

YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ

6.Hafta

TASARIM



Bölüm Hedefi

- Bu bölümde, yapısal sistem geliştirme metodolojisine göre yapılan bir çözümlemenin tasarıma dönüştürülmesi, tasarım kavramları, veri tasarım ve süreç tasarım yöntemleri açıklanmaktadır. Her tür yazılım için gerekecek ortak alt sistemler tanıtılmaktadır.



Tasarıma Giriş

- 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.

Çözümleme-Tasarım İlişkisi



Kuramsal

Teknik

SRS→SAD

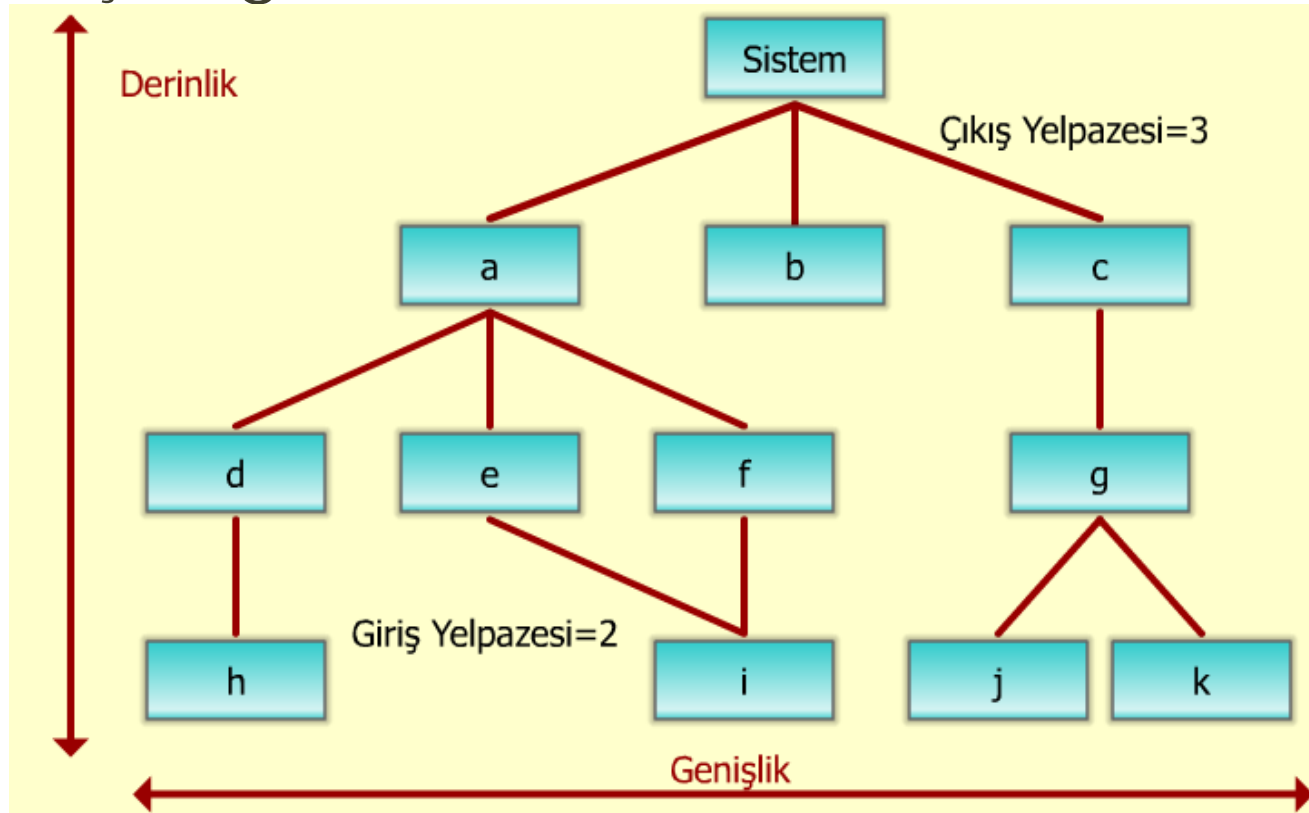
Tasarım Kavramları

- 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.

