Yazılım Mühendisliği
~~Yazılım Geliştirme Süreç Modelleri~~

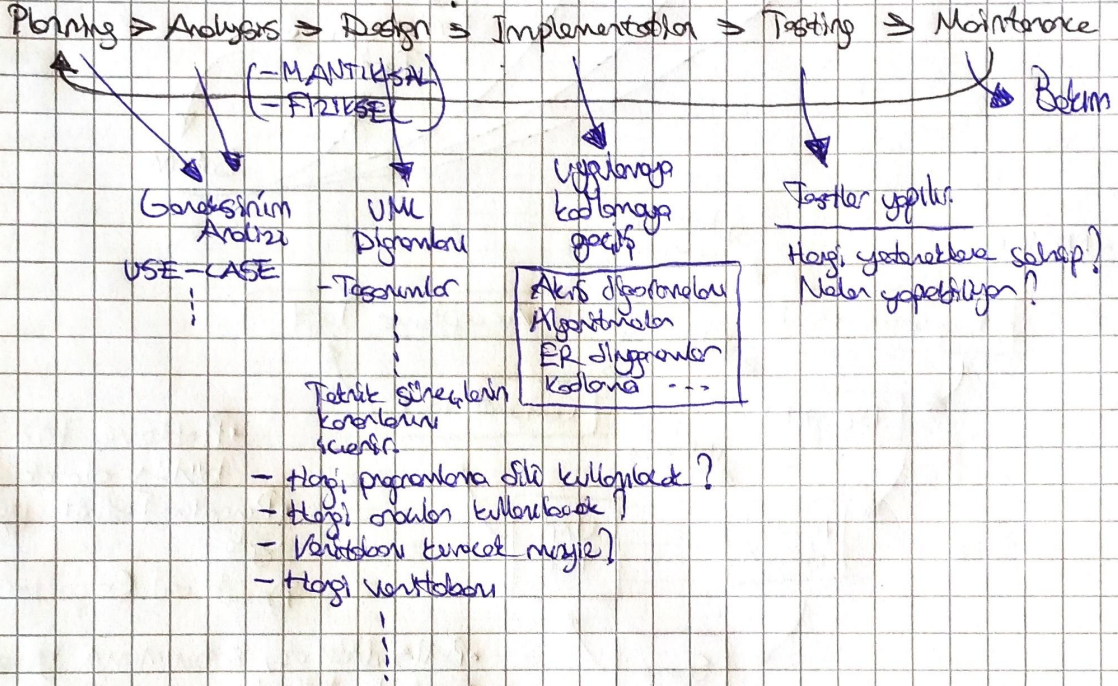Bu ders için yazılım endüstriyel üründür.
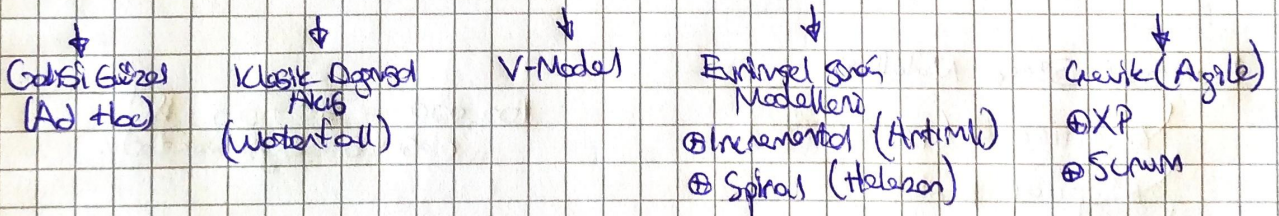• Standartları sağlayarak, kaliteli yazılım üretmektir

Kalite gereksinimlerine en çok uyan yazılımın kaliteli olduğunu varsayıyoruz.

Yazılım mühendisliği, eldeki kaynak, zaman ve imkanlar ile en iyi yazılımı üretmeye çalışır.

The Software Development Life Cycle (SDLC)

Planning → Analysis → Design → Implementation → Testing → Maintenance

(—MANTIKSAL)
(—FIZIKSEL)

→ Bakım

Gereksinim Analizi — UML Diyagramları — Uygulamaya kodlamaya geçiş — Testler yapılır.

USE-CASE — Tasarımlar

Hangi yeteneklere sahip?
Neler yapabiliyor?

| Akış diyagramları |
| Algoritmalar |
| ER diyagramları |
| Kodlama --- |

Teknik süreçlerin kararlarını içerir.

