik disiplini ve uzun yıllar boyunca mühendisliğin diğer dallarında oluşturulmuş mühendislik ilkelerini kullanarak yazılım sistemlerinin tasarımını, üretimini ve işletimini amaçlamaktadır. Yazılım mühendisliği, hemen hemen tüm disiplinlerde uygulamaları olan bir alan olarak karşımıza çıkmaktadır [6].

2. YAZILIM MÜHENDİSLİĞİ DERSİ

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde, “Yazılım Mühendisliği” dersi kapsamında genel olarak öğrencilere bir yazılım projesinin hangi evrelerden geçerek geliştirildiği gösterilmektedir [7]. Dersin verildiği eğitim-öğretim dönemi boyunca; ilk hafta, öğrencilere yazılım mühendisliğinin genel yapısı ve mekanizması anlatılmaktadır. İkinci hafta, yazılım projesi ile ilgili olarak gereksinim (ister, ihtiyaç) analizinin nasıl yapılması gerektiği öğrencilere gösterilmektedir. Üçüncü hafta, geliştirilen yazılım projesi ile ilgili olarak iç ve dış (iç ve dış yazılım mimarisi) tasarımın ne şekilde olması ve yapılması gerektiği ve ayrıca, hangi tasarım geliştirme yöntemlerinin uygulanması gerektiği öğrencilere aktarılmaktadır. Dördüncü hafta, yazılım geliştirilirken bakım, onarım, sına ve raporlama işlemlerinin hangi hassasiyetle yapılması gerektiği sınıfta öğrencilere uygulamalı bir şekilde gösterilmektedir. Beşinci ve altıncı hafta, yazılım geliştirme yöntemleri ayrıntılı bir şekilde ağır-siklet ve hafif-siklet olmak üzere örnek projeler esas alınarak öğrencilere anlatılmaktadır. Yedinci hafta, yazılım proje yönetimi ile ilgili olarak süreç planlaması ve tahminleme üzerinde durulmakta ve örnek vakalar analiz edilmektedir. Sekizinci hafta, ölçme ve değerlendirme ile ilgili olarak bir yazılım projesinin karmaşıklığının ve boyutunun ölçümleme işlemi yapılmakta ve matematiksel formüller kullanılmaktadır. Dokuzuncu hafta, yazılım kalitesi ve CMII (Capability Maturity Model Integration) kavramları detaylandırılmaktadır. Onuncu hafta, önemli tasarım desenleri üzerinde durularak nesneye dayalı programlama ve tasarım mantığı öğrencilere örneklerle beraber gösterilmektedir. Bir eğitim-öğretim dönemi boyunca dersin haftalık işleyişine ek olarak, “Yazılım Mühendisliği” dersi kapsamında dönem boyunca iki defa, piyasada ya da akademide çalışan yazılım mühendisliği ile ilgisi olan profesyoneller ile öğrenciler buluşturulmakta ve gerçek dünyada neler olup bittiği birinci ağızdan sınıfta anlatılıp dinlenmekte ve soru-cevap yapılmaktadır [7].

3. YAZILIM MÜHENDİSLİĞİ DERSİNİN UYGULAMASI

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde, “Yazılım Mühendisliği” dersinin uygulama saatlerinde genel olarak öğrencilerin gerçek bir müşteriyle yaptıkları görüşme sonucunda istenilen yazılımın nasıl geliştirilmesi gerektiği – aşama aşama – öğrencilere gösterilmektedir. Bu uygulama çerçevesinde, yazılım projesinin dersin verildiği eğitim-öğretim dönemi boyunca uygun bir şekilde tasarlanıp geliştirilebilmesi için, öğrenciler ilk olarak 5 ya da 6 kişi olacak şekilde kendi gruplarını oluşturmaktadırlar. Aşağıdaki tablolarda 2014-2015 (Tablo 1), 2015-2016 (Tablo 2) ve 2016-2017 (Tablo 3) eğitim öğretim dönemlerine ait oluşturulan gruplar, ilgilendikleri müşteriler ve yapılan projeler yer almaktadır.

Tablo 1. 2014-2015 eğitim öğretim dönemi yazılım mühendisliği dersinin uygulama grupları.