Modülerlik

- 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.

- Şekil sistemin modüllerden oluşmasını simgelemektedir. Aynı zamanda, sistemin '**kontrol hiyerarşisi**'ni de belirtmektedir. Bir modülün hangi modülleri kullanarak işlevini gerçekleştirdiği bu hiyerarşide görünmektedir.



İşlevsel Bağımsızlık

- Modüllere parametre ile veri gönderilir ve sonuç değer 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 tasarımından önce yapılması gereken ilk tasarım veri tasarımıdır.
- Bilgi saklama ve soyutlama bu işlem için önemli kavramlardır.



Hizmetlerimiz

Veri tasarımı yaparken 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

Ayrıntı Tasarım- 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 yapıları kullanılır;
 - Ardışıl İşlem yapısı
 - Koşullu işlem yapısı
 - Döngü yapısı
- GOTO kullanımı uygun değildir.

Ardışıl İşlem Yapısı

Şartlı İşlem Yapısı

Tekrarlı İşlem Yapıları

Sayı oku

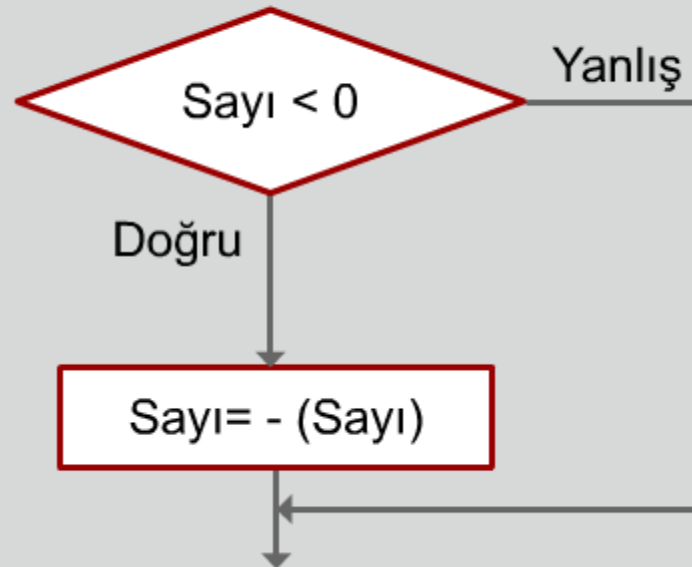
2 ile Çarp

Sonucu Göster

Ardışıl İşlem Yapısı

Şartlı İşlem Yapısı

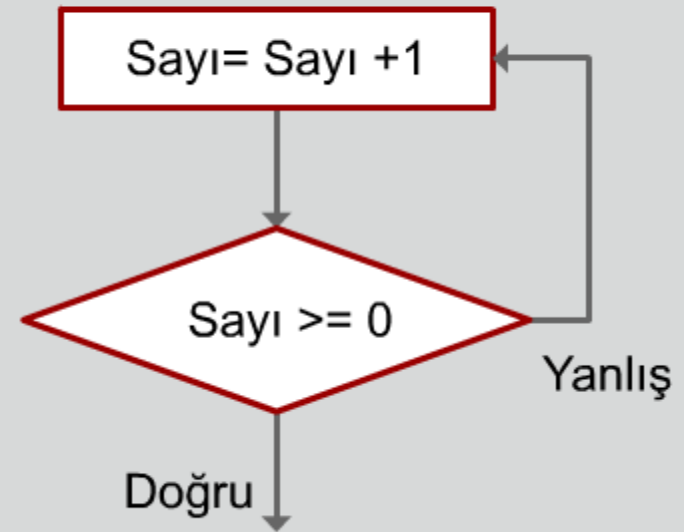
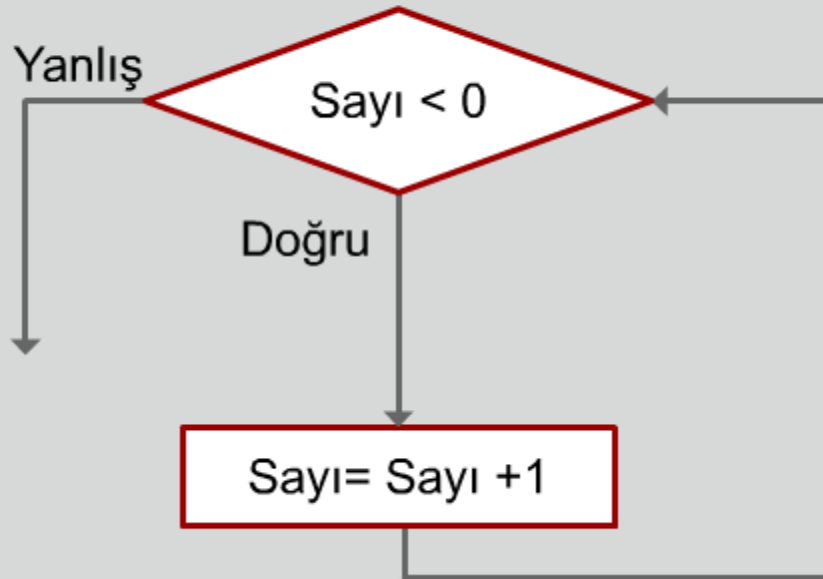
Tekrarlı İşlem Yapıları



