

BLM2732 Gr.2

Sistem Analizi ve Tasarımı

Yrd.Doç.Dr. Göksel Biricik

goksel@ce.yildiz.edu.tr

1.Ders

Ders Planı-1

Hafta	Tarih	İçerik
1	13 Şubat 2018	Giriş, Ders Planı, Genel Bilgiler, Sistem Tanımı
2	20 Şubat 2018	Sistem Tipleri, Tarafları, Sistem Geliştirme Süreci
3	27 Şubat 2018	Ön İnceleme – Fizibilite Çalışmaları
4	06 Mart 2018	Fizibilite Laboratuvar Uygulaması (MS Project)
5	13 Mart 2018	Sistem Analizi – Veri Toplama, İş Analizi, Kullanım Senaryoları

Ders Planı-2

Hafta	Tarih	İçerik
6	20 Mart 2018	Sistem Analizi – Veri Modelleme, Fonksiyonel Çözümleme
7	27 Mart 2018	Sistem Analizi Laboratuvar Uygulaması (VAD+Kavramsal Sınıf D.)
8	03 Nisan 2018	Sistem Tasarımı – Girdi, Çıktı, Veri Yapısı, Arabirim Tasarımları
9	10 Nisan 2018	Ara Sınav
10	17 Nisan 2018	Sistem Tasarımı Laboratuvar Uygulaması (VAD→Yapı D. + Ardışıl D.)

Ders Planı-3

Hafta	Tarih	İçerik
11	24 Nisan 2018	Veritabanı Tasarımı, Kodlama, Test Süreçleri
12	01 Mayıs 2018	Emek ve Dayanışma Günü
13	08 Mayıs 2018	Yeni Sisteme Geçiş, Bakım, Destek Süreçleri
14	15 Mayıs 2018	Proje Sunumları
15	22 Mayıs 2018	Proje Sunumları, Telafi Sınavı

Değerlendirme Kriterleri

Tipi	Ağırlığı	İçerik	Tarih
Vize	% 25	Derste Anlatılan Konuların Değerlendirilmesi	02/06 Nisan 2018 Haftası
Proje	% 20	Seçilen Bir Konuda Ön İnceleme, Fizibilite, Analiz, Tasarım, Kodlama, Veritabanı Tasarımı ve Destek Aktiviteler de Dahil Olacak Şekilde Bir Bilgi Sisteminin Gerçekleştirilmesi	Öneriler: 27 Şubat 2018 Değerlendirme Sonuçları: 02 Mart 2018 Rapor Teslimi: 08 Mayıs 2018
Laboratuvar	% 15	Laboratuvar Çalışmalarındaki Görev Performansı	06 Mart – 27 Mart – 17 Nisan 2018
Final	%40	Dersteki Konuların Tümünün Genel Değerlendirilmesi	28 Mayıs - 08 Haziran 2018 Arası

Kaynaklar

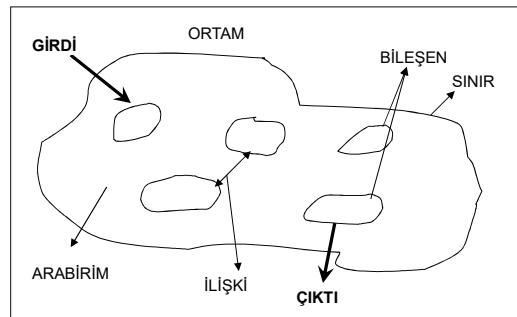
- **Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı 3.Basım**, O.Kalıpsız, A.Buharalı, G.Biricik, Papatya Yayıncılık.
- **System Analysis and Design in a Changing World Sixth Edition**, J.W.Satzinger, Cengage Learning.

Giriş

- İnsan, hammadde kaynakları önemli
- Günümüzde, bilgi en önemli kaynak
- Bilginin doğru kullanımı, rekabet gücünü arttırmır
 - Bilgi üretimi, dağıtımı, işlenmesi, güvenliği, depolanması önemli
 - Internet, elle üretilen verinin çok daha fazlasını üretiyor
 - Verilerin organizasyonu, kullanım maliyetleri arttı
- Çözüm? → Bilgi Sistemleri

Sistem

- Belli bir amacı gerçekleştirmek için
- Bir arada çalışan, birbiriyile ilişkili olan parçalardan oluşan
- Girdi ve çıktıları olan
- Sınırları belirlenmiş bir bütür

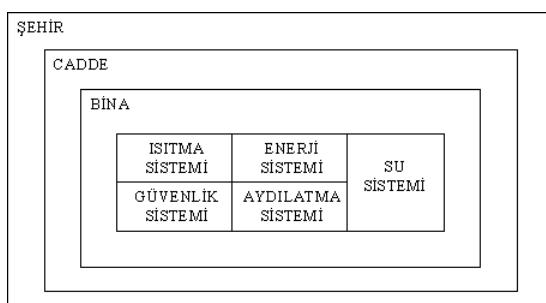


Sistemin Temel Özellikleri

- **Bileşenler:** Sistemi oluşturan parçalar
- **Değişkenler:** Değişik değer alan özellikler
- **Parametreler:** Sabit değerler
- **İlişkiler:** İlk üç madde arasındaki bağlantılar
- **Sınır:** Sistem için ortam ayrıacı
- **Arabirim:** Ortam veya alt sistemlerle karşılaşma noktası
- **Kısıtlar:** Değişken değerleri ve kaynak tahsis sınırlamaları
- **Ölçüt:** Hedef amaçların değerlendirme standarı
- **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



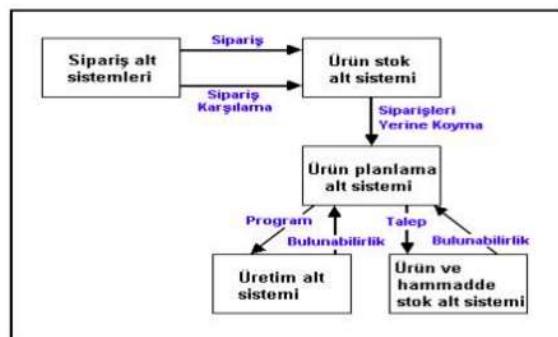
Bilgi Sistemi

- 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, tasarılayanlar, 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 disiplinlerarasıdır
 - Bir dalda 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 üzerinde sistemler vardır
- Sistem ortama göre düzenlenmelidir
 - Entropi ortamla ilişkiye 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

Gelecek Ders

- Bilgi Sistemi Tipleri
- Bilgi Sistemi Tarafları
- Bilgi Sistemi Geliştirme Süreci

Sistem Analizi ve Tasarımı

2.Ders

Göksel Biricik

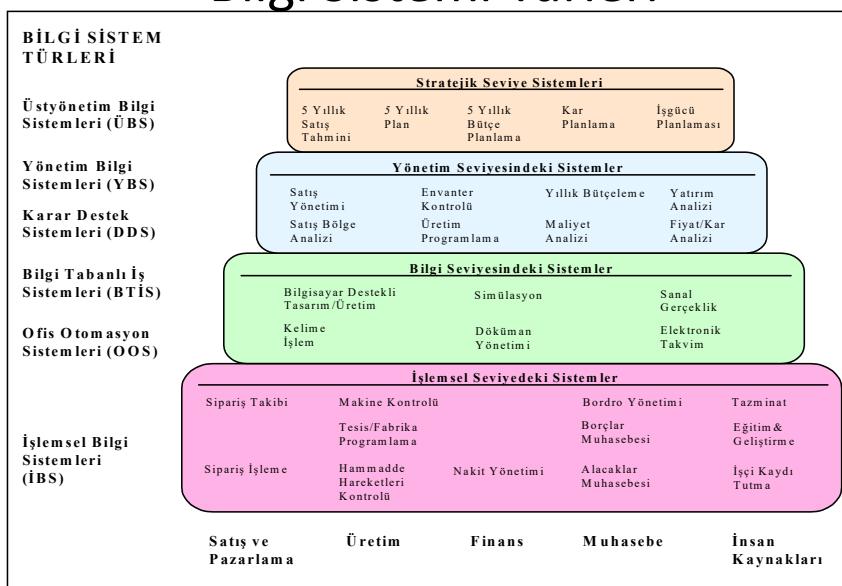
Bu Derste

- Bilgi Sistemi Türleri
- Bilgi Sistemi Tarafları
- Bilgi Sistemi Geliştirme Süreci

Bilgi Sistemleri

- Amaç: Problemlerin analizi, yeni ürün oluşturulması, 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



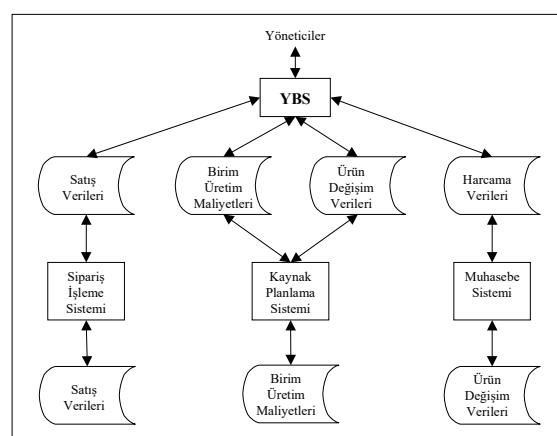
İşlemsel Bilgi Sistemleri

- Başka sistemlere giriş ve temel oluşturur
- İşlenmiş veriyi:
 - Sınıflar
 - Saklar
 - Bakımını Yapar
 - Değiştirir
 - Güncelleştirir
 - Geri çağrıır
- Örnekler
 - Satış izleme
 - Envanter düzenleme
 - Fatura hazırlama
 - Kartlı geçiş



Yönetim Bilgi Sistemleri

- İşlem boyutu: Rapor, analiz, KDS
 - Uygulama tabanı
 - VTYS
 - VT
- Yönetim boyutu
 - Alt yönetim için işlemsel kontrol ve güncelliliği,
 - Orta yönetim için kontrollü kaynak paylaşımı
 - Üst yönetim için stratejik planlama ve hedefler
- İşlevsel boyut
 - Kurumsal işlevler için birimler arası bilgi farkları

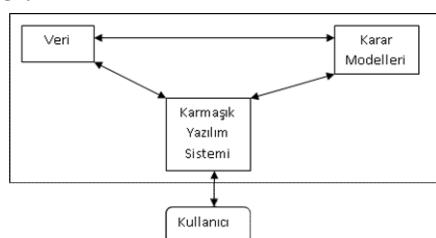


Ofis Otomasyon Sistemleri

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

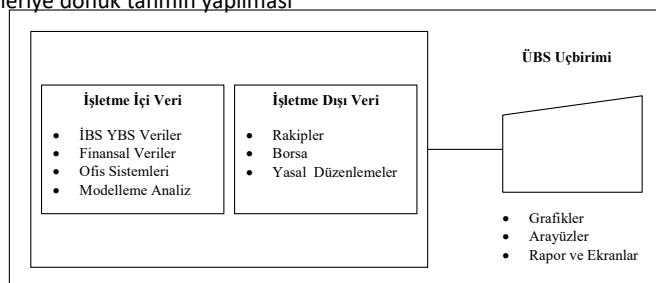
Karar Destek Sistemleri

- 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



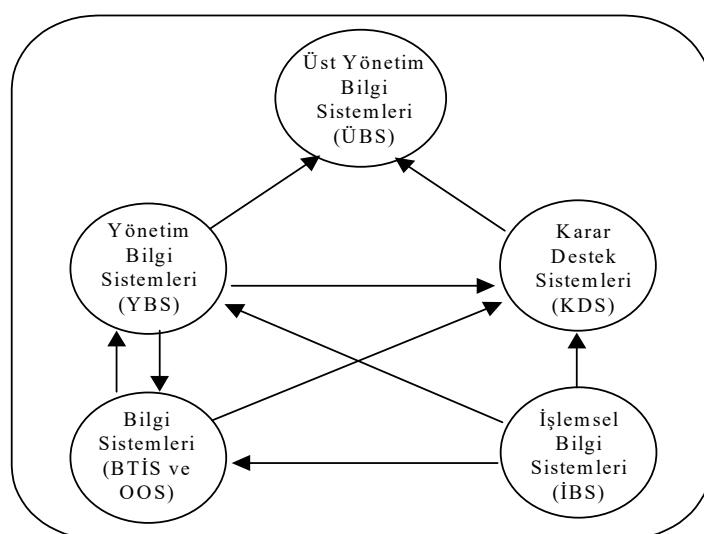
Bilgi Tabanlı İş Sistemleri

- İnsanların yerini almaz
- Çalışma – araştırma sonuçlarını bilgi sistemiyle birleştirir
- CAD/CAM sistemleri
- MR sistemleri