Grup Numarası	Grup İsmi	Müşteri	Geliştirilen Yazılım
1	PASBAS	Destek Otomasyon Endüstriyel Alternatif Enerji	Solaris Güneş Enerjili Aracı Verim Analiz Yazılımı
2	PowerSoft	Günay Kafe	Kafe Hesap Otomasyonu
3	Qualinno	Kibele PIMS Industrial IT & Automation Incorporated Company	Software Development on Quality Control
4	PentagonSoft	PAGOS TRAVEL	PagosVize
5	SKATNESS	Özel Tınaztepe Hastanesi	Hastane Randevu Sistemi
6	Pear Software	DIGITAL BÜRO MAKİNELERİ – UTAX	UTAX PRODUCT TRACKING SOFTWARE
7	PROSOFT	Karşıyaka Nüfus Müdürlüğü	Numaratör Yapımı
8	Business Software	Semafor teknoloji	“Sixt rent a car” android uygulaması
9	BYE-CS	Unividyo	Oteller arası iletişim projesi
10	Softrick.co	innolife Mekatronik Mühendislik ve Danışmanlık	Online Patient Appointment System
11	BC-COM	Fitsport Center	Spor merkezi için web sitesi
12	Snow Game Development	İzmo Bilişim	Flash Game Development
13	PHOSYS	Bornova Cinemaximum	Sinema Otomasyonu
14	UHIEM Automation	HERMES IPS Danışmanlık	Ambalaj Üretim Otomasyonu
15	AEGA BİLİŞİM	İnceoğlu Unlu Mamulleri	Adisyon Yazılımı
16	BMW SOFTWARE	MEGAfitGYM	Tourniquets Automation System

Tablo 2. 2015-2016 eğitim öğretim dönemi yazılım mühendisliği dersinin uygulama grupları.

Grup Numarası	Grup İsmi	Müşteri	Geliştirilen Yazılım
1	NYLON Software Solutions	İz Kent 7 No’lu Aile Sağlığı Merkezi	Hasta Takip Sistemi
2	ScorpTech	T.C. Ziraat Bankası Hasan Tahsin Şubesi	Mobil Sıramatik
3	TEKMESOFT	DEÜ Bilgisayar Mühendisliği Bölümü	Mobil “lectures” Uygulaması
4	Deuterium	EMAKİNA	Versiyon Kontrol Sistemi
5	AnkaTech	EMAKİNA	Tekstil Alışveriş Sitesi
6	SkySoft	Nucleo Ar-Ge Yazılım Donanım Ltd. Şti	Imaginary Waiter
7	EgeSoft	KARINCA Otomasyon	Su Kontrol Sistemi

8	DELTASOFT	Casa Mia	Restoran Otomasyon Sistemi
9	SPY SOFTWARE TECHNOLOGIES	İnceoğlu Unlu Mamulleri	Restoran Muhasebe Sistemi
10	DELTOID SOFTWARE	VENÜS Veteriner Kliniği	Veteriner Takip Sistemi
11	MSS	Media Markt	Ürün Talep Analiz Sistemi
12	Easybug	MİMEKS Gayrimenkul Değerleme Hizmetleri Ltd. Şti.	Gayrimenkul Otomasyon Sistemi
13	SOFTWRITE SOFTWARE	Game City	Oyun Rezervasyon Sistemi
14	KEEP CALM SOFTWARE	Üstün Prodüksiyon	Üstün Prodüksiyon Sitesi
15	DAMLA YAZILIM	NEMTAŞ DENİZCİLİK	Lojistik Otomasyon Sistemi
16	SURFWARE SOFTWARE	Spr Restaurant & Bar	Mobil Restoran Otomasyonu

Tablo 3. 2016-2017 eğitim öğretim dönemi yazılım mühendisliği dersinin uygulama grupları.