Ardışıl İşlem Yapısı

Şartlı İşlem Yapısı

Tekrarlı İşlem Yapıları



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.

Karar Tablosu Örneđi

Kurallar	1	2	3	4
Hesap Geçerli	X	X	X	X
Şifre Geçerli	X		X	X
Yeterli Nakit Var	X			
Yeterli Bakiye Var	X		X	
Bakiye Bildirme İşlemi				X
Para Çekme İşlemi	X			
Para Yatırma İşlemi		X		
Para Gönderme İşlemi			X	

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
- Oracle veri tabanına erişim konusunda yetkilendirme yapmaktadır.

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.



Veri İletişim Alt Sistemi

- 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.



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 deaktivleştirilmeli
- 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
 - Sekreterden oluşan bir ekip kullanılır.

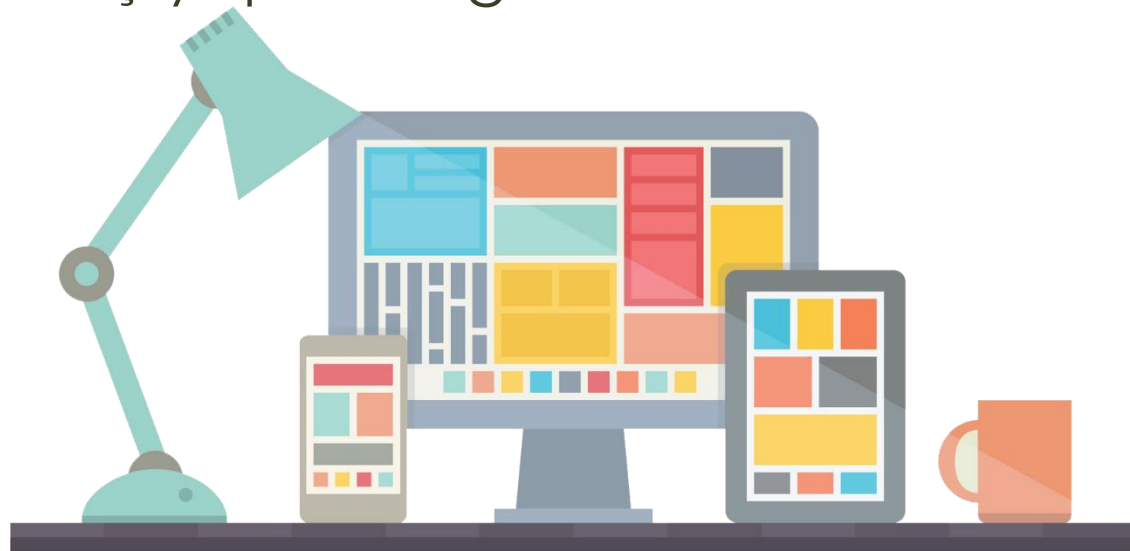
Tasarım Kalite Ölçütleri

- Bağlaşım (Coupling)

Tasarımı oluşturan modüller arası ilişki ile ilgilidir.

