



2025-2026 AKADEMİK YILI GÜZ DÖNEMİ

BİL403 YAZILIM MÜHENDİSLİĞİ

2. DERS

Dr. Öğr. Üyesi Ertürk ERDAĞI
erturk.erdagi@medeniyet.edu.tr

- Yazılım mühendisliği tarihsel olarak birkaç evreden geçmiştir:
- **1940–1950’ler:** Bilgisayarlar yeni ortaya çıkmış, programlar makine diliyle yazılıyordu. Yazılım geliştirmek çok zahmetliydi.
- **1960’lar:** Yazılım projeleri büyüdükçe ciddi sorunlar ortaya çıktı. Süreler aşıyor, bütçeler yetmiyor, hatalar kontrol edilemiyordu. Bu döneme “yazılım krizi” denir.
- **1970’ler:** Yazılım mühendisliği kavramı NATO konferanslarında ortaya çıktı. Sistematik yöntemler, süreçler ve standartlar geliştirilmeye başlandı.
- **1980’ler:** Nesne yönelimli programlama ve yazılım araçları önem kazandı. Yazılım geliştirme daha modüler hale geldi.
- **1990’lar:** İnternetin yayılması ile yazılım ölçeği katlanarak arttı. Büyük sistemler ve dağıtık yapılar gündeme geldi.
- **2000’ler:** Çevik yöntemler doğdu. Esneklik, müşteri memnuniyeti ve hızlı teslim ön plana çıktı.
- **2010 sonrası:** Bulut bilişim, mobil uygulamalar, DevOps ve sürekli entegrasyon kavramları yazılım mühendisliğinin temel parçası oldu.
- **Bugün:** Yapay zekâ destekli geliştirme, otomatik kod üretimi ve mikroservis mimarileri yazılım mühendisliğinin güncel trendleridir.

Yazılım Geliştirme Yaşam Döngüsü (SDLC)

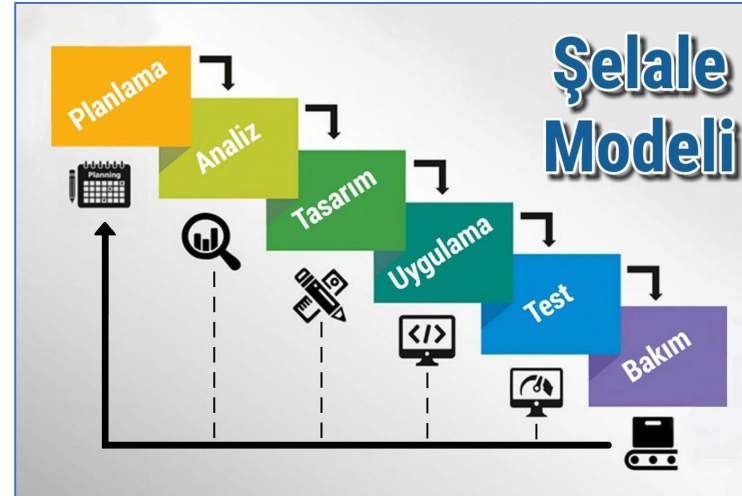
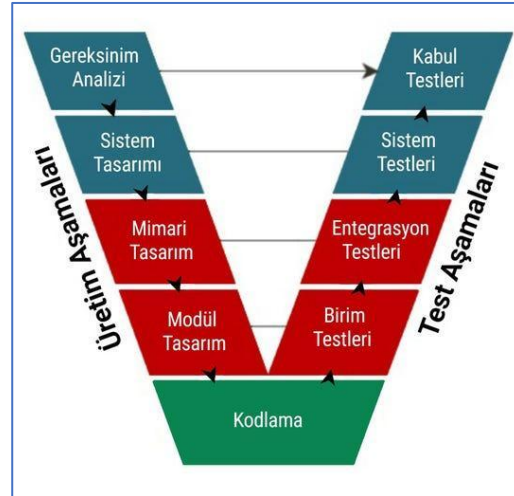
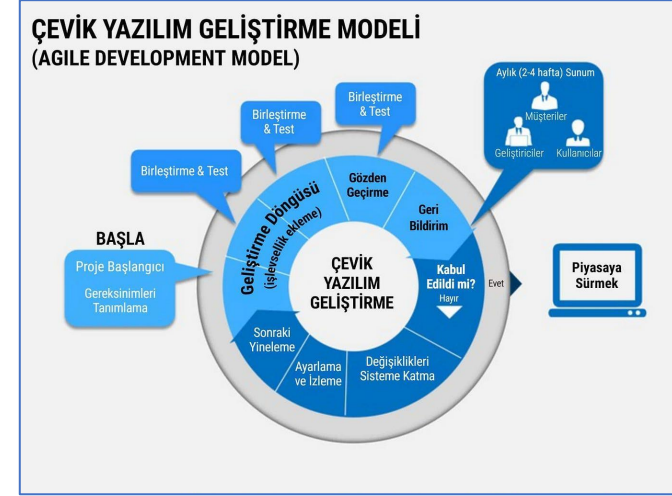
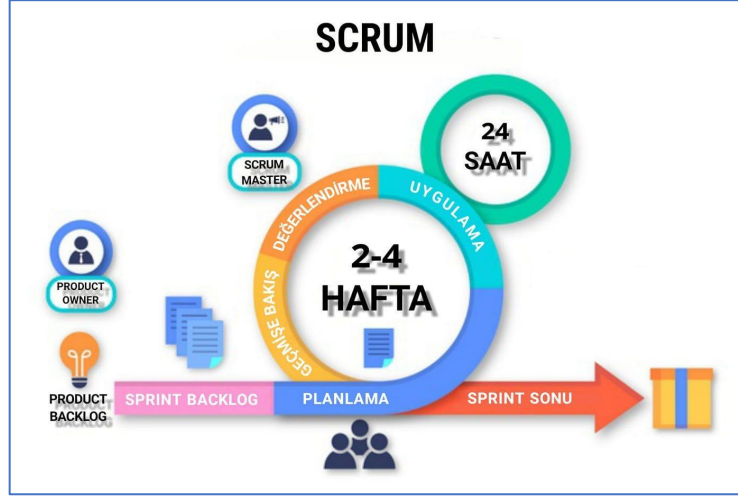
- Yazılım geliştirme süreci belirli adımlardan oluşur. Bu adımların tümüne Yazılım Geliştirme Yaşam Döngüsü (Software Development Life Cycle - SDLC) denir.
- Tipik yaşam döngüsü şu aşamalardan oluşur:
 - Gereksinimlerin belirlenmesi
 - Tasarımın yapılması
 - Kodlama ve entegrasyon
 - Test ve doğrulama
 - Dağıtım
 - Bakım ve geliştirme



