



Yazılım Mühendisliği

Temel Süreçler - *Tasarım*



HEDEFLER

Tasarım, Sistem Analizi çalışması sonucunda üretilen Mantıksal Modelin Fiziksel Modele dönüştürülme çalışmasıdır.

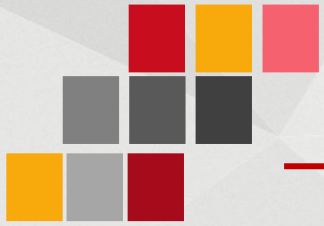
- ✓Tasarım kavramları: Soyutlama, İyileştirme ve Modülerlik olmak üzere 3 çeşittir.
- ✓Yapı Tasarımı, arayüz tasarımı ve süreç tasarımından önce yapılması gereken ilk tasarım veri tasarımıdır.
- ✓Veri Akışları Üç parçada incelenebilir: Girdi Akışı, Çıktı Akışı ve İşlem Akışı
- ✓Süreç tasarımı; veri, yapı ve ara yüz tasarımından sonra yapılır.
- ✓Program Akış Diyagramları: Tekrarlı, ardışıl ve koşullu şeklindedir.
- ✓Tasarlanması Gereken Ortak Alt Sistemler

- Yetkilendirme altsistemi
- Güvenlik altsistemi
- Yedekleme altsistemi
- Veri transferi altsistemi
- Arşiv altsistemi
- Dönüştürme altsistemi



HEDEFLER

- ✓ Tasarım Kavramları
- ✓ MODÜL-işlevsel Bağımsızlık
- ✓ Veri Tasarımı
- ✓ Tasarlanması Gereken ortak alt sistemler
- ✓ Tasarım Kalitesi
- ✓ Bağlılık -Yapışıklık



TASARIM

- ✓ Tasarım, Sistem Analizi çalışması sonucunda üretilen **Mantıksal Modelin Fiziksel Modele dönüştürülme çalışmasıdır.**
- ✓ Fiziksel Model geliştirilecek yazılımın;
 - hangi parçalardan oluşacağını,
 - bu parçalar arasındaki ilişkilerin neler olacağını,
 - parçaların iç yapısının ayrıntılarını,
 - gerekecek veri yapısının fiziksel biçimini (dosya, veri tabanı, hash tablosu, vektör, vs)tasarımını içerir



TASARIM KAVRAMLARI

- ✓ **Soyutlama (abstraction):** Detayları gizleyerek yukarıdan bakabilme imkanı sağlanır.
- ✓ **İyileştirme (enhancement):** Soyutlama düzeyinde irdeleme bittikten sonra, daha alt seviyelere inilerek tanımlamalarda ayrıntı, bazen de düzeltme yapılarak tasarımın daha kesinlik kazanması sağlanır.
- ✓ **Modülerlik (modularity):** Sistemi istenen kalite faktörleri ışığında parçalara ayırıştırma sonucu elde edilir.