- Yapışıklık (Cohesion)

Modüllerin iç yapısı ile ilgilidir.



Bağlaşım

- Modüller arası bağıllığın 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 Baęlaşımı

- Herhangi iki mod l arası iletiřim yalın veriler (tamsayı, karakter, boolean, vs) aracılığı ile ger ekleřtiriliyorsa bu iki mod l yalın veri baęlaşımıdır řeklinde tanımlanır.



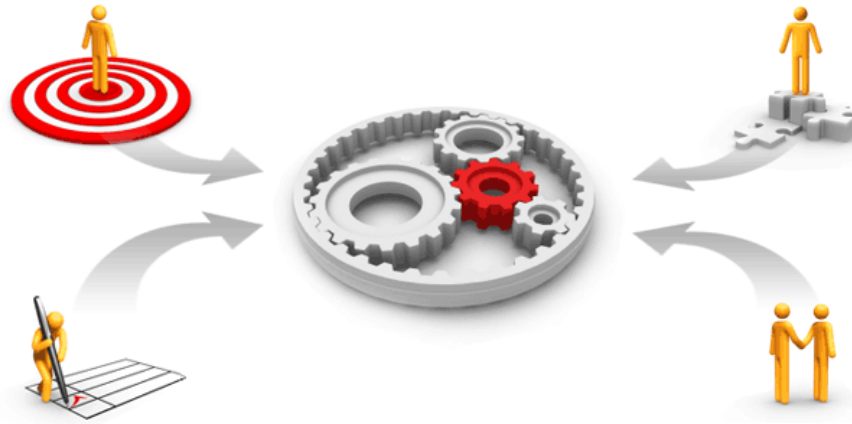
Karmaşık Veri Bağlaşımı

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



Denetim Baęlaşımı

- İki Modül arasında iletişim parametresi olarak denetim verisi kullanılıyorsa bu iki modül denetim baęlaşımli olarak tanımlanır.

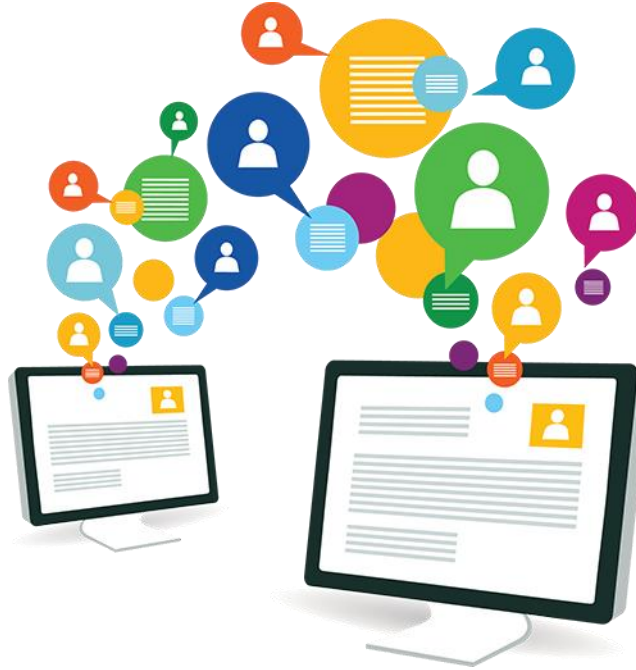


Ortak Veri Bağlaşımı

- 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.