- Hangi programlama dili kullanılacak?
- Hangi araçlar kullanılacak?
- Veritabanı kurcağı mıyız?
- Hangi veritabanı
⋮

Yazılım geliştirme süreç modelleri

↓ Gelişi Güzel (Ad hoc)
↓ Klasik Aşamalı Akış (waterfall)
↓ V-Model
↓ Evrimsel Süreç Modelleri
  ⊕ Incremental (Artımlı)
  ⊕ Spiral (Helezon)
↓ Çevik (Agile)
  ⊕ XP
  ⊕ Scrum

Gelişi güzel geliştirme

- Kervan yolda düzülür mantığı
- Herhangi bir yöntem ve plan yok
- Geliştirici kişiye bağlı
- Bakım ve takibi zor
- Genellikle tek kişilik üretim ortamı
- Basit programlama
- Dökümantasyon yoktur

BAROK Model

İnceleme
Analiz
Tasarım
Kodlama
Modül Testleri
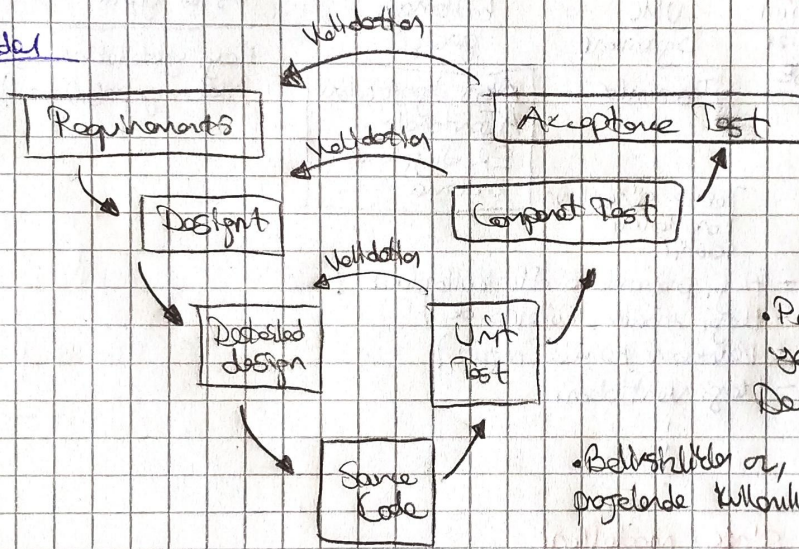Alt sistem testleri
Sistem testleri
* Belgeleme
Kurulum.
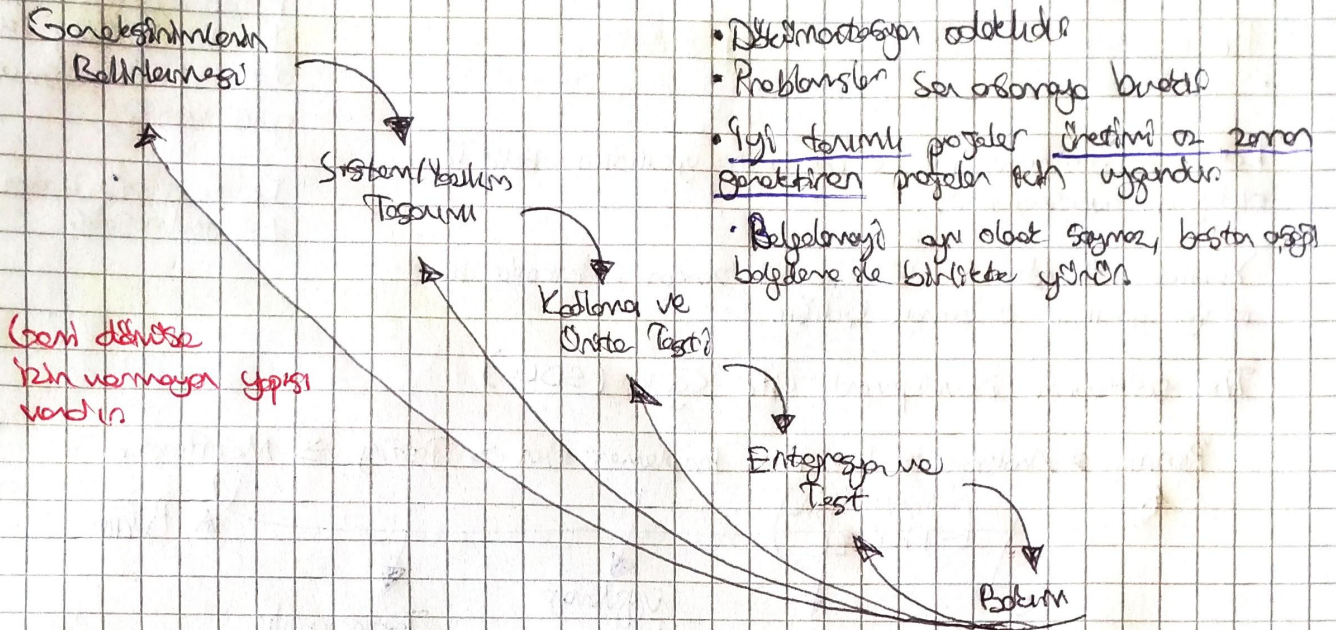
Kalite eşyanın özelliği değildir. Kalite, istenilene ulaşılmışmıdır, sorusuna yanıt verir.

