Sistem Analizi ve Tasarımı

BLM2042 GR.1/GR.2

2025-2026 BAHAR YARIYILI

DR.ÖĞR.ÜYESİ YUNUS EMRE SELÇUK / GÖKSEL BİRİCİK

Ön Bilgi

Ders: Salı 13.30-15.20

Gr.1 YES D-111 yselcuk@yildiz.edu.tr

Gr.2 G1 D-012 gbiricik@yildiz.edu.tr

Kopya Kuralları:

- Herhangi bir şekilde proje veya sınavlarda hazır kaynaklardan / başkalarından kopyalama, ortak çözüm ve hile yapılması durumunda, ilgili tüm taraflar projeden/sınavdan 0 alırlar.
- Bu gibi işlemler disiplin yönetmeliği uyarınca değerlendirilecektir.

Kaynaklar

Sınıfta sunulan materyallerden sorumlusunuz.

Ders kitapları ve diğer ek materyaller sadece bilgilendirme ve yol gösterme amaçlıdır.

Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı 4.Basım, O.Kalıpsız, A.Buharalı Olcaysoy, G.Biricik, Papatya Yayıncılık.

System Analysis and Design in a Changing World 5th Edition, J.W.Satzinger, Cengage Learning.

Ders Planı-1

Hafta	Tarih	Konu
1	24.Şub	Giriş, Ders Planı, Genel Bilgiler, Sistem Tanımı, Sistem Tipleri, Tarafları, Sistem Geliştirme Süreci (Waterfall)
2	03.Mar	Ön İnceleme – Fizibilite Çalışmaları, Zaman Fizibilitesi
3	10.Mar	Sistem Analizi – Veri Toplama, İş analizi, Kullanım Senaryoları
4	17.Mar	Sistem Analizi – Kullanım Senaryoları, Veri Modelleme
5	24.Mar	Sistem Analizi - Fonks.Çözümleme
6	31.Mar	Sistem Tasarımı – Mimari Tasarım (Yapı D.)
7	7.Nis	Sistem Tasarımı – Girdi, Çıktı, Veri Yapısı, Arabirim Tasarımları

Ders Planı-2

Hafta	Tarih	Konu
8	14.Nis	Vize Sınavı
9	21.Nis	Veritabanı Tasarımı, Kodlama, Test Süreçleri
10	28.Nis	Yeni Sisteme Geçiş, Bakım, Destek Süreçleri, Sistem Analizi ve Tasarımı Örnek Vaka Çalışması
11	5.May	Proje Sunumları
12	12.May	Proje Sunumları
13	19.May	Resmi Tatil
14	26.May	Resmi Tatil

Değerlendirme Kriterleri

Tipi	Ağırlığı	İçerik	Tarih
Vize	% 30	Derste Anlatılan Konuların Değerlendirilmesi	14 Nisan Haftası
Proje	% 30	Seçilen Bir Konuda Ön İnceleme, Fizibilite, Analiz, Tasarım, Kodlama, Veritabanı Tasarımı ve Destek Aktiviteler de Dahil Olacak Şekilde Bir Bilgi Sisteminin Gerçekleştirilmesi	Öneriler: 10 Mart 2026 Değerlendirme Sonuçları: 13 Mart 2026 Rapor Teslimi: 03 Mayıs 2026 Pazar 23.45
Final	%40	Dersteki Konuların Tümünün Genel Değerlendirilmesi	Final Haftaları

Sistem Analizi ve Tasarımına Giriş

Sistem - TDK sözlük anlamı:

sistem
isim *Fransızca système*

1. *isim* Düzen
"*Açıklamasının arkasına yeni bir ücretlendirme sistemi getirdiğini ekledi.*" - **L. Tekin**

2. Bir sonuç elde etmeye yarayan yöntemler düzeni
"*Servet, nasıl kazanılmış olursa olsun, onun kontrolüne girecek rejim ve sistem memleketi mahvedecektir.*" - **H. E. Adıvar**

3. Yol, yöntem
"*Eski bir sistem.*"

4. Bir aracı oluşturan düzen, düzenek, tertibat
"*Fren sistemi.*"

5. Model, tip
"*Son sistem, pırıl pırıl bir rotatif almışlar.*" - **Y. Z. Ortaç**

6. *felsefe* Dizge

Giriş

İnsan, hammadde vb kaynaklar önemli

Günümüzde, bilgi en önemli kaynak

Bilginin doğru kullanımı, rekabet gücünü arttırır

- Bilgi üretimi, dağıtımı, işlenmesi, güvenliği, depolanması önemli
- İnternet, elle üretilen verinin çok daha fazlasını üretiyor
- Verilerin organizasyonu, kullanım maliyetleri arttı

Çözüm? →Bilgi Sistemleri

Bilgisayar programından farkı ne?

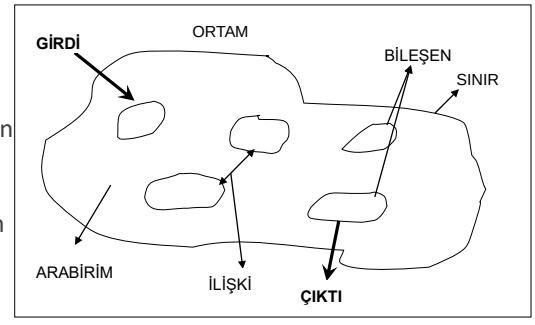
Sistem

Belli bir amacı gerçekleştirmek için

Bir arada çalışan, birbiriyle ilişkili olan parçalardan oluşan

Girdi ve çıktıları olan

Sınırları belirlenmiş bir bütün yapı/oluşum/çözüm



Sistemin Temel Özellikleri

Bileşenler: Sistemi oluşturan parçalar

Değişkenler: Değişik değer alan özellikler

Parametreler: Sabit değerler

İlişkiler: İlk üç madde arasındaki bağlantılar

Sınır: Sistem için ortam ayracı

Arabirim: Ortam veya alt sistemlerle karşılaşma noktası

Kısıtlar: Değişken değerleri ve kaynak tahsis sınırlamaları

Ölçüt: Hedef-amaçların değerlendirme standardı

Ortam: Sistem dışındaki her şey

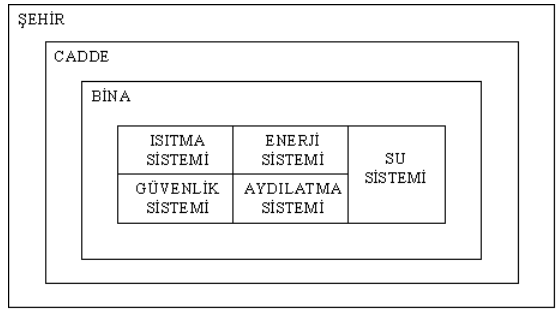
Geri Besleme: Çıktı kontrolü ve ölçme-değerlendirme ile girdiler ve sistem üzerinde iyileştirme yapma aracı

Sistem ve Ortam

Sistem ortama bağlıdır

- Aralarında doğrudan veya dolaylı ilişki vardır

Sistem ortamda değişiklik yapabilir



Sistem Örneği

Sistem Örneği

Bilgi Sistemi

İşletmenin ihtiyaçlarını desteklemek için

Veri toplayan

İşleyen

Depolayan

İnsan, veri, süreçler ve bilgi teknolojilerinin etkileşimde bulunduğu yapı.

Bilgi Sistemi Bileşenleri

Donanım Kaynakları
- Sunucu, bilgisayar, monitor, klavye, yazıcı, tablet, vs. sayısal ürünler

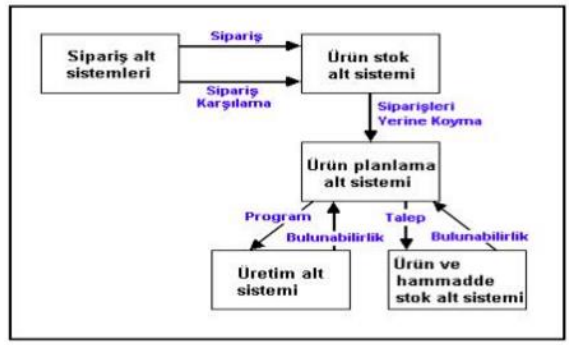
Yazılım Kaynakları
- Veri düzenleme, işleme, analiz programları. Bunlara ait süreçler ve yordamlar

İnsan Kaynakları
- Bilgi sistemi sahibi, tasarlayanlar, kuranlar, kullananlar.

Veri Kaynakları
- Sistemin kullandığı ve ürettiği verileri tutan veritabanı ve bilgi tabanı

Ağ Kaynakları
- İşletme içi ve dışı birimlerin bilgi sistemine bağlanma yapıları

Bilgi Sistemi Örneği



Genel Sistem Teorisi

Sistem; girdiyi çıktıya dönüştürür
- Bilgi sisteminde veri→bilgi

Sistemler disiplinler arasıdır
- Bir daldaki ürün başka dalda kullanılabilir

Sistem elemanları arası etkileşim vardır
- Bir parçadaki etki diğer parçaları da etkiler

Sistemler farklı elemanlardan oluşur
- Taşımacılık→ demiryolu, denizyolu, havayolu..

Sistemler hiyerarşiktir
- Her sistem alt ve üstünde sistemler vardır

Sistem ortama göre düzenlenmelidir
- Entropi ortamla ilişkiyi belirler

Sistem amaç yönelimlidir
- Her sistemin belli bir amacı vardır

İnceleyin:

Örnek bir sistem nedir, bileşenleri, ortamı, sınırları, arabirimleri, … nelerdir?

Nasıl işler?

1.Kaynakta, Hastane Sistemi Örneği

Bilgi Sistemleri

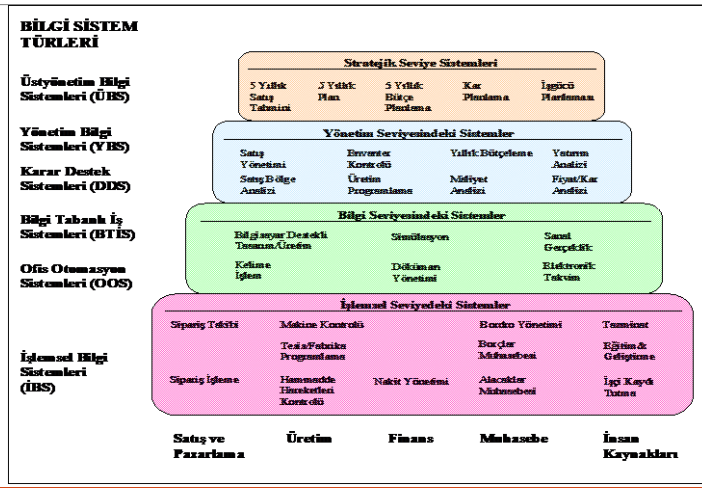
Amaç: Problemlerin analizi, işlerin yürütülmesi, yeni ürün oluşturulması, yönetim faaliyetlerine destek, karmaşık nesnelerin canlandırılmasında kontrolü ve koordinasyonu sağlayarak karar almaya destek olmak, …

Girdi: Her türlü bilginin toplanması

İşlem: Bilginin işlenmesi, saklanması, dağıtılması

Çıktı: İşlenmiş ve ilişkilendirilmiş bilgi

Bilgi Sistemi Türleri



İşlemsel Bilgi Sistemleri

Başka sistemlere giriş ve temel oluşturur

İşlenmiş veriyi:

- Sınıflar
- Saklar
- Bakımını Yapar
- Değiştirir
- Güncelleştirir
- Geri çağırır

Örnekler

- Satış izleme
- Envanter düzenleme
- Fatura hazırlama
- Kartlı geçiş

Girdiler:
Veri toplama, hazırlama, giriş

↓

İşlem:
Saklama, geri çağırma, sınıflama, sıralama, güncelleme

↓

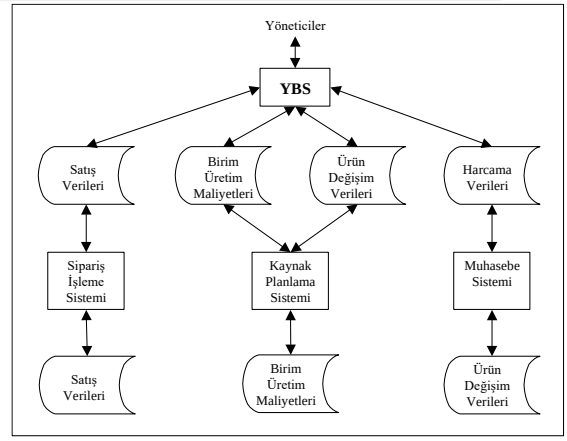
Çıktılar:
Raporlar, Sorgu cevapları, Diğer sistem girdileri, ürün

Yönetim Bilgi Sistemleri

İşlem boyutu: Rapor, analiz, KDS
- Uygulama tabanı
- VTYS
- VT

Yönetim boyutu
- Alt yönetim için işlemsel kontrol ve güncelliği,
- Orta yönetim için kontrollü kaynak paylaşımı
- Üst yönetim için stratejik planlama ve hedefler

İşlevsel boyut
- Kurumsal işlevler için birimler arası bilgi farkları



Ofis Otomasyon Sistemleri

Büro işlemleri
- Kelime işlemcileri
- Belge hazırlama, kopyalama, saklama, düzenleme, yazdırma, vs.
- Tablolama
- Grafik çizimi
- Elektronik posta
- Fax
- Sunumlar
- Telekonferans
- …

Karar Destek Sistemleri

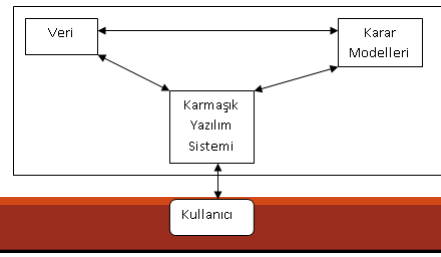
Planlı-plansız karar vermek için tüm aşamaları destekleyen sistem

Raporlardan çıkan planlı bilgi akışı ile çözüme özel

İçerebilecekleri:

- Özelleştirilmiş programlama dili
- İstatistiksel – optimizasyon – olasılık – finans analizleri
- Veritabanı
- Grafik gösterim
- Tahmin, Hedef arama, Rapor
- …

Uzman Sistemler?



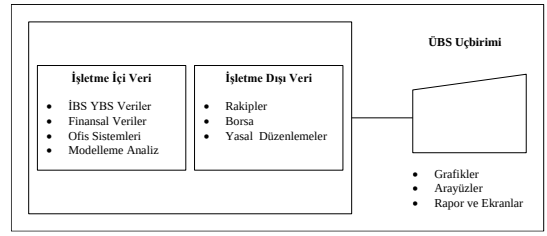
Üst Yönetim Bilgi Sistemleri

Stratejik seviyede:

- İşletme dışı veriler
- YBS + KDS ile işletme içi verileri alır
- Veri madenciliği: OLAP, Drill down analizleri

Hedefler:

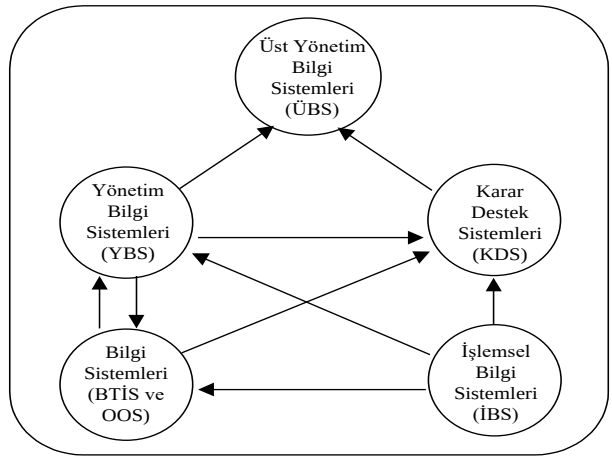
- İşletmenin performansı ve rakip aktivite takibi
- Sorunların görülmesi
- Fırsatların değerlendirilmesi
- İleriye dönük tahmin yapılması



Bilgi Sistemleri Arası İlişkiler

Sistem	Giriş	İşlem	Çıktı	Kullanıcı
ÜBS, EIS	Dahili ve harici veri	Grafikler; simülasyonlar; etkileşimli	Görsel; sorgulamalara cevaplar	Üst yönetim
KDS, DSS	Düşük yoğunlukta veri ya da analiz için optimize edilmiş veritabanları; analitik modeller ve veri analiz araçları	Etkileşimli; simülasyonlar; analiz	Özel raporlar; karar analizleri; sorgulamalara cevaplar	Uzmanlar; yöneticiler
YBS, MIS	Özet kayıtlar; büyük miktarda veri; basit modeller	Rutin raporlar; basit modelleme; düşük düzeyde analiz	Özet ve istisna raporları	Orta kademe yönetim
BTİS, KWS	Tasarım özellikleri; bilgi tabanı	Modelleme; simülasyon	Modeller; grafikler	Uzmanlar; teknik personel
OOS, OFFICE SYS	Dokümanlar; planlar	Doküman yönetimi; planlama; elektronik posta	Dokümanlar; planlar; elektronik posta	Sekreterler; muhasebeciler; alt kademe yöneticiler
İBS, TPS	İşlemler; olaylar	Sıralama; listeleme; birleştirme; güncelleme	Detaylı raporlar; listeler; özetler	İşletim personeli; şefler

Bilgi Sistemleri Arası Etkileşim



Bilgi Sistemi Tarafları

Kullanıcı

Yönetici

Sistem Analisti

Sistem Tasarımcıları

Yazılımcı

Destek Personeli

Kullanıcı

Sistemin varoluş sebebi, en önemli parça

MÜŞTERİ

Sistemden beklentileri TAM belirlenmeli

Sistemin nasıl başarıya ulaşacağı tespit edilmeli

Teknik değil, işlev ve kullanım önemli

Yönetici

Proje yöneticisi

- Projenin başarıya ulaşması
- Proje ekibinin sevk ve idaresi
- İşlev ve plan önemli

Üst düzey yönetici

- Sistemin geliştirilmesi için gerekli kaynak sağlayıcı
- Maliyet ve fayda önemli

Sistem Analisti

Dersimizin odak noktası

Projelerin geleceği için kilit üye

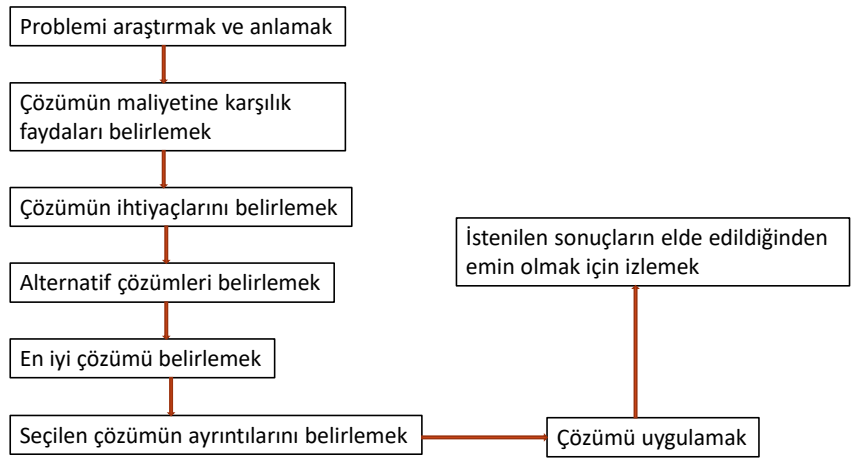
Hem işletme yönetimi, hem bilgi sistemi bilgisi olmalı

İşletmenin sorunları, ihtiyaçları, iş akışlarını belirlemeli

Bilgi sistemi ile çözümünü sağlatmalı, uygulamaya karar vermeli

Problemi görüp anlamalı, çözüm için bilgi sistemini metodolojiye uygun olarak oluşturmalı

Bir Analistin Problem Çözümü



Analistin Beceri ve Görevleri

Analitik düşünme bilgi ve becerisi

- Problemi belirleyip tanımlayabilmeli
- Sistemin parçalarını inceleyip ilişkileri ortaya koyabilmeli
- Çözüm alternatifleri sunup değerlendirebilmeli