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_Düzelt

Duzeltilmis_Ham_Veri_Kaydini_Dogrula

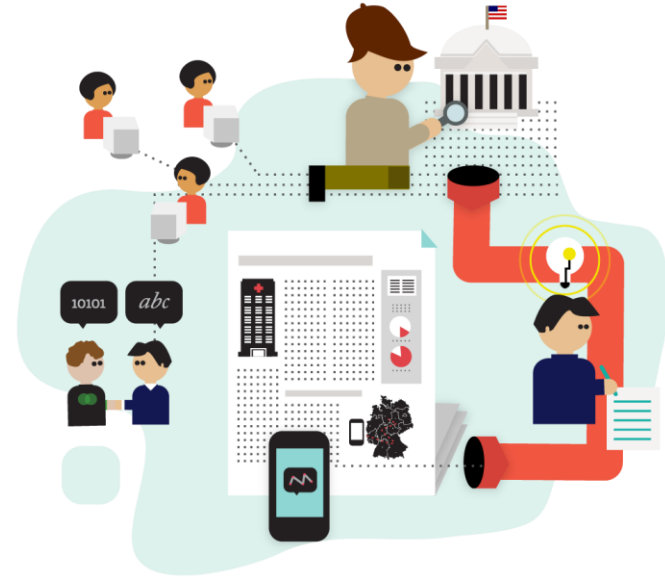
Dogrulanmis_Kaydi_Gonder



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



Zamansal Yapışıklık

- Bir modül içindeki işlemlerin belirli bir zamanda uygulanması gerekiyor ve bu işlemlerin kendi aralarında herhangi bir ilişkisi yok, yani işlemlerin sırası önemli değil ise, **zamansal yapıışıklık** vardır.

Alarm_Zilini_Ac

Kapiyi_Ac

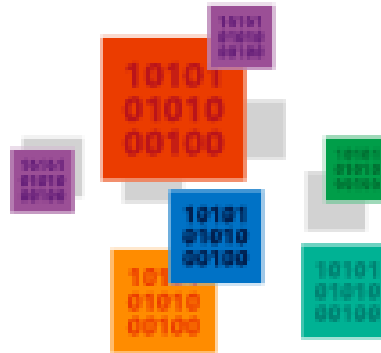
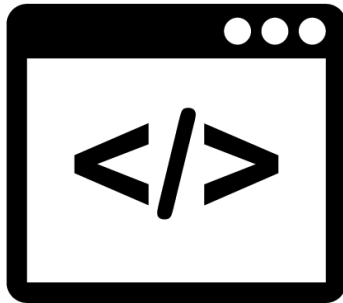
Kamerayi_Calistir



Mantıksal Yapışıklık

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

Dizilere değer atama işlemleri



Gelişigüzel Yapışıklık

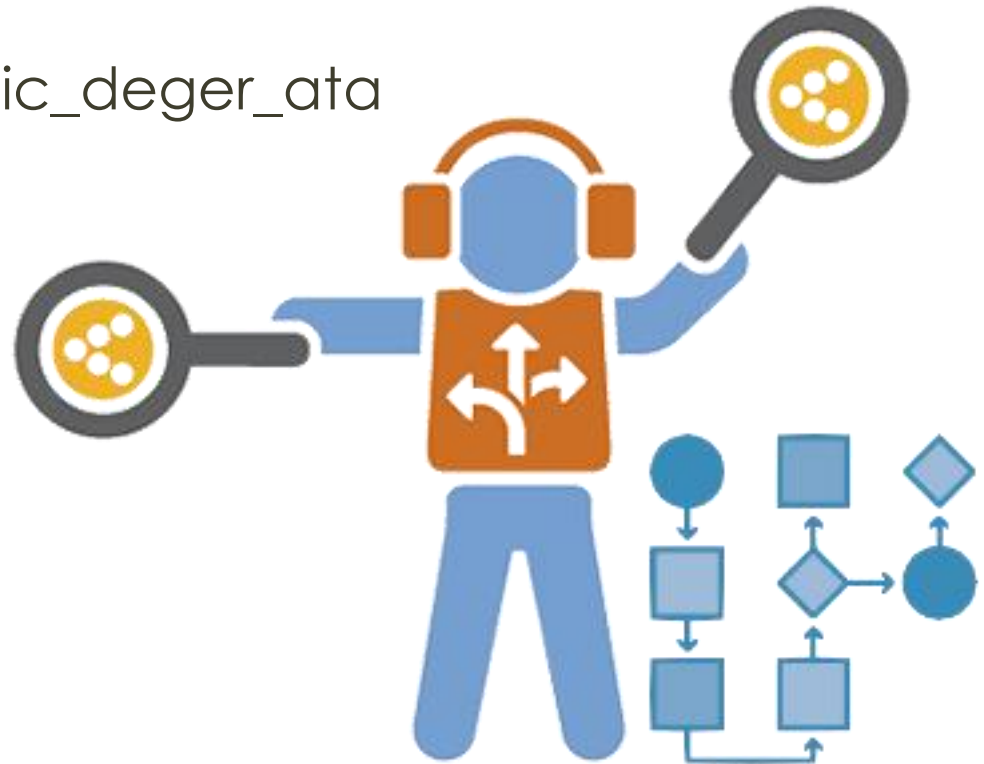
- İşlemler arasında herhangi bir ilişki bulunmaz.