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; analistik 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
- Programcı
- Destek Personeli

Kullanıcı

- Sistemin varoluş sebebi, en önemli parça
- MÜŞTERİ
- Sistemden beklenileri TAM belirlemeli
- Sistemin nasıl başarıya ulaşacağı tespit edilmeli
- Teknik değil, işlev ve kullanım önemli

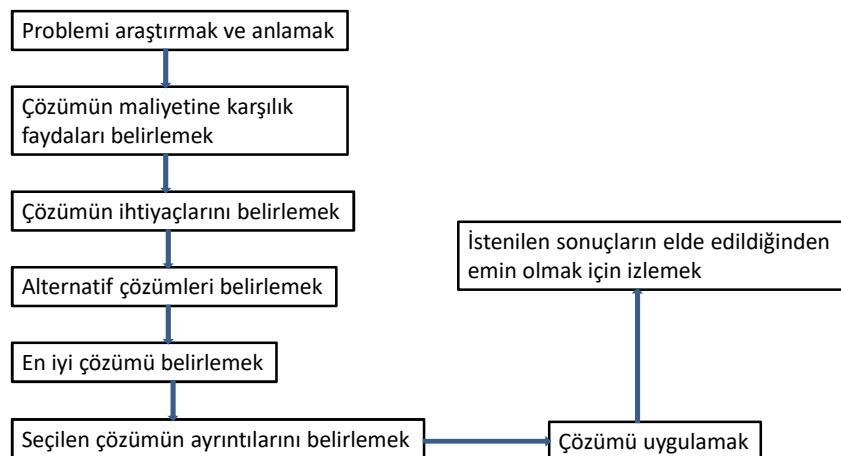
Yönetici

- 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

- Dersin odak noktası
- Projenin kilit üyesi
- Hem işletme yönetimi, hem bilgi sistemi bilgisi olmalı
- İşletmenin sorunları, ihtiyaçları, iş akışlarını belirlemeli
- Bilgi sistemi ile çözümünü sağlatmalı, uygulamaya karar vermelii
- Problemi görüp anlamalı, çözüm için bilgi sistemini metodolojiye uygun olarak oluşturmalı

Bir Analistin Problem Çözümü



Analistin Beceri ve Görevleri

- Analistik 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 protokoller 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

Programcı

- 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 iletişim
 - 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	Programı yazma, yükleme ve test	Çalışan sistem ve dokümantasyon
Bakım	Sistemin bakımını yaparak desteklemek	Çalışan Sistem

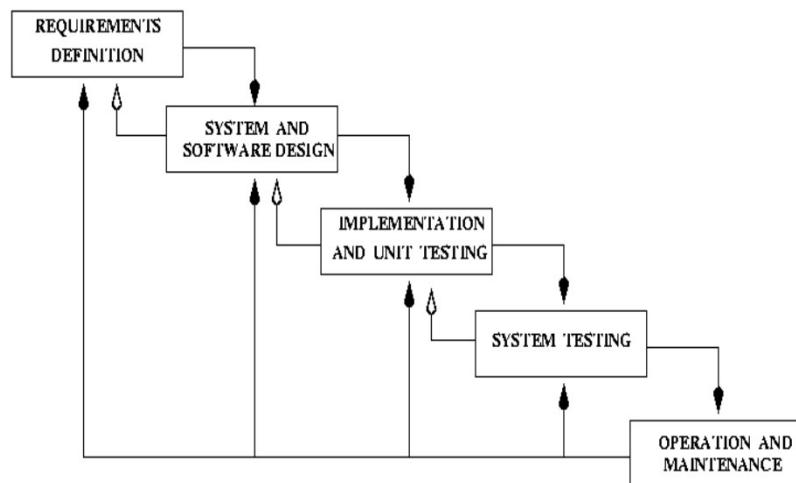
Sistem Geliştirme Yaşam Döngüsü



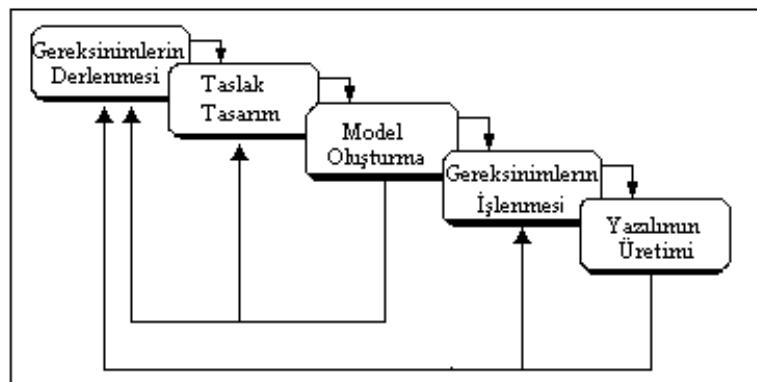
Süreç Modelleri

- Klasik Süreç (Waterfall)
- Model Oluşturma
- Evrimsel Süreçler
 - Artımlı Model
 - Spiral Model
- RUP Modeli
- Aykırı Programlama

Klasik Süreç - Waterfall

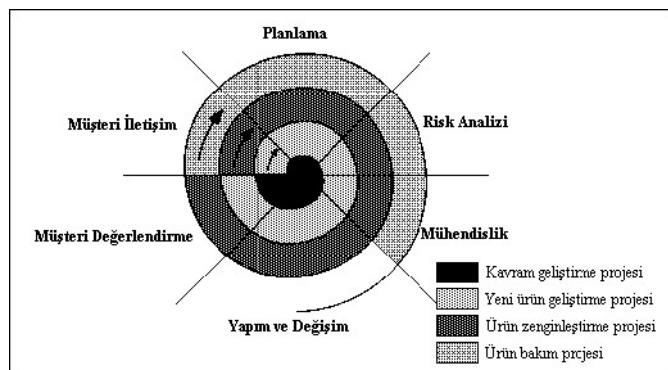


Model (Prototip) oluşturma



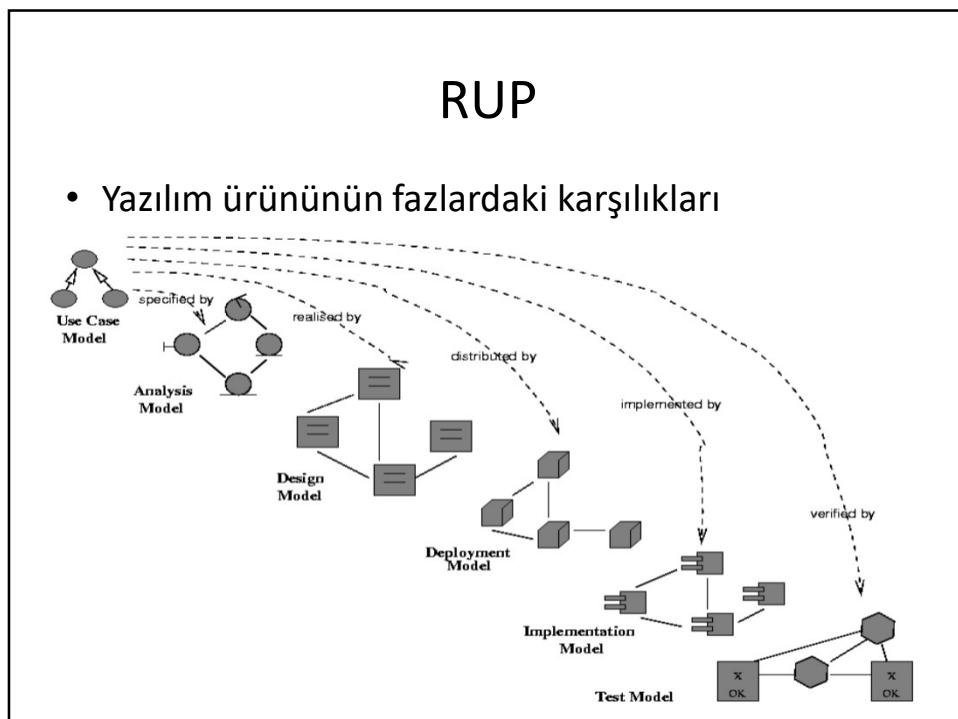
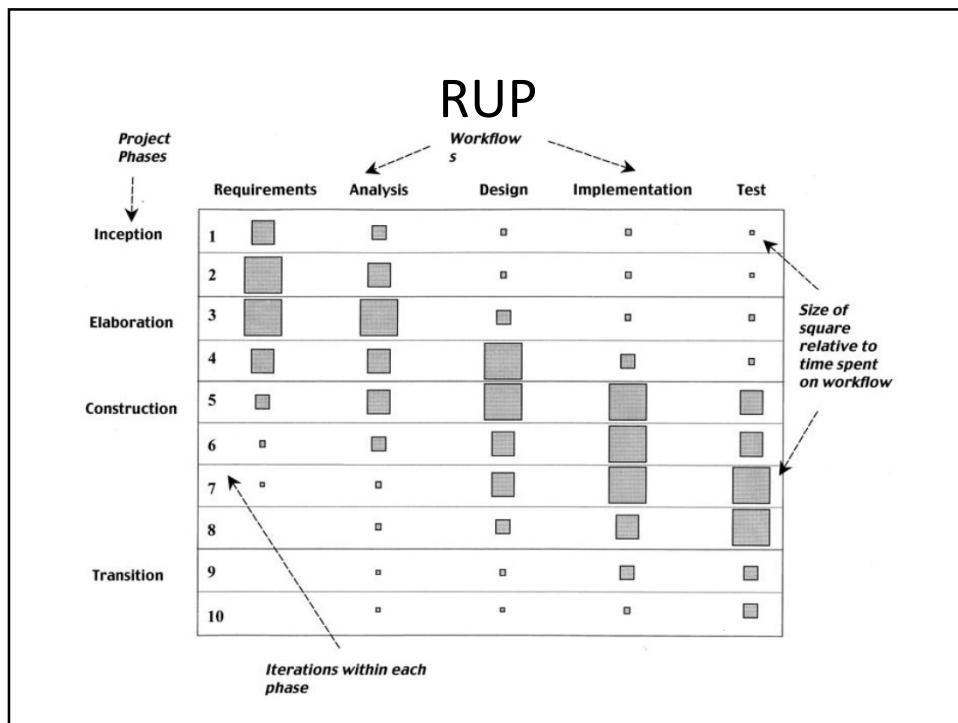
Evrimsel Süreçler

- Artımlı: İteratif olarak waterfall uygulanır
- Spiral: Prototipten versiyona doğru evrim ilerler



Rational Unified Process (RUP)

- Kullanım-senaryosu güdümlü
- Mimari yapı merkezli
- Artımlı ve iteratif



Aykırı Programlama

- 4 temel değer
 - İletişim
 - Basitlik
 - Geri besleme
 - Cesaret
- 12 Pratik
 - Sistem metaforu
 - Ekip üyesi müşteriler
 - 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
- İnceleyin

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

3.Ders

Göksel Biricik

Bu Derste

- Ön İnceleme
- Fizibilite

Ön İnceleme

- Fizibilitenin ilk aşaması
- Projenin olabilirliği belirlenir
 - Projeye(yeni sisteme) gerçekten 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