Kalite, Standartın tutturulmasıdır.

Selale (Waterfall) Model

Gereksinimlerin
Belirlenmesi

Sistem/Yazılım
Tasarımı

Kodlama ve
Ünite Testi

Entegrasyon ve
Test

Bakım

Geri dönüşe
izin vermeye yapısı
vardır.

- Dökümantasyon odaklıdır
- Problemler sıra oBoraya bakılır
- İyi tanımlı projeler üretimi az zaman gerektiren projeler için uygundur.
- Belgelemeyi ayrı olarak saymaz, baştan aşağı belgeleme ile birlikte yürür.

V - Model

| | Validation | |
| Requirements | | Acceptance Test |

Validation

| Design | | Component Test |

Validation

| Detailed design | | Unit Test |

| Source Code |

- Probleme bir bütün olarak bakılır.
- Parçala-Fethet yaklaşımı yok. Divide and Conquer.
- Belirsizlikler az, iş tanımının iyi olduğu projelerde kullanılır.

100.000 - 500.000 kod arasında öngörülmektedir.
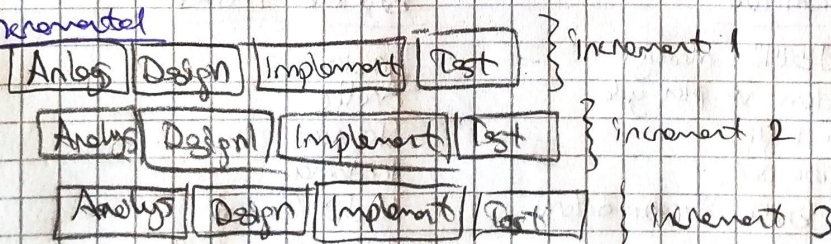
Evrimsel Süreç Modelleri

- Arttırımsal (Incremental)
- Spiral Model
- RUP

Incremental

| Analiz | Design | Implement | Test | } increment 1

| Analiz | Design | Implement | Test | } increment 2

| Analiz | Design | Implement | Test | } increment 3

- Müşteri prototipleri kullanmak zorunda olur. Sahip olur. Bir sonraki prototip sırasında bilgi vardır.

Spiral

Risklerin
Belirlenmesi

Hedeflerin
belirlenmesi

Risk analizi olası ortaya çıkabilir

Sonraki aşamanın planlanması

Geliştirme ve Testler
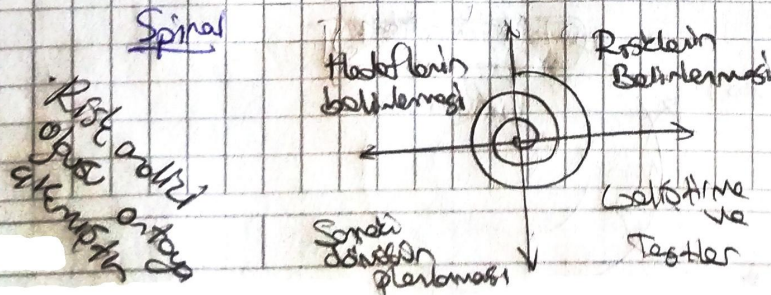
Çevik Modeller AGILE

- Adapte olma ve esneklik ön plana çıktı
- İnsan odaklı ön plana çıktı.

⭐ Agile yaklaşımlarda küçük küçük yazılımlar üretiyoruz. Klasik yaklaşımlarda yazılımın küçük parçasını üretiyoruz.

AGILE MANIFESTOSU

- Bireyler ve onlar arasındaki etkileşimi süreç ve araca tercih etmek
- Çalışan bir yazılımı, detaylı bir dökümantasyona tercih etmek
- Müşteri ile işbirliğini, anlaşma görüşmelerine tercih etmek
- Değişikliklere istenildiğinde cevap verebilmeyi sınırı belirli bir plana tercih etmek