Ara_Kayit_Oku

B_dizisine_baslangic_deger_ata

Stok_kutugu_oku

Hata_iletisi_yaz



Yapışıklık Hesaplama

- Genel olarak bir modülün yapışıklık düzeyini hesaplamada bazı güçlüklerle karşılaşılabilir. Çünkü, bir modüldeki işlemler hep aynı tür özellik göstermeyebilir.

Yapışıklık Türü	Ölçek	Kalite
İşlevsel	1	En iyi  En kötü
Sırasal	2	
İletişimsel	3	
Yordamsal	4	
Zamansal	5	
Mantıksal	6	
Gelişigüzel	7	

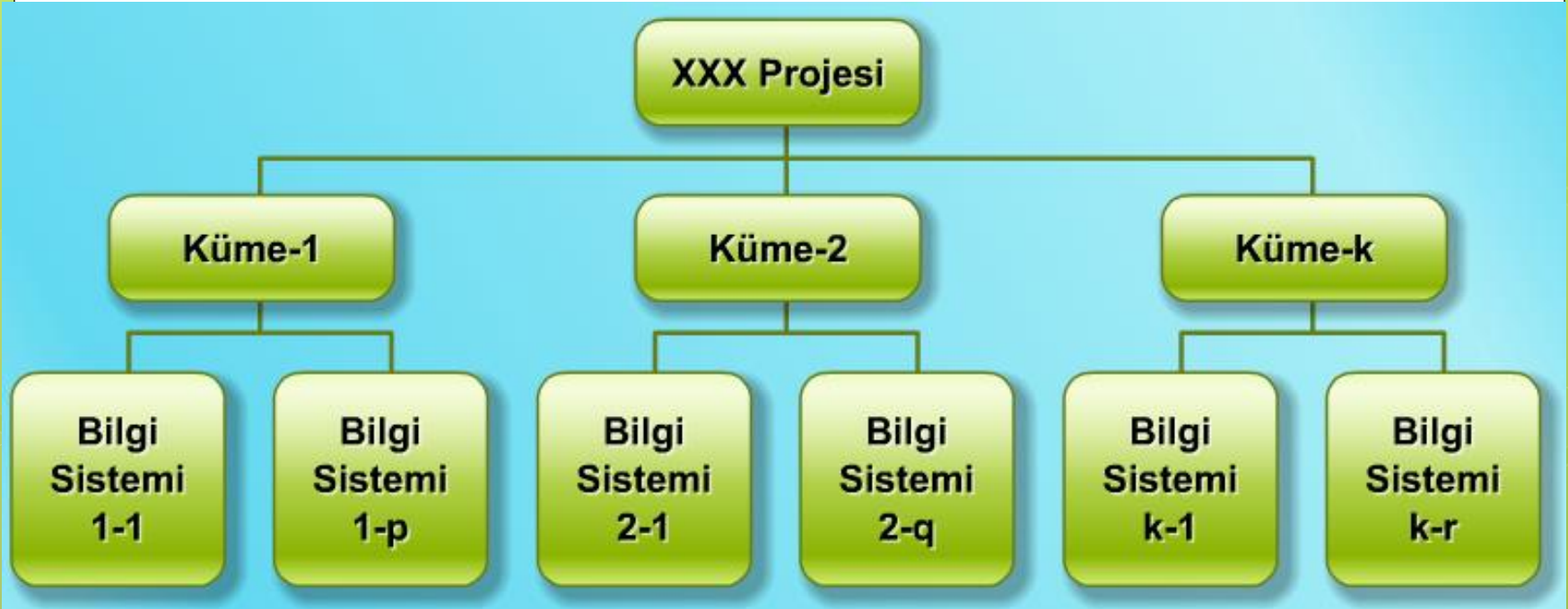
Yapışıklık Düzeyine Karar Verilmesi



TASARIM RAPORU İÇERİĞİ

- İçeriğin hazırlanmasında IEEE/ANSI standartları kullanılmıştır. Büyük boyutlu projeler için burada verilen içerik, genel mantık aynı kalmak koşulu ile detaylandırılabilir. Örneğin 200 kişi-yıllık işgücü gerektiren bir BT projesi, Şekil'deki gibi bölümlenebilir.
- Ödev IEEE 1016-1998

Tasarım Raporu İçeriği



Örnek Tasarım Raporu İçeriği

1. Giriş

- 1.1. Kapsam
- 1.2. Tanımlar, Kısa Adlar ve Kısaltmalar
- 1.3. Referanslar
- 1.4. Belgenin Yapısı

2. Genel Tasarım Bilgileri

- 2.1. Genel Sistem Tanımı
- 2.2. Varsayımlar ve Kısıtlamalar
- 2.3. Sistem Mimarisi
- 2.4. Dış Arabirimler
 - 2.4.1. Kullanıcı Arabirimleri
 - 2.4.2. Veri Arabirimleri
 - 2.4.3. Diğer Sistemlerle Arabirimler
- 2.5. Veri Modeli
- 2.6. Testler
- 2.7. Performans

Örnek Tasarım Raporu İçeriği

3. Veri Tasarımı

- 3.1. Tablo tanımları
- 3.2. Tablo- İlişki Şemaları
- 3.2. Veri Tanımları
- 3.3. Değer Kümesi Tanımları

4. Süreç Tasarımı

- 4.1. Genel Tasarım