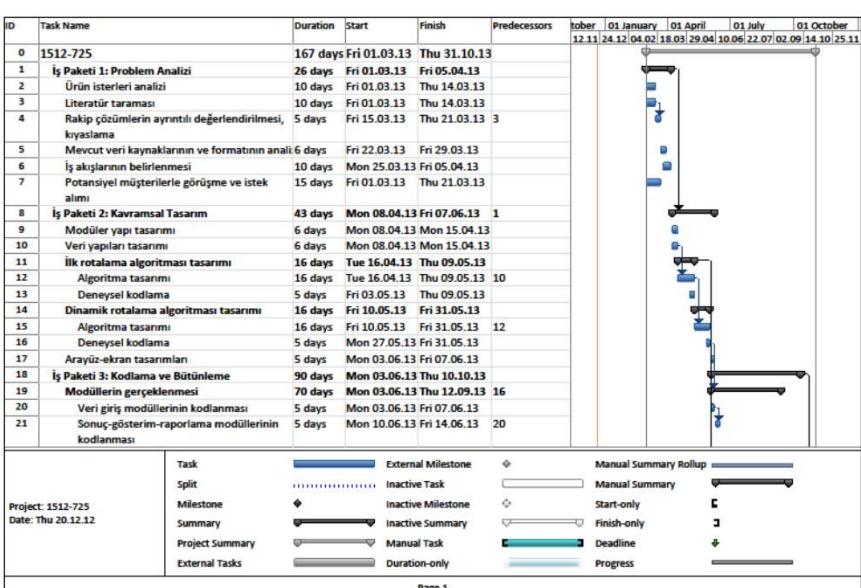
Zaman Fizibilitesi

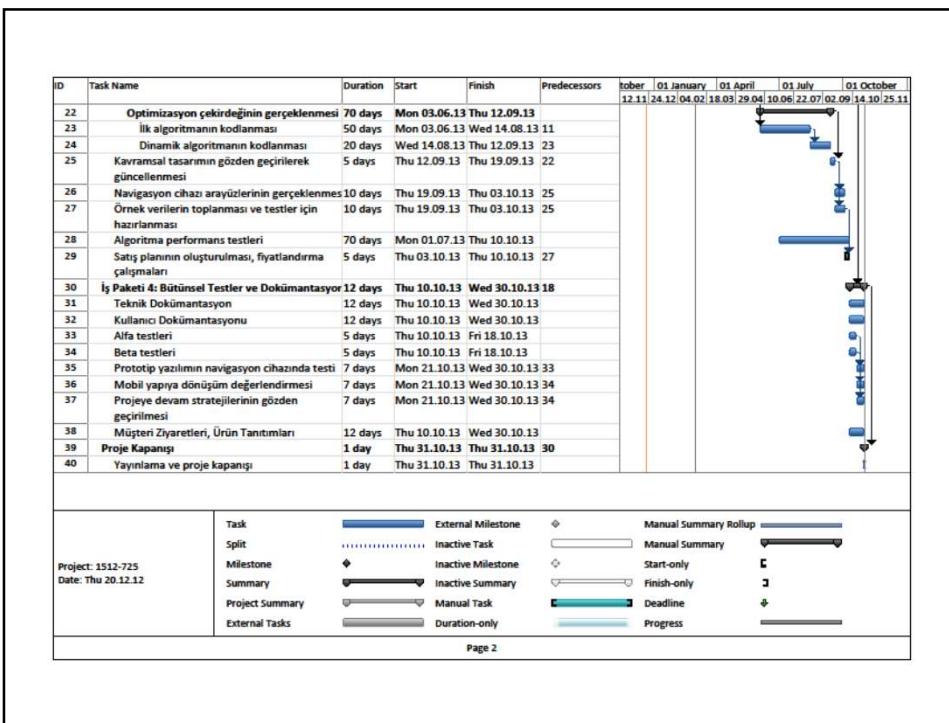
- Belirlenen zaman içinde projenin nasıl tamamlanacağı belirlenir
- Gantt ve PERT teknikleri kullanılır

Gantt

- Problem tanımından işletme 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



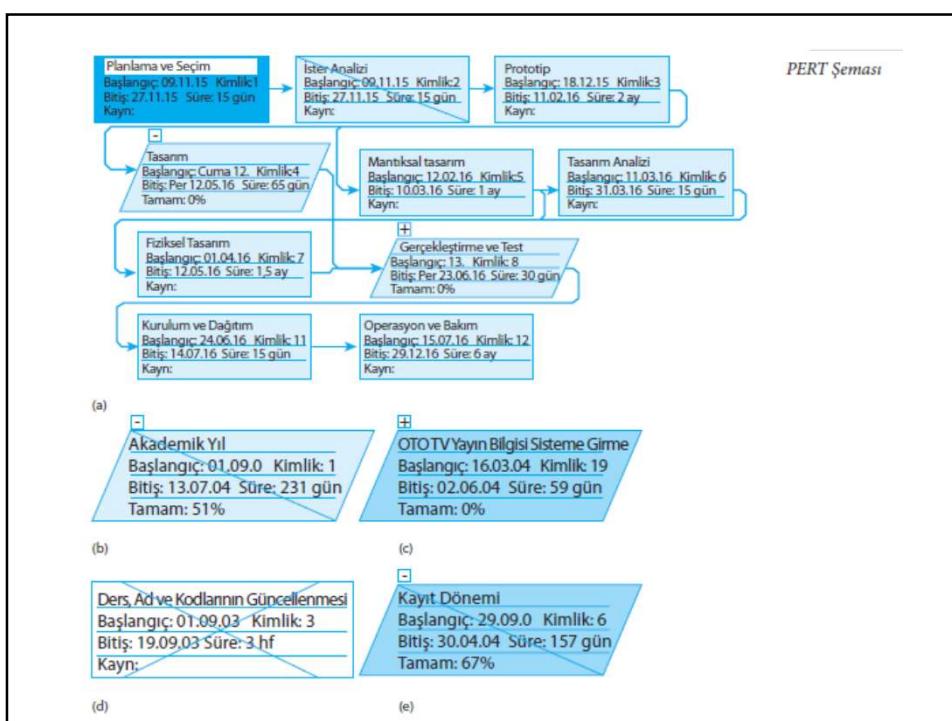
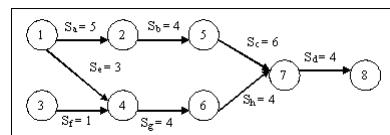


PERT

- Program Evaluation and Review Techniques
- Zaman – araç – maliyet – kontrol gereklerini cevaplar
- Sistemin olmasını sağlayan çok fazla faaliyet arasındaki ilişki ve tarih sayesinde ağ şeklinde düzene konulmasını sağlar
- Olayların mantıksal bir süre izleyeceği, bir olaya ancak önceki işlemler bitince ulaşacağı kabul edilir

PERT

- Süre tahminleri:
 $S_b = (S_i + 4 \cdot S_e + S_h) / 6$
 - 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 değerlendirdir



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 oalrak da görebilirsınız.

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 (ışgücü, 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 indirmek için iskonto oranı kullanılır
 - Farklı yöntemleri mevcuttur
 - En pratigi: Piyasadaki ağırlıklı sektörlerin uyguladığı faiz oranlarını ortalaması
- Q yıl, i iskonto oranı, V indirgenmiş nakit akışı toplamı ise,
- $$V = Q_0 + \frac{Q_1}{1+i} + \frac{Q_2}{(1+i)^2} + \cdots + \frac{Q_n}{(1+i)^n}$$

İç Verim Oranı Yöntemi

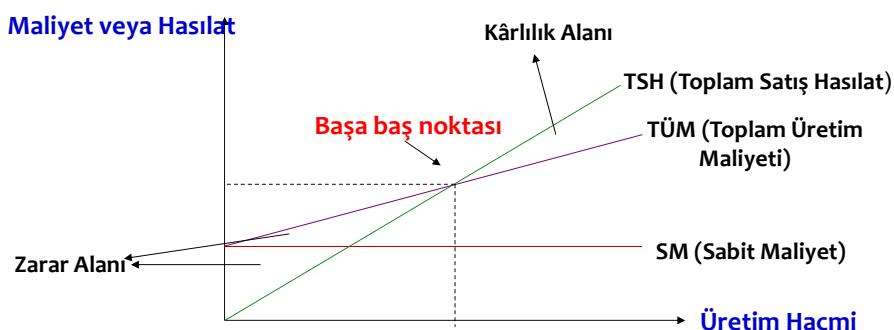
- Paranın zaman değeri ve yatırımın ekonomik ömrü dikkate alınır
- İVO: Nakit girişleri ile yatırım maliyetini eşitleyen iskonto oranı
- T: Yatırımın عمر yılı, x: toplam maliyet, R_t: t yılındaki net getiri, d: iskonto oranı için

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

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 sora 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

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 Sisteme hangi işlemlerin gerçekleştirileceği			
Coğrafik Yapı İşlemlerin ve verilerin sisteme nasıl dağıtıldığı			

Fizibilite Matrisi

- Yapılan çalışmalar değerlendirilir
- Yerde ağırlıklarla puanlandırma yapılır

	Önerilen Sistem 1	Önerilen Sistem 2	Önerilen Sistem 3
Teknik Fizibilite			
Ekonomik Fizibilite			
Zaman Fizibilitesi			
Sosyal Fizibilite			
Yönetim Fizibilitesi			
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 elemanya ya da mevcut elemanların bu konuda eğitime ihtiyaç var. Puan : 50	Sayısız yeterli olmayan mevcut teknik personelin sadece Powerbuilder tecrübesi olmasına rağmen proje yöneticisi, sistemi MS Visual Basic'e çevirmenin basit olduğunu ve VB bilen elemen bulmanın Powerbuilder'e göre daha kolay ve ucuz olduğunu 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 getirişi 210.000 \$'dır Puan : 60	Yaklaşık 418.040 \$'a mal olan ve geri dönüşümü 3,5 yıl olan sistemin net getirişi 306.748 \$'dır Puan : 85	Yaklaşık 400.000 \$'a mal olan ve geri dönüşümü 3,3 yıl olan sistemin net getirişi 325.500 \$'dır Puan : 90
Zaman Fizibilitesi	%10	3 aydan az Puan : 95	9 – 12 ay Puan : 80	9 ay Puan : 85
Operasyonel Fizibilite	%30	Sadece servis üyelerinin ihtiyaçlarını karşılamakta ve var olan iş süreçlerini yazılıma göre 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

- Fizibilite Laboratuvar Çalışması
- Sonraki hafta:
- Gereksinim Analizi
- İş Analizi
- Kullanım Senaryoları
- Bilgi Toplama Yöntemleri

Sistem Analizi ve Tasarımı

4.Ders

Göksel Biricik

Bu Derste

- Gereksinim Analizi
- İş Analizi
- Kullanım Senaryoları
- Bilgi Toplama Yöntemleri

Gereksinim Analizi

- Tanım: Problem nedir? Çözümleri nelerdir? Sistem nasıl Çalışır? Sorularını cevaplamak
- Amaç: En uygun çözümü bulmak için ana öğeler ve işlevler ayrıntılarıyla tanımlanır
 - Gelecekteki hedefler detaylandırılır
 - Bilgi kaynakları ve ihtiyaçlar belirlenir
- Çıktılar:
 - Kullanıcının tüm ihtiyaçları
 - Varolan sistemin durumu
 - Seçilen en uygun çözümün değerlendirilmesi
 - Varolan sistemin nasıl iyileştirileceği önerileri

