

GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	Software Engineering / Software Engineering	
Ders Kodu / Course Code	BTT517	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	Master with Thesis / Master with Thesis	
Ders Akts Kredi / ECTS	6.00	
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	1	
Öğretim Sistemi / Teaching System	Daytime Class / Daytime Class	
Eğitim Dili / Education Language	Turkish / Turkish	
Ön Koşulu Olan Ders(ler) / Precondition Courses	Yok	None
Amacı / Purpose	Bu dersin amacı, öğrencilere yazılım mühendisliği disiplininin temel kavramlarını, yöntem ve tekniklerini öğretmek, yazılım geliştirme sürecindeki adımları anlamalarını sağlamak ve kaliteli ve etkili yazılım üretebilmeleri için gerekli becerileri kazandırmaktır.	The aim of this course is to provide students with the fundamental concepts, methods, and techniques of software engineering, enable them to understand the steps in software development process, and equip them with the necessary skills to produce quality and effective software.
İçeriği / Content	Bu dersin amacı, öğrencilere yazılım mühendisliği disiplininin temel kavramlarını, yöntem ve tekniklerini öğretmek, yazılım geliştirme sürecindeki adımları anlamalarını sağlamak ve kaliteli ve etkili yazılım üretebilmeleri için gerekli becerileri kazandırmaktır.	The aim of this course is to provide students with the fundamental concepts, methods, and techniques of software engineering, enable them to understand the steps in software development process, and equip them with the necessary skills to produce quality and effective software.
Önerilen Diğer Hususlar / Recommended Other Considerations	Yok	None
Staj Durumu / Internship Status	Yok	None
Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	Pressman R., Software Engineering, 6th ed. (2005), Mc Graw Hill. Larman C., Applying UML and Patterns, 2nd ed. (2002), Prentice Hall. Kalıpsız Oya, Yazılım Mühendisliği, G.Ü. Yayını Sommerville I., Software Engineering, 8th ed. (2006), Prentice Hall.	Pressman R., Software Engineering, 6th ed. (2005), Mc Graw Hill. Larman C., Applying UML and Patterns, 2nd ed. (2002), Prentice Hall. Kalıpsız Oya, Yazılım Mühendisliği, G.Ü. Yayını Sommerville I., Software Engineering, 8th ed. (2006), Prentice Hall.
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)		

ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	Yazılım mühendisliği disiplini ve temel prensiplerini anlayabilme.	Understand the discipline of software engineering and its fundamental principles.
2	Yazılım yaşam döngüsü süreçlerini ve yöntemlerini uygulayabilme.	Apply software life cycle processes and methodologies.
3	Yazılım gereksinimlerini analiz edebilme ve tasarım yapabilme becerisini geliştirebilme.	Develop the ability to analyze software requirements and perform design.
4	Yazılım kodlama ve test etme süreçlerini yönetebilme.	Manage software coding and testing processes.
5	Yazılım projesi yönetimi ve kalite güvencesi konularını değerlendirebilme.	Evaluate software project management and quality assurance topics.

HAFTALIK DERS İÇERİĞİ / DETAILED COURSE OUTLINE

Hafta / Week					
1	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Mühendisliğine Giriş				
	Introduction to Software Engineering				
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Geliştirme Süreci ve Modelleri				
	Software Development Process and Models				
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Gereksinim Analizi ve Gözden Geçirme				
	Requirement Analysis and Review				
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Tasarım ve Mimari				
	Design and Architecture				
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Programlama ve Kodlama				
	Programming and Coding				

6	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Test Etme ve Doğrulama				
	Testing and Verification				
7	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Kalitesi ve Ölçümü				
	Software Quality and Measurement				
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Proje Yönetimi				
	Software Project Management				
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Bakımı ve Evrimi				
	Software Maintenance and Evolution				
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Güvenliği				
	Software Security				
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Mühendisliği Araçları ve Ortamları			Software Engineering Tools and Environments	
	Software Engineering Tools and Environments			Software Engineering Tools and Environments	

12	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Dokümantasyonu				
	Software Documentation				
13	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Etiği ve Mesleki Sorumluluklar				
	Software Ethics and Professional Responsibilities				
14	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Mühendisliği Uygulamaları ve Geleceği				
	Software Engineering Applications and Future Trends				

DEĞERLENDİRME / EVALUATION

Yarıyıl (Yıl) İçerik Etkinlikleri / Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Ara Sınav / Midterm Examination	1	80
Proje Hazırlama / Project Preparation	1	20
Toplam / Total:	2	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		60

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(%):		40

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	25.00	25.00
Derse Katılım / Attending Lectures	1	60.00	60.00
Final Sınavı / Final Examination	1	25.00	25.00
Proje Sunma / Project Presentation	1	30.00	30.00
Toplam / Total:	4	140.00	140.00
Dersin AKTS Kredisi = Toplam İş Yüğü (Saat) / 25.00 (Saat/AKTS) = 140.00/25.00 = 5.60 ~ / Course ECTS Credit = Total Workload (Hour) / 25.00 (Hour / ECTS) = 140.00 / 25.00 = 5.60 ~			

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	1.1.12	1.1.13	1.1.14	1.1.15
1.Yazılım mühendisliği disiplinini ve temel prensiplerini anlayabilme. / Understand the discipline of software engineering and its fundamental principles.	4	4	4	4	4	5	5	4	4	5	4	4	5	5	5
2.Yazılım yaşam döngüsü süreçlerini ve yöntemlerini uygulayabilme. / Apply software life cycle processes and methodologies.	5	4	5	4	4	5	4	4	4	4	4	4	4	4	4
3.Yazılım gereksinimlerini analiz edebilme ve tasarım yapabilme becerisini geliştirebilme. / Develop the ability to analyze software requirements and perform design.	5	4	4	4	4	5	5	5	5	5	4	5	5	5	5
4.Yazılım kodlama ve test etme süreçlerini yönetebilme. / Manage software coding and testing processes.	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5
5.Yazılım projesi yönetimi ve kalite güvencesi konularını değerlendirebilme. / Evaluate software project management and quality assurance topics.	5	4	5	5	5	4	5	4	5	5	5	4	5	5	5

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