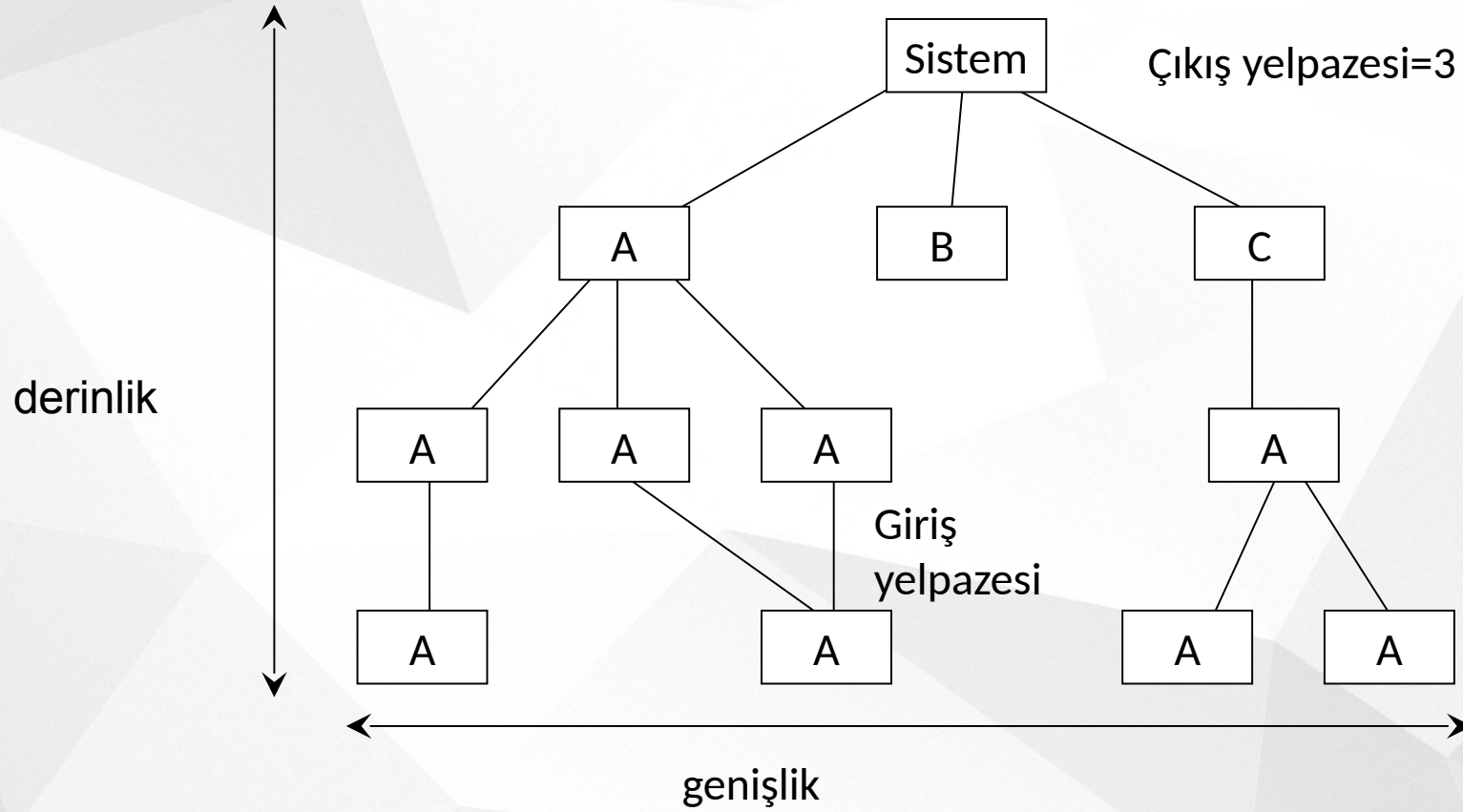


MODÜL

- ✓ Bütün karmaşıklığın tek bir modülde toplanması yerine, anlaşılabilir ve dolayısıyla projenin zihinsel kontrol altında tutulması için sistem bir çok modüle ayrılır.
- ✓ Modüller, isimleri olan tanımlanmış işlevleri bulunan ve hedef sistemi gerçekleştirmek üzere tümleştirilen birimlerdir.



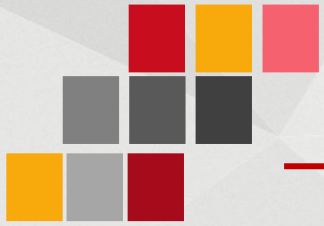
MODÜL





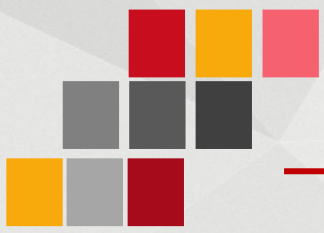
İşlevsel Bağımsızlık

- ✓ Modüllere parametre ile veri gönderilir ve sonuç değeri alınır. Bu modülü çağıran program parçası sadece bu sonucu kullanabilir. Çağrılan modülün işlevsel olarak yaptıkları ile ilgili değildir.