Gereksinim Analizi

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

Varolan Sistemin Analizi

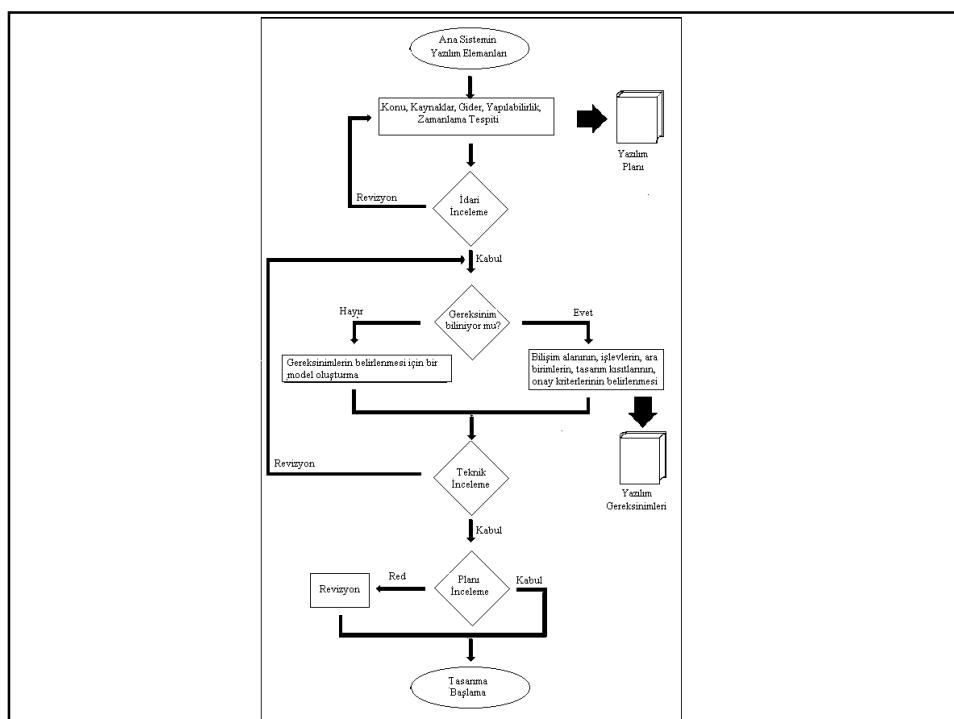
- Varolan sistemin nasıl çalıştığı ve maliyetinin anlaşılması çok önemli
- Sistem analisti varolan sisteme tamamen hakim olmalı
 - Varolan bilgi kaynakları
 - Kullanılan donanım-yazılım
 - İşlemlerde kullanılan bilgi ve miktarı
 - İç-dış bilgi etkileşimi
 - Sistem döngüsü ve süresi
 - Arşiv bilgisi ve 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



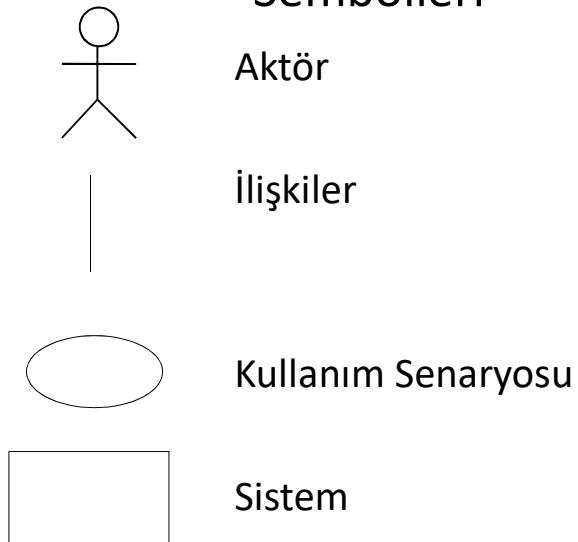
İdari İncelemede Önemli Noktalar

- İhtiyaçların, işletmenin genel stratejileriyle uyumlu olduğu sürekli kontrol edilmelidir
- Karar verirken deneyim, bilgi ve öngörü birlikte önemlidir
- Rekabet varsa daha iyi olan sistem tercih edilecektir
- Yeni sistemin sağlayacağı finansal fayda ispatlanmalıdır
- Organizasyondaki değişiklik beceri ve iş yükünü etkiler, değişiklik olmasa da geliştirme sürecinde kaynak ihtiyacı artabilir
- Hızlı/ayrıntılı raporlama esnek-verimli yönetim için gerekir
- Bilgi akışını iyileştirme ve güncellemeler kullanıcıların daha bilgili olmasını sağlar
- Yeni sistem var olan sisteme göre daha iyi ve hızlı olmalıdır
- Teknoloji bilginin hacmiyle uyumlu olmalıdır
- Kullanıcı dostu olmayan arabirimler ve bilginin yanlış tutulması başarısızlık getirir

Kullanım Senaryosu Modellemesi

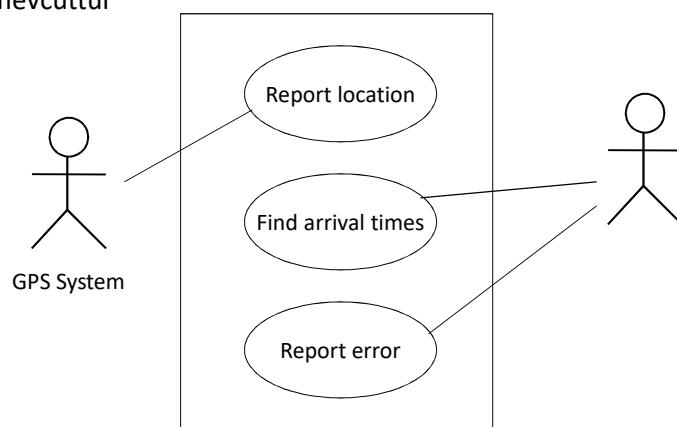
- Kullanım Senaryosu Diyagramları
- Kullanıcının bakış açısından sistemin fonksiyonel çözümlemesini tanımlar
- Sistemin ihtiyaçlarını yerine getirmek için gereken davranışlar için kullanılır
- Aktörler: Kullanıcılar, dış sistemler, fiziksel çevre etmenleri, ...
- Senaryolar: Bir olay akışı şeklinde sistemce sağlanan fonksiyonellik
 - Ön koşullar ve son koşulları, senaryo, birincil ve ilgili aktörleri olmalıdır.

Kullanım Senaryosu Modellemesi Sembollerı



Örnek Kullanım Senaryosu Diyagramı

- Aktörler insan olmak zorunda değildir
- Aktörler ve sistem etkileşimini gösterir
- 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 dışında, neler içinde olmalı?

Kullanım Senaryolarında Yer Alan Bilgiler

- Önsöz
- Birincil Aktör
- İlgililer ve Beklentileri: Sistemin çalışmasından etkilenen ve beklenenleri 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 beklenenleri)
- 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 belki olmalı. Bunlarla tüm amaçlar sağlanmış olur.
- Sıra Dışı Durumlar: Hata olunca yapılacaklar.
- Özel istekler: istenen hız, güvenlik vs gibi kalite kısıtları.
- Teknolojik beklenenler: dil-platform vs.

Senaryolar Arası İlişkiler

- **İçerme:** Birçok senaryo grubunda kullanılan başka bir senaryo grubu. Alt programa dallanıp geri dönmek gibi.
- **Genişletme:** Belirli koşulda ana senaryodan ayrılma noktasından sonra çalışan senaryo
- **Özelleştirme:** Daha genel senaryodan özel senaryolar türetilir (ödeme → nakitle, kartla, çekle, ...)

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.

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
 - iş ortamı gözlenmesi
- Kişisel Görüşme Yöntemi (Toplantı, Mülakat, Ortak Gereksinim Planlaması)
 - Yapılandırılmış görüşme
 - Yapılandırılmış görüşme
- Anket Yöntemi
 - Açık uçlu sorular
 - Kapalı uçlu sorular

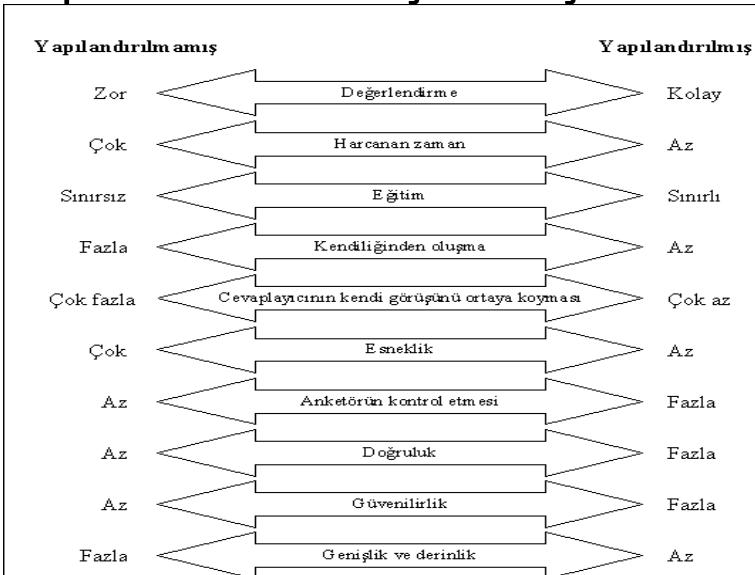
Kişisel Görüşme Yöntemi

- 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 degenilmemiş 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.
- Tümdeğelim: Açık uçlardan 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: Göresel öğelerle istenen cevap desteklenebilir
- Cevaplayıcıyı şartlandırmama: 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

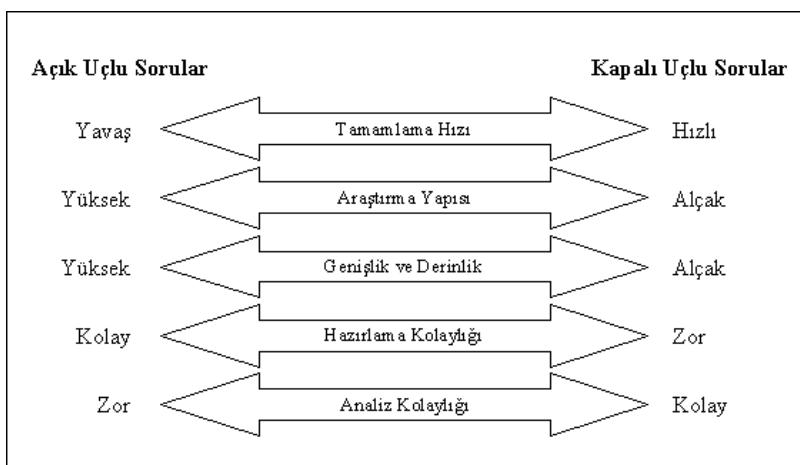
Anket Soruları: Açık Uçlu Sorular

- Cevaplayıcıdan kendi ifadelerinden oluşan yanıt ister
- Avantajlar
 - Önceden hazırlanmış cevaplarla alınamayacak bilgiler alınabilir
 - Cevaplayıcı bakış açısı ile yanıldığından 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 kümelerinden 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, cavaplayıcının yorumlama etkisini devre dışı bırakır
 - Hatırlamayı kolaylaştırır: Hangi eğitimleri aldınız sorusu şıklarıyla olunca insan kolay hatırlar
- Dezavantajlar
 - Cevaplayanlar konuya ilgili gerçek duygularını yansıtmayan seçeneklere maruz kalabilir. Gerçek cevabı benzeyen şıkları seçerek ince ama önemli farklılıklar yokeder.
 - Kişinin cevabı yoksa da, soruyu anlamasa da cevaba zorlar
 - Soruların çoğu kapalı uçlu olursa görüşlerin ifadesine izin verilmemişinden 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 sağlanmalıdır
- Sorular arasında bağlantı kurularak anı konu değişikleri yapılmamalı

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, bibliyografiya)
- 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

5.Ders

Göksel Biricik

Bu Derste

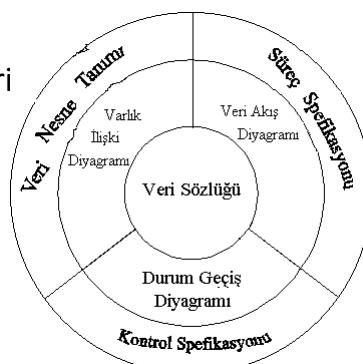
- Sistem Analizi
- Fonksiyonel Çözümleme
- SRS
- Kavramsal Modelin Oluşturulması

Gereksinim Analizi

- Sistem analizi modelinin amacı:
 - İhtiyaçları açıklamak
 - Tasarımın nasıl oluşturulacağına temelini oluşturmak
 - Oluşturulan yazılımın ihtiyaçları karşılayıp karşılamadığını onaylayan unsurları belirlemek