Grup Numarası	Grup İsmi	Müşteri	Geliştirilen Yazılım
1	COBRA SOFTWARE	Stackpole International	KOI Management System
2	POWTECH SOFTWARE	Günay Kafe	Kafe Personel Otomasyonu Mobil Uygulaması
3	PineApple	Domino's Pizza	Pizza Servisi Otomasyon Sistemi
4	portatech	LemonSOFT	Çocuklara kod yazmadan program yapmayı sağlayan mobil uygulama
5	MacroSoft	EDVAN VANTİLATÖR SANAYİ ve TİCARET LTD. ŞTİ.	Ürün Dağıtım Ağı Takip Sistemi
6	Lotus Game	S&I Law Office	Law Office Automation
7	Alliance Software	Ottoman Nargile Atölyesi	Nargile Atölye Yazılımı
8	Zagreussoft Company	Ericsson	Employee Follow System
9	WHITESOFT	Seferihisar Necat Hepkon Devlet Hastanesi	Hastane Otomasyon Sistemi
10	MOBİGO	Bağcı Tütün	Şirketin ürünlerinin stok kontrolü ve satışını denetleyen yazılım
11	Pear Software	Windoform Kapı Pencere Aksesuarları Sanayi Ticaret Ltd. Şti.	Tedarik zincir sistem yazılımı
12	Callisto	Miss Coffee	Product selling and Publicity website
13	KASHZILLA	KARACA SİNEMASI	Sinema Yönetim ve Otomasyon Sistemi
14	PHARAOH	ASTRON-Yaşar Bilgi İşlem ve Ticaret A.Ş.	Pınar Su için stok takip yazılımı
15	Change Software	BeerAber	Restoran otomasyonu
16	Pixel Inc.	Cummins Filtration	Motor yakıt filtresi üretim hattında veri toplama ve raporlama yazılımı
17	PENTASOFT	EGEMEN BİLGİSAYAR TEKNOLOJİLERİ	TECHNICAL SERVICE TRACKING SOFTWARE
18	PlusSoft	DEÜ Bilgisayar Mühendisliği Bölümü	Mobil Oyun

19	APPOTECH	TIQ TAQ COFFEE	Kafe Otomasyonu ve Yönetim Sistemi
20	EAGLE SOFTWARE	REY OTOMOTİV	ARAÇ-MÜŞTERİ TAKİP SİSTEMİ
21	Lö Soft	ERBAŞ MÜHENDİSLİK	Email System
22	Veritas Software	myNEVA GAYRİMENKUL	Satış için Masaüstü/Mobil/Tablet Uygulaması

Öğrenciler, gruplarını oluşturduktan sonra kendilerini birer yazılım firması gibi tanıtan raporlarını – Şekil 1 – teslim etmektedirler.

Date: 05.11.2016


1- Company Name
POWTECH SOFTWARE

2- Logo
Our company name is abridgment of Power of Women in Technology and logo is first letter of Women.

3- Slogan
"Your Technology Power"


4- Mission
Our mission is to provide quality and reliable solutions which supply ideally the needs of customers about developing software and hardware and system integration.

5- Vision
Our vision is to be reputable and leader company which provides software and hardware solutions to problems of customers for every sector in Turkey and all over the world.



Şekil 1. Yazılım firması tanıtım raporu.

Gruplar, yazılım firması oluşturma işleminin ardından yazılımsal bir ürüne ihtiyaç duyan gerçek bir müşteriyle bir araya gelerek tasarlanacak ve geliştirilecek yazılım ile ilgili istekleri (ihtiyaçlar, gereksinimler) alıp gereksinim raporunu – Şekil 2 – sunmaktadırlar.



CALLISTO SOFTWARE SYSTEMS
ENGINEERED TO PERFECTION

CLASSIFICATION OF REQUIREMENTS

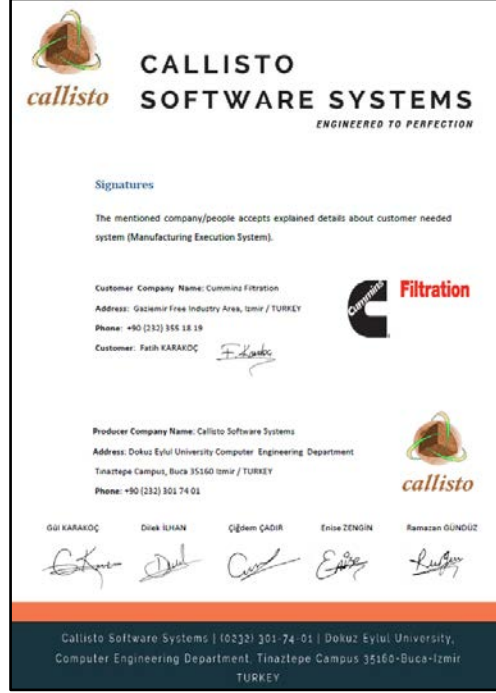
1. Functional Requirements List

- ✦ The needed system must be a part of the production lines.
- ✦ The needed system must create a record at database for all processes which applied onto any product.
- ✦ After semi products set into machine on any production line, all of semi products information must record in the database of the needed system. The recording must be according to product traceability number.
- ✦ The system must keeps data about semi products that: when it was produced and supplier of it, lot number etc.
- ✦ The needed system must keep all data about machines; operator of the machine (ID Number of the operators) which product produced (Traceability Number of the product), which semi-products used (Barcode Number of the semi-products), starting and finishing times of the product process on that machine.
- ✦ The needed system must communicate with PLCs before processes and after processes. They must work in combination. Before processes PLC sends information about who is the operator and when started to work. During processes, PLC sends information about what process occurs on any time and on any machine, which product is producing. After processes, how many product was produced, faults of the products if exists, which semi-products used etc. And the needed system must record