Teknik bilgi ve beceri

- Donanım, programlama dilleri, işletim sistemleri, vtys, haberleşme protokolleri gibi konularda temelleri bilmeli
- Farklı teknolojilerin kullanım amaçlarını, nasıl kullanıldığını, nasıl entegre edilebileceklerini bilmeli
- Bilgi sistemi geliştirme sürecinin tamamı hakkında bilgi sahibi olmalı

Analistin Beceri ve Görevleri

Yönetim ve iş bilgi-becerisi

- İşin işleyişine ve organizasyon yapısına hakim olmalı
- Sektör, hedefler, stratejiler, planlar, kurum kültürüne bağlı değerleri bilmeli
- Kaynak yönetimi, proje yönetimi, risk yönetimi, değişim yönetimi bilgisi olmalı

İnsan ilişkileri

- Müşteri ihtiyaçlarını ve isteklerini, kullanıcıların işleri nasıl yaptığını, davranış biçimlerini belirleyebilmeli
- Tüm çalışanların becerileri, bilgi ve teknolojileri ile ilgili fikri olmalı
- Rahat iletişim kurmalı, dinlemeli, sorunları anlamalı
- İlerlemeleri belgelemeli ve paydaşlara aktarmalı

Tasarımcılar

Veritabanı yöneticileri

Ağ mimarları

Web mimarları

Grafik sanatçıları

Güvenlik uzmanları

Teknolojik uzmanlar

Sistem Tasarımcısı – Yazılım Mimarı

Gereksinimleri belirlenmiş sistemin bilişim modelini kurgular

Yazılım mimarilerine hakim olmalı

Ağ ve donanım kaynaklarını iyi bilmeli

Veri yönetimi konusunda deneyimli olmalı

Yazılımcı

Sistemi yazanlar

İş süreçleri ve problem gereksinimlerini bilgisayar diline çevirir

Analist (tasarımcı) girdilerini kullanır

Sistemin çalışan halini üretir

CASE araçları ile kod üretimi kolaylaştı

Kod optimizasyonu ve entegrasyon önemli

Sistem programcıları

Veritabanı programcıları

Test personeli

Destek Personeli

Sistemin devamlılığını ve sürekliliğini sağlar

Sorumlulukları:

- Ağ iletişimi
- Donanım
- Yazılım ve parametreleri
- Çıktıları düzenleme
- Ürün desteği
- Güvenlik
- Web arayüzleri
- Dış sistem entegrasyonları

Bilgi Sistemi Geliştirme Süreci

Sistemin geliştirilmesi için izlenen süreç ve uyulan metodoloji

ADIM	İŞLEM	ÇIKTILAR
Problemin Tanımı	Problemi ortaya koymak	İhtiyaçlar belirlenir
Fizibilite Çalışması	Projenin kapsamı ve hedefleri ortaya konarak olabilirliğini belirlemek	Fizibilite çalışması raporu
Analiz	Problemin çözümlerini ortaya koymak	Çözümün lojik modeli
Genel Tasarım	Sistemin nasıl gerçekleştirileceğini belirleme	Sistemin maliyeti ve üst düzey dizaynı
Ayrıntılı Tasarım	Genel tasarımda belirlenen sisteme ait alt sistemlerin tanımlanması	Sistemin özellikleri ve ayrıntılı tasarım
Gerçekleştirme	Kodlama, yükleme ve test	Çalışan sistem ve dokümantasyon
Bakım	Sistemin bakımını yaparak desteklemek	Çalışan Sistem

Sistem Geliştirme Yaşam Döngüsü

Süreç Modelleri

Klasik Süreç (Waterfall)

Model Oluşturma

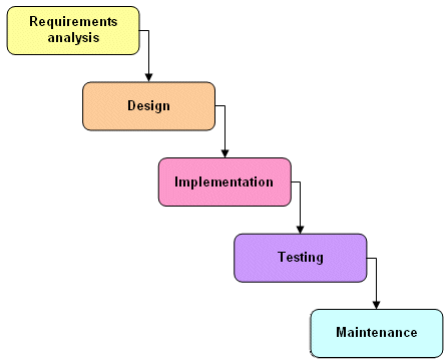
Evrimsel Süreçler
- Artımlı Model
- Spiral Model

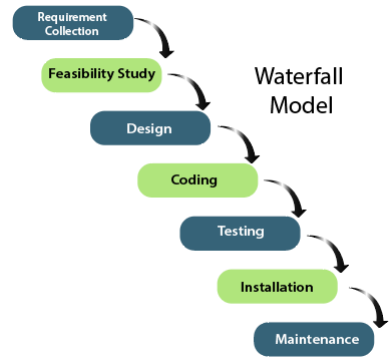
RUP Modeli

Aykırı Programlama

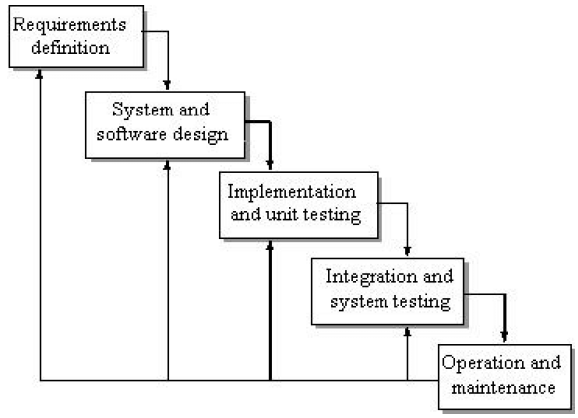
Çevik Süreçler

Klasik Süreç - Waterfall

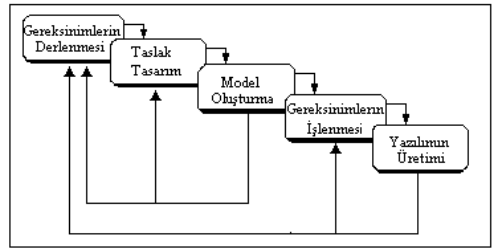




İteratif Waterfall (Artımlı)



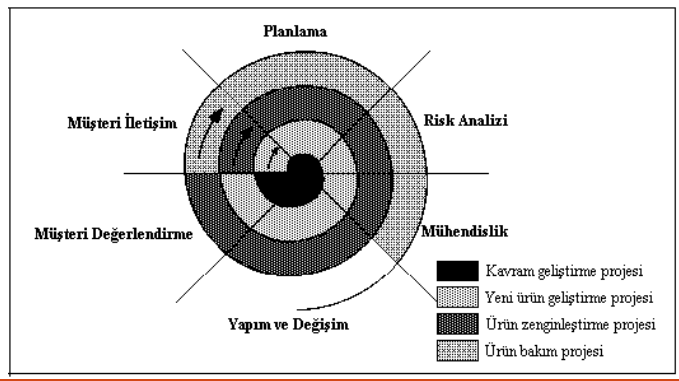
Model (Prototip) oluşturma



Evrimsel Süreçler

Artımlı: İteratif olarak waterfall uygulanır (gördük)

Spiral: Prototipten versiyona doğru evrim ilerler



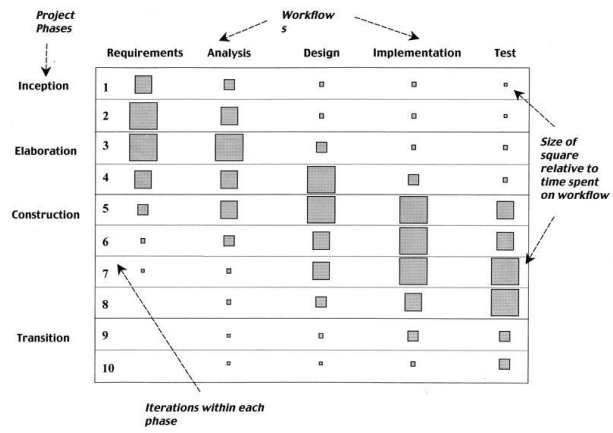
Rational Unified Process (RUP)

Kullanım-senaryosu güdümlü

Mimari yapı merkezli

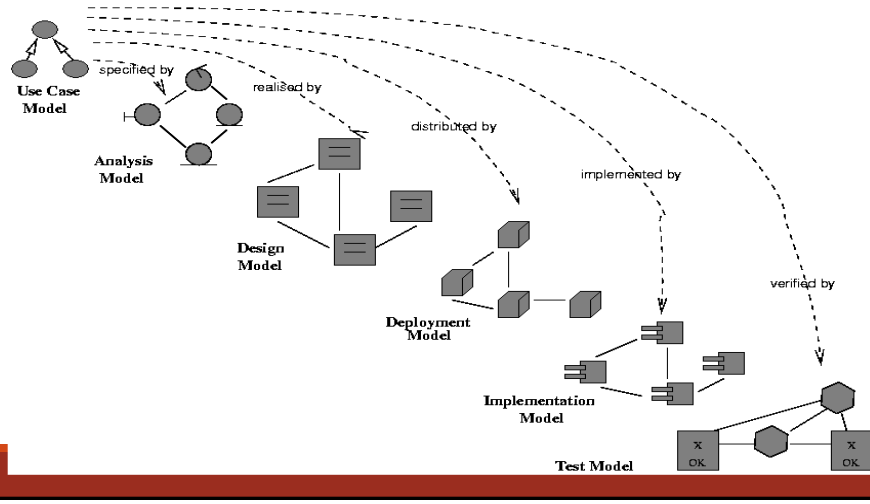
Artımlı ve iteratif

RUP



RUP

Yazılım ürününün fazlardaki karşılıkları



Aykırı Programlama

4 temel değer
- İletişim
- Basitlik
- Geri besleme
- Cesaret

İnceleyin

- 12 Pratik
  - Sistem metaforu
  - Ekip üyesi müşteri
  - Kısa aralıklı yayımlar
  - Planlama oyunu
  - Basit tasarım
  - Ortak kod mülkiyeti
  - Kodlama standartları
  - Eşli programlama
  - Test
  - Sürekli tümleştirme
  - Devamlı yeniden tasarım
  - Devam ettirilebilir hız: 40saat/hafta

Aykırı Programlama

Çevik manifestoya uygundur

Özellikleri:

- Müşteri de işin içinde
- Artımlı planlama, sürekli test ve entegrasyon, küçük sürümler yayımlama, basit tasarımlar
- Test güdümlü geliştirme
- Refactoring
- Eşli Programlama
- Birlikte sahiplenme ve sürdürülebilir gelişme hızı
- Çok fazla doküman gerektirmeme

Gelecek Ders

Ön İnceleme Safhası

Fizibilite Çalışması

Sistem Analizi ve Tasarımı

BLM2042 GR.1/GR.2

2025-2026 BAHAR YARIYILI

DR.ÖĞR.ÜYESİ YUNUS EMRE SELÇUK / GÖKSEL BİRİCİK

Bu Derste

Ön İnceleme

Fizibilite

Ön İnceleme

Fizibilitenin ilk aşaması

Projenin olabilirliği belirlenir

- Projeye(yeni sisteme) gerçekte ihtiyaç var mı?
- Sistemin gerçekleştirilmesi için neye ihtiyaç var?
- Ne kadar süreye ihtiyaç var?
- Tahmini bütçe nedir?
- Faydaları ve zorlukları nelerdir?

Kısaca: Tamam mı? Devam mı?

Kapsam Tanımlama

Ürün: Ne isteniyor?

Kalite: Ne kadar iyi olmalı?

Zaman: Ne zaman isteniyor?

Maliyet: Proje bütçesi ne kadar?

Kaynaklar: Hangi kaynaklar kullanılacaktır?

Fizibilite Çalışması

Amaç: Projenin olabilirliğinin araştırılması

Çıktı: Genel hatlarıyla proje planı ve tahmini bütçe

Fizibilite Tipleri

Teknik Fizibilite

Zaman Fizibilitesi

Sosyal Fizibilite

Yönetim Fizibilitesi

Yasal Fizibilite

Ekonomik Fizibilite

Teknik Fizibilite

Sistemin her türlü teknik olanaklarının belirlenmesi. Örneğin;

- Yazılım Fizibilitesi
  - İşletim Sistemi, Yazılım Geliştirme Ortamı, VTYS, Uygulama Sunucu, Destek Yazılımlar, …
- Donanım Fizibilitesi
  - Geliştirme ihtiyaçları, kısa-orta vadeli büyüyecek ihtiyaçlar, …
- İletişim Fizibilitesi
  - Haberleşme ihtiyaçları, protokoller, …

Riskler göz önünde tutulmalı

Teknik yetersizlikler tamamlanmalı

Sosyal Fizibilite

Oluşturulacak sistemin hedef kitlesi tarafından kabul edilip edilemeyeceğinin araştırılması

Sistemden direk olarak etkilenecek kullanıcı grupları incelenip kontrol edilir

İstekleri karşılaması önemli

Operasyonel Fizibilite olarak da görebilirsiniz.

Yönetim Fizibilitesi

Geliştirilecek olan sistem yönetimi nasıl etkiliyor?

Yönetim sistemden ne kadar fayda sağlayacak?

Yasal Fizibilite

Sistem kanun ve yönetmeliklere uygun mu?

Mevcut patent ve fikri sınai hakları ihlal ediyor mu?

Belli kaynakların kullanımı için özel izinlere – lisanslara gerek var mı?

Sistem işletmenin yaptığı anlaşmalara uygun mu?

Ekonomik Fizibilite

Geliştirilecek sisteme ait tüm maliyetler göz önüne alınıp maliyet-fayda analizi yapılır

- Teknik (donanım-yazılım), zaman (işgücü, personel), yasal fizibiliteden (patent hakkı-lisanslar) maliyetler doğar

Maliyet fayda analizi yöntemleriyle bugün yapılan birim yatırımın gelecekte kaç birim olacağı bulunur

- Bugünkü değer (net present value)
- İç verim oranı (internal rate return)
- Başabaş noktası (break-even)
- Geri ödeme süresi (payback period)

Bugünkü Değer Yöntemi

Projenin bütün nakit giriş ve çıkışları bugünkü değere indirilip karşılaştırılır

Net nakit girişleri > Yatırım harcamaları ?

- Evet: Proje kabul
- Hayır: Proje red

Bugünkü değere indirgemek için iskonto oranı kullanılır

- Farklı yöntemleri mevcuttur
- En pratiği: Piyasadaki ağırlıklı sektörlerin uyguladığı faiz oranlarını ortalaması

Q yıl, i iskonto oranı, V indirgenmiş nakit akışı toplamı ise,

$$V = Q_0 \frac{Q_n}{(1+i)^n}$$

Bugünkü Değer Yöntemi

1 TL'nin Bugünkü Değeri :PVIF =1/(1+i)^n

Yıllar(%)	1	2	3	4	5	6	7	8	9
1	0,990	0,980	0,971	0,962	0,952	0,943	0,935	0,926	0,870
2	0,980	0,961	0,943	0,925	0,907	0,890	0,873	0,857	0,842
3	0,971	0,942	0,915	0,889	0,864	0,840	0,816	0,794	0,772
4	0,961	0,924	0,889	0,855	0,823	0,792	0,763	0,835	0,708
5	0,951	0,906	0,863	0,822	0,784	0,747	0,713	0,681	0,650

İç Verim (Karlılık) Oranı Yöntemi

Paranın zaman değeri ve yatırımın ekonomik ömrü dikkate alınır

İVO: Nakit girişleri ile yatırım maliyetini eşitleyen iskonto oranı

T: Yatırımın ömür yılı, x: toplam maliyet, Rt t yılındaki net getiri, d iskonto oranı için

$$x = \sum_{t=1}^{T} \frac{R_t}{(1+d)^t}$$

Net Bugünkü Değeri (önceki yöntem) 0 yapan iskonto oranıdır.

Başabaş Noktası Yöntemi

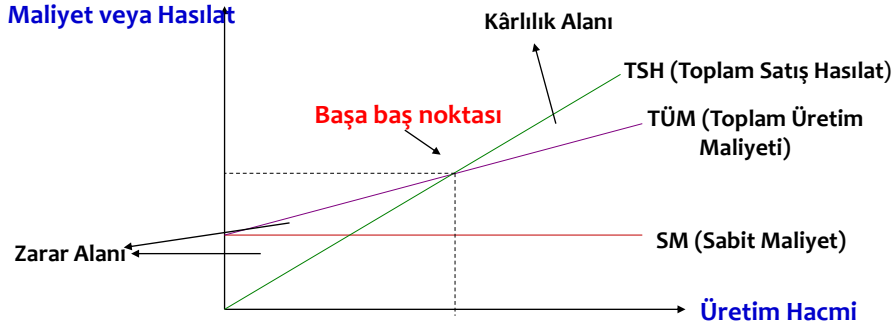
Proje işletme aşamasına geçtikten sonra;

- Toplam satışlar ile toplam giderlerin eşit olduğu satış tutarı, miktarı ya da kapasite kullanım oranını bulmak için kullanılır

Harcamalar, maliyet ve faydalar aynı grafikte gösterilir. İki eğrinin çakıştığı nokta başabaş noktasıdır

Bu noktadan sonra gelir maliyetten yüksek olur

Başabaş Noktası Grafiği



Geri Ödeme Süresi Yöntemi

Gerekli yatırım ve yıllık yarar arası ilişki

Yıllık yararın yatırım miktarına bölünmesi ile geri ödeme süresi bulunur

Başlangıçtaki nakit yatırımların kaç yılda elde edilebileceğini belirler

Geri ödeme süresi = Yatırım tutarı / Yıllık net nakit girişi

Zaman Fizibilitesi

Belirlenen zaman içinde projenin nasıl tamamlanacağı belirlenir.

Gantt ve PERT teknikleri kullanılır.

Gantt

Gantt çizelgesi; 1910'larda proje planlama ve kontrolü için Henry Gantt tarafından önerilmiş bir diyagramdır.

Gantt çizelgesinde, projenin işlem adımları faaliyet olarak isimlendirilir. Her bir faaliyetin süresi, başlangıç-bitiş zamanları, hangi kaynakların bu faaliyet için kullanılacağı, diyagramdan çıkarılabilecek verilerdir.

Gantt

Problem tanımından işletime kadar süredeki tüm aktivitelerin zamanları ve ilişkileri gösterilir.

Durak noktaları belirlenebilir.

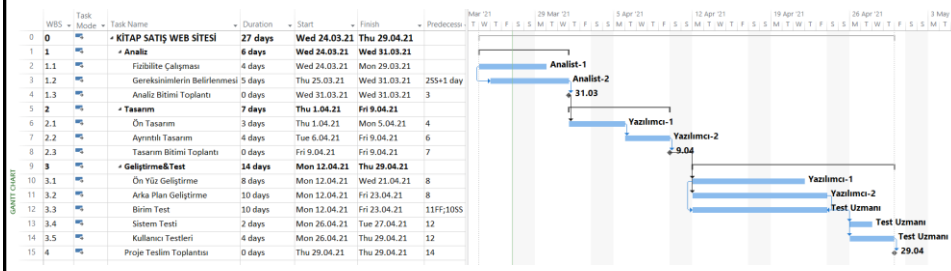
Microsoft Project gibi araçlarla; proje verisi işlem ağı olarak verildiğinde;

- Yapılacak işleri tarih sırası ile gösteren, durak yerlerini ve iş yoğunluğunun az olduğu işlemleri belirleyen bir Gantt çizelgesi,
- Yöneticinin proje sürecini izleyebilmesi için, her işlemin süresini en erken ve en geç başlama ve bitme tarihlerini gösteren bir görev durumu çizelgesi,
- İşlem sırasına göre görev paylaşımını gösteren bir çizelge,
- Üzerinde kritik yolu koyu çizgilerle belirleyen bir işlem ağı

Otomatik olarak elde edilir.