Yazılım Geliştirme Yaşam Döngüsü (SDLC)

4

- SDLC'nin uygulanması için farklı modeller geliştirilmiştir:



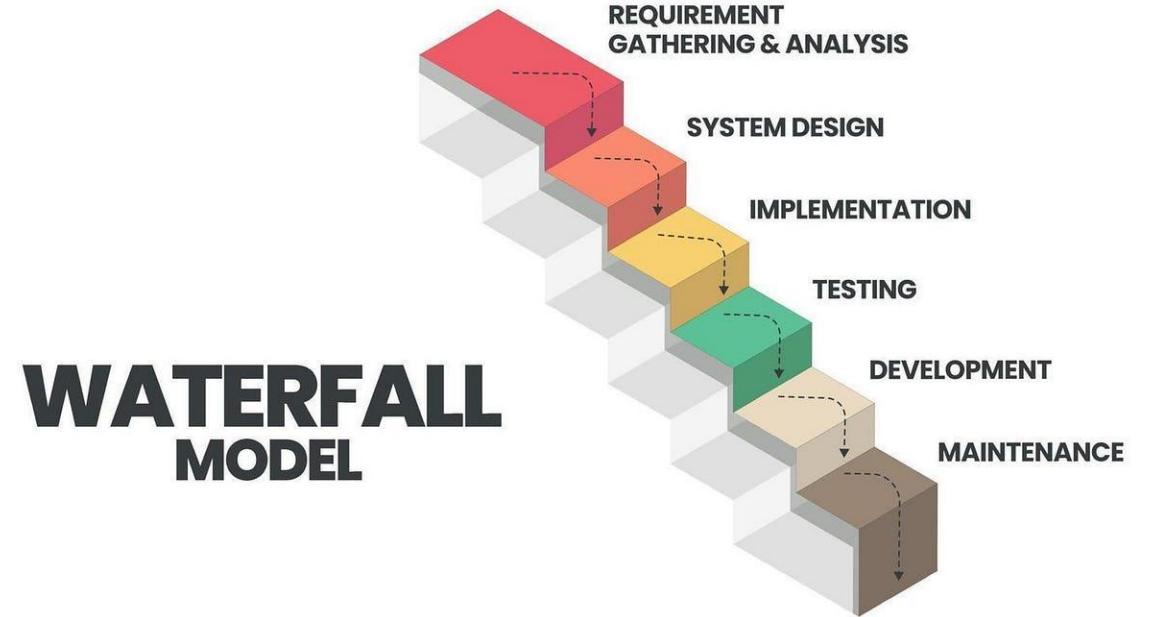
- Yazılım mühendisliği sürekli gelişen bir alandır. Günümüzde öne çıkan trendler şunlardır:
 - **Çevik yöntemler:** Scrum, Kanban gibi yaklaşımlar proje yönetiminde hız ve esneklik sağlar.
 - **DevOps ve CI/CD:** Yazılım geliştirme ve sistem yönetimi birleştirilir, sürekli entegrasyon ve teslim sağlanır.
 - **Mikroservis mimarisi:** Yazılım küçük, bağımsız parçalar halinde geliştirilir, ölçeklenebilirlik kolaylaşır.
 - **Bulut tabanlı geliştirme:** Yazılım bulut üzerinde çalıştırılır ve dağıtılır.
 - **Yapay zeka destekli geliştirme:** Kod tamamlama, hata tespiti ve test otomasyonu için yapay zekâ araçları kullanılır.
 - **Açık kaynak yazılımlar:** Topluluk desteği ile sürekli gelişen yazılımlar yaygın hale gelmiştir.