- 4.2. Modüller
 - 4.2.1. XXX Modülü
 - 4.2.1.1. İşlev
 - 4.2.1.2. Kullanıcı Arabirimi
 - 4.2.1.3. Modül Tanımı
 - 4.2.1.4. Modül iç Tasarımı
 - 4.2.2. YYY Modülü
- 4.3. Kullanıcı Profilleri
- 4.4. Entegrasyon ve Test Gereksinimleri

Örnek Tasarım Raporu İçeriği

5. Ortak Alt Sistemlerin Tasarımı

- 5.1. Ortak Alt Sistemler
- 5.2. Modüller arası Ortak Veriler
- 5.3. Ortak Veriler İçin Veri Giriş ve Raporlama Modülleri
- 5.4. Güvenlik Altsistemi
- 5.5. Veri Dağıtım Altsistemi
- 5.6. Yedekleme ve Arşivleme İşlemleri

EKLER

- EK-A Rapor Görünümleri
- EK-B Test Belirtileri
- EK-C Kullanıcı Kılavuzu Örneği

Alınan Dersler 1

- Ortak alt sistemler oldukça önemlidir. Tasarlanması ve geliştirilmesi için ayrı bir yazılım ekibi kullanılması gerekir. Aksi durumda, tutarsızlık ve işlem tekrarı gündeme gelir, işgücü kaybına neden olunur.

Alınan Dersler 2

- Özellikle, coğrafik olarak birden çok alana yayılmış kuruluşlar için geliştirilecek bilgi sistemlerinde gerek merkez gerekse bağlı birimlerde çalışacak yazılımın aynı yazılım olarak tasarlanıp geliştirilmesi uygulama ve bakım kolaylığı açısından önerilmektedir.

Alınan Dersler

- Genellikle bu tür uygulamalarda, merkezdeki yazılım, birimlerdeki yazılımlara oranla daha farklı unsurlar içermekte, ancak birimlerdeki uygulamaları da kapsamaktadır.
- Bu durumda, neden merkezdeki yazılımın işlevlerini, kullanılmayacaklarını bile bile birimlere de taşıyalım sorusu akla gelebilir.