Veri Tasarımı

- ✓ Yapı Tasarımı, arayüz tasarımı ve süreç tasarımıdan önce yapılması gereken ilk tasarım veri tasarımıdır.
- ✓ Bilgi saklama ve soyutlama bu işlem için önemli kavramlardır.



Veri tasarımımda dikkat edilecek konular

- ✓ Değişik veri yapıları değerlendirilmelidir.
- ✓ Bütün veri yapıları ve bunlar üzerinde yapılacak işlemler tanımlanmalıdır.
- ✓ Alt düzeyde tasarım kararları tasarım süreci içerisinde geciktirilmelidir.
- ✓ Bazı çok kullanılan veri yapıları için bir kütüphane oluşturulmalıdır.
- ✓ Kullanılacak programlama dili soyut veri tiplerini desteklemelidir.



Yapısal Tasarım

- ✓ Yapısal Tasarımın ana hedefi modüler bir yapı geliştirip modüller arasındaki kontrol ilişkilerini temsil etmektir.
- ✓ Ayrıca yapısal tasarım bazen de veri akışlarını gösteren biçime dönüştürülebilir.
- ✓ Veri Akışları Üç parçada incelenebilir
 - Girdi Akışı
 - Çıktı Akışı
 - İşlem Akışı



AYRINTI TASARIM- Süreç Tasarımı

- ✓ Süreç tasarımı; veri, yapı ve arayüz tasarımından sonra yapılır.
- ✓ İdeal şartlarda bütün algoritmik detayın belirtilmesi amaçlanır.
- ✓ Ayrıca süreç belirtiminin tek anlamı olması gerekir, değişik şahıslar tarafından farklı yorumlanmamalıdır.
- ✓ Doğal diller kullanılabilir (açıklamalarda, çünkü doğal dil tek anlamlı değildir)
- ✓ Süreç Tanımlama Dili (PDL)



Yapısal Program Yapıları

✓ Yapısal programlamanın temel amacı;

- program karmaşıklığını en aza indirmek,
- program anlaşılabilirliğini artırmaktır.

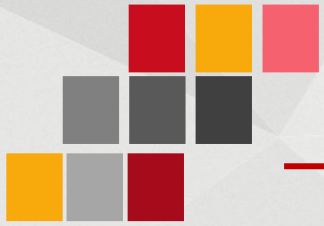
✓ Bu amaçla şu yapılar kullanılır;

- Ardışıl İşlem yapısı
- Koşullu işlem yapısı
- Döngü yapısı



Karar Tabloları

- ✓ Bazen karmaşık koşul değerlendirmeleri yapmak gerekir. Bunların düzenli bir gösterilimi karar tablolarında yapılabilir.
- ✓ Öncelikle, bütün işlemler saptanmalı, sonra ön koşullar belirlenmelidir.
- ✓ Belirli işlemler ile belirli koşulları birleştirerek tablo oluşturulur.
- ✓ Alt tarafta ise işlemler benzer satırlar olarak gösterilir.



Program Tanımlama Dili

- ✓ **Program Tanımlama Dilleri** süreç belirtiminde doğal dillerin programlama dili ile sentezlenmesi şeklinde ortaya çıkmıştır.
- ✓ Hangi programlama dilinin kullanılacağından bağımsız özellikler bulunmalıdır.

DO

Hesap Numarasını Oku

IF (hesap numarası geçerli değil) başlangıca dön
işlem türünü iste

IF (para yatırma işlemi) { para_yatir(); Başlangıca dön }

IF (yeterli bakiye yok) başlangıca dön

WHILE



Tasarlanması Gereken Ortak Alt Sistemler

- ✓ Yetkilendirme altsistemi
- ✓ Güvenlik altsistemi
- ✓ Yedekleme altsistemi
- ✓ Veri transferi altsistemi
- ✓ Arşiv altsistemi
- ✓ Dönüştürme altsistemi



Yetkilendirme Alt Sistemi

- ✓ Özellikle kurumsal uygulamalarda farklı kullanıcıların kullanabilecekleri ve kullanamayacakları özellikleri ifade eder.
 - İşlev bazında yetkilendirme
 - Ekran bazında yetkilendirme
 - Ekran alanları bazında yetkilendirme



Güvenlik Alt Sistemi

- ✓ Yapılan bir işlemde, işlemi yapan kullanıcının izlerinin saklanması işlemleri.
- ✓ LOG files (Sistem günlüğü)



Yedekleme Alt Sistemi

- ✓ Her bilgi sisteminin olağandışı durumlara hazırlıklı olmak amacıyla kullandıkları veri tabanı (sistem) yedekleme ve yedekten geri alma işlemlerinin olması gerekmektedir.



