

GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	Software Engineering / Software Engineering	
Ders Kodu / Course Code	EBLG348	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	Bachelor / Bachelor	
Ders Akts Kredi / ECTS	5.00	
Haftalık Ders Saati (Kuramsal) / Course Hours For Week (Theoretical)	3.00	
Haftalık Uygulama Saati / Course Hours For Week (Objected)	0.00	
Haftalık Laboratuar Saati / Course Hours For Week (Laboratory)	0.00	
Dersin Verildiği Yıl / Year	3	
Öğretim Sistemi / Teaching System	Daytime Class / Daytime Class	
Eğitim Dili / Education Language		
Ön Koşulu Olan Ders(ler) / Precondition Courses	yok	no
Amacı / Purpose	Bu ders kapsamında yazılım mühendisliğinin esasları ele alınacaktır. Öğrenciler, yazılım geliştirme süreç modelleri, sistem gereksinim mühendisliği ve ilgili tanımlama ile gerçekleştirme yöntemleri konusunda bilgilendirilecektir. Öğrencilerin gerçekçi problemler karşısında geçerli mühendislik kararları alarak, tasarım, gerçekleştirme ve test konusunda teorik bilginin yanı sıra pratik deneyim kazanmasını hedeflenmektedir.	In this course, the fundamentals of software engineering will be covered. Students will be informed about software development process models, system requirements engineering and related definition and implementation methods. It is aimed that students gain practical experience as well as theoretical knowledge about design, implementation and testing by making valid engineering decisions in the face of realistic problems.
İçeriği / Content	Yazılım geliştirme süreç modelleri ve yazılım yaşam döngüsü, gereksinim analiz ve belirtimi, Unified Modeling Language ile nesne-tabanlı yazılım modelleme ve CASE araçları ile nesne-tabanlı yazılım geliştirme, yazılım kalite kontrolü, teslim ve bakım süreci, dönem projesi ile yazılım mühendislik süreçlerinin uygulanması.	Software development process models and software lifecycle, requirements analysis and specification, object-oriented software modeling with Unified Modeling Language and object-oriented software development with CASE tools, software quality control, delivery and maintenance process, term project and implementation of software engineering processes.
Önerilen Diğer Hususlar / Recommended Other Considerations		
Staj Durumu / Internship Status		
Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	Yazılım Mühendisliği (Yazar: Ian Sommerville Yayınevi: Nobel Akademik Yayıncılık)	Yazılım Mühendisliği (Yazar: Ian Sommerville Yayınevi: Nobel Akademik Yayıncılık)
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)	Öğr. Üyesi Reşat Buğra Erkartal	

ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	Bir yazılım ürünü geliştirmeye yönelik temel adımları uygular.	Applies the basic steps for developing a software product.
2	Yazılım geliştirme yöntemlerini belirler.	Determines software development methods.
3	Yazılım ve ortam özelliklerine göre en olası-başarılı metodolojiyi planlar.	Plans the most likely-successful methodology according to software and environment characteristics.
4	Bir yazılım projesi tanımlayabilir.	Can define a software project.
5	Büyük programcı grupları tarafından yazılımların arşivlenmesi ve geliştirilmesi için dokümantasyon standartlarını ve örneklerini hazırlar.	Prepares documentation standards and samples for archiving and development of software by large groups of programmers.

HAFTALIK DERS İÇERİĞİ / DETAILED COURSE OUTLINE

Hafta / Week					
1	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Giriş: Yazılım mühendisliği nedir?				
	Introduction: What is software engineering?				
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Gereksinim ve tasarım modelleme aracı olarak UML				
	UML as a requirements and design modeling tool				
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım yaşam döngüsü				
	Software lifecycle				
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım geliştirme süreç modelleri: Şelale, hızlı prototipleme, spiral çevik modeller				
	Software development process models: Waterfall, rapid prototyping, spiral agile models				
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım geliştirme süreç modelleri: Şelale, hızlı prototipleme, spiral çevik modeller				
	Software development process models: Waterfall, rapid prototyping, spiral agile models				

6	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım geliştirme süreç modelleri: Şelale, hızlı prototipleme, spiral çevik modeller				
	Software development process models: Waterfall, rapid prototyping, spiral agile models				
7	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım gereksinim analizi ve belirtimi				
	Software requirements analysis and specification				
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Ara Sınav				
	Midterm				
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım tasarımı - tasarım doğrulama				
	Software design - design validation				
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım gerçekleştirimi				
	Software implementation				
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım test ve kalite kontrolü				
	Software testing and quality control				

12	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılımın evrimi				
	Evolution of software				
13	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Proje Yönetimi				
	Project Management				
14	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Diyagramlar				
	Diagrams				

DEĞERLENDİRME / EVALUATION

Yarıyıl (Yıl) İçi Etkinlikleri / Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Ara Sınav / Midterm Examination	1	100
Toplam / Total:	1	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		40

Yarıyıl (Yıl) Sonu Etkinlikleri / End Of Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Final Sınavı / Final Examination	1	100
Toplam / Total:	1	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		60

Etkinliklerinin Başarı Notuna Katkı Yüzdesi(%) Toplamı / Total Percentage of Contribution (%) to Success Grade:	100
Değerlendirme Tipi / Evaluation Type:	

İŞ YÜKÜ / WORKLOADS

Etkinlikler / Workloads	Sayı / Number	Süresi (Saat) / Duration (Hours)	Toplam İş Yüğü (Saat) / Total Work Load (Hour)
Ara Sınav / Midterm Examination	1	1.00	1.00
Ara Sınav İçin Bireysel Çalışma / Individual Study for Mid term Examination	1	25.00	25.00
Bireysel Çalışma / Self Study	14	2.00	28.00
Derse Katılım / Attending Lectures	14	3.00	42.00
Final Sınavı / Final Examination	1	1.00	1.00
Final Sınavı için Bireysel Çalışma / Individual Study for Final Examination	1	30.00	30.00
Toplam / Total:	32	62.00	127.00
Dersin AKTS Kredisi = Toplam İş Yüğü (Saat) / 25.00 (Saat/AKTS) = 127.00/25.00 = 5.08 ~ / Course ECTS Credit = Total Workload (Hour) / 25.00 (Hour / ECTS) = 127.00 / 25.00 = 5.08 ~			

PROGRAM VE ÖĞRENME ÇIKTISI / PROGRAM LEARNING OUTCOMES

Öğrenme Çıktıları / Learning Outcomes	Program Çıktıları / Program Outcomes										
	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7	1.1.8	1.1.9	1.1.10	1.1.11
1.Bir yazılım ürünü geliştirmeye yönelik temel adımları uygular. / Applies the basic steps for developing a software product.	5	3	5	5				4	5		
2.Yazılım geliştirme yöntemlerini belirler. / Determines software development methods.	5	5	5	5				5			
3.Yazılım ve ortam özelliklerine göre en olası-başarılı metodolojiyi planlar. / Plans the most likely-successful methodology according to software and environment characteristics.	5	5	5	5	4			4			
4.Bir yazılım projesi tanımlayabilir. / Can define a software project.	5	5	5	5	4			5			
5.Büyük programcı grupları tarafından yazılımların arşivlenmesi ve geliştirilmesi için dokümantasyon standartlarını ve örneklerini hazırlar. / Prepares documentation standards and samples for archiving and development of software by large groups of programmers.	5	5	5	5	4		3	5			

Katkı Düzeyi / Contribution Level : 1-Çok Düşük / Very low, 2-Düşük / Low, 3-Orta / Moderate, 4-Yüksek / High, 5-Çok Yüksek / Very high