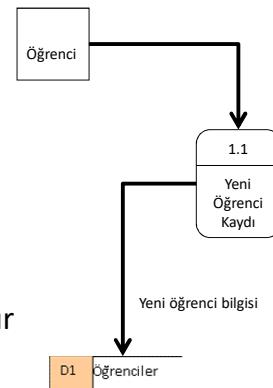
Gereksinim Analizi Modeli

- Veri Sözlüğü: Tüm veri nesnelerinin tanımları (metadata)
- Varlık İlişki Diyagramı (ERD): Veri nesneleri arası ilişkiler
- Veri Akış Diyagramı (VAD): Verilerin nasıl taşındığı, veri akışını sağlayan fonksiyonların neler olduğu
- Durum Geçiş Diyagramı (STD): Sistem dışındaki olaylar sonucunda sistemin nasıl hareket ettiği



Veri Akış Diyagramı

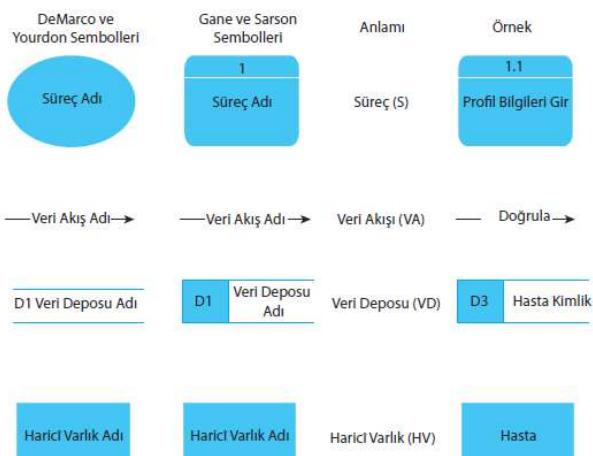
- Sistemdeki
 - Varlıklar
 - Süreçler - işlemler
 - Veri depoları
 - Aralarında verinin akışı
- Birden fazla varlık olabilir
- Okların çakışmaması için aynı varlık tekrarlayabilir
- Oklar tek yönlüdür
- Okların tek bir kaynağı ve hedefi vardır
- İşlemler hiyerarşije uygun numaralandırılır



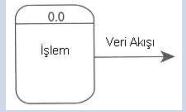
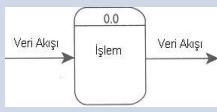
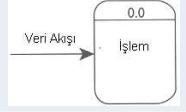
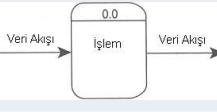
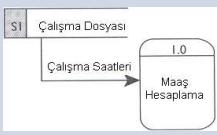
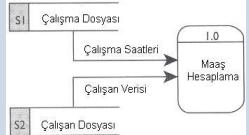
VAD Öğeleri

- **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 Sembollereri



VAD Kuralları

Kural	Yanlış	Doğru
İşlemenin sadece çıkışı olamaz.	 0.0 İşlem Veri Akışı →	 0.0 İşlem Veri Akışı → Veri Akışı →
İşlemenin sadece girişi olamaz.	 0.0 İşlem ← Veri Akışı →	 0.0 İşlem ← Veri Akışı → Veri Akışı →
İşlem girişleri istenen çıkışı verecek kadar yeterli olmalıdır.	 S1 Çalışma Dosyası Çalışma Saatleri → I.0 Maaş Hesaplama Veri Akışı →	 S1 Çalışma Dosyası S2 Çalışan Dosyası Çalışma Saatleri Çalışan Verisi → I.0 Maaş Hesaplama Veri Akışı → Veri Akışı →

VAD Kuralları

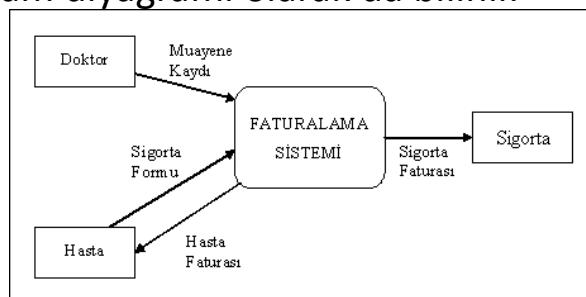
Kural	Yanlış	Doğru
Her veri deposu bir işlemle ilgili olmalıdır.	<p>Diagram illustrating a common mistake: two data stores, S1 and S2, are connected by a single data flow labeled "Veri Akışı".</p>	<p>Diagram illustrating the correct way: "Çalışma Dosyası" and "Çalışma Saatleri" feed into "Maaş Hesaplama".</p>
Veri deposu bir varlıkla doğrudan ilişkide olamaz.	<p>Diagram illustrating a common mistake: an Entity Source/Sink block is connected to a data store, with a data flow labeled "Veri Akışı".</p>	<p>Diagram illustrating the correct way: an "İşlem" block connects to a data store via two data flows, one labeled "Veri Akışı" entering and one labeled "Veri Akışı" exiting.</p>
Bir işlemle oku iki yönlü olamaz. Bir işlemle veri deposu arasında karşılıklı veri akışı varsa farklı tek yönlü oklarla gösterilmelidir.	<p>Diagram illustrating a common mistake: an "İşlem" block connected to a data store via two data flows, one labeled "Veri Akışı" entering and one labeled "Veri Akışı" exiting.</p>	<p>Diagram illustrating the correct way: two "İşlem" blocks are connected to a data store via two separate data flows, each labeled "Veri Akışı".</p>

VAD Kuralları

Kural	Yanlış	Doğru
Bir işlemden farklı iki işleme gidecek olan aynı veri, aynı yönde iki uçlu okla gösterilmelidir.	<p>Diagram illustrating a common mistake: an "İşlem" block has two outgoing data flows, both labeled "Veri Akışı", pointing to two separate "İşlem" blocks.</p>	<p>Diagram illustrating the correct way: a single "İşlem" block has two outgoing data flows, each labeled "Veri Akışı", pointing to two separate "İşlem" blocks.</p>
Veri, hiçbir işlemden geçmeden çıktıgı işleme doğrudan dönemez.	<p>Diagram illustrating a common mistake: an "İşlem" block has a data flow labeled "Veri Akışı" entering from below and another labeled "Veri Akışı" exiting to the right.</p>	<p>Diagram illustrating the correct way: an "İşlem" block with an incoming data flow labeled "Veri Akışı" and an outgoing data flow labeled "Veri Akışı".</p>
Veri akış okları üzerinde gösterilen veri, sadece isim formatında olmalıdır.	<p>Diagram illustrating a common mistake: two parallel horizontal arrows: "Siparişi Doğrula" above and "Zaman Çizelgesini Tamamla" below.</p>	<p>Diagram illustrating the correct way: two parallel horizontal arrows: "Siparişi Doğrula" above and "Zaman Çizelgesini Tamamla" below.</p>

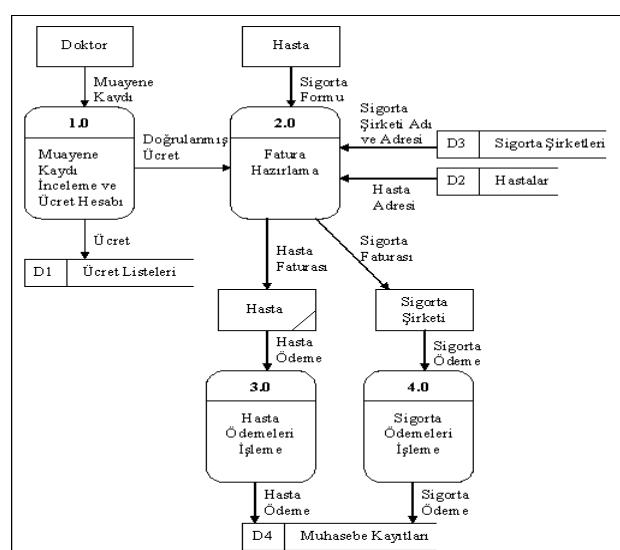
VAD Düzeyleri: Taslak

- Ön inceleme sonucunda belirlenir
- Sistemle varlıklar arası ilişkiyi gösterir
- Ayrıntılı süreç ve veri depoları bulunmaz
- Bağlam diyagramı olarak da bilinir.



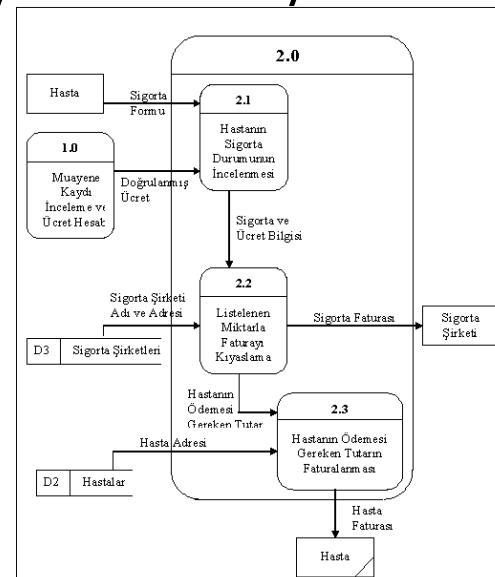
VAD Düzeyleri: 1.Düzey

- 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

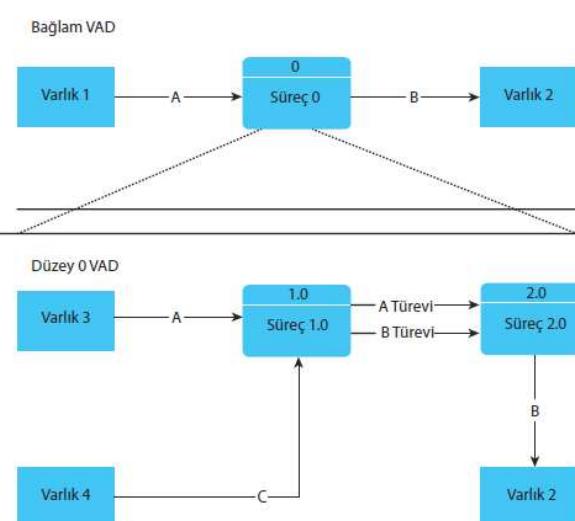


VAD Düzeyleri: 2.Düzey

- 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

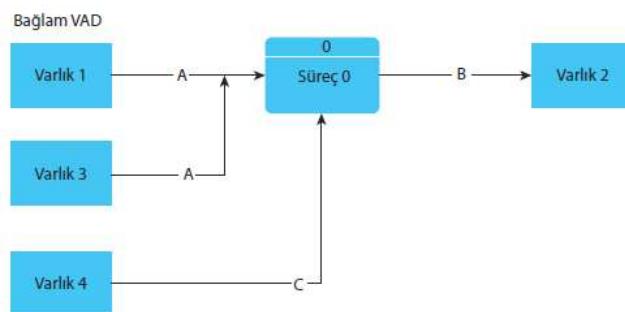


Hata Nerede?

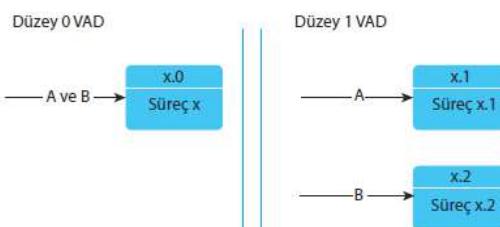


Çözümü

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



Burada sorun var mı?



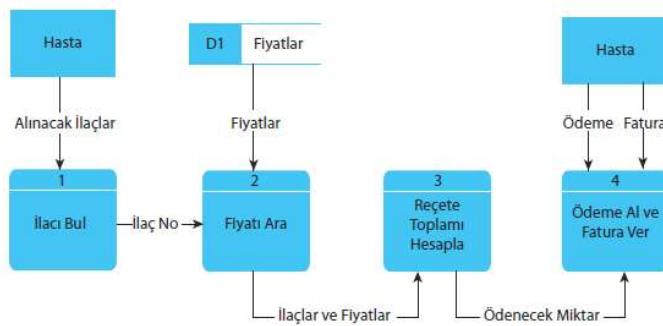
Kavramsal Veri Modeli - Varlık İlişki Diyagramları

- 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

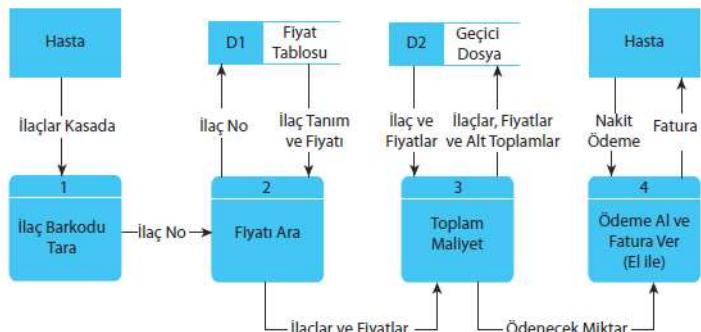
Mantıksal ve Fiziksel VAD

- 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 Modellenenecek?