- ✓ Coğrafi olarak dağıtılmış hizmet birimlerinde çalışan makineler arasında veri akışının sağlanması işlemleri
- ✓ Çevirim içi veri iletimi (real-time)
- ✓ Çevirim dışı veri iletimi (disketler, teypler)



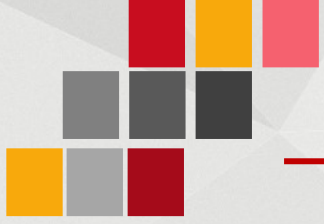
Arşiv Alt Sistemi

- ✓ Belirli bir süre sonrasında sık olarak kullanılmayacak olan bilgilerin ayrılması ve gerektiğinde bu bilgilere erişimi sağlayan alt sistemlerdir.
- ✓ Aktif veri tabanı.



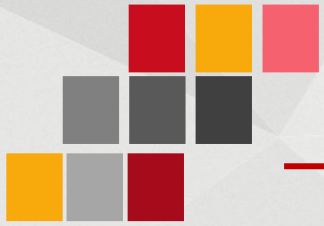
Dönüştürme Alt Sistemi

- ✓ Geliştirilen bilgi sisteminin uygulamaya alınmadan önce veri dönüştürme (mevcut sistemdeki verilerin yeni bilgi sistemine aktarılması) işlemlerine ihtiyaç vardır.



Dönüştürme Alt Sistemi

- ✓ Geliştirilen bilgi sisteminin uygulamaya alınmadan önce veri dönüştürme (mevcut sistemdeki verilerin yeni bilgi sistemine aktarılması) işlemlerine ihtiyaç vardır.



Kullanıcı Arayüz Tasarımı

✓ Kullanıcı ile ilişkisi olmayan arayüzler

- Modüller arası arayüz
- Sistem ile dış nesneler arası arayüz

✓ Kullanıcı arayüzleri

- Kullanım kolaylığı ve öğrenim zamanı esastır
- Program=arayüz yaklaşımı vardır



Genel Prensipler

- ✓ Veri giriş formlarının tutarlı olması
- ✓ Önemli silmelerde teyit alınmalı
- ✓ Yapılan çoğu işlem geri alınabilmeli
- ✓ Hataların affedilmesi, yanlış girişte kırılmama
- ✓ Komut isimlerinin kısa ve basit olması
- ✓ Menülerin ve diğer etkileşimli araçların standart yapıda kullanımı



Bilgi Gösterimi

- ✓ Yalnızca içinde bulunan konu çerçevesi ile ilgili bilgi gösterilmeli
- ✓ Veri çokluğu ile kullanıcı bunaltılmamalı, grafik ve resimler kullanılmalı
- ✓ Tutarlı başlık, renkleme ve kısaltma kullanılmalı
- ✓ Hata mesajları açıklayıcı ve anlaşılır olmalı
- ✓ Değişik tür bilgiler kendi içinde sınıflandırılmalı
- ✓ Rakamsal ifadelerde analog görüntü verilmeli (%89 değil)



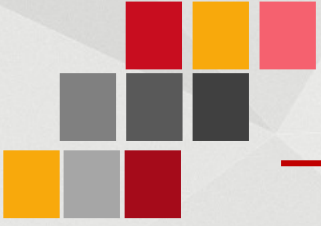
Veri Girişi

- ✓ Kullanıcı hareketleri en aza indirilmeli
- ✓ Gösterim ve girdi sahaları birbirinden ayrılmalı (renk)
- ✓ Kullanıcı uyarlamasına izin verilmeli, kullanıcı bazı özellikleri tanımlayabilmeli
- ✓ Kullanılan konu ile ilgili gereksiz komutlar deaktifleştirilmeli
- ✓ Bütün girdiler için yardım kolaylıkları olmalı