Callisto Software Systems | (0232) 204-74-01 | Dokuz Eylül University,
Computer Engineering Department, Tinaztepe Campus 35160-Buca-İzmir
TURKEY

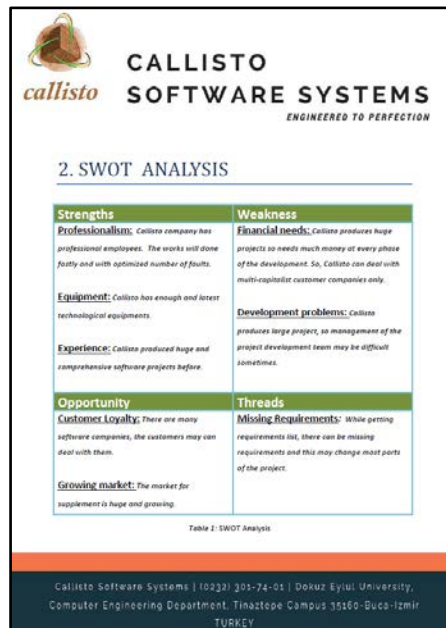
Şekil 2. Gereksinim raporu.

Gereksinim raporlarının hazırlanıp teslim edilmesinin ardından, proje grupları muhatap oldukları müşteriler ile tekrar bir araya gelerek ellerinde hali hazırda bulunan gereksinim raporlarını, açık ve net bir şekilde anlaşılır ifadelerle yazılım isterlerinin yer aldığı yazılım gereksinim spesifikasyon raporlarına dönüştürüp müşteriler ile kontrat imzalamaktadırlar – bu şekilde yazılımdan beklenen ihtiyaçlar kesinlik kazanmaktadır. Bu kontratın yapıldığına dair delili de proje kapsamında öğrenciler sunmaktadırlar. Kontrata ait örnek aşağıdaki Şekil 3’de gösterilmiştir.



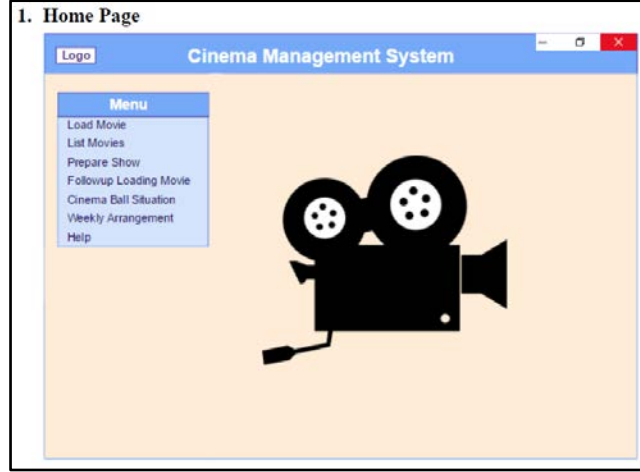
Şekil 3. Örnek bir kontrat.

Kontratın imzalanmasıyla tasarlanacak ve geliştirilecek yazılımla ilgili gereksinimler tamamen ortaya çıktıktan sonra ters mühendislik kavramı uygulanıp geliştirilecek yazılımla ilgili fizibilite çalışması – Şekil 4 – yapılabilmektedir ve akabinde teslim edilmektedir.