- 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	Rxettedeki ilaçların toplam maliyetine bağlı olarak hasta ödemesi nakit ya da kart ile tahlil edilecek ve fatura verilecek.
Girdi Veri Akışı	<ul style="list-style-type: none">• Süreç 3'ten hesaplanarak gelecek olan "ne kadar ödeme yapılacağı" bilgisi,• Hastadan yapılacak tahsilat.
Çıktı Veri Akışı	Fatura
Süreç Tipi	<input type="checkbox"/> Çevrimiçi <input type="checkbox"/> Otomatik <input checked="" type="checkbox"/> El İle
Süreç Mantığı	<pre>READ ÖDENECEK_MİKTAR ÖDEME AI SELECT CASE CASE 1 (ÖDEMEbüyütür ÖDENECEK_MİKTAR) DO Para üstü ver ÖDEME=Tamam CASE 2 (ÖDEME eşittir ÖDENECEK_MİKTAR) DO ÖDEME=Tamam CASE 3 (ÖDEME küçütür ÖDENECEK_MİKTAR) DO ÖDEME=Tamam değil END SELECT IF ÖDEME eşittir Tamam THEN Fatura ver ELSE Hastaya bilgi ver END IF</pre>
Çözülmeyen Sorunlar	<ul style="list-style-type: none">• 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 blogunun özel bir tipidir. Bir koşulun izleyebileceği birden fazla durum varsa ve bu durumlardan biri olduğunu diğerleri oluşamayorsa kullanılan blok yapısıdır.	READ Kontrol-edilecek-değer SELECT CASE CASE 1 (Kontrol-edilecek-değer=Koşul1) DO Eylem#1 CASE 2 (Kontrol-edilecek-değer=Koşul2) DO Eylem#2 CASE 3 (Kontrol-edilecek-değer=Koşul3) DO Eylem#3 CASE 4 (Kontrol-edilecek-değer=Koşul4) DO Eylem#4 END CASE
Döngü Blok Yapısı: Koşul sağlanana kadar tekrarlanması gereken eylemler varsa kullanılır.	DO WHILE Koşul1 doğru olunda 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

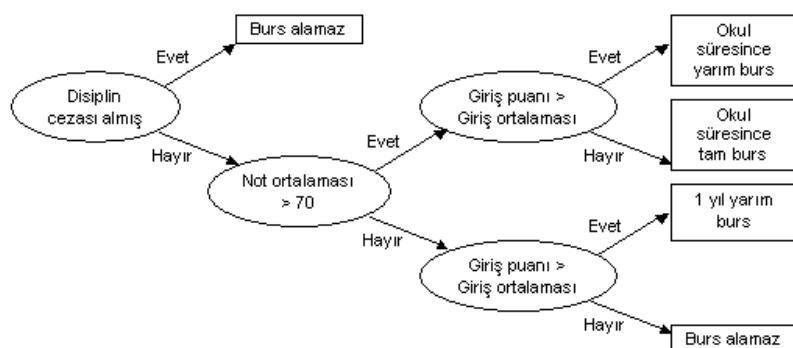
		KURALLAR				
		1	2	3	4	5
D U R U M	Disiplin cezası almış	E	H	H	H	H
	Not ortalaması > 70	H	E	H	E	H
	Giriş puanı > Giriş ortalaması	H	E	E	H	H
İ Ş L E M	Okul süresince tam burs		X			
	1 yıl yarım burs			X		
	Okul süresince yarım burs				X	
	Burs alamaz	X				X

Karar Tablosu Örneği

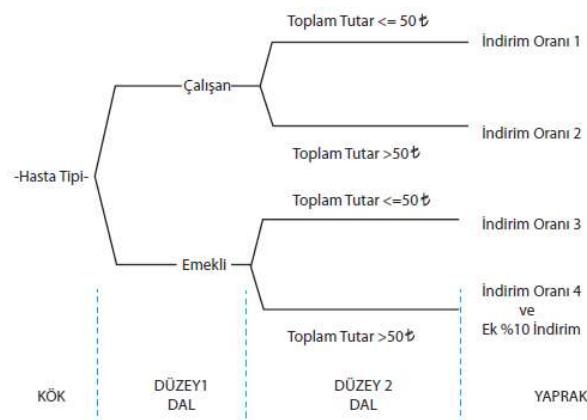
İlaç İndirim Hesapla		KURALLAR			
KOŞULLAR		1	2	3	4
Çalışan Hasta	E	E	H	H	
Emekli Hasta	H	H	E	E	
Toplam Tutar 50 TL'den küçük ya da eşit	E	H	E	H	
Toplam Tutar 50 TL'den büyük	H	E	H	E	
EYLEMLER					
İndirim Oranı 1 Uygula	X				
İndirim Oranı 2 Uygula		X			
İndirim Oranı 3 Uygula			X		
İndirim Oranı 4 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 Girişи
 - Veri Deposu Sözlük Girişи
 - Veri Yapısı Sözlük Girişи
 - Veri Elemanı Sözlük Girişи
 - İşlem Sözlük Girişи
- 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

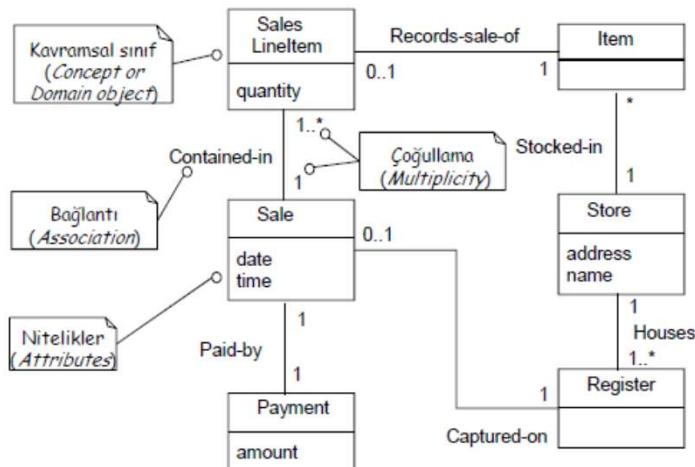
İşlem Tanımı : ... yap, eğer değilse yap vb.

Cıktı : Girilmiş satış siparişleri

Problem Uzayının Modellenmesi

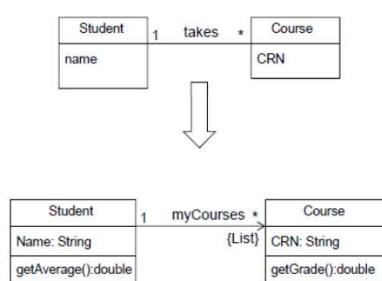
- Amaç gerçek dünyanın (problemin) doğru, anlaşılır, sınanabilir modelini oluşturmak
- Problemin anlaşılması aşamasıdır
- Teknik Yorumdan çok müşteri isterleri ön plandadır
- İsteklerin modellenmesi nd değil, burası nd.
- Gerçek dünyadaki kavramsal sınıflar ve nesneler yer alır (nesneler, o anda görülen özellikleri, aralarındaki ilişkiler (bağıntıları)). Yazılım nesneleri henüz düşünülmez
- UML ile görselleştirilir.
- Oluşturacak 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 uyarlanması yapılır.

Örnek: İsim İşaretleme

Main Success Scenario (or Basic Flow):

1. Customer arrives at a **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.
6. Cashier tells Customer the total, and asks for **payment**.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external **Accounting** system (for accounting and **commissions**) and **Inventory** system (to update inventory).
9. System presents **receipt**.
10. Customer leaves with receipt and goods (if any).

Extensions:

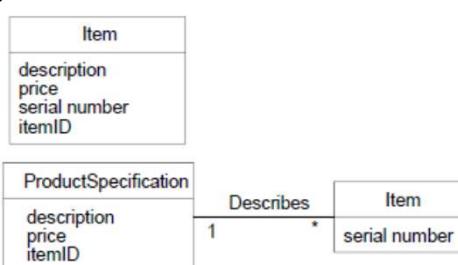
- 7a. Paying by cash:
 1. Cashier enters the cash **amount tendered**.
 2. System presents the **balance due**.

Gereksiz Sınıfların Elenmesi

- **Fazlalık sınıflar:** aynı unsuru ifade edenlerden, daha tanımlayıcı olan. Kişi-müşteri: Müşteri
- **İlgisiz sınıflar:** çözümle ilgisi olmayan ya da o aşamada ilgilenmeyeceğimiz sınıflar: Kredi Kartı
- **Belirsiz sınıflar:** sınırları iyi çizilmemiş, kaba-geniş tanımlı sınıflar. Genellikle birden fazla sınıfın oluşturmuşlardır ya da başka sınıfın parçasıdır: Muhasebe Sistemi
- **Nitelikler:** kendi başına varlığı anlamlı olmayanlar, sınıfların özellikleridir: Miktar
- **İşlemler:** Sadece nesnelere uygulanan işlemler sınıf olmaz. Kendi nitelikleri olmalı ve hizmet alışverişini yapmalıdır: Ödeme(miktar, para birimi, tarih, ...)
- **Roller:** Sınıflar arası ilişki olan roller sınıf olamaz.
- **Gerçekleme Elemanları:** Yazılım sınıfları bu uzayda yer almaz.
- **Sınıf olup olmadığına 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?

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

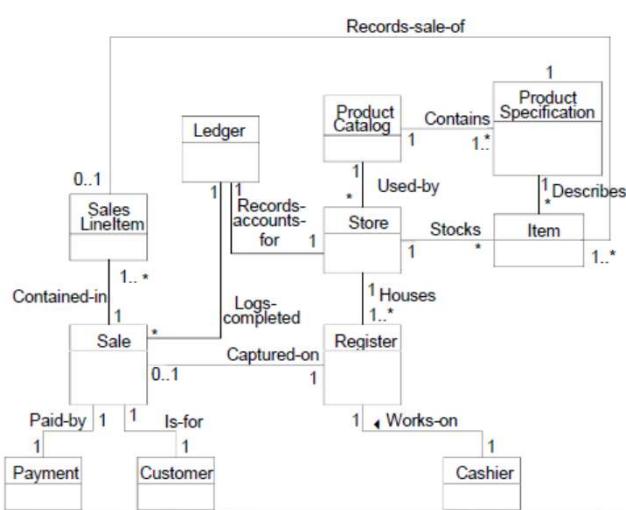
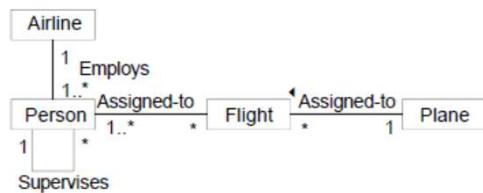
- Bir nesnenin kaç tane nesneyle ilişki içinde olacağı çoğullama ile gösterilir.
- Bağlantılara doğru isimler verilmelidir. Tip-fiil-tip
- Bazı bağlantıların unutulması tasarımcı çok etkilemez.
- Kavramsal sınıfların doğru bulunması daha önemlidir.
- Fazla bağlantı anlaşılırlığı azaltır.
- Yaygın bağlantılar 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.

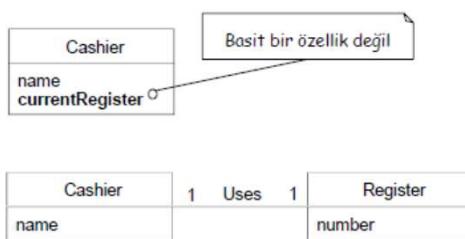
Örnek: Bağlantılar

- Tasarımı ilgilendiren bağlantıları ortaya çıkarın
- Tip-fiil-tip doğru isim ataması yapın.