Kullanıcı Arayüz Prototipi

- ✓ Tasarım çalışması sonucunda, daha önceden gereksinim çalışması sırasında hazırlanmış olan kullanıcı arayüz prototipi, ekran ve rapor tasarımları biçimine dönüşür. Ekranlar son halini alır, raporlar kesinleşir. Kullanıcıya gösterilerek onay alınır.
- ✓ Tüm programın tek elden çıktığının ifade edilebilmesi açısından tüm ekranların aynı şablon üzerine oturtulması önerilmektedir.
 - Menü Çubuğu
 - Araç Çubuğu
 - Gövde (Değişebilir)
 - Durum Çubuğu



Başlangıç Tasarım Gözden Geçirme

✓ Yapılan tasarım çalışmasının bir önceki geliştirme aşaması olan analiz aşamasında belirlenen gereksinimleri karşılayıp karşılamadığının belirlenmesidir.

- Sistem gereksinimlerine yardımcı olan kullanıcılar
 - Sistem analizini yapan çözümleyiciler
 - Sistemin kullanıcıları
 - Tasarımcılar
 - Yönlendirici
 - Sekreter
 - Sistemi geliştirecek programcılar
- dan oluşan bir grup tarafından yapılır.



Ayrıntılı Tasarım Gözden Geçirme

- ✓ Başlangıç tasarımı gözden geçirme çalışmasının başarılı bir biçimde tamamlanmasından sonra, tasarımın teknik uygunluğunu belirlemek için **Ayrıntılı Tasarım Gözden Geçirme** çalışması yapılır. Bu çalışmada;
- Çözümleyiciler
 - Sistem Tasarımcıları
 - Sistem Geliştiriciler
 - Sekreter
- den oluşan bir ekip kullanılır.



Tasarım Kalite Ölçütleri

✓ Bağlaşım (Coupling)

Tasarımı oluşturan modüller arası ilişki ile ilgilidir.

✓ Yapışıklık (Cohesion)

Modüllerin iç yapısı ile ilgilidir.



Bağlaşım

- ✓ Modüller arası bağılılığın ölçülmesi için kullanılan bir ölçüttür.
- ✓ Yüksek kaliteli bir tasarımda bağlaşım ölçümü az olmalıdır.
- ✓ Bağlaşımın düşük olması
 - Hatanın dalgasal yayılma özelliğinin azaltılması
 - Modüllerin bakım kolaylığı
 - Modüller arası ilişkilerde karmaşıklığın azaltılmasınedenleri ile istenmektedir



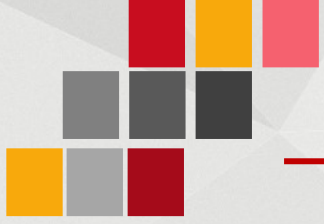
Yalın Veri Bağlaşımı

- ✓ Herhangi iki modül arası iletişim yalın veriler (tamsayı, karakter, boolean, vs) aracılığı ile gerçekleştiriliyorsa bu iki modül yalın veri bağlaşımlıdır şeklinde tanımlanır.



Karmaşık Veri Bağlaşımı

- ✓ Herhangi iki modül arasındaki iletişimde kullanılan parametrelerin karmaşık veri yapısı (*kayıt, dizi, nesne, vs*) olması durumunda modüller karmaşık veri paylaşımı olarak tanımlanır.



Denetim Bağlaşımı

- ✓ İki Modül arasında iletişim parametresi olarak **denetim verisi** kullanılıyorsa bu iki modül denetim bağlaşımlı olarak tanımlanır.