- Sizce yazılım mühendisliği ile yalnızca kod yazmak arasındaki fark nedir?
- Yazılım krizini nedir, günümüzde hâlâ yaşıyor muyuz?
- Yapay zekâ, yazılım geliştiricilerin yerine geçebilir mi, yoksa onların gücünü artıran bir araç mı olacak?

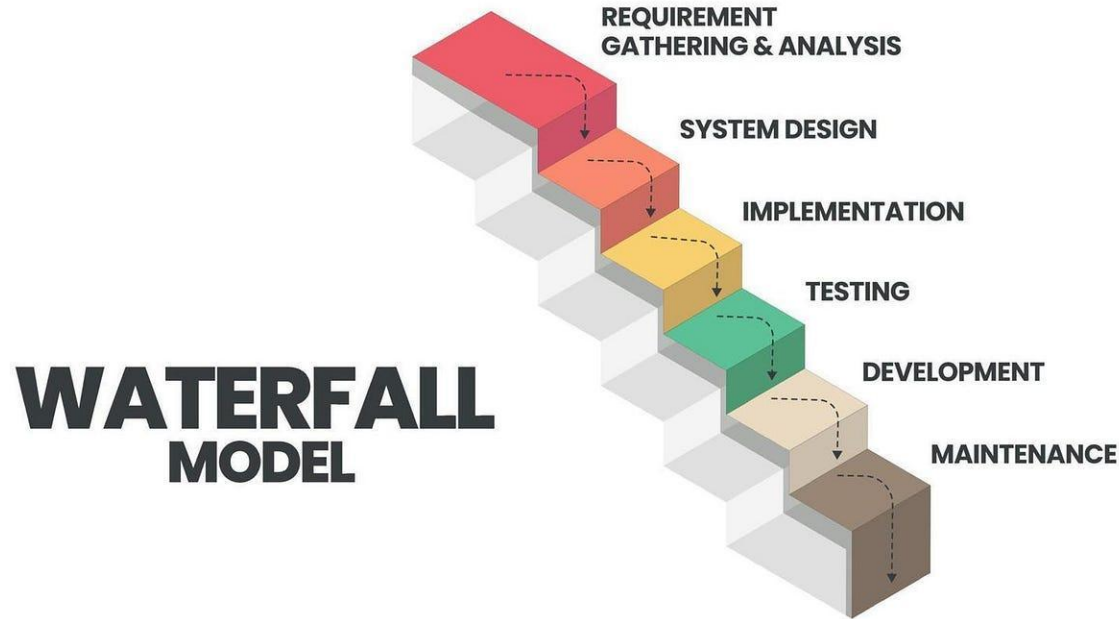
- Yazılım geliştirme süreç modelleri, yazılımın planlanması, tasarlanması, kodlanması, test edilmesi ve bakımının sistematik bir şekilde yürütülmesini sağlar. Doğru süreç modelinin seçimi, projenin başarısını doğrudan etkiler.
- 1. Şelale Modeli (Waterfall Model)
- 2. V-Model (Doğrulama ve Geçerleme Modeli)
- 3. Artımlı Model (Incremental Model)
- 4. Spiral Model
- 5. Çevik (Agile) Yaklaşımlar

Şelale Modeli (Waterfall Model)

- Yazılım geliştirme sürecinin adım adım ve lineer olarak ilerlediği klasik modeldir. Her aşama bir öncekine bağlıdır ve genellikle geri dönmek zordur
- **Aşamalar:**
 - Gereksinim Analizi
 - Sistem ve Yazılım Tasarımı
 - Uygulama (Kodlama)
 - Test ve Doğrulama
 - Kurulum ve Dağıtım
 - Bakım
- **Avantajlar :**
 - Basit ve anlaşılır
 - Yönetimi kolay
 - Dokümantasyon odaklı
- **Dezavantajlar :**
 - Değişime kapalı
 - Hatalar erken aşamada fark edilmezse maliyetli olur
 - Müşteri geri bildirimi süreç sonunda alınır



Şelale Modeli (Waterfall Model)



- Bu modelde bir aşama tamamlanmadan diğerine geçilmez. Sizce bu yaklaşımın proje başarısını artırdığı durumlar nelerdir?
- Şelale modelinde değişen gereksinimler nasıl yönetilir? Sizce bu model hangi tür projeler için uygundur?



KATILIMINIZ İÇİN TEŞEKKÜR EDERİM.

**YOKLAMA İÇİN İMZANIZI ATMAYI UNUTMAYINIZ.
CLASSROOM ÜZERİNDEN SINIFA DAHİL OLMAYI UNUTMAYINIZ**

Sınıf Kodu : pua2tnoe

Dr. Öğr. Üyesi Ertürk ERDAĞI
erturk.erdagi@medeniyet.edu.tr