

Nesne Yönelimli Analiz ve Tasarım

Hazırlayan : M. Kemal Güvenç

Öğretmen : Celal Çeken



McCall Kalite Üçgeni

Product Revision

- Maintainability
- Flexibility
- Testability

Product Operation

- Efficiency
- Correctness
- Reliability
- Integrity
- Usability

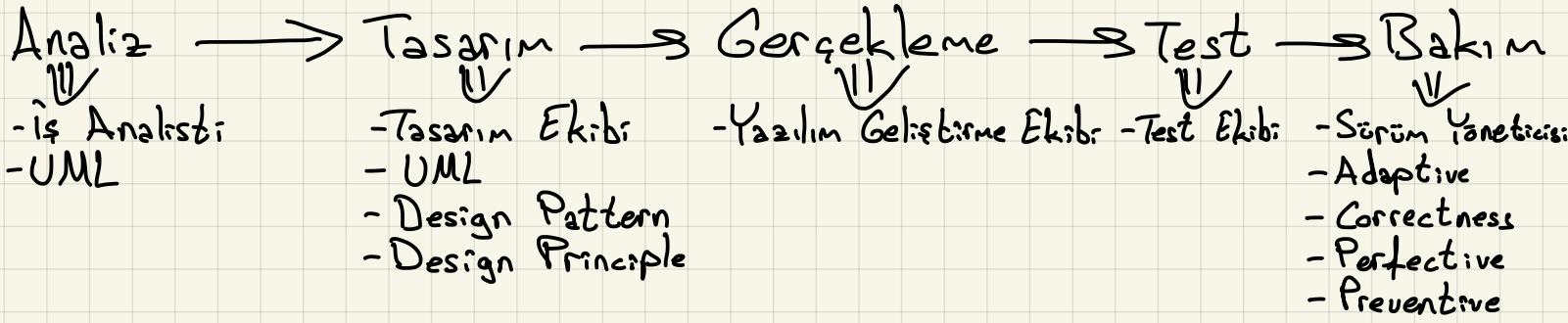
Product Transition

- Portability
- Reusability
- Interoperability

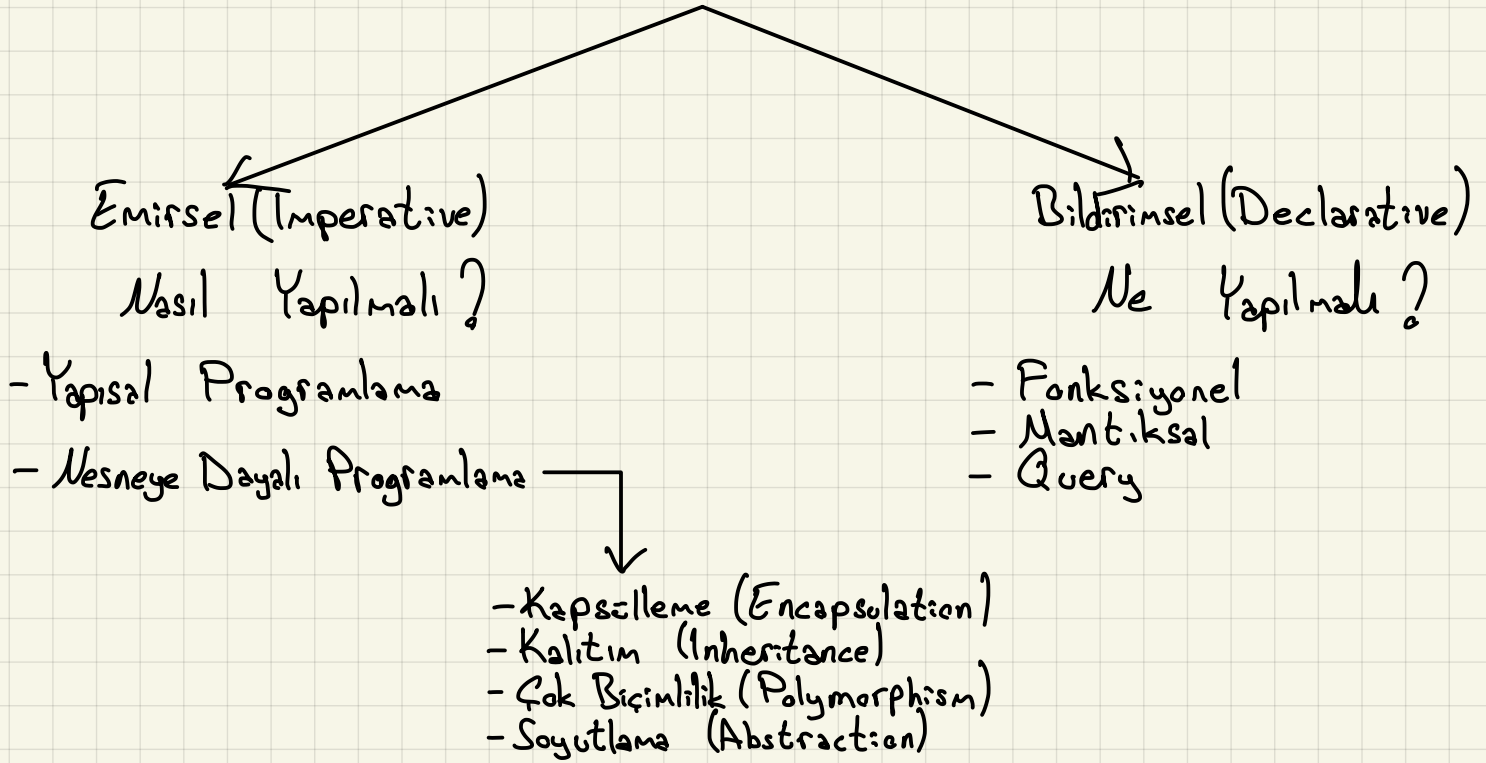
High Coherence

Low Coupling

Yazılım Geliştirme Döngüsü



Programlama Paradigmaları



Tasarım Prensipleri (Design Principles)

Yazılım geliştirirken ihtiyaçlarımıza yönelik olarak kullandığımız prensiplerdir.

S.O.L.I.D: Nesneye Dayalı Programlama'da, sonradan değişikliğe açık yazılım geliştirmemizi sağlayan prensiplerdir.

Tek Sorumluluk Prensipleri (The Single Responsibility Principle):

Bu prensibe göre bir sınıfa birden fazla sorumluluk yüklenmekten kaçınılmalıdır. Bir sınıfın sadece bir sorumluluğu olmalıdır.

Açık-Kapalı Prensipleri (The Open-Closed Principle):

Bu prensibe göre bir yazılımın mevcut kodları değiştirilmeden genişletilebilir.

Liskov Yer Değiştirme Prensipleri (The Liskov Substitution):

Bu prensibe göre bir yazılımda temel sınıfın nesnelerinin kullanıldığı yerlerde türetilmiş sınıfa ait nesneler sıkıntısız çalışmalıdır.

Arayüzleri Ayırma Prensipleri (The Interface Segregation Principle):

Bu prensibe göre bir sınıf gerçekleştirdiği arayüzün (interface) bazı metotlarını kullanmıyorsa gerçekleştirilen arayüz başka arayüzlere ayrılmalıdır.

Bağımlılıkların Azaltılması (The Dependency Inversion):

Bu prensibe göre sınıfların direkt olarak kendisini kullanmak yerine onlar için tasarladığımız arayüzleri (interface) kullanmalıyız.