

## GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	FUNDAMENTALS OF SOFTWARE ENGINEERING / FUNDAMENTALS OF SOFTWARE ENGINEERING	
Ders Kodu / Course Code	501003362023	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	First Cycle / First Cycle	
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)	1.00	
Haftalık Laboratuvar Saati / Course Hours For Week (Laboratory)	0.00	
Dersin Verildiği Yıl / Year	3	
Öğretim Sistemi / Teaching System	Face to Face / Face to Face	
Eğitim Dili / Education Language	English / English	
Ön Koşulu Olan Ders(ler) / Precondition Courses	Yok	None
Amacı / Purpose	Bu dersin amacı yazılım mühendisliğinin temel ilkelerini incelenmesi ve bir proje çerçevesinde uygulanmasıdır. Bu kapsamda, yazılım geliştirme yaşam döngüsünün aşamaları ve farklı aşamalarda kullanılan yöntemler, teknikler ve araçlar tanıtılmaktadır.	Acquisition of the basic principles of software engineering and the ability to apply them in a software development project. In this context the stages of the software development lifecycle, and the tools and techniques used in each stage are introduced.
İçeriği / Content	Yazılım Mühendisliğinin Temel Kavramları, Süreç Modelleri, Proje Yönetimi, Gereksinimler Analizi, Sistem Modelleri, Gereksinim Tanımlama, Prototipleme, Yazılım Tasarımı, Yazılım Güvenilirliği, Sınama Ve Doğrulama, Yazılım Bakımı	Basic concepts of Software Engineering, Process Models, Project Management, Requirements Analysis, System Models, Requirements Specification, Prototyping, Software Design, Software Reliability, Validation and Verification, Software Maintenance
Önerilen Diğer Hususlar / Recommended Other Considerations	Yok	None
Staj Durumu / Internship Status	Yok	None
Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	Sommerville, I., 'Software Engineering', Addison Wesley.	Sommerville, I., "Software Engineering", Addison Wesley.
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)	Prof.Dr.N. Yasemin TOPALOĞLU	

## ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	Yazılım tasarımında UML modellerini geliştirebilme.	Ability to develop UML models
2	Yazılım mimarisi modellerini kavrayabilme.	Perception of software architecture models
3	Bir yazılım projesine ait tasarımı gerçekleştirebilme.	Ability to design a software project.
4	Yazılım geliştirmede takım çalışması yapabilme.	Ability to participate in a teamwork in a software project.
5	Bir yazılım projesine ait kullanıcı arayüzlerini belirleyebilme.	Ability to specify user interfaces
6	Bir yazılım projesinde sına ma etkinliklerini kavrayabilme ve uygulayabilme.	Ability to apply test-driven development method in a software project
7	Bir yazılım projesine ait bakım etkinliklerini değerlendirebilme.	Ability to manage software maintenance activities
8	Yazılım süreçlerini kavrayabilme.	Understanding software development processes
9	Yazılım süreç modellerini(Çağlayan, gelişimsel, spiral) değerlendirebilme.	Ability to comprehend major software development processes (waterfall, evolutionary and spiral).
10	Yazılım proje yönetimini kavrayabilme ve yazılım projelerinde risk analizi yapabilme.	Ability to manage software projects and perform risk analysis.
11	Bir yazılım projesine ait gereksinimleri belirleyebilme.	Ability to specify the requirements of a software project.
12	Yazılım projelerinde prototiplemeyi değerlendirebilme.	Ability to implement prototyping in a software project.

## 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 mühendisliğinin tanımı, önemi, yazılım süreç ve ürün tipleri.	İlgili CASE araçlarının tanıtımı			
	Introduction: The definition and importance of software engineering, Software process and product types	Introduction of related CASE tools			
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Süreç modelleri: Çağlayan modeli, gelişimsel model ve spiral model.	Proje konusunun açıklanması, Yazılım süreç modelleri alıştırmaları			
	Software Process Models: Waterfall model, evolutionary model and spiral model	Description of the term project, Software Process Models Exercises			
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım proje yönetimi ve risk analizi.	Proje konusunun açıklanması			
	Software project management and risk analysis	Explanations on term project			
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Gereksinim Mühendisliği, aşamalar	Gereksinim türleri, belgelemesi alıştırmaları			
	Requirements engineering and its phases briefly	Exercises about types of requirements and requirements documentation			
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Gereksinim Mühendisliği: Gereksinim mühendisliği sürecinin aşamaları. Gereksinimler analizi ve sistem modelleri, UML.	Proje kullanım durumlarının anlatılması			
	Requirements Engineering: Phases of Requirements Engineering, Requirements Analysis and system models and UML	Explanation of the project use cases			