Örnek bir Gantt Şeması

Kitap satışı yapılacak bir web sitesinin tasarlanmasından kullanıma sunulmasına kadar olan süreç Gantt şeması ile şöyle ifade edilebilir:



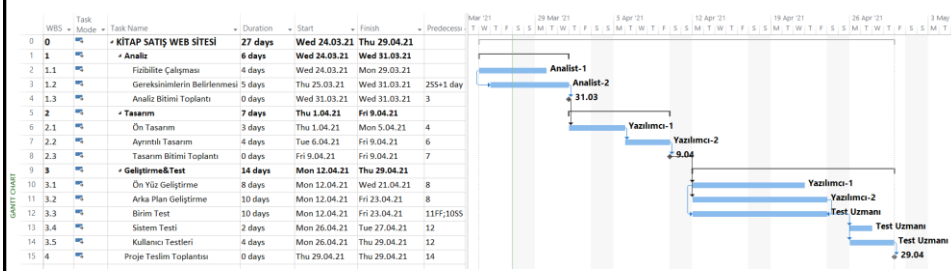
	WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors
0	**0**		**KİTAP SATIŞ WEB SİTESİ**	**27 days**	**Wed 24.03.21**	**Thu 29.04.21**	
1	**1**		**Analiz**	**6 days**	**Wed 24.03.21**	**Wed 31.03.21**	
2	1.1		Fizibilite Çalışması	4 days	Wed 24.03.21	Mon 29.03.21	
3	1.2		Gereksinimlerin Belirlenmesi	5 days	Thu 25.03.21	Wed 31.03.21	2SS+1 day
4	1.3		Analiz Bitimi Toplantı	0 days	Wed 31.03.21	Wed 31.03.21	3
5	**2**		**Tasarım**	**7 days**	**Thu 1.04.21**	**Fri 9.04.21**	
6	2.1		Ön Tasarım	3 days	Thu 1.04.21	Mon 5.04.21	4
7	2.2		Ayrıntılı Tasarım	4 days	Tue 6.04.21	Fri 9.04.21	6
8	2.3		Tasarım Bitimi Toplantı	0 days	Fri 9.04.21	Fri 9.04.21	7
9	**3**		**Geliştirme&Test**	**14 days**	**Mon 12.04.21**	**Thu 29.04.21**	
10	3.1		Ön Yüz Geliştirme	8 days	Mon 12.04.21	Wed 21.04.21	8
11	3.2		Arka Plan Geliştirme	10 days	Mon 12.04.21	Fri 23.04.21	8
12	3.3		Birim Test	10 days	Mon 12.04.21	Fri 23.04.21	11FF;10SS
13	3.4		Sistem Testi	2 days	Mon 26.04.21	Tue 27.04.21	12
14	3.5		Kullanıcı Testleri	4 days	Mon 26.04.21	Thu 29.04.21	12
15	4		Proje Teslim Toplantısı	0 days	Thu 29.04.21	Thu 29.04.21	14

Örnek bir Gantt Şeması

WBS (work breakdown structure) ifadesi ile her bir faaliyet alt faaliyet kırılımlarına sahip olup numaralandırılır.

Faaliyetler için başlangıç tarihi ve süresi belirtildiğinde otomatik olarak bitiş tarihi hesaplanır.

Bir işin başlangıç zamanı başka bir işe bağlı olabilir (öncül tanımlama)

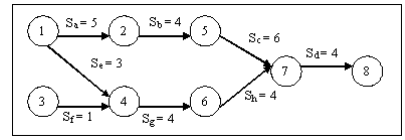


	WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors
0	**0**		**KİTAP SATIŞ WEB SİTESİ**	**27 days**	**Wed 24.03.21**	**Thu 29.04.21**	
1	**1**		**Analiz**	**6 days**	**Wed 24.03.21**	**Wed 31.03.21**	
2	1.1		Fizibilite Çalışması	4 days	Wed 24.03.21	Mon 29.03.21	
3	1.2		Gereksinimlerin Belirlenmesi	5 days	Thu 25.03.21	Wed 31.03.21	2SS+1 day
4	1.3		Analiz Bitimi Toplantı	0 days	Wed 31.03.21	Wed 31.03.21	3
5	**2**		**Tasarım**	**7 days**	**Thu 1.04.21**	**Fri 9.04.21**	
6	2.1		Ön Tasarım	3 days	Thu 1.04.21	Mon 5.04.21	4
7	2.2		Ayrıntılı Tasarım	4 days	Tue 6.04.21	Fri 9.04.21	6
8	2.3		Tasarım Bitimi Toplantı	0 days	Fri 9.04.21	Fri 9.04.21	7
9	**3**		**Geliştirme&Test**	**14 days**	**Mon 12.04.21**	**Thu 29.04.21**	
10	3.1		Ön Yüz Geliştirme	8 days	Mon 12.04.21	Wed 21.04.21	8
11	3.2		Arka Plan Geliştirme	10 days	Mon 12.04.21	Fri 23.04.21	8
12	3.3		Birim Test	10 days	Mon 12.04.21	Fri 23.04.21	11FF;10SS
13	3.4		Sistem Testi	2 days	Mon 26.04.21	Tue 27.04.21	12
14	3.5		Kullanıcı Testleri	4 days	Mon 26.04.21	Thu 29.04.21	12
15	4		Proje Teslim Toplantısı	0 days	Thu 29.04.21	Thu 29.04.21	14

PERT

Süre tahminleri: $S_b=(S_i+4*S_e+S_k)/6$

- İyimser Süre: İşlemlerde gecikme ve aksama olmadan tamamlanmayı sağlayan en kısa süre
- Kötümser Süre: Gecikme ve aksamaları dikkate alan en uzun süre
- En Yaklaşık Süre: Normal gecikme ve aksamaları değerlendirir



Farklı Çözüm Önerilerinin Değerlendirilmesi

Her çözüm için fizibilite çalışması gerçekleştirilir

Aday sistemler matrisi ile alternatifler karşılaştırılır

	Önerilen Sistem 1	Önerilen Sistem 2	Önerilen Sistem 3
Teknoloji **Sistemin teknik alt yapısı**			
Ara Birim **Sistemin kullanıcılarla ve diğer sistemlerle olan iletişimi**			
Veri **Sisteme hangi verilerin nasıl girileceği ve buna göre elde edilen çıktılar**			
İşlemler **Sistemdeki veriler üzerinde hangi işlemlerin gerçekleştirileceği**			
Coğrafik Yapı **İşlemlerin ve verilerin sistem içinde nasıl dağıtıldığı**			

Fizibilite Matrisi

Yapılan çalışmalar değerlendirilir

Yüzde ağırlıklarla puanlandırma yapılır

	Önerilen Sistem 1	Önerilen Sistem 2	Önerilen Sistem 3
Teknik Fizibilite %V			
Ekonomik Fizibilite %W			
Zaman Fizibilitesi %X			
Sosyal Fizibilite %Y			
Yönetim Fizibilitesi %Z			
Sıralama			

	Ağırlık	Önerilen Sistem 1	Önerilen Sistem 2	Önerilen Sistem 3
Teknik Fizibilite	%30	Kullanılan ürünün son sürümü sadece 6 haftadır piyasada. Dolayısıyla ürünün gelişimi risk taşımaktadır. Sistemle ilgili gerekli entegrasyonları yapmak için C++ bilen elemana ya da mevcut elemanların bu konuda eğitime ihtiyaç var. Puan : 50	Sayısı yeterli olmayan mevcut teknik personelin sadece Powerbuilder tecrübesi olmasına rağmen proje yöneticisi, sistemi MS Visual Basic'e çevirmenin basit olduğuna ve VB bilen eleman bulmanın Powerbuilder'e gore daha kolay ve ucuz olduğuna inanmaktadır. Puan : 95	Şirket bünyesinde MS SQL Server kullanılmaktadır. Ancak istemci/sunucu veri tabanı yönetimi sektöründeki rekabetten ötürü Powerbuilder'in yeni versiyonlarının SQL Server ile ne kadar uyumlu olacağı belli olmaması risk taşımaktadır. Puan : 60
Ekonomik Fizibilite	%30	Yaklaşık 350.000 $'a mal olan ve geri dönüşümü 4,5 yıl olan sistemin net getirisi 210.000 $'dır Puan : 60	Yaklaşık 418.040 $'a mal olan ve geri dönüşümü 3,5 yıl olan sistemin net getirisi 306.748 $'dır Puan : 85	Yaklaşık 400.000 $'a mal olan ve geri dönüşümü 3,3 yıl olan sistemin net getirisi 325.500 $'dır Puan : 90
Zaman Fizibilitesi	%10	3 aydan az Puan : 95	9 – 12 ay Puan : 80	9 ay Puan : 85
Operasyonel Fizbilite	%30	Sadece servis üyelerinin ihtiyaçlarını karşılamakta ve var olan iş süreçlerini yazılıma gore güncellenmesi gerekmektedir. Puan : 60	Kullanıcının tüm ihtiyaçlarını karşılamaktadır. Puan : 100	2. sistem ile aynı Puan : 100
Sıralama	%100	60,5	92	83,5

Gelecek Ders

Gereksinim Analizi

İş Analizi

Kullanım Senaryoları

Bilgi Toplama Yöntemleri

Bu yansı ders notlarının düzeni için boş bırakılmıştır.

Sistem Analizi ve Tasarımı

BLM2042 GR.1/2

2025-2026 BAHAR YARIYILI

DR.ÖĞR.ÜYESİ YUNUS EMRE SELÇUK / GÖKSEL BİRİCİK

Bu Derste

Gereksinim Analizi

İş Analizi

Kullanım Senaryoları

Bilgi Toplama Yöntemleri

Gereksinim Analizi

Tanım: Problem nedir? Çözümleri nelerdir? Sistem nasıl çalışır? Sorularını cevaplamak için yapılan çalışmalardan oluşan faz.

Amaç: En uygun çözümü bulmak için ana öğeler ve işlevler ayrıntılarıyla tanımlanır

- Gelecekteki hedefler detaylandırılır
- Bilgi kaynakları ve ihtiyaçlar belirlenir

Çıktılar:

- Kullanıcının tüm ihtiyaçları
- Var olan sistemin durumu
- Seçilen en uygun çözümün değerlendirilmesi
- Var olan sistemin nasıl iyileştirilebileceği önerileri

Gereksinim Analizi

Öncelikle:

- Fizibilite raporları gözden geçirilmeli
- Tüm teknik terimler incelenmeli
- Ayrıntılı inceleme planı yapılmalı
- Proje grubunun görev dağılımı yapılmalı (Gantt)
- Denetleme mekanizması kurulmalı

Varolan Sistemin Analizi

Varolan sistemin nasıl çalıştığı ve maliyetinin anlaşılması çok önemli

Sistem analisti varolan sisteme tamamen hakim olmalı

- Varolan bilgi kaynakları
- Kullanılan donanım-yazılım
- İşlemlerde kullanılan bilgi ve miktarı
- İç-dış bilgi etkileşimi
- Sistem döngüsü ve süresi
- Arşiv bilgisi ve araçları
- Raporlar ve formatları
- Personel
- Maliyetler

Sistem Analizi Yöntemleri

Gereksinimler önceden müşteri ile saptanırsa **Akış Diyagramları Gösterimi**

- Bilişim alanı, işlevler, ara birimler, kabul kriterleri belirlenip diyagramlarla gösterilir

Gereksinimler önceden saptanmazsa **Prototip** oluşturulur

- Prototip üzerinden müşteri ile tartışarak gereksinimler kararlaştırılır

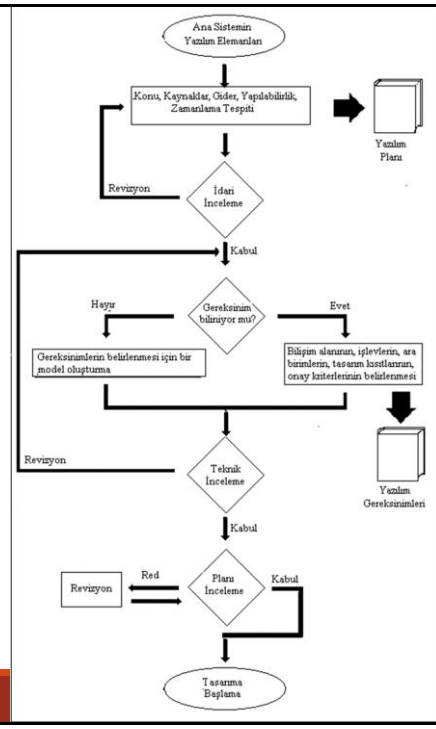
Sistem Gereksinimleri Spesifikasyonu

Analist ve müşterinin ortak çalışması ile sistem gereksinimlerinin özellikleri ortaya konulur:

Belge haline gelir →**Sistem gereksinimleri spesifikasyonu**, sistem prototipi

Sistem geliştirme planı da gözden geçirilir

Sistem geliştirme planı ile sistem analizi birlikte iç içe gerçekleştirilir: Tanımlama aşaması



İdari İncelemede Önemli Noktalar

İhtiyaçlar, işletmenin genel stratejileriyle uyumlu mu?

Kararlarda bilgi/deneyim ve öngörü birlikte kullanıldı mı?

Çekişen gereksinimlerden hangisi daha çok fayda sağlar?

Yeni çözümün somut mali faydası nedir?

Organizasyonda değişiklik(-/+) olasılığı nedir?

Hızlı/ayrıntılı raporlama yapılıyor mu?

Bilgi akışı/güncelleme planlı mı?

Yeni çözüm mevcuttan somut biçimde iyi/hızlı mı?

Teknolojiler bilginin hacmiyle uyumlu mu?

Kullanıcı etkileşimi / veri organizasyonu düzgün mü?

Kullanım Senaryoları

Kullanım Senaryosu Modellemesi

Kullanım Senaryosu Diyagramı

Kullanıcının bakış açısından sistemin fonksiyonel çözümlemesini tanımlar

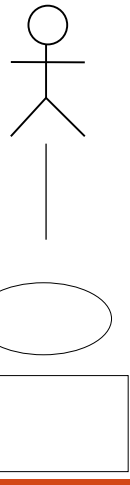
Sistemin ihtiyaçlarını yerine getirmek için gereken davranışlar için kullanılır

Aktörler: Kullanıcılar, dış sistemler, fiziksel çevre etmenleri, …

Senaryolar: Bir olay akışı şeklinde sistemce sağlanan fonksiyonellik

- Ön koşullar ve son koşulları, senaryo, birincil ve ilgili aktörleri olmalıdır.

Kullanım Senaryosu Modellemesi Sembolleri



Aktör

İlişkiler

Kullanım Senaryosu

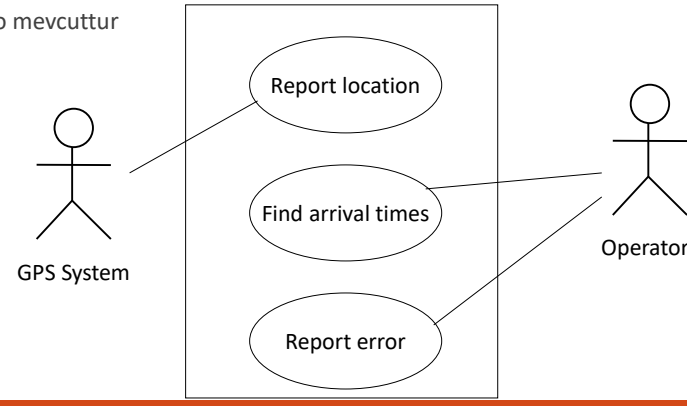
Sistem

Örnek Kullanım Senaryosu Diyagramı

Aktörler insan olmak zorunda değildir

Aktörler ve sistem etkileşimini gösterir

Burada üç senaryo mevcuttur



Kullanım Senaryoları

Use-case model: isteklerin anlaşılıp ifade edilmesini sağlayan bir yöntem

Aktör: sistemin kullanıcıları (insan, başka sistem veya cihaz). Aktörler hizmet ister ya da verir.

Birincil aktör: asıl faydayı sağlayan, işlemleri başlatan kullanıcı

Destek aktörü: bilgi-destek sağlayan aktör(ler)

Senaryo: Anlamlı bir sonuca ulaşmak için aktörle sistem arasında gerçekleşen olaylar zinciri

Senaryoların İfade Edilmesi

İhtiyaç-özellik listesi değil

Sistem **kara kutu** olarak alınır, sorumlulukları ifade edilir

Aktörler-sistem etkileşimi, **etken** cümlelerle söylenir

Ne yapar? Cevaplanır. Nasıl yapar? Tasarımın sorusudur

Sistemin **bitmiş hali** hayal edilir. Oluşabilecek senaryolar yazılır.

Sistemin **sınırları** doğru belirlenmelidir. Neler dışarıda, neler içeride olmalı?

Kullanım Senaryolarında Yer Alan Bilgiler

Önsöz

- **Birincil Aktör**
- **İlgililer ve Beklentileri**: Sistemin çalışmasından etkilenen ve beklentileri olan diğer aktörler. İlk başta doğru tanımlanmazsa sistemin sınırları çizilemez. Bazı durumlar unutulabilir.
- ***Ön koşullar****: senaryonun başlaması için gerekli koşullar*
- ***Son koşullar****: senaryo tamamlanınca sistemin olacağı durum (ilgililerin beklentileri)*

Ana Başarılı Senaryo: Sistemin en doğal çalışma şekli, adım adım yazılır. Koşul-dallanma olmaz. Etken cümlelerle, kimin ne yaptığı açık şekilde belirtilir. Üç tipi varır: kullanıcı-sistem etkileşimi-tetikleme, onaylama, durum değişikliği-bilgi kaydı.

Alternatif Akış: Ana senaryo dışındaki başarılı-başarısız sonuçlara götüren kısım. Şart belli olmalı. Bunlarla tüm amaçlar sağlanmış olur.

Sıra Dışı Durumlar*: Hata olunca yapılacaklar.*

Özel istekler*: istenen hız, güvenlik vs gibi kalite kısıtları.*

Teknolojik beklentiler*: dil-platform vs.*

Senaryolar Arası İlişkiler

İçerme (include): Birçok senaryo grubunda kullanılan başka bir senaryo grubu. Alt programa dallanıp geri dönmek gibi.

Genişletme (extend): Belirli koşulda ana senaryodan ayrılma noktasından sonra çalışan senaryo

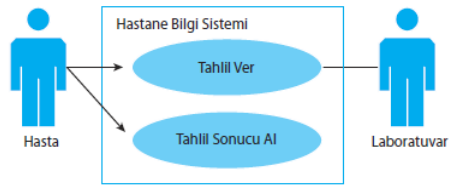
Özelleştirme (inheritance): Daha genel senaryodan özel senaryolar türetilir (ödeme→ nakitle, kartla, çekle, …)

Bağıntı İlişkisi

Senaryo ve aktörler belirlendikten sonra ilişkilendirilirler.

Bağıntı İlişkisi (Association relation): bir aktörle senaryo arasındaki ilişki.

- Bir aktör en az bir senaryoyla ilişkilidir.
- Bir senaryo birden fazla aktörle bağıntılı olabilir.
- İlişki yönlü (ok) ise birincil aktörler, yönsüz ise harici aktörler anlaşılır.

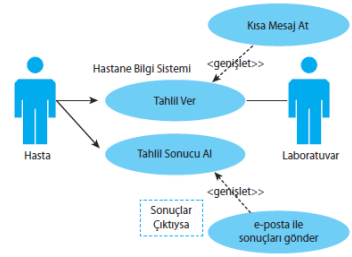


Genişletme İlişkisi

(Extend Relation) bir senaryonun başka senaryolar ile ilişkilendirilerek yeni işlevler kazandırılması.

Birkaç adımdan oluşan karmaşık işlevler için alt senaryolar **eklenti senaryolar** (extension use case) ile gösterilir.

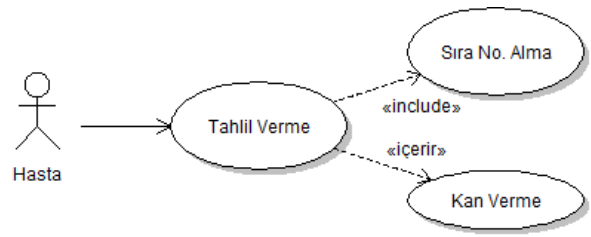
Genişletilmiş senaryo
tek başına anlamlı olmalıdır.
Eklenti olmadan da davranış
sergileyebilmelidir.



Ekleme İlişkisi

(Include Relation): Bir senaryo içindeki adımların diğer senaryolarda yer alması durumu.

Ortak işlevler tüm bileşenlerde aynı tanımlanır.