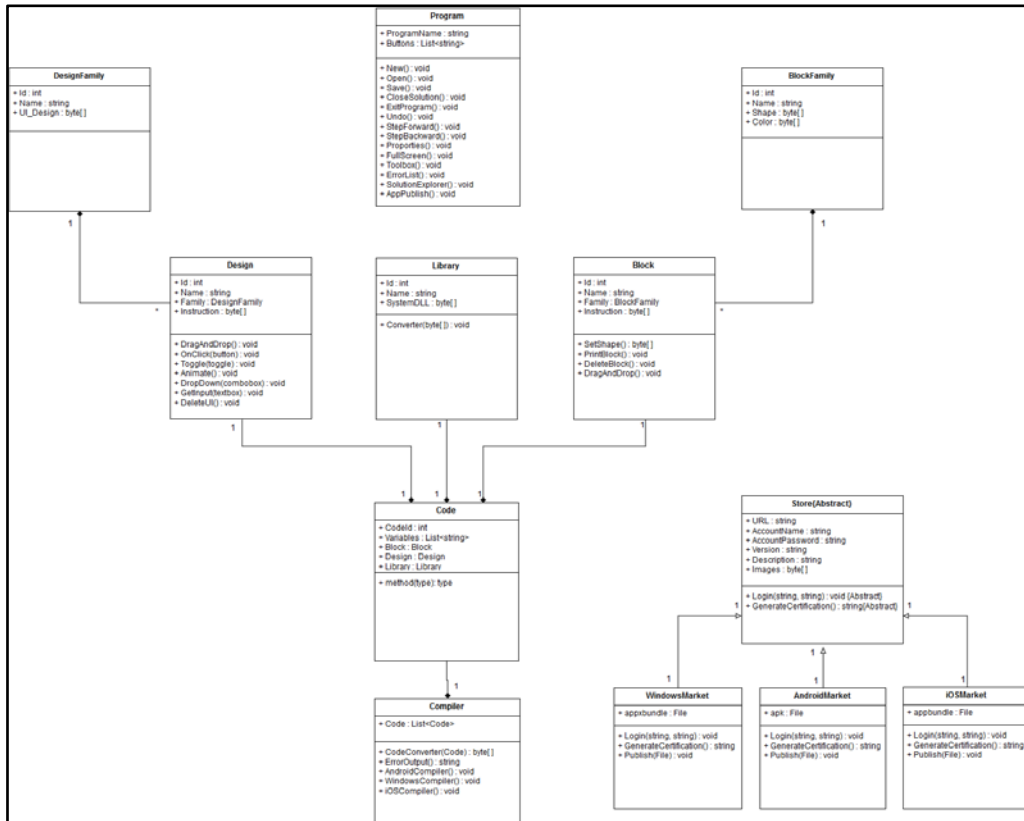
Şekil 4. Örnek bir fizibilite raporunda SWOT analiz çalışması.

Ters mühendislik çalışmasının sonucu olarak ortaya çıkan fizibilite çalışma raporunun ardından tasarlanacak ve geliştirilecek yazılım ürünü ile ilgili dış tasarım (mockup) çalışmaları yapılmakta ve ilgili ara yüzler raporlanıp teslim edilmektedir. Bu kapsamda ortaya çıkan örnek ara yüz çalışması aşağıdaki Şekil 5’de verilmiştir.



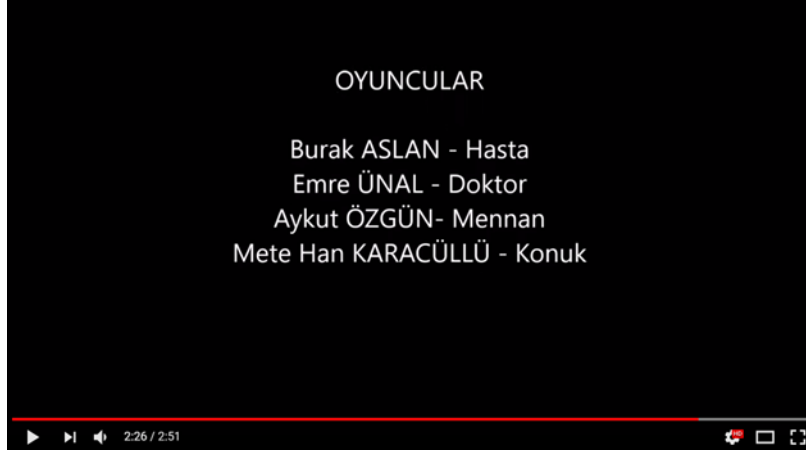
Şekil 5. Sinema yönetim ve otomasyon sistemine ait örnek bir mockup çalışması.

Tasarlanan bu prototiplerin geliştirilmesinin ardından yazılımın iç tasarımı ile ilgili olarak çeşitli mimari çalışmalar yapılmakta ve geliştirilecek yazılımın kodlama evresinin temelini oluşturacak ve buna ışık tutacak mimari ile ilgili olarak UML (Unified Modeling Language) diyagramları (use case diyagram, class diyagram, sequence diyagram ve activity diyagram) çizilip rapor şeklinde sunulmaktadır. Bu kapsamda örnek bir class diyagram çalışması aşağıdaki Şekil 6’da sunulmuştur.