	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
6	Gereksinimler: Gereksinim tipleri, gereksinim tanımlama.	Alan Modelinin anlatılması			
	Requirements: Requirements types, Requirements definition.	Explanation of the domain model			
7	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Prototipleme: Prototipleme tipleri, araçları ve yazılım süreçlerindeki yeri.	Alıştırma			
	Prototyping: Types of prototyping, tools and their usage in the software process	Exercises			
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Tasarımı: Tasarım süreci, tasarım yaklaşımları ve yöntemleri.	Proje sunumları			
	Software Design: Design process, design approaches and methods	Project demonstrations			
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Mimari Tasarım: Sistem yapılandırma, modüler ayrıştırma.	Mimari Desenler: MVC, Arabulucu, Katmanlı mimarilerin anlatımı			
	Architectural Design: system structuring and modular decomposition	Architectural Patterns: MVC, Broker and Layered architecture			
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Ara sınav				
	Midterm				
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Desenler: Mimari desenler ve tasarım desenleri	Uygulama			
	Patterns: Architectural patterns and design patterns	Training			

	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
12	Kullanıcı arayüz tasarımı, yardım sistemi, kullanıcı dökümantasyonu.	Alıştırma			
	User interface design,help system, user documents.	Exercises			
13	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Sinama ve Doğrulama: Durağan sinama.	Test Güdümlü Geliştirme Anlatımı			
	Validation and Verification: Static verification.	Instruction on test driven development			
14	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Sinama ve Doğrulama: Sinama türleri ve aşamaları.	Proje sunumları ve değerlendirme			
	Validation and Verification: Types and phases of verification.	Project demonstrations and evaluation			
15	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Bakımı: Bakım türleri, bakım maliyeti, konfigürasyon yönetimi.	CVS(Concurrent Versioning System) anlatımı			
	Software Maintainence: Types of maintainence, maintainence cost, configuration management.	Introduction to CVS(Concurrent Versioning System)			
16	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Final Sınavı				
	Final Exam				

## DEĞERLENDİRME / EVALUATION

Yarıyıl (Yıl) İçi Etkinlikleri / Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Ara Sınav / Midterm Examination	1	50
Proje Hazırlama / Project Preparation	1	20
Proje Sunma / Project Presentation	1	10
Proje Tasarımı /Yönetimi / Project Design/Management	1	20
Toplam / Total:	4	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	2.00	2.00
Final Sınavı / Final Examination	1	2.00	2.00
Derse Katılım / Attending Lectures	14	3.00	42.00
Problem Çözümü / Problem Solving	10	2.00	20.00
Takım/Grup Çalışması / Team/Group Work	3	14.00	42.00
Rapor Hazırlama / Report Preparation	4	8.00	32.00
Ara Sınav İçin Bireysel Çalışma / Individual Study for Mid term Examination	1	16.00	16.00
Final Sınavı için Bireysel Çalışma / Individual Study for Final Examination	1	16.00	16.00
<b>Toplam / Total:</b>	<b>35</b>	<b>63.00</b>	<b>172.00</b>
Dersin AKTS Kredisi = Toplam İş Yüğü (Saat) / 30.00 (Saat/AKTS) = 172.00/30.00 = 5.73 ~ / Course ECTS Credit = Total Workload (Hour) / 30.00 (Hour / ECTS) = 172.00 / 30.00 = 5.73 ~			

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.Yazılım tasarımında UML modellerini geliştirebilme. / Ability to develop UML models	2	3	5		4						
2.Yazılım mimarisi modellerini kavrayabilme. / Perception of software architecture models	2		5		4						
3.Bir yazılım projesine ait tasarımı gerçekleştirebilme. / Ability to design a software project.	2	3	5		4						
4.Yazılım geliştirmede takım çalışması yapabileme. / Ability to participate in a teamwork in a software project.	2					5					5

5.Bir yazılım projesine ait kullanıcı arayüzlerini belirleyebilme. / Ability to specify user interfaces	2	3	5		4						
6.Bir yazılım projesinde sınam etkinliklerini kavrayabilme ve uygulayabilme. / Ability to apply test-driven development method in a software project	2				4						
7.Bir yazılım projesine ait bakım etkinliklerini değerlendirebilme. / Ability to manage software maintainence activities	2	3					4	4		4	
8.Yazılım süreçlerini kavrayabilme. / Understanding software development processes	2								4		
9.Yazılım süreç modellerini (Çağlayan, gelişimsel, spiral) değerlendirebilme. / Ability to comprehend major software development processes (waterfall, evolutionary and spiral).	2	3									
10.Yazılım proje yönetimini kavrayabilme ve yazılım projelerinde risk analizi yapabilme. / Ability to manage software projects and perform risk analysis.	2						5				
11.Bir yazılım projesine ait gereksinimleri belirleyebilme. / Ability to specify the requirements of a software project.	2	5	5		4				4	4	
12.Yazılım projelerinde prototiplemeyi değerlendirebilme. / Ability to implement prototyping in a software project.	2	4	4								

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