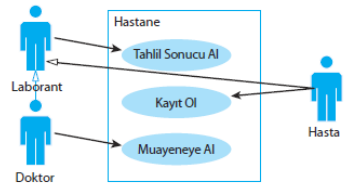


Kalıtım İlişkisi (Genelleştirme)

(Inheritance Relation [Generalization]): Bir aktörün kendisiyle aynı işlevleri meydana getiren aktörlerle olan ilişkisi.

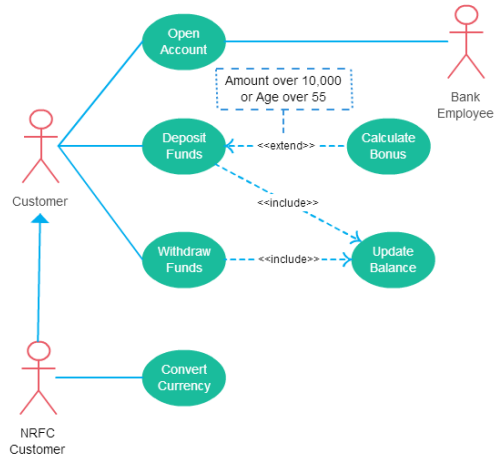
Aynı işlevleri kullanan aktörler için bir **soyut aktör** tanımlanır.

- (Abstract Actor): Kullanım senaryolarında birden fazla aktöre ait olabilecek ortak özellikleri taşıyan temel aktör



Sistem sınırı, sistemin adının yazıldığı bir dörtgendir. Önceki bileşenlerin tümünü kapsar.

Örnek Kullanım Senaryosu Diyagramı



Sözleşmeler (Contracts)

Bazı durumlarda karmaşık işlemlerin daha iyi anlaşılabilmesi için.

Ön koşullar sağlandığında sistemin alacağı son durumun daha iyi açıklanmasını sağlar.

Aktör-sistem etkileşimi yanında sistem içi nesnelerdeki değişim de belirtilir.

İşlemler belirlenir. Karmaşık olanları için sözleşme yazılır.

Önemli olan son koşullardır. (bir nesne yaratma yok etme, bir özellik güncellenmesi, bir bağlantı oluşturma-koparma)

Modele bazı eklentiler (nitelik gibi) yapılmasına neden olabilirler.

İşin nasıl yapılacağını göstermezler. O, tasarımın işidir.

Örnek Sözleşme

Sözleşme SO1: Çalışan Çıkarma

İşlem: calisanCikart(c:Calisan)

Referans: Kullanım Senaryosu KS3: Bölüm Kapama

Ön Koşullar: KS3 (Bölüm Kapama) kullanım senaryosunun çalışması gereklidir. Çalışan uygun bir bölüme atanamamış olmalıdır.

Son Koşullar: Sistem tüm bölüm çalışanlarının tazminatını hesaplamıştır.

Çalışanın işyeri ile bağlantısı kesilmiştir.

Bilgi Toplama

Araştırma ve Bilgi Toplama Yöntemleri

Araştırma sürecinde var olan sistemin eksik ya da yanlış anlaşılmaması için bilgi toplama önemlidir

Gözleme Yaklaşımı

- Mevcut belge-form-dosya örneklemesi
- İş ortamı gözlenmesi

Kişisel Görüşme Yöntemi (Toplantı, Mülakat, Ortak Gereksinim Planlaması)

- Yapılandırılmamış görüşme
- Yapılandırılmış görüşme

Anket Yöntemi

- Açık uçlu sorular
- Kapalı uçlu sorular

Kişisel Görüşme Yöntemi

Analistin kullanıcı ile işyerinde karşılıklı görüşmesi ve soru sorması

Randevu alınmalı ve hazırlıklı olunmalı

Başlangıç konuşması ve konular belirlenmiş olmalı

Görüşme süresince notlar alınmalı

Hitap biçimi ve duruşa dikkat edilmeli

Sonunda görüşmenin başlıca noktaları, sonra konuşulacak ve hiç değinilmemiş konular belirlenmeli, gelecek görüşmeler planlanmalı

Yapılandırılmış Görüşme

Amaçlar saptanmış

Kimle görüşüleceği belli

Zamanı ve yeri belli

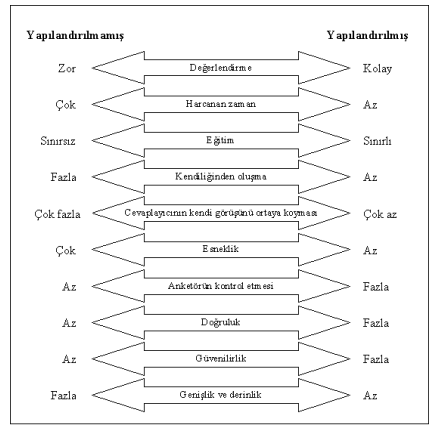
Taraflara amaç, süre ve gerekli dokümanlar bildirilmiş

Sorular belli

Görüşmeden sonra en kısa zamanda sonuçlar değerlendirilip yorumlanmış

Sonuçlar formal olarak yazılıp tüm ilgililere dağıtılmış

Yapılandırılmamış Görüşme Farkı



Soru Sıralaması

Tümevarım: piramit yapı. Ayrıntılı-kapalı sorulardan açık uçlu sorulara. Mülakat veren konuya ısınır.

Tümdengelim: Açık uçludan kapalı uçluya sorular dizilir. Mülakat kolay başlar.

Mülakat veren konu hakkında istekliyse, fikirlerini serbestçe ifade ediyorsa.

Anket Yöntemi

Sorular ve cevap alanlarından oluşan form

Açıklık: Sorular tam ve açık olmalı, eksik olmamalı

Hatırlatma: Değerlendirmede bilgi istenen olaydan geçen zaman, cavaplayan için önemi dikkate alınmalı

Cevap arzusu oluşturma: Özel yaşamla ilgisiz, cevaplaması kolay, ilgi çekici sorular olmalı

Hataya engel olma: Hatasız cevap için eleme sorusu ya da çoktan seçmeli cevaplama tercih edilmeli

İfade kolaylığı: Görsel öğelerle istenen cevap desteklenebilir

Cevaplayıcıyı şartlandırma: Şartlandırma olmamalı, seçim özgürlüğü bırakılmalı

Anket Yöntemi

Çok sayıda kişiden veri toplanabilir.

Avantajlar:

- Hızlı cevaplanabilir.
- Çok sayıda kişiden maliyetsiz veri toplanır.
- Kimlikler gizli kaldığı için katılanlar düşüncelerini rahat aktarır
- Analiz hızlı yapılabilir

Dezavantajlar:

- Az kişi cevaplar
- Herkes tüm soruları cevaplamayabilir
- Gönüllü bilgi edinme kısıtlıdır, esneklik azdır
- Vücut dili analiz edilemez
- Bulanık soru varsa cevap açıklığa kavuşmaz

Anket Soruları: Açık Uçlu Sorular

Cevaplayıcıdan kendi ifadelerinden oluşan yanıt ister

Avantajlar

- Önceden hazırlanmış cevaplarla alınamayacak bilgiler alınabilir
- Cevaplayıcı bakış açısı ile yanıtladığından gerçek görüşleri daha iyi anlaşılır, endişenin içeriği ve öncelik sırası anlaşılır
- Cevaplayıcının duygularını ifade etmesini ve aktarmasını sağlar

Dezavantajlar

- Tamamlanması zahmetlidir, zaman alır, yorucudur
- Cevaplayıcı çabadan kaçarak eksik yanıt verebilir
- Verilerin işlenmesi ve analizi çok çaba ve zaman gerektirir
- Cevaplayanın eğitim seviyesi cevabın kapsamını etkiler
- Sayısal bilgi almak için en iyi-en kötü üç madde istenebilir, bunlar sıralanarak değerlendirilebilir

Anket Soruları: Kapalı Uçlu Sorular

Cevaplayıcıdan sabit cevap seçenek kümesinden seçim ister. Çoktan seçmeli, iki cevaplı, sıralamalı olabilir.

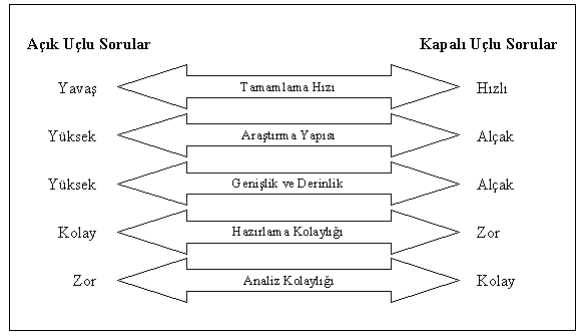
Avantajlar

- Cevap çok kolay ve hızlı bulunur
- Kodlanması ve işlenmesi sorunsuzdur
- Cevap olasılıklarını anketin amacıyla ilgili olacak şekilde sınırlar
- Herkese aynı bakış açısını iletir, cevaplayıcının yorumlama etkisini devre dışı bırakır
- Hatırlamayı kolaylaştırır: Hangi eğitimleri aldınız sorusu şıklarıyla olunca insan kolay hatırlar

Dezavantajlar

- Cevaplayanlar konuyla ilgili gerçek duygularını yansıtmayan seçeneklere maruz kalabilir. Gerçek cevabına benzeyen şıkkı seçerek ince ama önemli farklılıkları yok eder.
- Kişinin cevabı yoksa da, soruyu anlamasa da cevaba zorlar
- Soruların çoğu kapalı uçlu olursa görüşlerin ifadesine izin verilmediğinden cevaplayıcılar kendilerini engellenmiş hisseder

Soru Tiplerinin Karşılaştırılması



Anket Formu Düzeni

Sorular genelden özele ya da özelden genele sıralanabilir

Cevaplama direnciyle karşılaşmamak için genelden özele tercih edilir

Sorular basitten zora dizilmeli

Önceki soru(lar) sonraki soruya hazırlayıcı olmalı

Cevaplayıcının ilgisi ve işbirliği bu şekilde sağlanmalı

Sorular arasında bağlantı kurularak ani konu değişikleri yapılmamalı

İş Gereksinimleri

Toplanan bilgiler İş Gereksinimleri Dokümanı ile rapor haline getirilir.

- Özet
- Amaç, hedef
- İhtiyaç (Sistem neden lazım)
- Kapsam (Sınırlar)
- Mali analiz
- İşlevsel Gereksinimler (Üst seviyeli)
- Kaynak ihtiyacı
- İş Zaman Planı
- Maliyet-Fayda Analizi

Alternatif: Araştırma Raporu

- Teknik personel için bilgi toplama sonuçlarının teknik raporu
  - Başlık: yürütücü
  - İçindekiler
  - Araştırma amacı
  - Araştırma metodolojisi (yöntem, işlem, analiz, yorum)
  - Sonuç ve öneriler
  - Ekler (veri toplama formları, tablolar, grafikler, bibliyografya)
- Yönetici için özet rapor (Teknik rapordan sonra)
  - Araştırmanın amacı
  - Kullanılan yöntem
  - Elde edilen sonuçlar

Gelecek Ders

Sistem Analizi

Fonksiyonel Çözümleme

SRS

Kavramsal Modelin Oluşturulması

Sistem Analizi ve Tasarımı

BLM2042 GR.1/2

2025-2026 BAHAR YARIYILI

DR.ÖĞR.ÜYESİ YUNUS EMRE SELÇUK / GÖKSEL BİRİCİK

Bu Derste

Sistem Analizi

Fonksiyonel Çözümleme

SRS

Kavramsal Modelin Oluşturulması

Gereksinim Analizi

Sistem analizi modelinin amacı:

- İhtiyaçları açıklamak
- Tasarımın nasıl oluşturulacağının temelini oluşturmak
- Oluşturulan yazılımın ihtiyaçları karşılayıp karşılamadığını onaylayan unsurları belirlemek

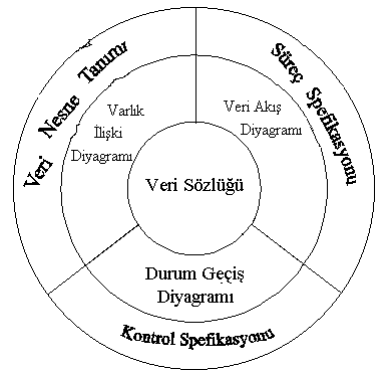
Gereksinim Analizi Modeli

Veri Sözlüğü: Tüm veri nesnelerinin tanımları (metadata)

Varlık İlişki Diyagramı (VİD) (ERD: Entity-Relationship Diag.): Veri nesneleri arası ilişkiler

Veri Akış Diyagramı (VAD) (DFD: Data Flow Diag.): Verilerin nasıl taşındığı, veri akışını sağlayan fonksiyonların neler olduğu

Durum Geçiş Diyagramı (DGD) (STD: State Transition Diag.): Sistem dışındaki olaylar sonucunda sistemin nasıl hareket ettiği



Veri Akış Diyagramı

Sistemdeki
- Varlıklar
- Süreçler - işlemler
- Veri depoları
- Aralarında verinin akışı

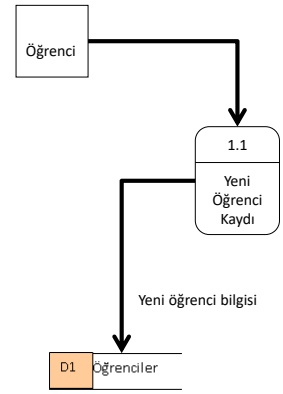
Birden fazla varlık olabilir

Okların çakışmaması için aynı varlık tekrarlayabilir

Oklar tek yönlüdür

Okların tek bir kaynağı ve hedefi vardır

İşlemler hiyerarşiye uygun numaralandırılır



VAD Öğeleri

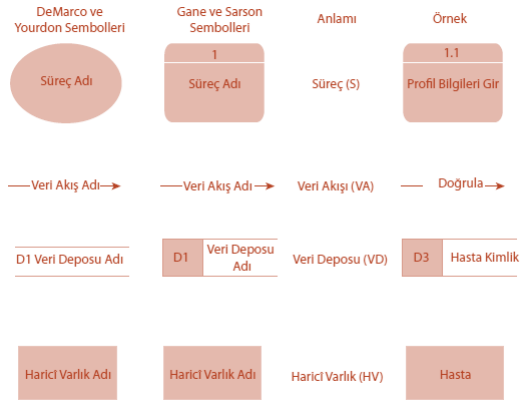
Varlıklar: Kişi, kurum, birim, sistem olabilir. Birincil aktörlere karşılık gelir. Sisteme veri sağlar ya da veri alır.

Veri Akışı: sistemde bir yerden başka bir yere hareket eden veri (ör. barkod no) ya da mantıksal veri koleksiyonu (ör. Rapor çıktısı içeriği).

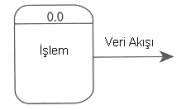
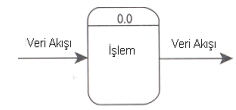
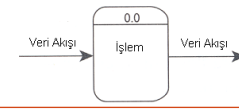
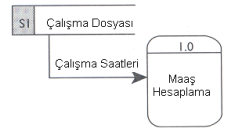
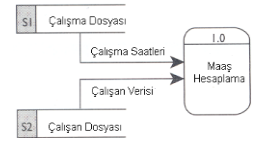
Süreç: Belirli bir işi gerçekleştirmek amacıyla elle veya bilgisayarla yürütülen etkinlik/fonksiyon. Emir kipinde yüklemle isimlendirilir. (ör. Randevu al) Her süreçte tek etkinlik gerçekleşir.

Veri Deposu: Verilerin kalıcı olarak bulunduğu yerler (dosya, klasör, veritabanı, form, çıktı, rapor, karekod, …)

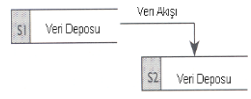
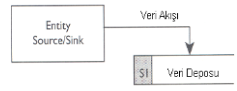
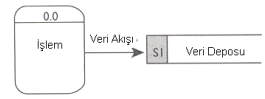
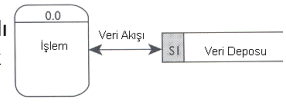
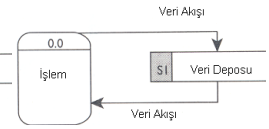
VAD Sembolleri



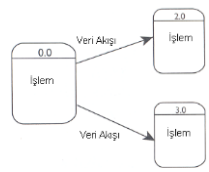
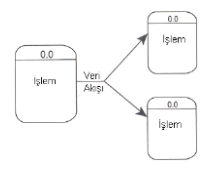
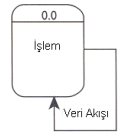
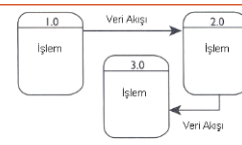
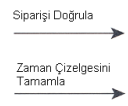
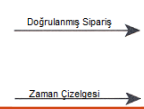
VAD Kuralları

Kural	Yanlış	Doğru
İşlemin sadece çıkışı olamaz.		
İşlemin sadece girişi olamaz.		
İşlem girişleri istenen çıkışı verecek kadar yeterli olmalıdır.		

VAD Kuralları

Kural	Yanlış	Doğru
Her veri deposu bir işlemle ilgili olmalıdır.		
Veri deposu bir varlıkla doğrudan ilişkide olamaz.		
Veri akışı oku iki yönlü olamaz. Bir işlemle veri deposu arasında karşılıklı veri akışı varsa farklı tek yönlü oklarla gösterilmelidir.		

VAD Kuralları

Kural	Yanlış	Doğru
Bir işlemden farklı iki işleme gidecek olan aynı veri, aynı yönde iki uçlu okla gösterilmelidir.		
Veri, hiçbir işlemden geçmeden çıktığı işleme doğrudan dönemez.		
Veri akış okları üzerinde gösterilen veri, sadece isim formatında olmalıdır.		

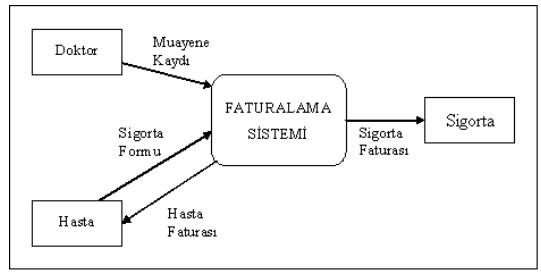
VAD Düzeyleri: Taslak

Ön inceleme sonucunda belirlenir

Sistemle varlıklar arası ilişkiyi gösterir

Ayrıntılı süreç ve veri depoları bulunmaz

Bağlam diyagramı olarak da bilinir. 0. Düzey olarak değerlendirilir.

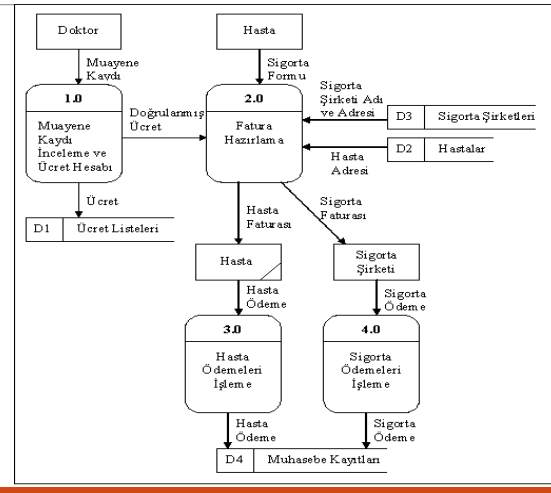


VAD Düzeyleri: 1.Düzey

Süreçleri ve işlemleri, ilişkili veri depoları, varlıklar ve depolarla işlemler arası ilişkileri gösterir

Sistemdeki tüm süreçlerin birbiriyle ve dış kaynaklarla olan ilişkisi belirlenir

Öncesinde «Bağlam VAD» kullanıldıysa, «Düzey 0 VAD» olarak de isimlendirilir



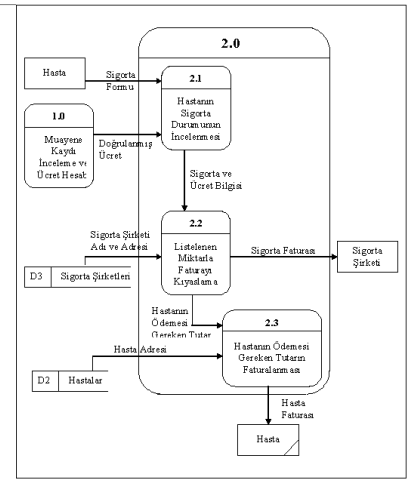
VAD Düzeyleri: 2.Düzey

Her sürecin alt işlemleri ayrıntılarıyla gösterilir

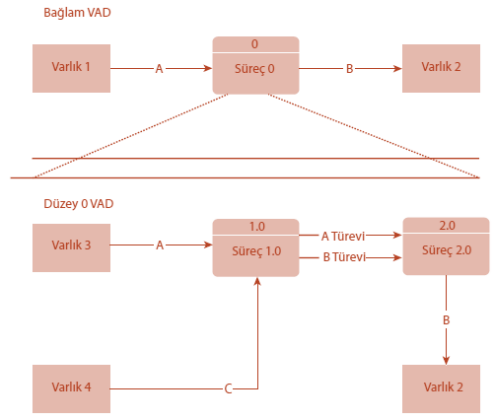
1.Düzeydeki her süreç için bir 2.Düzey VAD çizilir

Tek bir süreçle veri kaynakları arası ilişki detaylı gösterilir

Öncesinde «Düzey 0 VAD» kullanıldıysa, «Düzey 1 VAD» olarak de isimlendirilir

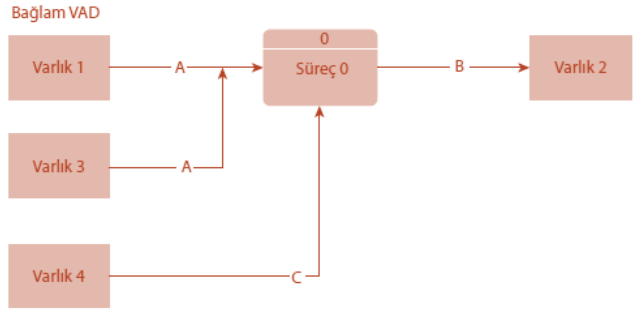


Hata Nerede?