Şekil 6. Örnek bir class diyagramı.

Tasarlanan yazılım projesinin iç mimari tasarımının geliştirilmesinin ardından, yazılımın programlanabilmesi için “Yazılım Mühendisliği” dersinin verildiği eğitim-öğretim dönemi içinde yeterli zaman kalmadığından dolayı yazılım kodlanmış gibi varsayılarak geliştirme sonucunda ortaya çıkacak yazılım ürünü ile ilgili olarak bir ya da iki dakikalık bir pazarlama reklam filmi (dijital pazarlama) – Şekil 7 – çekilip sunulmakta ve bu dönem projesi, öğrencilerin heyecan ve merakına uygun olarak geliştirilmeye (lisans bitirme projesi olabilecek seviyede) açık bir şekilde sonlandırılmaktadır.



Şekil 7. Nylon Software Solutions grubunun yazılım reklamı
(https://www.youtube.com/watch?v=8_yOtCIITFY).

4. SONUÇ

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde, “Yazılım Mühendisliği” dersi kapsamında 2014-2015, 2015-2016 ve 2016-2017 eğitim-öğretim dönemlerinde, uygulama saatleri gerçek müşterilerle beraber tasarlanmaya ve geliştirilmeye çalışılan yazılım projeleri ışığında işlenmiştir. Bu projeler kapsamında ve 3 eğitim-öğretim döneminde, toplamda 54 farklı yeni proje ortaya konmuş ve 291 öğrenci bu projelerin tasarlanmasında ve geliştirilmesinde aktif rol oynamıştır. Gerçek müşteriler ile yapılan bu projeler sonucunda “Yazılım Mühendisliği” dersinin teori kısmı ile uygulamasındaki farklılar daha net bir şekilde ortaya konmuş ve teoride, pratik ile teorinin aynı olduğu belirtilmesine rağmen; pratikte birçok gri noktanın (teoride yazılım geliştirme akışına uymayan durumlar, problemler, aksaklıklar, riskler, vb.) yazılım geliştirmede ortaya çıktığı saptanmıştır.

Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği öğrencileri, “Yazılım Mühendisliği” dersini 5. veya 6. dönemde aldıklarından dolayı, son dönemlerine (7. ve 8. dönem) geçtikleri zaman geliştirecekleri bitirme projeleri (lisans tezleri) ile ilgili uygulamalı bir şekilde ön bilgiye sahip olmaktadır. Ayrıca, “Yazılım Mühendisliği” dersi kapsamında ortaya koydukları yazılım projeleri sayesinde, bitirme tezleriyle ilgili raporlama ve süreç hakkında deneyime sahip olup son senelerinde bitirme projeleriyle ilgili zorlanma kat sayılarının azaldığı gözlenmiştir. Sonuç olarak, “Yazılım Mühendisliği” dalının sadece kavramlardan ve terimlerden ibaret olmadığı, gerçek hayatta ne gibi artılar ve eksiler yarattığı tasarlanan ve geliştirilen bu gerçek projeler sayesinde somut bir şekilde gözler önüne serilmiştir.

KAYNAKLAR

- [1] Bourque, P., & Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Version 3.0, Third Edition. Los Alamitos, CA, USA: IEEE Computer Society Press.
- [2] Sommerville, I. (2011). *Component Based Software Engineering*. Boston: Pearson.
- [3] Johanson, A., & Hasselbring, W. (2018). Software engineering for computational science: past, present, future. *Computing in Science & Engineering*, 20(2), 90-109. DOI: 10.1109/MCSE.2018.021651343
- [4] Ramakrishnan, L., & Gunter, D. (2017). Ten Principles for Creating Usable Software for Science. *2017 IEEE 13th International Conference on e-Science (e-Science), Auckland, New Zealand*. DOI: 10.1109/eScience.2017.34
- [5] Storer, T. (2017). Bridging the chasm: a survey of software engineering practice in scientific programming. *ACM Computing Surveys*, 50(4), 1-32. DOI: 10.1145/3084225.
- [6] Heaton, D. (2015). *Software Engineering for Enabling Scientific Software Development*. PhD Thesis, The University of Alabama, Tuscaloosa, Alabama.
- [7] Yazılım Mühendisliği.
Erişim Linki: https://debis.deu.edu.tr/ders-katalog/2020-2021/tr/tr_1210_1210_4065.html Erişim Tarihi: 10.08.2021