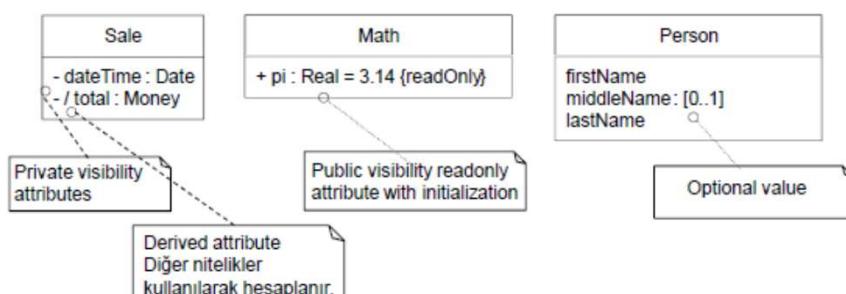
Özelliklerin Belirlenmesi

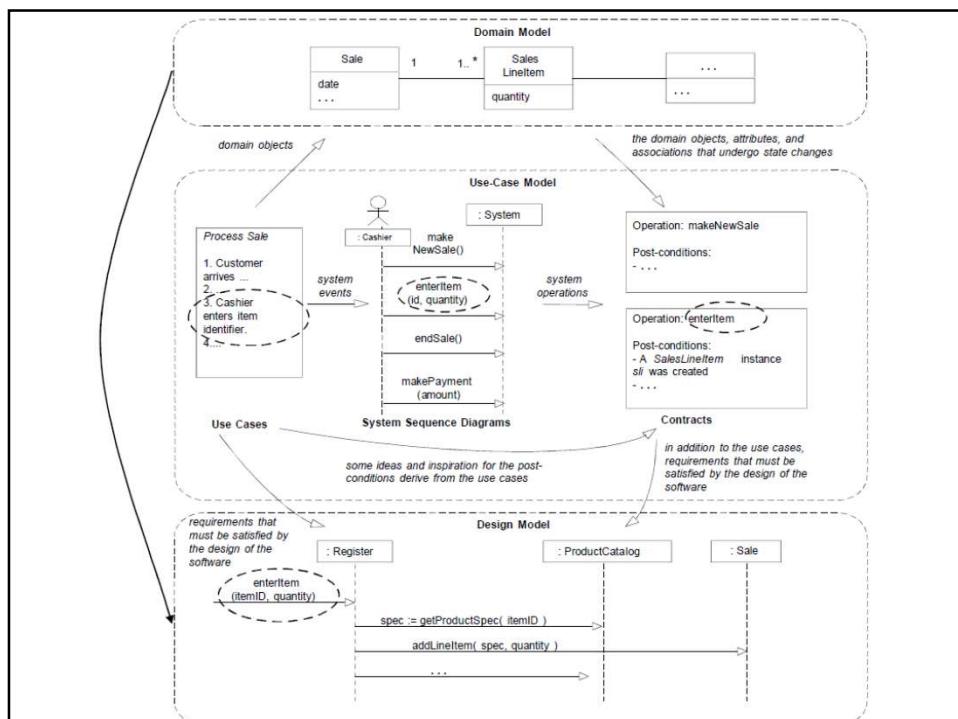
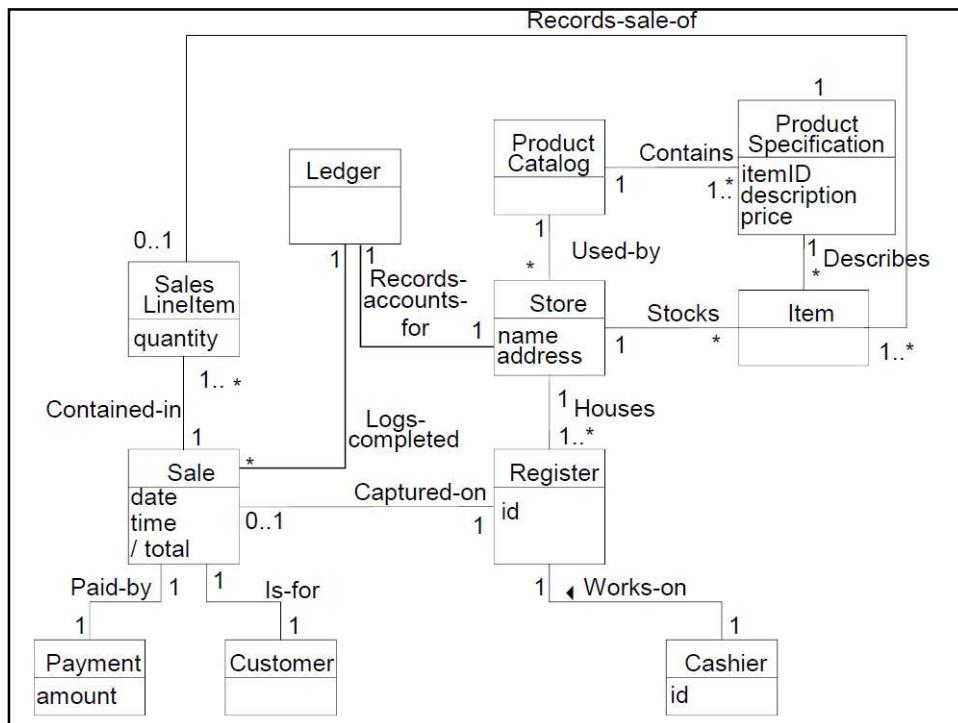
- Özellik: nesne yaratıldığında nesneye özgü değer alabilen veri.
- Senaryolarla ilgili nitelikler bulunmalıdır.
- Basit veri tipleri ile ifade edilirler.
 - birimleri varsa doğru değil. (para gibi)
- Karmaşık tipteyse, başka bir sınıf olma olasılığı yüksektir.
 - birden fazla alansa (tel no, ad soyad)
 - işlem yapılmıysa (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 Analizi Laboratuvar Uygulaması
(VAD+Kavramsal Sınıf D.)
- Sistem Tasarımı
 - Girdi, Çıktı, Veri Yapısı, Arabirim Tasarımları
- Ara Sınav

Sistem Analizi ve Tasarımı

6.Ders

Göksel Biricik

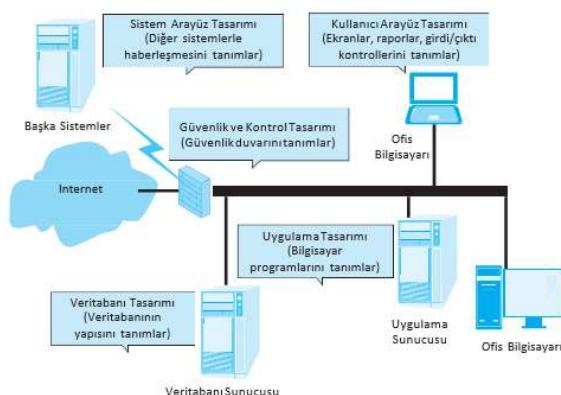
Bu Derste

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

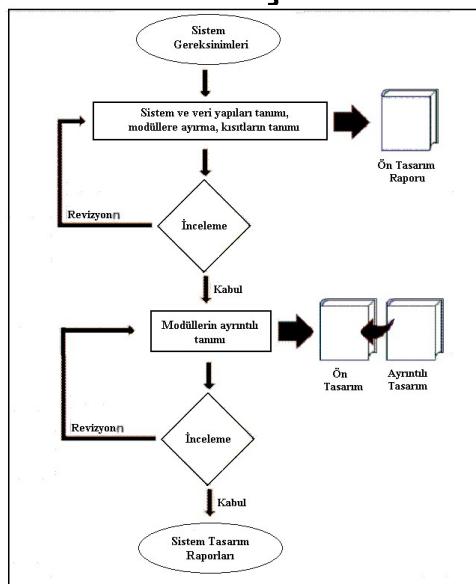
Sistem Tasarımı

- Ön Tasarım
 - Alt yapı belirleme
 - Modül mimarisinin oluşturulması
- Ayrıntılı Tasarım
 - Sistem Etkileşimi tasarımlı
 - Çıktı tasarımlı
 - Girdi tasarımlı
 - Arabirim tasarımlı
 - Program tasarımlı
 - Veritabanı tasarımlı

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 gerçekleştirmeye 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 gerçekleştirir.

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 tasarıımı 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

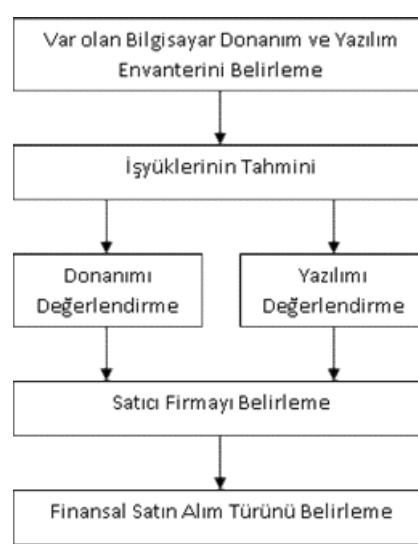
Sistem Mimari Modellemesi

- Bilgi sistemi, merkezi mi yoksa dağıtık mı olacak? (Birçok sistem, ağ üzerinden dağıtık çalışmaktadır.)
- Bir ağ üzerinden depolanan verinin dağıtımını nasıl olacak? (Birçok modern veritabanı, dağıtık ya da ağ üzerinde çok kopyalı olarak bulunabilmektedir.)
- Geliştirilecek yazılım için uygulama teknolojileri ne olacak? Hangi programlama dili ve araçları kullanılacaktır?
- Satın alabilecek ticari yazılımların bütünlendirilmesi 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.

İş Yüklerinin Tahmini

- Donanım altyapı kapasitesinin yeterliliğini belirlemek için yapılır.
- Zaman-maliyet kıyaslaması yapılır, gereksiz sistem kurulumu engellenmeye çalışılır.

	Mevcut Sistem	Önerilen Sistem
İş	Bayı satışlarının aylık raporu	Bayı satışlarının aylık raporu
Yöntem	Dosya takibi	Bilgisayar tabanlı takip
Personel	Bayı yöneticisi	Bilgisayar uzmanı
Maliyet / Saat	30 TL	15 TL
Ne Zaman? Nasıl?	Günlük : Her bayi için faturaların dosyalanması Aylık : Hesap makinası kullanarak günlük kayıtların özetlenmesi ve rapor hazırlanması	Günlük : Fatura bilgilerinin girilip toplamın alınacağı programın çalıştırılması Aylık : Raporlamayı yapıp yazdırın programın çalıştırılması
Gerekli Olan İnsan Zamanı	Günlük : 20 dakika Aylık : 8 saat	Günlük : 4 dakika Aylık : 20 dakika
Gerekli Olan Bilgisayar Zamanı	Yok	Günlük : 4 dakika Aylık : 20 dakika

Donanım Değerlendirme

- İş yükü ve mevcut envantere göre proje ihtiyaçları belirlenir.
- İş yüklerinin farklı sistemlerde simülasyonu yapılarak karşılaştırılır.
- Karşılaştırma kriterleri:
 - İşlemenin gerçekleşmesi için geçen ortalama süre (Bu süreye giriş verisinin sisteme girilmesi ve çıkış elde etme de dahildir.)
 - Sistemin toplam kapasitesi (Herhangi bir problem olmadan aynı zamanda kaç işlem gerçekleştirildiği.)
 - CPU'nun atıl zamanı
 - Önerilen belleğin büyüklüğü