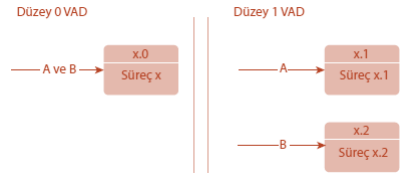


Çözümü

Varlık 3 ve Varlık 4 bağlam VAD'da yer almalıdır.



Burada sorun var mı?



Kavramsal Veri Modeli - Varlık İlişki Diyagramları

Yüksek seviyelidir, kullanıcı topluluğuyla iletişim için kullanılır.

Veri yapısı, donanım, yazılım bilgisi yer almaz.

Tam ve yeterli bileşenler ile sistemin bilgi gereksinimlerinin gerçek gösterimini oluşturur.

Varlık ilişki diyagramları: verinin kavramsal gösterimini sunar.

- Veri nesneleri ile aralarındaki ilişkilerin grafiksel gösterimi
- Kavramsal veri modellemesinin yapılmasını sağlar

İleriki haftalarda

Mantıksal ve Fiziksel VAD

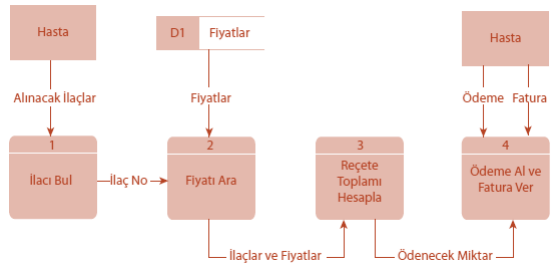
Mantıksal VAD: işler nedir?

- Faaliyetler, varlıklar ve üretilip kullanılacak veriler tanımlanır
- Analiz safhası

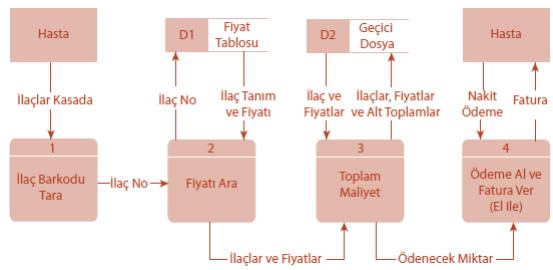
Fiziksel VAD: sistem nasıl uygulanacak,

- Hangi donanım, yazılım, dosya, insan kaynağı nasıl kullanılacak?
- Tasarım safhası

Hastane Bilgi Sistemi - Eczaneden ilaç temin etme Mantıksal VAD



Hastane Bilgi Sistemi - Eczaneden ilaç temin etme Fiziksel VAD



Süreçler Nasıl Modellenecek?

Süreçler tanımlandı ama içindeki işlemlerle ilgili açıklama yok.

Mantıksal modellemede süreç iç yapıları ve işlevleri ifade edilir.

Süreç tanımlama formları kullanılır.

- Mantık tanımlamak için 3 yöntem vardır:
  - Yapısal dil (sözde kod)
  - Karar tabloları
  - Karar Ağaçları

Süreç Tanımlama Formu

Süreç Tanımlama Formu			
No	4		
Ad	Ödemeyi Al ve Fatura Ver		
Tanım	Reçetedeki ilaçların toplam maliyetine bağlı olarak hasta ödemesi nakit ya da kart ile tahsil edilecek ve fatura verilecek.		
Girdi Veri Akışı	• Süreç 3'ten hesaplanarak gelecek olan "ne kadar ödeme yapılacağı" bilgisi, • Hastadan yapılacak tahsilat.		
Çıktı Veri Akışı	Fatura		
Süreç Tipi	☐ Çevrimiçi	☐ Otomatik	☒ El İle

Süreç Mantığı	READ ÖDENECEK_MİKTAR
	ÖDEME Al
	SELECT CASE
	CASE 1 (ÖDEME büyüktür ÖDENECEK_MİKTAR)
	DO Para üstü ver
	ÖDEME=Tamam
	CASE 2 (ÖDEME eşittir ÖDENECEK_MİKTAR)
	DO ÖDEME=Tamam
	CASE 3 (ÖDEME küçüktür ÖDENECEK_MİKTAR)
	DO ÖDEME=Tamam değil
	END SELECT
	IF ÖDEME eşittir Tamam
	THEN Fatura ver
	ELSE
	Hastaya bilgi ver
	END IF		

Çözülmeyen Sorunlar	• Nakit ödeme dışında bir ödeme biçiminin kabul edilip edilemeyeceği belli değil. • Reçete kayıp ya da unutulmuş ise nasıl bir işlem yapılacak?		

Yapısal Dil

Süreçleri mantıksal olarak ifade edebilmek için standart İngilizce kelimelerden oluşan bir alt küme.

Sıralı (ardışık), karar, durum ve döngü işlemleri için yapı blokları kullanılır.

Girintili yazı stili kullanılır.

Yapısal Dil Blokları

Yapısal Dil Bloğu	Örnek
Ardışık Blok Yapısı: Herhangi bir yönlendirmenin bulunmadığı, özel işlemler ya da kontrol gerektirmeyen işlem bloklarıdır.	Eylem #1 Eylem#2 Eylem#3 Eylem#4
Karar İşlem Bloğu: IF ifadesinden sonra verilen koşul doğru ise THEN ifadesinden sonraki eylemler uygulanır; aksi durumda ELSE ifadesinden sonraki eylemler uygulanır.	IF Koşul 1 Doğru THEN Eylem#1 uygula ELSE Eylem#2 uygula END IF
Durum Kontrol Bloğu: Karar işlem bloğunun özel bir tipidir. Bir koşulun izleyebileceği birden fazla durum varsa ve bu durumlardan biri oluştuğunda diğerleri oluşamıyorsa kullanılan blok yapısıdır.	READ Kontrol-edilecek-değer SELECT CASE CASE 1 (Kontrol-edilecek-deger=Koşul1) DO Eylem#1 CASE 2 (Kontrol-edilecek-deger=Koşul2) DO Eylem#2 CASE 3 (Kontrol-edilecek-deger=Koşul3) DO Eylem#3 CASE 4 (Kontrol-edilecek-deger=Koşul4) DO Eylem#4 END CASE
Döngü Blok Yapısı: Koşul sağlanana kadar tekrarlanması gereken eylemler varsa kullanılır.	DO WHILE Koşul1 doğru oldukça Eylem#1 ENDDO Veya DO Eylem#1 UNTIL Koşul1 doğru olunca

Yapısal Dil Örneği

```
READ ILAÇ_STOK_MIKTARI
    SELECT CASE ILAÇ_STOK_MIKTARI
        CASE 1 (ILAÇ_STOK_MIKTARI büyükse KRITIK_STOK)
        Herhangi bir eylem uygulama
        CASE 2 (ILAÇ_STOK_MIKTARI esitse KRITIK_STOK)
                Eczane çalışanına bildirimde bulun
        CASE 3 (ILAÇ_STOK_MIKTARI küçükse KRITIK_STOK)
        Otomatik olarak sipariş üret
        CASE 4 (ILAÇ_STOK_MIKTARI esitse SIFIR)
                Muadil ilaçları göster
        IF ILAC_URETIMI_DURDU
                THEN İlacın reçeteye yazılmasını engelle
        END IF
    END CASE
```

Karar Tabloları

Karmaşık kararların mantığını belirleme mekanizması

Durumlar, Kurallar, İşlemler, Kararlardan oluşur

		KURALLAR				
		1	2	3	4	5
DURUM	Disiplin cezası almış	E	H	H	H	H
	Not ortalaması > 70	H	E	H	E	H
	Giriş puanı > Giriş ortalaması	H	E	E	H	H
İŞLEM	Okul süresince tam burs		X			
	1 yıl yarım burs			X		
	Okul süresince yarım burs				X	
	Burs alamaz	X				X

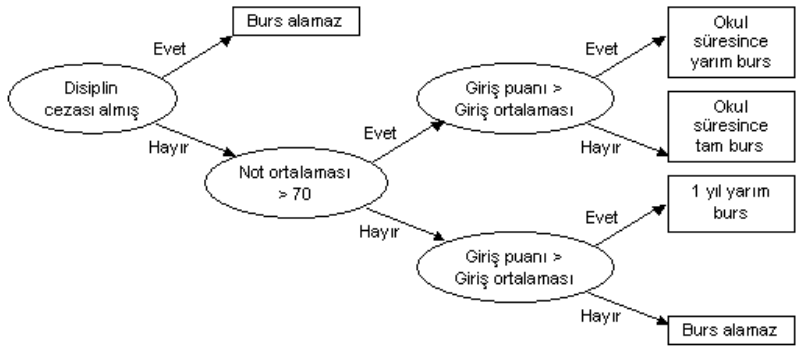
Karar Tablosu Örneği

İlaç İndirim Hesapla	KURALLAR			
KOŞULLAR	1	2	3	4
Çalışan Hasta	E	E	H	H
Emekli Hasta	H	H	E	E
Toplam Tutar 50 TL'den küçük ya da eşit	E	H	E	H
Toplam Tutar 50 TL'den büyük	H	E	H	E
EYLEMLER				
İndirim Oranı 1 Uygula	X			
İndirim Oranı 2 Uygula		X		
İndirim Oranı 3 Uygula			X	
İndirim Oranı 4 Uygula				X
Ek %10 İndirim Uygula				X

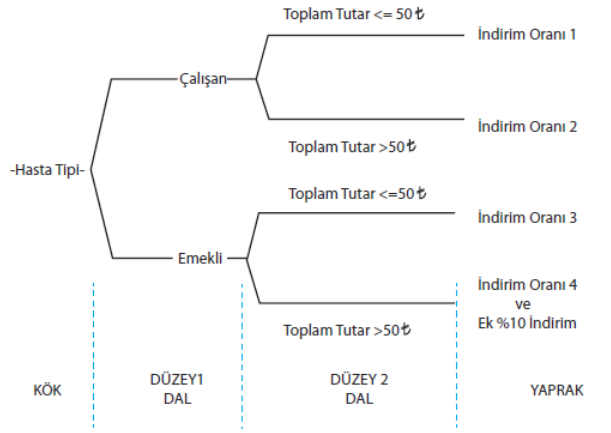
Karar Ağaçları

Daha az karmaşık yapılar için uygun

Olasılıklar kullanılmaz



Karar Ağacı Örneği



Veri Sözlüğü

Sadece VAD yeterli değil

Tüm bilişim maddeleri tanımlanmalı

- Veri Akış Sözlük Girişi
- Veri Deposu Sözlük Girişi
- Veri Yapısı Sözlük Girişi
- Veri Elemanı Sözlük Girişi
- İşlem Sözlük Girişi

Veri akış ve işlemler (kısa tanımlanmışlardı) sözlükte tanımlanmalı

Bileşik veriler öğelerine göre, basit veriler anlamlarına göre tanımlanır

Veri Akış Sözlük Girişi

Veri Akış Sözlük Girişi

Veri Akış Adı	**:** FATURA
Tanım	**:** Müşteriye fatura edilecek doküman için gerekli bilgiler
Nereden	**:** 1.1 Faturayı Hazırla
Nereye	**:** 1.2 Fatura numarasını hazırla
Veri Yapıları	**:** Fatura Detayları (K) Müşteri Detayları (K) (K: Kompozit, E: Elemanter)
Açıklama	**:** .....

Veri Deposu Sözlük Girişi

Veri Deposu Sözlük Girişi

Veri Depo Adı	**:** SATIŞ SİPARİŞ FORM DOSYASI
Tanım	**:** Satış sipariş formlarının saklandığı arşiv dosyasıdır
Veri Yapıları	**:** Satış sipariş kaydı
Miktar	**:** Günde yaklaşık 100 kayıt
Erişim	**:** Sipariş bölümü personeli
Açıklama	**:** .....

Veri Yapısı Sözlük Girişi

Veri Yapısı Sözlük Girişi

(Her kompozit veri yapısı için olmalıdır)

Veri Yapı Adı	: SATIŞ SİPARİŞ KAYDI	
Tanım	: Müşterinin mal siparişi için kullandığı satış sipariş formu	
Veri Elemanları	: MusteriNo	(E)
	SiparisNo	(E)
	SiparisTarihi	(K)
	* ParcaNo	(E)
	* Miktar	(E)
	* BirimFiyat	(E)
Açıklama	: * olanlar, her bir parça kaydı için oluşur.	

Veri Elemanı Sözlük Girişi

Veri Elemanı Sözlük Girişi

Veri Elemanı Adı	: MusteriNo
Tanım	: Müşteriyi tanımlayan numara
Tip	: Numerik
Uzunluk	: 4
Değer Aralığı	: 0001-6999
Diğer Detaylar	: .....

Her elemanter veri yapısı için. Sadece değer ise kod tablosu da olabilir

İşlem Sözlük Girişi

İşlem Sözlük Girişi

İşlem Adı	**:** 2.0 Sipariş Satış Verisini Gir
Girdi	**:** Satış siparişleri
İşlem Tanımı	**:** ... yap, eğer değilse ..... yap vb.
Çıktı	**:** Girilmiş satış siparişleri

Problem Uzayının Modellenmesi

Problem Uzayının Modellenmesi

Amaç gerçek dünyanın (problemin) doğru, anlaşılır, sınanabilir modelini oluşturmak

Problemin anlaşılması aşamasıdır

Teknik yorumdan çok müşteri isterleri ön plandadır

İsteklerin modellenmesi nesneye dayalı değildi, burası ise nesneye dayalı.

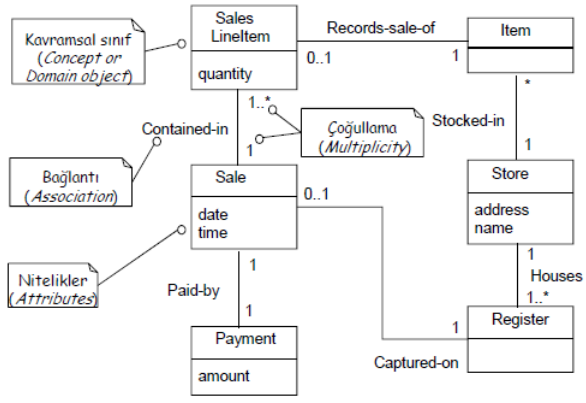
Gerçek dünyadaki kavramsal sınıflar ve nesneler yer alır (nesneler, o anda görülen özellikleri, aralarındaki ilişkiler (bağlantılar)). Yazılım nesneleri henüz düşünülmez

UML ile görselleştirilir.

Oluşacak sistem anlaşılır.

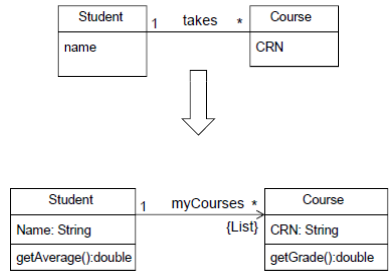
Tasarıma girdi sağlar (sorumlulukları atamak üzere)

Kavramsal Sınıf Diyagramı



Problem ile Yazılımın Yakınlaşması

Tasarımda yazılım sınıfları oluşturulurken ve sorumluluklar atanırken uygulama modeli kullanılır



Kavramsal Sınıfların Bulunması

Gerçek dünyadaki somut/soyut varlıklar

Kategori listesi: Deneyim sonucu sık karşılaşılan kategoriler. Zaten görülen sınıfları bulur. Gözden kaçırma olasılığı yüksek.

Senaryolardaki isimler (isim tamlamaları): Senaryolardaki tüm isim ve tamlamalar aday sınıf olarak alınır. Çoğunlukla fazla sınıf çıkar, elenir.

Varolan model güncellemesi: Yayımlanmış modellerin uyarlaması yapılır.

Örnek: İsim İşaretleme

Ar-Ge Projeleri Kuluçkalandırma Bilgi Sistemi

Firmamızın (TGB: Teknoloji Geliştirme Bölgesi) yönetim kurulu araştırma-geliştirme projelerine (Ar-Ge'ye) verilen ağırlığı arttırmayı kararlaştırmıştır. Kendi Ar-Ge ekibimize sahip olmakla birlikte, girişimci gençlerimizin ve araştırmacılarımızın geliştireceği yenilikçi fikirlere dayanan yeni firmaların (start-up) kurulmasına da destek vermek istemekteyiz. Bu destek Ar-Ge binamızda ayrılacak olan kuluçka alanında yapılacaktır. Ar-Ge projelerinin ihtiyaçlarına ve girişimci firmanın gereksinimlerine göre fiziksel mekan, teknolojik donanım, finansman ve benzeri kaynakların girişimci firmaya tahsisi söz konusu olacaktır. Verilecek kuluçka desteğinin firmamıza yeni kurulan firmadan hisse sahibi olmak, Ar-Ge projesinin sonunda ortaya çıkacak ürüne önemli maliyet avantajları ile sahip olmak gibi getirileri olacaktır.

Girişimciler kuluçka desteği başvurularını yapmak için, bilgi sistemimize web tarayıcısı ile erişerek bir form doldururlar. TGB Ar-Ge ekip liderimizin de katılacağı genişletilmiş yönetim kurulu toplantısında başvurular ön elemeden geçirilir. Dikkate değer bulunan projeler, adımları Ar-Ge destek personeli tarafından izlenecek olan ayrıntılı değerlendirme sürecine girer. Bu süreçte üniversitelerde ilgili konularda görev yapan akademisyenlerden seçilecek en az bir, en fazla üç hakemin değerlendirmesine yine bilgi sistemimiz üzerinden sunulur. Bu amaçla önceden oluşturulmuş ve sürekli güncellenen hakem havuzundan hakem adayları belirlenerek kendilerine özet bilgisi verilen projeyi değerlendirmeyi isteyip istemedikleri sorulur. Değerlendirme belli bir sürede bitirilmek zorundadır, bu süre aşıldığı zaman hem hakeme hem de Ar-Ge destek personeline uyarıcı mesajlar otomatik olarak gönderilmelidir. Hakemlerin oy çokluğu olursa proje kabul edilir, eğer olumlu ve olumsuz karar sayısı eşit olursa son kararı Ar-Ge ekip lideri verir. Değerlendirme süreci olumlu sonuçlanan projelerin sahibi firmalarla karşılıklı iyi niyet anlaşması imzalanır ve projenin izleme süreci başlar. İzlemeyi genişletilmiş yönetim kurulu yürütür.