Alınan Dersler

- Zira bu durum kaynak kullanımı açısından birimlere ek yük getirmektedir. Bu durumda bakım ve kurulum kolaylığı ve fazla kaynak kullanımı arasında bir tercih yapmak gerekmektedir.
- Günümüzde, bilgisayar kaynaklarının giderek ucuzladığı düşünüldüğünde bu tercihin, bakım ve kurulum kolaylığı yönünde yapılması önerilmektedir.

Alınan Dersler 3

- Genellikle kurumsal BT projelerinde veri iletişiminin tüm türlerine (Çevrim-içi anında, çevrim-içi kargo, çevrim dışı) rastlanmaktadır. Bu durumda, Veri İletişim alt sistemlerinin bu özellikleri içerecek biçimde tasarlanması gerekmektedir.
- Özellikle çevrim içi ve anında iletişim gerektiren durumlarda, işlemlerin izlenmesi ve herhangi bir aksaklık durumunda geri alınması gibi işlevlerin tasarımda içerilmesi gerekmektedir.

Alınan Dersler 4

- Dönüştürme işlemleri, yazılım gereksinim tanımlarında oldukça az ve yetersiz biçimde tanımlanmakta ve bu durum yüklenici ile iş sahibi arasında sorunlara yol açmaktadır.
- Geçmiş uygulamalarda, bilgisayar ortamında saklanan bilgilerin yeni uygulama ortamında oluşturulacak veri tabanına aktarılmasında ortaya çıkan temel sorun, geçmiş verilerdeki tutarsızlıklardır.

Alınan Dersler

- Bir taraftan bu tür tutarsızlıkların yeni sistemde bulunmaması öngörülürken, öte yandan tutarsızlık içeren geçmiş verilerin aktarılması istenir. Bu durum tasarımda içerilmemişse büyük zorluklarla karşılaşılır.
- Örneğin, nüfus uygulamasındaki kişilerin evlilikleri ile ilgili bağ alanlarının dolu olması bir zorunluluk olarak öngörüldüğünü varsayalım.

Alınan Dersler

- Öte yandan, bilgisayar ortamındaki mevcut verilere bakıldığında bir çok bağ alanının boş olduğu görüldüğünde mevcut verilerin aktarımında zorluklar çıkar.
- Veri tabanındaki kısıtlar kaldırılıp veriler olduğu gibi aktarılrsa bile, bu verilere ilişkin ileride ortaya çıkabilecek işlemlerde yine sorunlar oluşabilir.

Alınan Dersler

- Bu tür durumların, yazılım gereksinim tanımlarında ayrıntılı olarak tanımlanması, tasarım sırasında dikkatlice ele alınmalarını sağlar.

Sorular

1. Yazılım tasarım sürecinin temel işlemlerini sayınız. E-R diyagramı çizerek ilişkilerini gösteriniz.
2. Geliştireceğiniz bir uygulama için ekran şablonunuzu belirleyiniz.
3. Bağlaşım ve yapışıklık kavramlarını açıklayınız. Program bakımı ile ilişkilerini belirtiniz.
4. Bir sistem tümüyle bağlaşımsız biçimde tasarlanabilir mi? Yani sistemin tüm modülleri arasında hiç bağlaşım olmadan tasarım yapılabilir mi?
5. Tümüyle işlevsel yapışık modüllerden oluşan bir sistem tasarlanabilir mi? Neden yapılabilir? Neden yapılamaz?
6. Bağlaşım ile Yazılım Taşınabilirliği arasındaki ilişkiyi belirtiniz.

Genel

- Ders Kitabı: Yazılım Mühendisliği
Erhan Sarıdoğan- papatya Yayıncılık
(kitapyurdu.com)

Diğer Kaynaklar:

- Ders Notları.
- Ali Arifoğlu, Yazılım Mühendisliği. SAS bilişim Yayınları
- İnternet, UML Kaynakları
- Roger S. Pressman, Software Engineering – Practitioner's Approach