Ortak Veri Bağlaşımı

- ✓ Eğer iki modül ortak bir alanda tanımlanmış verilere ulaşabiliyorsa bu iki modül **ortak veri bağlaşımlı** olarak tanımlanır.
- ✓ Verilerin ortak veri bağlaşımlı olmaları şu nedenlerden dolayı fazla istenmez;
 - Ortak veri alanını izlemek zordur.
 - Ortak veri kullanan modüllerde yapılan değişiklikler diğer modülleri etkiler.
 - Ortak veri üzerinde yapılacak değişikliklerde bu veriyi kullanacak bütün modüller göz önüne alınmalıdır.



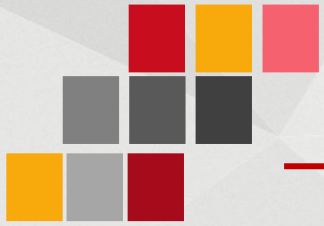
Ortak Veri Bağlaşımı

- ✓ Eğer iki modül ortak bir alanda tanımlanmış verilere ulaşabiliyorsa bu iki modül **ortak veri bağlaşımlı** olarak tanımlanır.
- ✓ Verilerin ortak veri bağlaşımlı olmaları şu nedenlerden dolayı fazla istenmez;
 - Ortak veri alanını izlemek zordur.
 - Ortak veri kullanan modüllerde yapılan değişiklikler diğer modülleri etkiler.
 - Ortak veri üzerinde yapılacak değişikliklerde bu veriyi kullanacak bütün modüller göz önüne alınmalıdır.



İçerik Bağlaşımı

- ✓ Modüllerin iç içe tasarlanması sonucu, bir modülün başka bir modül içerisinde tanımlanmış veri alanına erişebilmesi olanaklaşır ve bu durum **içerik bağlaşımına** yol açar.



Yapışıklık

- ✓ Bir modülün kendi içindeki işlemler arasındaki ilişkilere ilişkin bir ölçüttür. **Modül gücü** olarak ta tanımlanır.
- ✓ Tasarımda **yapışıklık** özelliğinin yüksek olması tercih edilir.
- ✓ Yapışıklık ile Bağlaşım ters orantılıdır.



İşlevsel Yapışıklık

- ✓ İşlevsel Yapışık bir modül, tek bir iş problemine ilişkin sorunu çözen modül olarak tanımlanır.
- ✓ Maas_Hesapla, Alan_Hesapla gibi



Sırasal Yapışıklık

- ✓ Bir modülün içindeki işlemler incelendiğinde, bir işlemin çıktısı, diğer bir işlemin girdisi olarak kullanılıyorsa bu modül **sırasal yapışık** bir modül olarak adlandırılır.

Ham_Veri_Kaydını_Düzeltil

Düzeltilmiş_Ham_Veri_Kaydını_Dogrula

Dogrulanmis_Kaydi_Gonder



İletişimsel Yapışıklık

- ✓ Bir modülün içindeki farklı işlemler aynı girdi ya da çıktıyı kullanıyorlarsa bu modül **iletişimsel yapışık** bir modül olarak adlandırılır.

Sicil_No_yu_Al
Adres_Bilgisini_Bul
Telefon_Bilgisini_Bul
Maas_Bilgisini_Bul



Yordamsal Yapışıklık

- ✓ Yordamsal Yapışık modüldeki işlemler arasında denetim ilişkisi bulunmaktadır.
- ✓ İşlemlerin birbirleri ile veri ilişkisi yoktur, ancak işlem sırası önemlidir.

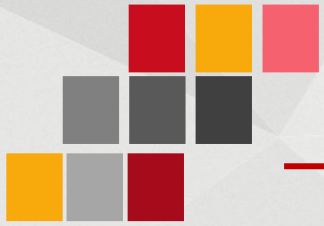
Ekran_Goruntusunu_Yaz
Giris_Kaydini_Oku



Mantıksal Yapışıklık

- ✓ Mantıksal olarak aynı türdeki işlemlerin bir araya toplandığı modüller **mantıksal yapışık** olarak adlandırılır.

Dizilere değer atama işlemleri



Gelişigüzel Yapışıklık

✓ İşlemler arasında herhangi bir ilişki bulunmaz.

Ara_Kayit_Oku

B_dizisine_baslangic_deger_ata

Stok_kutugu_oku

Hata_iletisi_yaz