Gereksiz Sınıfların Elenmesi

Fazlalık sınıflar: aynı unsuru ifade edenlerden, daha tanımlayıcı olan. Akademisyen - Hakem

İlgisiz sınıflar: çözümle ilgisi olmayan ya da o aşamada ilgilenmeyeceğimiz sınıflar: *Firma hissesi*

Belirsiz sınıflar: sınırları iyi çizilmemiş, kaba-geniş tanımlı sınıflar. Genellikle birden fazla sınıftan oluşmuşlardır ya da başka sınıfın parçasıdır:

Nitelikler: kendi başına varlığı anlamlı olmayanlar, sınıfların özellikleridir: *Miktar*

İşlemler: Sadece nesnelere uygulanan işlemler sınıf olmaz. Kendi nitelikleri olmalı ve hizmet alışverişi yapmalıdır: Ar-Ge Destek Ödemesi (miktar, para birimi, tarih, gider türü …)

Roller: Sınıflar arası ilişki olan roller sınıf olamaz.

Gerçekleme Elemanları: Yazılım sınıfları bu uzayda yer almaz.

Sınıf olup olmadığının testi için şu sorular sorulabilir:

- Kavramla ilgili veri saklanması gerekiyor mu?
- Değişik değerler alabilecek farklı özellikleri var mı?
- Kavramdan birçok nesne türeyebilir mi?
- Uygulamanın kapsama alanı içinde mi?
- Sınırları iyi çizilmiş mi, tanımı yapılabiliyor mu?

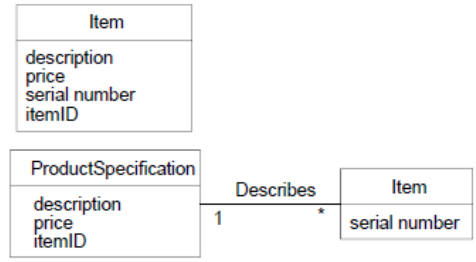
Not: Ders akışı sırasında üstteki ağır iki paragraflık metin üzerinden isimler belirlenip bu yansıdaki kriterlere tabi tutula bilir (uzun sürer) veya sonraki yansıdaki kullanım senaryosu metninden bu işler yapılabilir (alan modeli UML şeması buna göre hazırlanmıştır)

Betimleme Sınıfı (description-specification) İhtiyacı

Satış yaparken nesneler ürünleri tutsun.

Bir cinsten tüm malzeme satılırsa, nesne kalmaz. Bilgi kaybı olur.

Bu yüzden özellikleri ayrı bir sınıf olarak tutmak gerekebilir.



Bağlantıların Belirlenmesi

Bir nesnenin kaç tane nesneyle ilişki içinde olacağı çoğullama ile gösterilir.

Bağlantılara doğru isimler verilmelidir. Tip-fiil-tip

Bazı bağlantıların unutulması tasarımı çok etkilemez.

Kavramsal sınıfların doğru bulunması daha önemlidir.

Fazla bağlantı anlaşılırlığı azaltır.

Yaygın bağlantılar listesinden yararlanılabilir:

- Fiziksel barındırma, mantıksal barındırma, kayıt ilişkisi, kullanım ilişkisi, tanım, sahiplik, …
- Kullanım senaryolarındaki fiillerden yararlanılabilir.

Gereksiz Bağlantıların Elenmesi

Elenen sınıflar arası bağlantılar gereksizdir

Sistemin amacı dışındaki bağlantılar gereksizdir

Gerçeklemeyi ilgilendiren bağlantılar gereksizdir

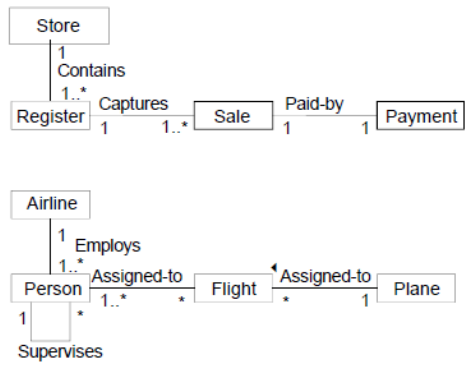
Faaliyetler bağlantı değil, etkileşimdir. (ATM kredi kartı kabul eder)

Üçlü bağlantılar ikili hale çevrilmelidir. Memur hesapla ilgili işlemleri girer → memur işlemleri girer. İşlemler hesapla ilgilidir.

Örnek: Bağlantılar

Tasarımı ilgilendiren bağlantıları ortaya çıkarın

Tip-fiil-tip doğru isim ataması yapın.



Örnek: İsim İşaretleme

KULLANIM SENARYOSU: KULUÇKA PROJE DEĞERLENDİRME

Birincil Aktör: Ar-Ge destek personeli

İlgililer:

Ar-Ge destek personeli: Değerlendirme sürecinin zamanında tamamlanmasını ister.

Ar-Ge Lideri: Oy kullanmasına ihtiyaç duyulduğu zaman bilgilendirilmek ister.

Hakemler: Değerlendireceği projenin bilgilerine rahatça ulaşıp raporunu kolayca yazabilmek ister.

Ön Koşullar:

Girişimcinin başvurusu ön elemeden geçmiştir.

Son Koşullar:

Değerlendirme süreci olumlu veya olumsuz olarak karara bağlanmıştır.

İzleme süreci başlatılmıştır.

Örnek: İsim İşaretleme

Ana Senaryo:

1. Ar-Ge destek personeli, projenin konusuna göre havuzdan hakem seçimi yapar.
2. Ar-Ge destek personeli hakemlerden atama kabulü yanıtı bekler.
3. Hakemler belirlenen süre içerisinde değerlendirmelerini sisteme girer.
4. Oy birliği veya çokluğu ile kabul edilen projelerin izleme süreci başlatılır.

Alternatif akışlar:

1.a. İlgili konularda uzman hakem havuzda bulunamaz

1. Ar-Ge destek personeli havuzdaki mevcut hakemlerden yeni hakem adayları talep eder.
2. İstekli adaylar Ar-Ge destek personeli tarafından havuza eklenir.

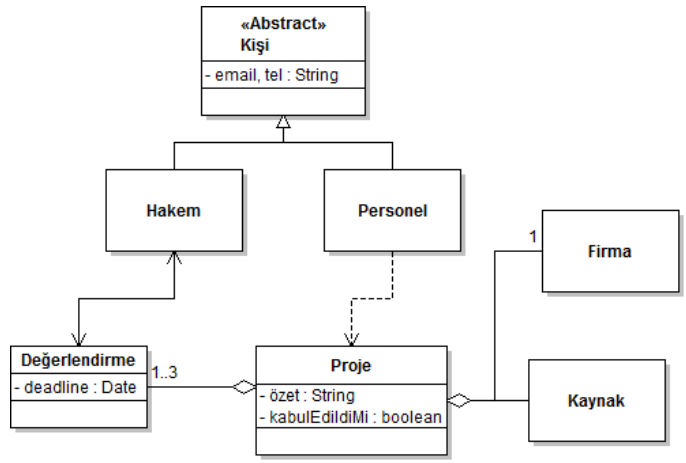
2.a. Hakem(ler) olumsuz yanıt verir.

1. Ar-Ge destek personeli havuzdan yeni hakem seçimi yapar.

3.a. Bir hakem süreyi zaman aşımına uğratır.

1. Sistem hakeme ve Ar-Ge destek personeline otomatik mesaj gönderir.
2. Ar-Ge destek personeli hakemi telefonla arar ve gerekirse süreyi bir miktar uzatır.

KULLANIM SENARYOSUNDAN ALAN MODELİNE



Özelliklerin Belirlenmesi

Özellik: nesne yaratıldığında nesneye özgü değer alabilen veri.

Senaryolarla ilgili nitelikler bulunmalıdır.

Basit veri tipleri ile ifade edilirler.

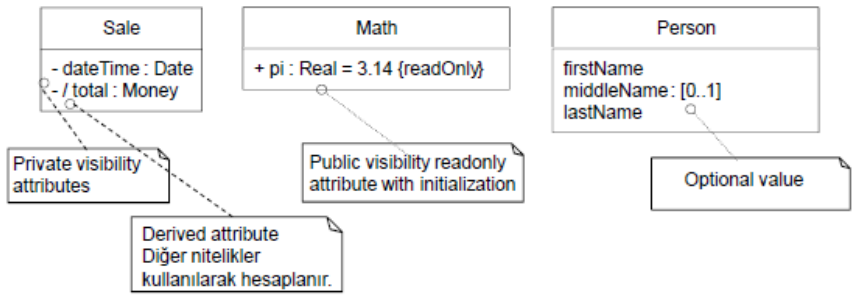
- birimleri varsa doğru değil. (para gibi)

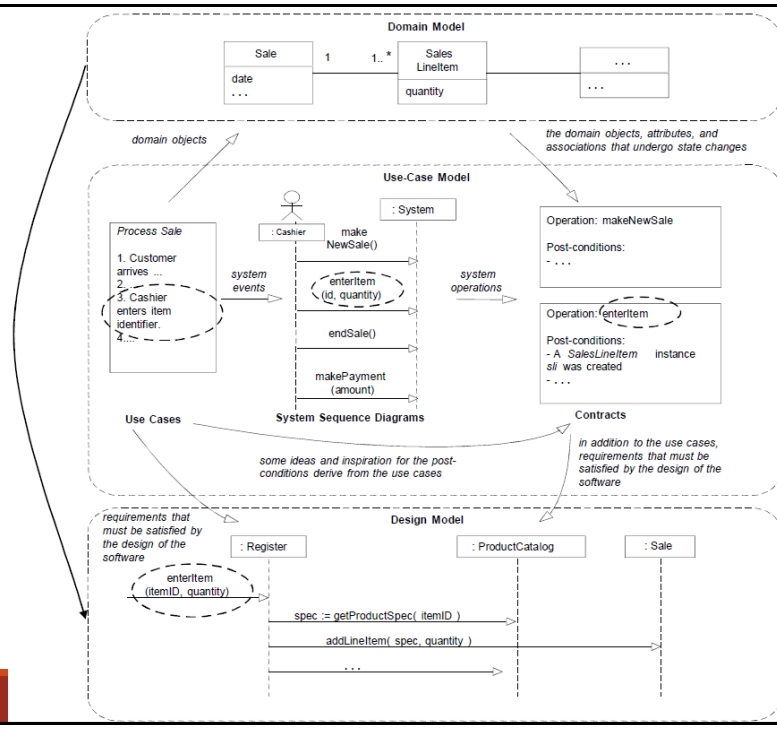
Karmaşık tipteyse, başka bir sınıf olma olasılığı yüksektir.

- birden fazla alansa (tel no, ad soyad)
- işlem yapılıyorsa (kredi kartı ve doğrulama)
- kendi nitelikleri varsa (fiyat: geçerlilik tarihi))

Özellikler-Detay

Analizde kavramsal sınıf özellikleri hakkında detay bilgi varsa, tasarıma kaynak olması için bunlar da belirtilebilir.





Gelecek Dersler

Sistem Tasarımı

- Girdi, Çıktı, Veri Yapısı, Arabirim Tasarımları

Bu yansı ders notlarının düzeni için boş bırakılmıştır.

Sistem Analizi ve Tasarımı

BLM2042 GR.1/2

2025-2026 BAHAR YARIYILI

DR.ÖĞR.ÜYESİ YUNUS EMRE SELÇUK / GÖKSEL BİRİCİK

Bu Derste

Sistem Tasarımı

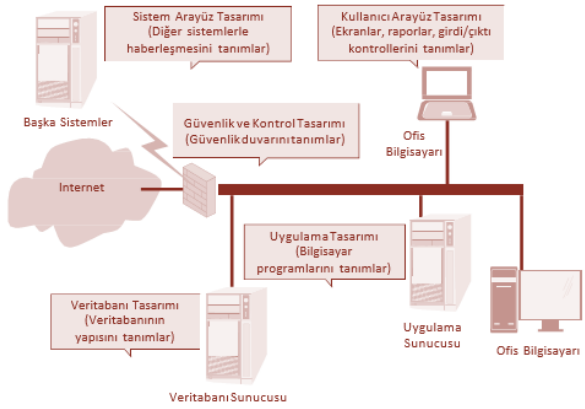
- Altyapı Belirleme
- Ön (Genel) Tasarım
- Ayrıntılı Tasarım
- Tasarım Modelinin Oluşturulması

Sistem Tasarımı

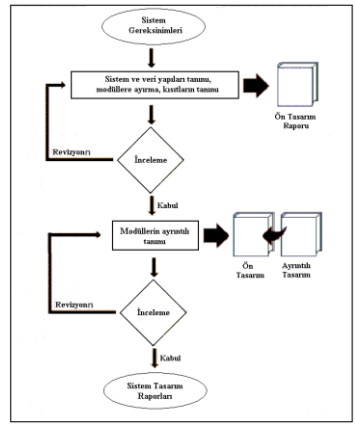
Ön Tasarım
- Alt yapı belirleme
- Modül mimarisinin oluşturulması

Ayrıntılı Tasarım
- Sistem Etkileşimi tasarımı
  - Çıktı tasarımı
  - Girdi tasarımı
  - Arabirim tasarımı
- Program tasarımı
- Veri tabanı tasarımı

Sistem Tasarımı Bileşenleri



Tasarım Aşamaları



Sistem Tasarım Yaklaşımları

Model Güdümlü Yaklaşımlar: Geliştirilen mantıksal modeller kullanılarak model güdümlü tasarım modelleri elde edilir. Bu modeller yeni kurulum ve gerçekleştirme için tasarıdır.

- Modern Yapısal Tasarım
- Bilgi Mühendisliği
- Prototipleme

Hızlı Uygulama Geliştirme: Model hazırla, prototip yap, model hazırla, prototip yap, vb. döngü. Ortak uygulama geliştirme oturumları

Ortak Uygulama Geliştirme: Tasarım tüm paydaşların katıldığı atölye çalışmaları ile gerçekleşir.

Ön (Genel) Tasarım

Sistem nasıl temin edilecek?

- Sistem sıfırdan oluşturulabilir: Kurum içi geliştirme +/-?
- Satın alınıp ihtiyaca göre özelleştirilebilir +/-?
- Dışarıdan (hizmet olarak) temin edilebilir +/-?

İşlevsel olmayan gereksinimler mimari tasarımı etkiler

- Sistem ne kadar hızlı çalışacak? Kapasitesi ne olacak? Şifreleme ve virüs kontrol ihtiyacı var mı? vb.
- Yeni sistemi desteklemek üzere alınacak donanım ve yazılımlara yönelik mimari kararlar

Sistem Mimari Modellemesi

Bilgi sistemi, merkezi mi yoksa dağıtık mı olacak? (Birçok sistem, ağ üzerinden dağıtık çalışmaktadır.)

Bir ağ üzerinden depolanan verinin dağıtımı nasıl olacak? (Birçok modern veri tabanı, dağıtık ya da ağ üzerinde çok kopyalı olarak bulunabilmektedir.)

Geliştirilecek yazılım için uygulama teknolojileri ne olacak? Hangi programlama dili ve araçları kullanılacaktır?

Satın alınabilen ticari yazılımların bütünleştirilmesi nasıl olacak? (Ticari yazılımın gereksinimlere göre düzenlenme ihtiyacı vardır.)

Kullanıcı arayüz uygulamasında kullanılacak teknolojiler neler olacaktır?

Diğer sistemlerle arayüz oluşturmak için kullanılacak teknolojiler neler olacaktır?

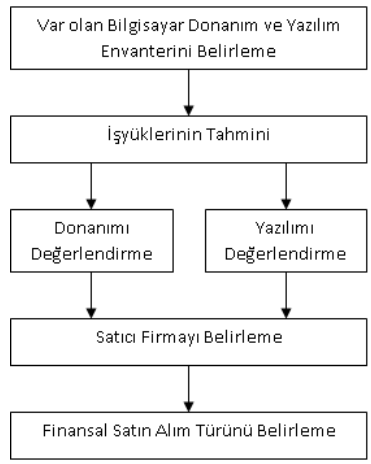
…

Mimari Tasarım Bileşenleri

Yazılım
- Veri Depolama
- Veri Erişim Mantığı
- Uygulama Mantığı
- Sunum Mantığı

Donanım
- İstemci bilgisayarlar
- Sunucu bilgisayarlar
- Ağ yapısı

Sistem Altyapısını Belirleme



Envanter Belirleme

Sistem analisti, var olan sistemin alt yapısının durumunu görmek için; teçhizatın

- modelini ve üreticisini,
- durumunu (kullanılmıyor, bakıma ihtiyacı var, çalışır durumda vb.),
- yaşını,
- planlanan yaşını,
- işletme içindeki fiziksel yerini,
- sorumlu çalışanı veya bölümü,
- finansal durumunu (işletmenin kendi malı, kiralık ve leasing yapılmış vb. şeklinde) belirler.

Mevcut donanım, gerekli altyapıyla kıyaslanır.

Ayrıntılı Tasarım

Çıktı Tasarımı

Girdi Tasarımı

Arabirim Tasarımı

Uygulama (Program) Tasarımı

Veri tabanı Tasarımı

Çıktı Tasarımı

Çıktı: Sistemin Kullanıcılara verdiği bilgi, üretilen raporlar

Amaçlar:

- Belirlenen amaca hizmet etme
- Kullanıcı için anlamlı olma
- Uygun sayıda olma
- Hangi kullanıcılara dağıtılacağının doğru belirlenmesi
- Zamanında sağlanma (günlük, aylık, yıllık veya koşula bağlı raporlar)
- Doğru çıktı yönteminin (ortamının) seçilmesi

Kullanıcıyı etkileyecek yönlendirme:

- Bilgilerin belirli kriterlere göre sıralanması
- Sınırların Belirlenmesi
- Grafik tipi – rengi – ölçeğinin belirlenmesi

Girdi Tasarımı

Kaliteli çıktı için girdi kalitesi önemli

Amaçlar

- Etkinlik: Form ve ekran görüntülerinin belli bir amacının olması
- Doğruluk: Analizde tanımlanan tüm işlemleri yerine getirmesi
- Kullanım kolaylığı: Bilgi girişi kullanıcılarının fazla zamanını almaması ve ergonomik olması
- Uyumluluk: Bir formda diğerine ya da ekran görüntüsüne geçişte düzenin değişmemesi
- Basitlik: Gereksiz ayrıntıya yer verilmemesi ve karmaşık olmaması
- Çekicilik: Ekran ve form yapılarının güzel görünmesi

Girdi Tasarımı Prensipleri

Çevrimiçi işleme ve toplu işleme girdileri uygun olarak kullanılır

Veri, kaynağında tutulur

Klavye tuşlamaları azaltılır

Doğrulama ve geçerleme yapılır
- Tamlık, Biçim, Aralık, Tutarlılık

Girdi Tasarımı - Ekranlar

Kolay kullanım ve basitlik: Gereksiz bilgi bulunmaması, pencereler içinde girilecek bilgilerle ilgili açıklamalar olması

Uyumluluk: Bilgi toplama formları ve diğer ekran görünümleri arası

Hareket kolaylığı: Ekrandan ekrana geçme ve başlık kolonunu sabit tutarak diğer kolonları kaydırma vs.

Çekici ekran tasarımı: Tüm ekranların belli bir düzene uygun hazırlanması ve imleç yapısı, font tipinin seçimi

Kullanıcı Arabirimi Tasarımı

Arabirim: Sistemin kullanıcıyla iletişime giren elemanı

Tipleri:

- Doğal dil arabirimleri
- Soru-cevap sorgulamalar
- Menüler
- Girdi-çıktı formları
- Komut dili
- Sistem bildirimleri

Amaç:

- Etkinlik: Kullanıcıların gereksinimlerine uygun olarak sisteme erişmelerini sağlama
- Verimlilik: Hataları azaltma, veri giriş hızını arttırma
- Kullanıcıların görüşlerinin alınabilmesi
- Ergonomik olması

Kullanıcı Arabirimi Tasarımı

Yerleşim planı: Ekran üç ana alana bölünür.

- En üst alan, sistemde gezinim sağlar
- Orta alan kullanıcı çalışmalarına ayrılır
- En alt alan, yapılanlarla ilgili durum bilgisi verir

Arabirim Tasarım Problemleri

Bilginin düzensiz görünümde yığınlar şeklinde olması

Kullanıcının yürütmesi gereken çeşitli görevler arasındaki geçişin uyumlu olmaması

Ekran üzerindeki komutlarda kullanılan terminolojinin karmaşık olması

Sistem tarafından verilen hata mesajlarının açıklayıcı olmaması

El kitaplarının anlaşılmaz derecede karmaşık olması

…

Sistem(Uygulama) Mimarisini Belirleme

Modüler program yapısı geliştirilir.

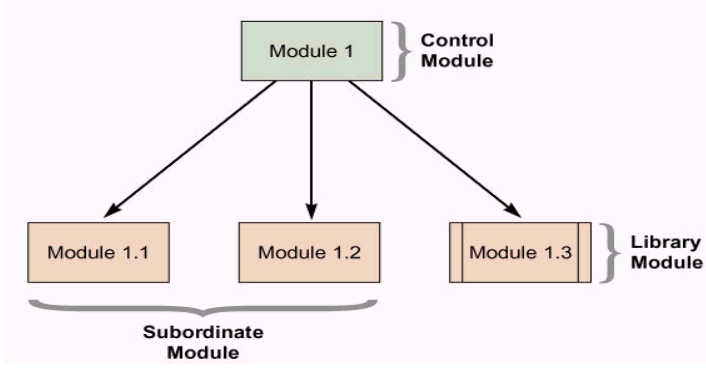
Aralarındaki ilişkilerin denetimi belirlenir.

Veri yapısı ve program birleştirilir, ara birimler tanımlanır.

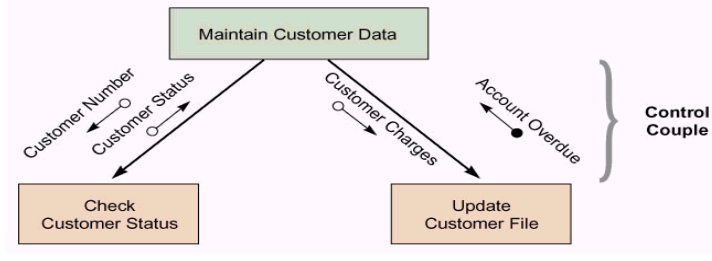
VAD ve veri sözlüğüne dayanan yapı diyagramı ile gösterilir.

- Modüller: Anlaşılması ve bakımı kolay olan, mantıksal program işlem birimleri
- Veri iletişimi: Modüller arası iletilen veri
- Kontrol mesajları: Modüller arasında bir durumu ya da hareketi başka modüle aktarma mesajı (örneğin: dosya sonu)
- Durumlar: Kontrol Modülünün hangi alt modülü çağıracağının gösterimi
- Döngüler: Bir veya daha fazla tekrar eden alt modül işlemesinin gösterimi

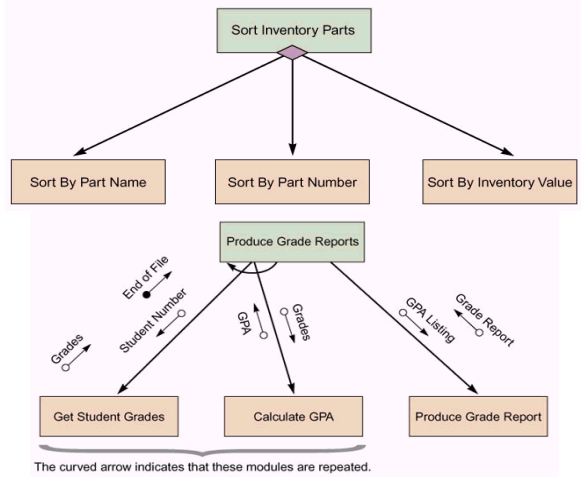
Ana Modül – Alt Modüller



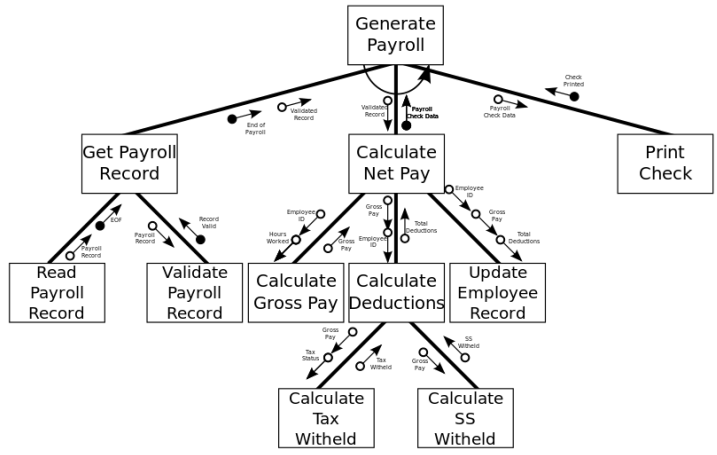
Veri İletişimi – Kontrol Mesajları



Durumlar - Döngüler



Yapı Diyagramı



DERS NOTLARINDA BU NOKTADAN SONRA DÜZENLEMELER OLACAKTIR

GÜNCELLENECEK İÇERİK ARA SINAVDAN SONRA PAYLAŞILACAKTIR.

Tasarım Spesifikasyonları

Ön tasarım spesifikasyonları: Yazılım sisteminin genel özellikleri ve ilişkileri

Mimari tasarım spesifikasyonları: Sistemin yapısı ve kuruluşu

Ayrıntılı tasarım spesifikasyonları: Modüller içerisindeki kontrol akışı, veri gösterimi ve diğer algoritmik ayrıntılar

Nesneye Dayalı Tasarım Modelinin Oluşturulması

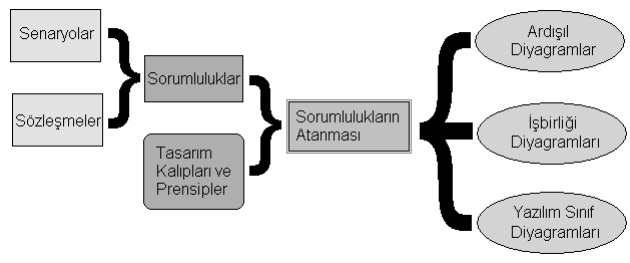
Problemin mantıksal çözümü oluşturulur.

YAZILIM SINIFLARI ve aralarındaki İŞBİRLİĞİ (etkileşim) belirlenir.

- Tasarım Sınıf Diyagramı
- Etkileşim Diyagramları

Etkileşim: sınıfların davranışlarının belirlenmesi → sorumlulukların atanması

Nasıl Yapılır?



Etkileşim Diyagramları

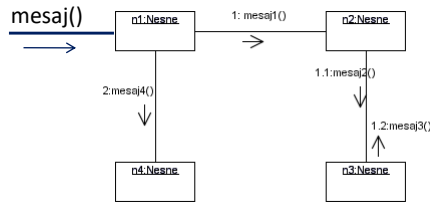
İşbirliği Diyagramları

- Az yer kaplar (+)
- Dallanma, paralellik ve iterasyonlar kolay gösterilir (+)
- Mesaj sırasını anlamak zor (–)

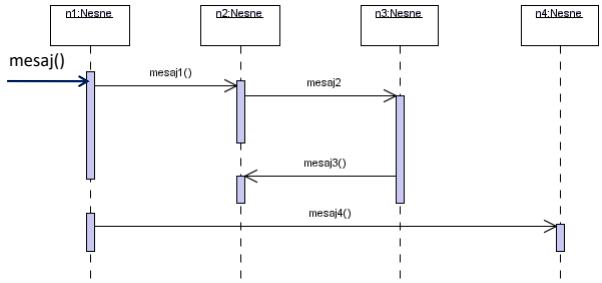
Ardışıl Diyagramlar (Sıralama Diyagramları)

- Fazla yer kaplar (–)
- Dallanma ve paralellik gösterilemez (–)
- Mesaj sırasını anlamak kolaydır (+)

Mesaj Sıra Numaraları

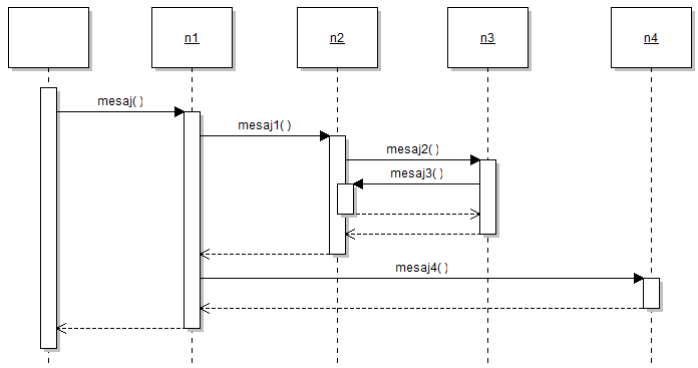


Mesaj Sıra Numaraları



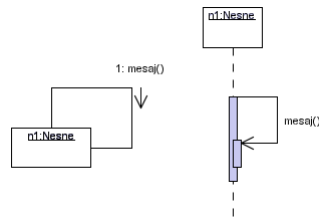
Geri dönüş okları olmadan, özellikle mesaj2-mesaj3 metot çağrılarının iç içe mi ardışıl mı olduğu, işbirliği diyagramı bilinmeden tam anlaşılamamaktadır.

Mesaj Sıra Numaraları

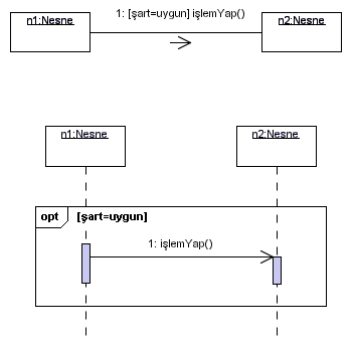


Violet UML ok başlangıcının boşlukta olmasına izin vermemektedir.

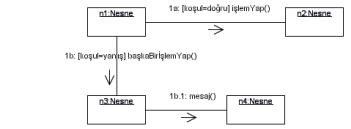
Kendine mesaj

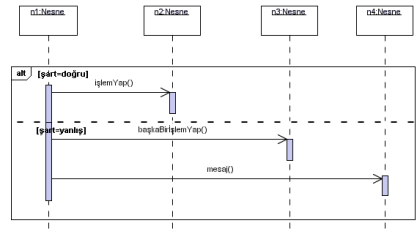


Koşullu mesajlar

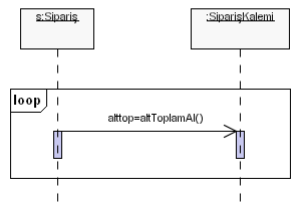


Karşılıklı dışlamalı mesajlar

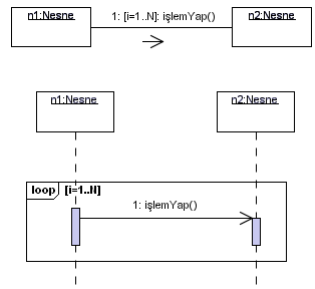




Döngüler



Döngüler



Kullanım senaryolarının gerçeklenmesi

Senaryodaki durumların modellenerek gerçeklenmesi

Tasarım: yazılım sınıflarına metotların eklenmesi ve istekleri yerine getirmek üzere nesneler arası mesajların belirlenmesi.

Sorumluluklar:

- Bilinmesi gerekenler
  - Kendi özel verileri
  - İlgili diğer nesneler
  - Üzerinde hesap yapabileceği, hesapla elde edebileceği veriler
- Yapılması gerekenler
  - Hesap yapma, nesne yaratma yok etme
  - Başka nesneleri harekete geçirme
  - Başka nesnelerin hareketlerini denetleme

Senaryoların gerçeklenmesi

Sorumlulukları yerine getirmek için metotlar oluşturulur

Bir sorumluluğu erine getirmek için bir metot başka metotlarla işbirliği yapabilir

İlk iterasyonda senaryoların ana akışları gerçekleştirilir

İkinci iterasyonda alternatif akışlar ele alınır ve gerçeklenir.

Büyük senaryolar birkaç iterasyon sürebilir. Daha küçükleri bir iterasyonda bitebilir.

Tasarımın sonunda tersine gidilerek, problem domeninin diyagramı çıkarılır. Bu sayede analiz diyagramlarının son hali de elde edilmiş olur.

Örnek senaryo – Ana Akış

Satışın toplam bedelinin hesaplanması:

Main Success Scenario (or Basic Flow):

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules. *Cashier repeats steps 3-4 until indicates done.*
5. System presents **total with taxes calculated.**

Senaryoya göre satışın toplam bedelinin hesaplanması gerekiyor.

GRASP Tasarım Kalıpları

Genel prensipler ve akla yatkın çözümlerin birleştirilmesiyle oluşturulan repertuar.

Probleme getirdiği çözümü belirten kolay isimler verilir (iletişimi kolaylaştırır).

Kalıplar, tanımlı bir problem için en iyi çözümü sağlar.

GoF kalıpları: 23 adet.

Bu derste 9 adet genel sorumluluk atama prensibi anlatılacaktır.

1-Uzman (Expert)

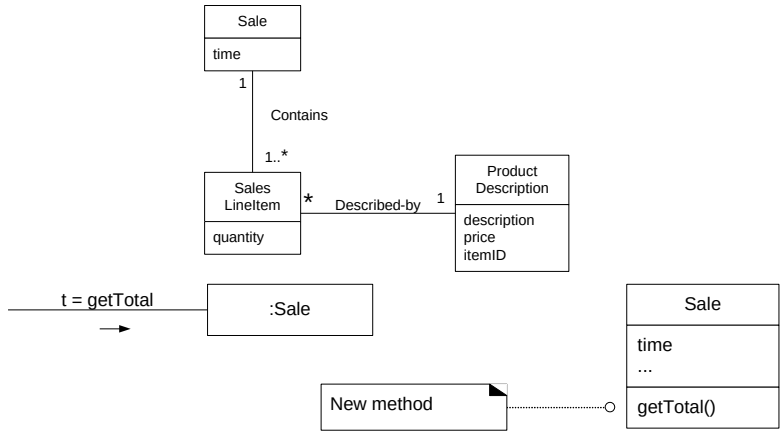
Nesnelere sorumluluk atamanın temel prensibi nedir?

Bir sorumluluğu bilginin uzmanına, onu yerine getirecek veriye sahip olan sınıfa atayın.

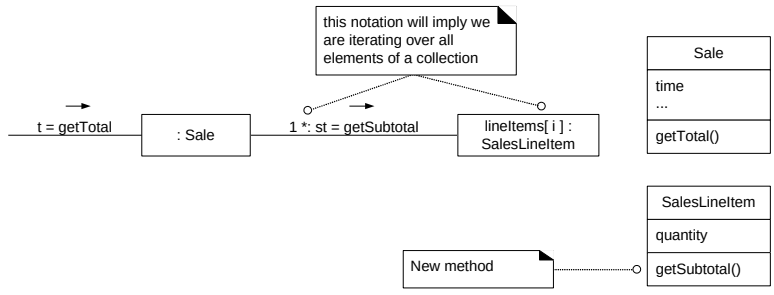
Arama önce yazılım sınıfları arasında yapılır. Henüz bu sınıf yoksa kavramsal sınıflar incelenir. En uygunu, yazılım sınıfı olarak modele alınır. (Low representional gap)

Örnek

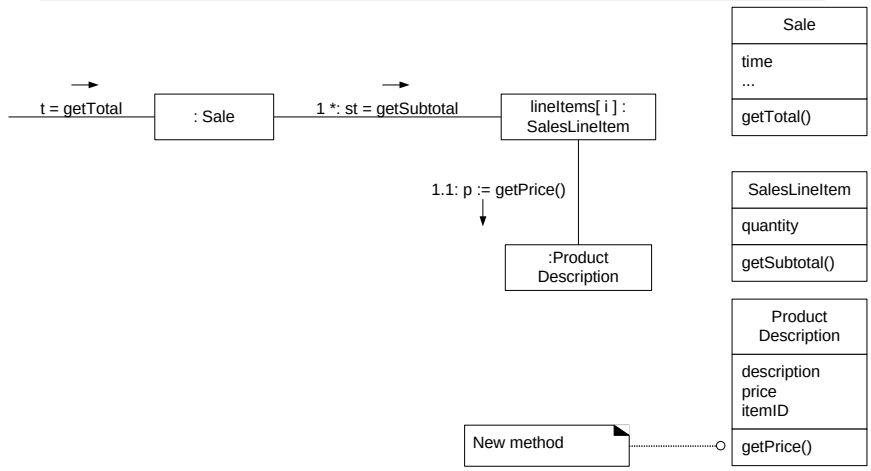
Satışın toplam bedelinin belirlenmesinden kim sorumludur?



Örnek



Örnek



2-Yaratıcı (Creator)

Bir sınıftan nesne yaratma sorumluluğu kimdedir?

B'ye A'dan nesne yaratma sorumululuğunu,

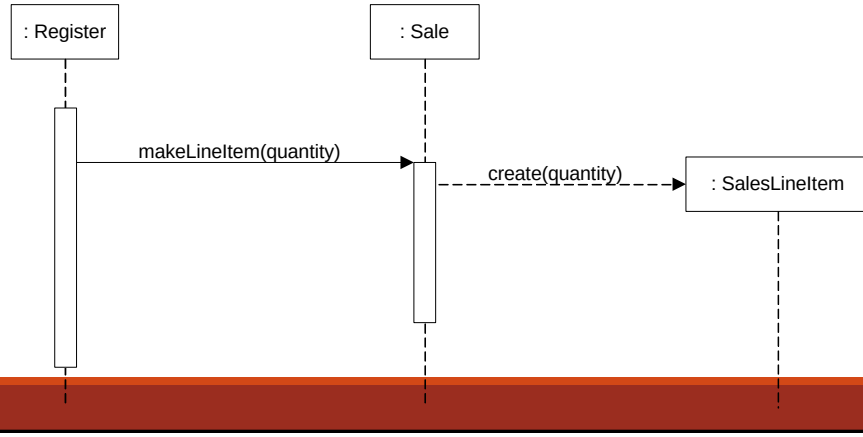
- B, A nesnelerini içeriyorsa
- B, A'ların kaydını tutuyorsa
- B, A'ları kullanıyorsa
- A'nın yaratılması için başlangıç verilerine B sahipse

Atayın.

Örnek

Satış kalemlerini kim yaratacak?

- Satış çok sayıda satış kalemi içeriyor.



3-Az bağımlılık (Low Coupling)

Diğer sınıfların değişimlerinden etkilenmeme, tekrar kullanılabilirlik nasıl sağlanır?

Sorumlulukları, sınıflar arası bağımlılık en az olacak şekilde atayın.

Bağımlılık: Bir sınıfın kendi işleri için başka sınıfları ne kadar kullandığı, onlar hakkında ne kadar bilgi içerdiği

Neden az bağımlılık?

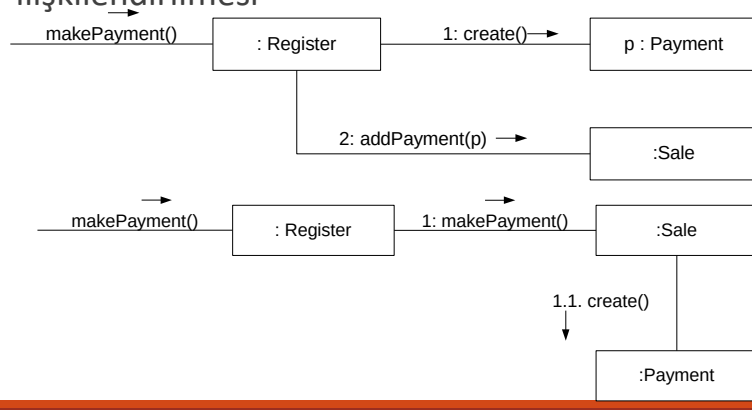
- Bir sınıftaki değişim, diğerlerini de etkiler
- Sınıfları ayrı anlamak zordur
- Yeniden kullanmak bağımlılık yüzünden zordur

Bağımlılık ne zaman oluşur? (A'nın B'ye)

- A'nın B cinsinden üyesi varsa
- A, B'nın metodunu çağırıyorsa
- A'nın metodunda B tipinden parametre varsa
- A'nın metodunda B tipinde değişken varsa
- A, doğrudan veya dolaylı B'den türetilmişse (alt sınıfı)

Örnek

Ödeme nesnesinin yaratılıp satış ile ilişkilendirilmesi



4-İyi Uyum (High Cohesion)

Karmaşıklık nasıl idare edilebilir?

Sorumlulukları, sınıf içinde iyi bir uyum olacak şekilde atayın.

Uyum: Sorumlulukların biribirleriyle ilgili olması ve aynı konuya yoğunlaşmaları

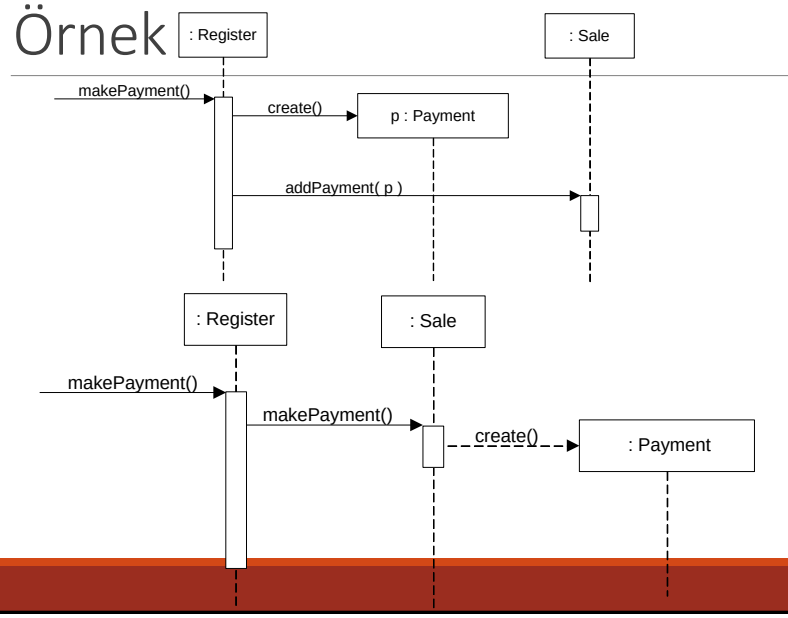
Bir sınıf birbiriyle ilişkisi olmayan işler yapıyorsa, veya çok fazla iş yapıyorsa uyum kötüdür.

- Sınıfın anlaşılması zorlaşır.
- Bakım yapmak zorlaşır.
- Tekrar kullanmak zorlaşır.
- Değişikliklerden çok etkilenir.

Veritabanı işlemleri gibi, bir işi tek bir programcının sorumluluğuna vermek için iyi uyum bozulabilir.

Dağıtık sistemlerde ağ trafiğini azaltmak için iyi uyum bozulabilir.

Örnek



5-Denetçi (Controller)

Sistem olayları ile ilgili işleri yapmakla kim sorumludur?

Sistem olaylarını algılayıp değerlendirme sorumluluğunu aşağıdaki iki seçenekten biri ile oluşturun:

- Tüm sistemi-cihazı veya alt sistemi temsil eden bir sınıf (Görüntü denetçi-facade controller)
- Bir senaryoyu temsil eden denetçi (senaryo ya da oturum denetçisi-session or use case controller)

Sistem olayları: aktörler tarafından üretilen, sistem işlemleri ile ilgili olaylar.

Örneğin, sipariş et butonuna tıklanması.

Olayı algılayıp bazı denetimler yaptıktan sonra olayları işleyecek nesnelere yönlendirirler.

Denetçi nasıl belirlenmeli?

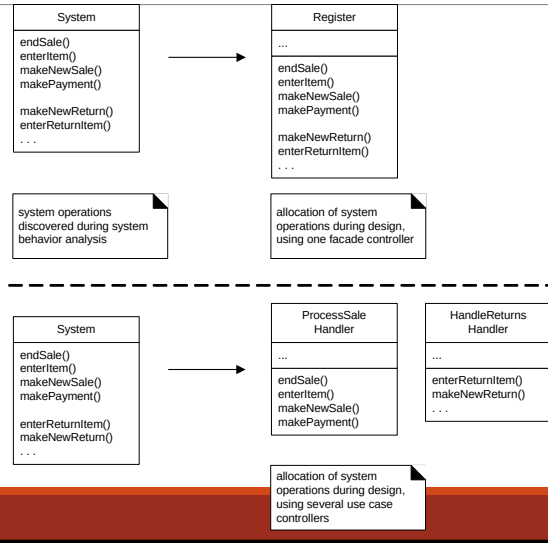
- Olaylar fazla değilse tek bir denetçi uygundur.
- Olay sayısı uyumu bozacak kadar fazlaysa birden fazla denetçi
- Senaryo içinde sistem kontrolleri yapılacaksa senaryo denetçisi uygun

İşleri denetçiler yapmaz. Uygun mesajlarla sorumlu nesnelere aktarırlar.

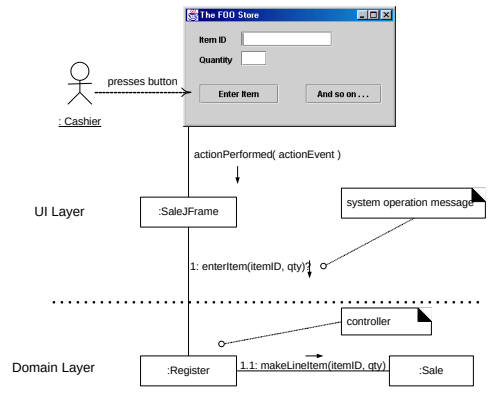
Yararı:

- Arayüzle uygulama katmanı ayrılır. Program arayüzden bağımsız olur.
- Denetçi görevi arayüz nesnesine de verilebilir. Yeniden kullanılabilirlik ve esneklik kaybolur.
- Senaryoda işlemlerin sırası denetim altında tutulur. Örneğin, ödeme mesajının siparişi ver'den önce gelmesi engellenir.

Denetçi Tipleri



Örnek



6-Çok şekillilik (Polymorphism)

Tiplere bağlı alternatifler nasıl ele alınmalı? Sisteme kolay monte edilen (pluggable) yazılım parçaları nasıl gerçeklenir?

Alternatif davranışlar sınıflara bağlı olarak değişiklik gösteriyorsa, bunları çok şekilli (polymorphic) metotlarla gerçekleyin.

Koşullu deyimlerle (if then else) tipi test edip ilgili metodu çağırmak iyi bir yöntem değildir.

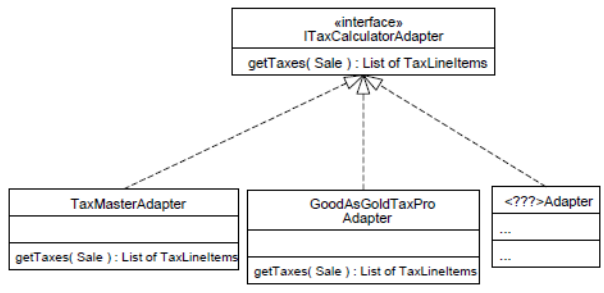
Kolay monte edilebilirlik: Bir sınıfın sistemden çıkarılması, yerine bir başkasının konması ve kullanıcıların bu değişimden etkilenmemesi

Tek bir davranış olsaydı, çok şekilliliğe gerek olmazdı.

Örnek: Vergi hesabı başka programlara yaptırılsın. Farklı programlarla ve ileride alınabilecek başka programlarla çalışabilsin.

Çözüm: Her bir programla ilişkiyi sağlayan arayüz (adapter) sınıfı oluşturulur. Hepsi bir interface'den türetilir. (Hepsinin içinde çok şekilli bir vergiHesapla() metodu bulunur.

Örnek



7-Yapay Sınıf (Pure Fabrication)

Uzman kalıbının çözümü iyi uyum ve az bağımlılıkla çelişiyorsa sorumluluklar hangi sınıfa atanmalıdır?

Bağımlılık azalıyor ve tekrar kullanılabilirlik artıyorsa, birbiriyle uyumlu sorumluluklar gerçekte var olmayan yapay bir sınıfa atanabilir.

NDMT'nin temeli, yazılım unsurlarının gerçek dünyadaki varlıklara benzer tasarlanmasıdır. Bazı durumlarda uyum ve bağımlılık sorunlarına yol açabilir. Bu durumlarda yapay sınıf kullanılır.

Örneğin veritabanı işlemleri yapay bir sınıfta gerçekleştirilir. Bunun başka projelerde kullanılma olasılığı da yüksektir.

Yapay sınıflar gereğinden fazla kullanılmamalıdır. Eski alışkanlıklara sahip bir programcı, gerçek dünyadaki varlıklara ayırmak yerine işlevlere ayırmayı seçebilir.

Çok kullanılırsa gerçek dünyadan uzaklaşılır, anlaşılırlık azalır, bağımlılık artar.

8-Dolaylılık-Arabirim (Indirection)

Kolay değişebilen iki birim arasındaki bağımlılık nasıl azaltılır?

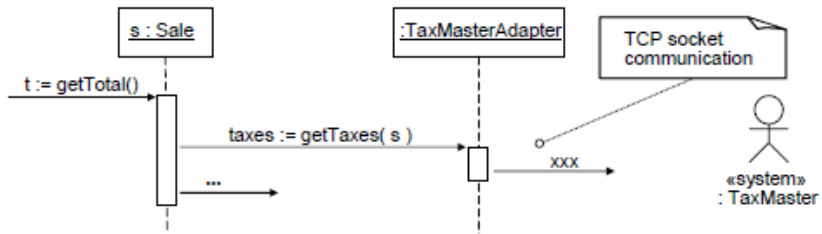
Sorumlulukları bir arabirim nesnesine atayın.

Yapay veritabanı sınıfı, sistemle veritabanı arasında arabirimdir. Örnek: Vergi adaptörleri

Adapter, Facade, Observer gibi kalıplar bu kalıbın özel halleridir.

Örnek

Vergi hesaplama programları ile POS sistemi arasına konan adaptörler: dış değişimlerden etkilenmeyi önler



9-Değişimlerden Korunma (Protected Variation)

Nesneler, değişim ve kararsızlıklar birbirini en az etkileyecek şekilde nasıl tasarlanmalıdır?

Değişime açık, kararsız noktalar belirlenir ve sorumluluklar bunların etrafında kararlı arayüz oluşturacak şekilde atanır.

Örnek: çok şekilli vergi adaptörleri

Birçok prensip ve kalıbın temelidir.

- Encapsulation (Özellik ve davranışların bir bütün oluşturması)
- Veri gizleme (data hiding) (security değil)
- Çok şekillilik (polymorphism) (hedef compile time değil, runtime belli olsun. Mesaj giden birim değişse bile etkilenmeyelim)

“Yabancılarla Konuşma” Prensibi

Don’t Talk To Strangers.

Bir nesne ancak tanıdık (sınırlı sayıda) bir hedefe mesaj göndermelidir.
- Kendisi (this)
- Metodun parametresi olan nesne
- Nesnenin üyesi (özelliği) olan nesne
- Nesnenin üyesi olan bir grubun (liste, vektör vs) elemanı olan nesne
- Metodun içinde yaratılan nesne

Dolaylı (tanıdığın tanıdığı) nesneler ise yabancı nesnelerdir.

Metot içinden yabancı nesnelere mesaj gönderilmesi, bağımlılık yaratır. Görülmesi zor, olmaması gereken bir bağımlılıktır. Bu yüzden tercih edilmez.

Örnek:Money amount= sale.getPayment().getTenderedAmount(); kötü

Money amount= sale.getTenderedAmountOfPayment(); iyi

Gerçekten gerekliyse, tanıdık nesneye sorumluluk olarak atamak gerekir.

Senaryoların gerçeklenmesi

Anlatılan kalıplara uygun olarak
- Belirlenen tüm senaryolar
- Tüm sözleşmeler

Gerçekleştirilir.

Kavramsal sınıflardan yola çıkılarak yazılım sınıfları oluşturulur.

Sorumluluklar, metotlarla gerçekleştirilir.

Her bir işbirliği, tasarım sınıf diyagramında bir bağlantı olarak gösterilir.

Başlangıç İşlemleri

Sistem ilk çalışmaya başladığında yapılacaklar ayrı bir senaryo grubu olarak yazılabilir.

Bu işleri tasarımın en son aşamasında belirlemek uygundur.

Bir başlangıç nesnesi (initial domain object) belirlenir.

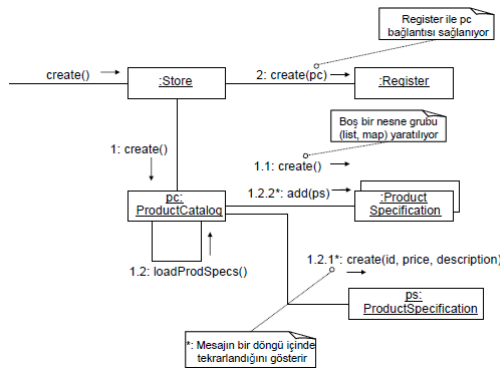
Program çalışmaya başlayınca bu nesne yaratılır.

Doğrudan içerdiği diğer nesneleri yaratma ve aralarındaki bağlantıyı sağlama sorumluluğuna sahiptir.

Tüm programın çalışmasını denetleyen temel bir nesne gibidir. Ancak bu denetim ana programda veya arayüz nesnesinde de olabilir.

Başlangıçta arayüz nesnelerine denetçinin referansı da gönderilir.

Örnek



Görünürlük (Visibility)

B, A'ya görünür ise, A nesnesi B'ye erişebilir. A nesnesi B'ye, ancak B A'ya görünür olduğunda mesaj gönderebilir.

Nitelik Görünürlüğü (Attribute): B, A'nın bir üyesidir. Nesnenin yaşam süresince devam eder.

Parametre Görünürlüğü (Parameter): B, A'nın bir metodunun bir parametresidir. Metot süresince geçerlidir. Bu yüzden bağımlılık daha azdır.

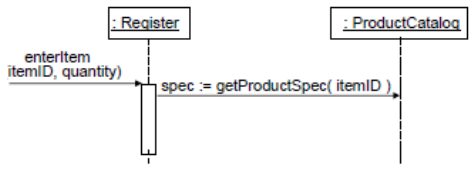
Eğer parametreyi yerel değişkenine atarsa yerel, bir özelliğine atarsa özellik görünürlüğüne dönüşür.

Yerel Görünürlük (Local): B, A'nın metodunda yerel bir değişkendir.

Genel Görünürlük (Global): B, A'nın global uzayındadır.

Nitelik Görünürlüğü

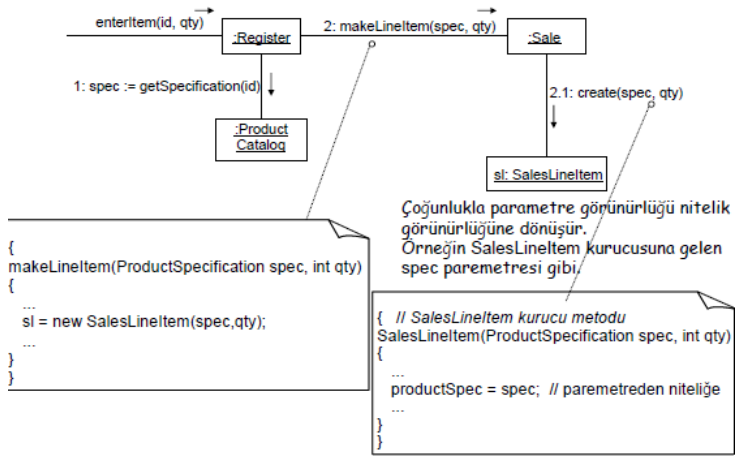
Nitelik Görünürlüğü: Bir nesne diğerinin üyesi olduğundan, görünürlük nesnenin yaşam süresince devam eder.




```
class Register
{
  .....
   private ProductCatalog catalog;
  .....
} // atama Kurucu metodta
```

```
public void enterItem(itemId,qty)
{
  .....
   spec= catalog.getProductSpec(itemID);
  .....
}
```

Parametre Görünürlüğü



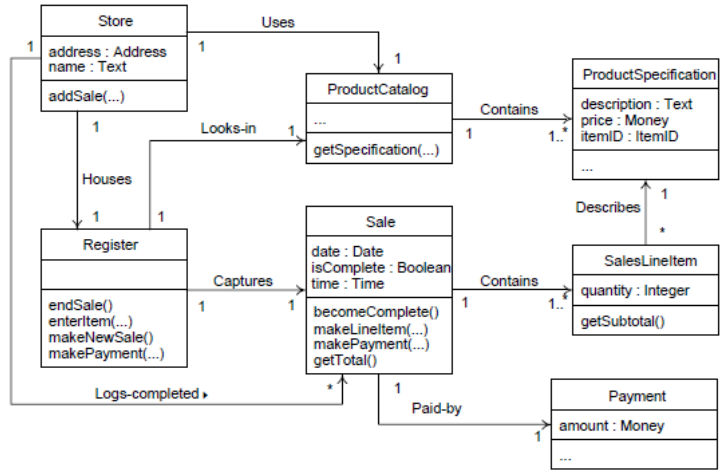
Tasarım Sınıf Diyagramı

Senaryoları gerçeklerken çizilen etkileşim diyagramlarına paralel olarak, yazılım sınıflarından oluşan diyagram(lar) da çizilir.

Yazılım sınıfları, özellikleri ve tipleri, metotların parametreleri ve erişim hakları büyük ölçüde belirtilir.

İlişkiler ve bağımlılıklar yönlü olarak gösterilir. (Yeni ilişkiler de bulunabilir)

Örnek



Yazılım Sınıfları Arası Bağlantılar (Navigability)

Çoğunlukla özellik görünürlüğünün sonucudur.

B, A'nın üyesidir ve A, B'ye mesaj gönderir.

Sınıf diyagramında bunu ifade etmek için A'dan B'ye bir ok çizilir.

Uygulama domenindeki bağlantılar, kavramsal sınıflar arasında gerçek dünyada ilişki olup olmadığını gösterir.

Buradaki bağlantılar, görünürlük ilişkisini gösterir.

Bağlantılar, işbirliği diyagramlarının incelenmesiyle bulunabilir.

Bağımlılık İlişkisi (Dependency)

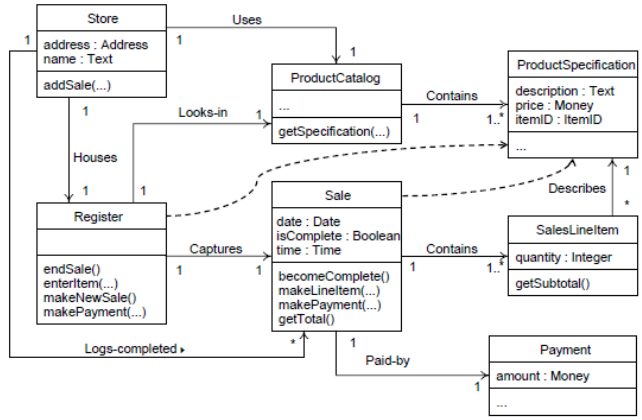
Bağımlılık: Bir birimin başka bir birimle ilgili bilgilere sahip olması gerekliliği

Özellik görünürlüğü (navigability) dışındaki görünürlükleri ifade eder.

Kesikli ok ile ifade edilir.

Örnek: Slayt 65'de Kasa, Katalog'a özellikAl mesajını gönderdiğinde ürünTanım cinsinden yanıt alır. Bu yanıtı da ürünGir metodunda yerel değişken olarak saklar. Bu yüzden kasa, ürünTanım'a bağımlıdır. (yerel görünürlük)

Örnek



Kodlama

Sınıf tanımları dcd'den yararlanılarak oluşturulur.

Metotlar etkileşim diyagramlarından yararlanılarak oluşturulur.

Sınıfları gerçeklemeye en az bağımlı sınıftan başlanır, en çok bağımlıya doğru gidilir.

Her sınıf mutlaka birim teste tabi tutulur.

Test programı daha önce yazılabilir. Bu program sınıflardan nesneler yaratır, tüm mesajları sonuçlarıyla kontrol eder.

Gelecek Ders

Veri Modelleme

Veri Yapısı ve Veri Tabanı Tasarımı

Bu yansı ders notlarının düzeni için boş bırakılmıştır.