Yazılım Değerlendirme

Yazılım Gereksinimleri	Yazılım Özellikleri	Yazılım Gereksinimleri	Yazılım Özellikleri
Performans Etkinliği	<ul style="list-style-type: none">▪ İhtiyaç duyulan tüm işleri yerine getirme▪ Tüm işleri istenildiği şekilde yerine getirme▪ İyi tasarlanmış ekran çıktıları▪ Yeterli kapasite	Esneklik	<ul style="list-style-type: none">▪ Girdi seçeneği▪ Çıktı seçeneği▪ Diğer yazılımlarla birlikte kullanılma
Performans Verimliliği	<ul style="list-style-type: none">▪ Hızlı cevap zamanı▪ Verimli giriş ve çıkış▪ Verimli veri saklama▪ Verimli yedekleme	Dökümantasyon Kalitesi	<ul style="list-style-type: none">▪ İyi organizasyon▪ Yeterli çevrim içi kaynak▪ Web sitesinde sık sorulan soruların(FAQ) bulunması
Kullanım Kolaylığı	<ul style="list-style-type: none">▪ İyi tasarlanmış kullanıcı arabirimi▪ Yardım menüleri▪ Esnek arabirim▪ Yeterli geri besleme▪ Hata düzeltme	Üretici Desteği	<ul style="list-style-type: none">▪ Teknik destek▪ Ürün güncellemesinin Web sitesi üzerinden yapılabilmesi

Satıcı Firmayı Belirleme

Temel Satıcı Hizmetleri	Satıcıının Sunduğu Hizmetler
Donanım Desteği	<ul style="list-style-type: none">▪ Ürün kalitesi▪ Garanti
Yazılım Desteği	<ul style="list-style-type: none">▪ Toplam yazılım ihtiyacı▪ Özel programlama gereksinimi▪ Garanti
Kuruluş ve Eğitim	<ul style="list-style-type: none">▪ Zaman programına uyma▪ Eğitim▪ Müşteri hizmetleri
Bakım	<ul style="list-style-type: none">▪ Rutin bakım işlemleri▪ Acil durumdaki cevap verme süresi

Satın Alma Şeklini Belirleme

	Avantajları	Dezavantajları
Satın Alma	<ul style="list-style-type: none">▪ Uzun dönemde leasing ve kiralamadan ucuz olması▪ Sistemin değişirebilinmesi▪ Vergi indirimini sağlama▪ Tam kontrol	<ul style="list-style-type: none">▪ İlk ödemenin yüksek olması▪ Eskime riski▪ Seçimin yanlış yapılması sonucu yatırımin batma riski▪ Tam sorumluluk
Leasing	<ul style="list-style-type: none">▪ Sermayenin bağlanmaması▪ Finanslamaya ihtiyaç olmaması▪ Kira bedelinden düşük olması	<ul style="list-style-type: none">▪ Sözleşme bittiğinde işletmenin sisteme sahip olamaması▪ Ödemenin gecikmesi durumunda yüksek faiz oranı▪ Satın almadan daha pahali olması
Kiralama	<ul style="list-style-type: none">▪ Sermayenin bağlanmaması▪ Finanslamaya ihtiyaç duyulmaması▪ Sistemin kolayca değiştirilmesi▪ Genelde bakım ve garantisinin fiyat kapsamında olması	<ul style="list-style-type: none">▪ İşletmenin sistemin gerçek sahibi olmaması▪ Satıcı kendi riskini üstlendiği için kira bedelinin çok yüksek olması

Ayrıntılı Tasarım

- Çıktı Tasarımı
- Girdi Tasarımı
- Arabirim Tasarımı
- Uygulama (Program) Tasarımı
- Veritabanı Tasarımı

Çıktı Tasarımı

- Çıktı: Sistemin Kullanıcılara verdiği bilgi, üretilen raporlar
- Amaçlar:
 - Belirlenen amaca hizmet etme
 - Kullanıcı için anlamlı olma
 - Uygun sayıda olma
 - Hangi kullanıcılara dağıtılacığının doğru belirlenmesi
 - Zamanında sağlanması (günlük, aylık, yıllık veya koşula bağlı raporlar)
 - Doğru çıktı yönteminin (ortamının) seçilmesi
- Kullanıcıyı etkileyeyecek yönlendirme:
 - Bilgilerin belirli kriterlere göre sıralanması
 - Sınırların Belirlenmesi
 - Grafik tipi – rengi – ölçüğünün 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 kullanıcının fazla zamanını almaması ve ergonomik olması
 - Uyumluluk: Bir formda diğerine ya da ekran görüntüsüne geçişte dönemin 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 tasarımları:** 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 artırma
 - Kullanıcıların görüşlerinin alınabilmesi
 - Ergonomik olması

Kullanıcı Arabirim Tasarımı

- Yerleşim planı: Ekran iç ana alana bölünür.
 - En üst alan, sistemde gezinim sağlar
 - Orta alan kullanıcı çalışmalarına ayrıılır
 - En alt alan, yapılanlarla ilgili durum bilgisi verir

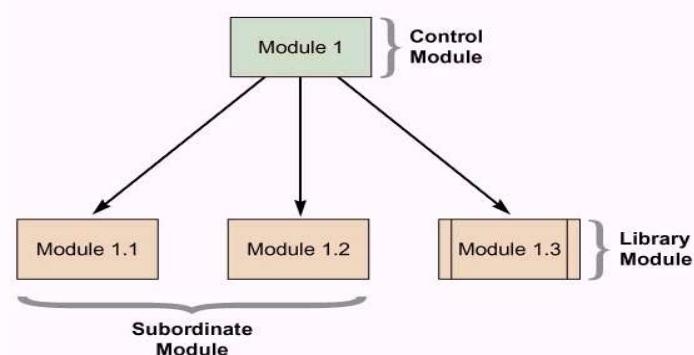
Arabirim Tasarım Problemleri

- Bilginin düzensiz görünümde yiğinlar 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
- ...

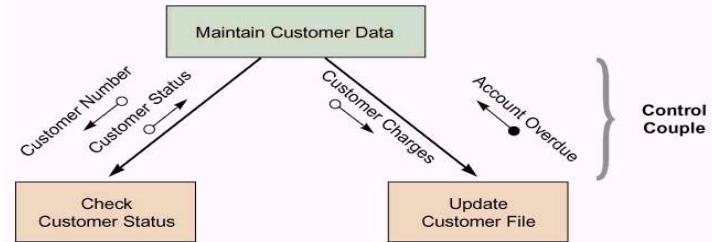
Sistem(Uygulama) Mimarısını Belirleme

- 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 iletişim: 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

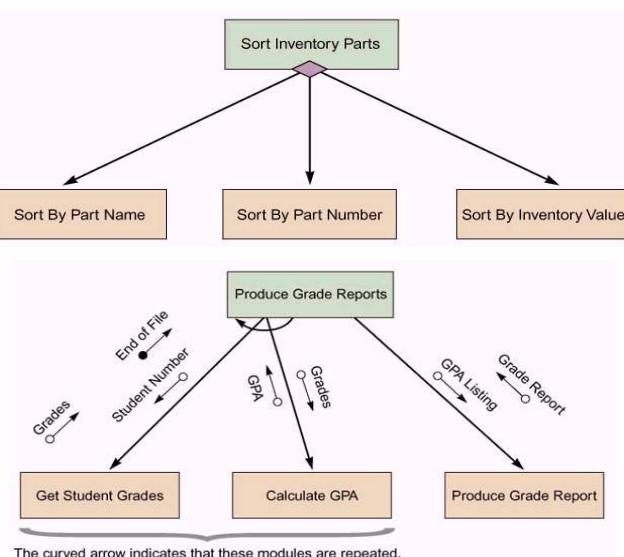
Ana Modül – Alt Modüller



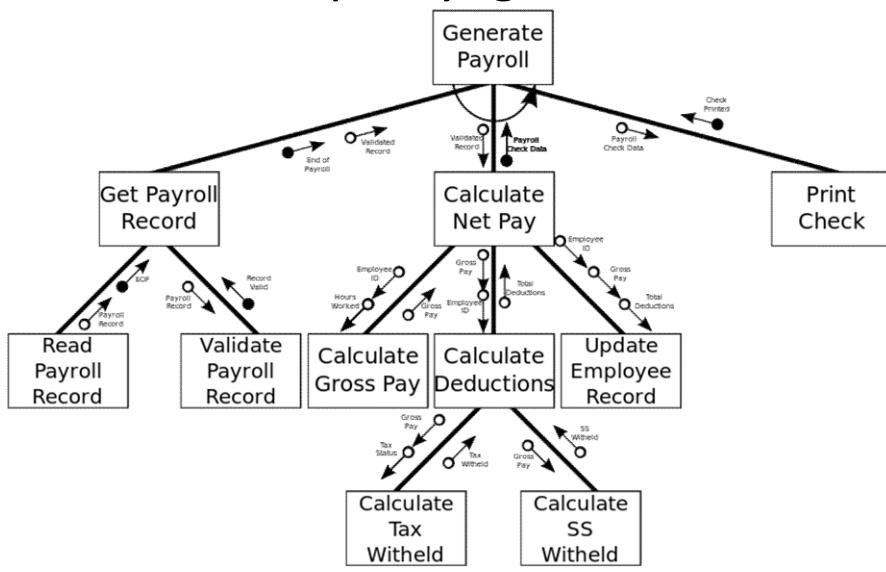
Veri İletişimi – Kontrol Mesajları



Durumlar - Döngüler



Yapı Diyagramı



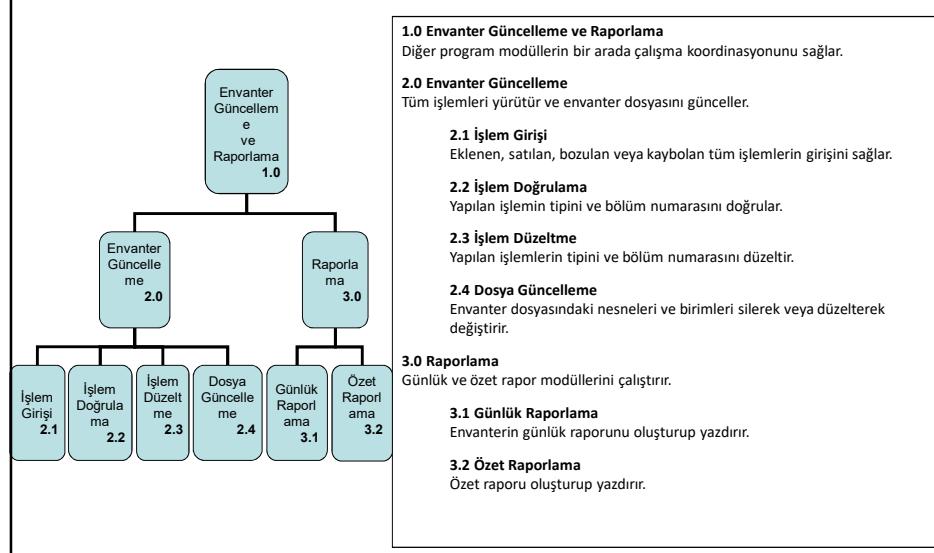
Tasarım Spesifikasyonları

- 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

Hierarchy & Input-Process-Output

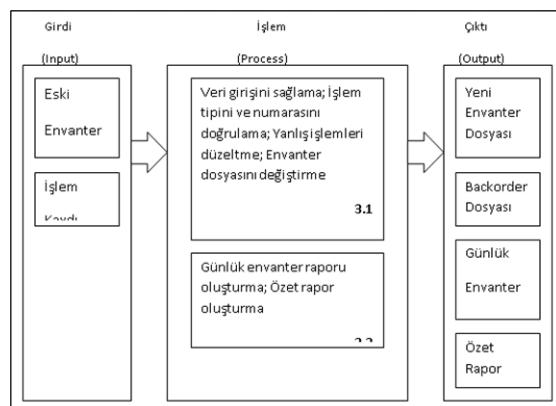
- HIPO
- Sistemdeki her fonksiyonun birbiriyle olan ilişkisini belirler.
- Temel fonksiyonlar belirlendikten sonra alt fonksiyonlar tanımlanarak fonksiyon hiyerarşisi oluşturulur.
- Diyagramlar:
 - Görsel içerik tablosu (Visual Table of Contents, VTOC)
 - Genel IPO (Input Process Output) Diyagramı
 - Ayrıntılı IPO Diyagramı

Görsel İçerik Tablosu

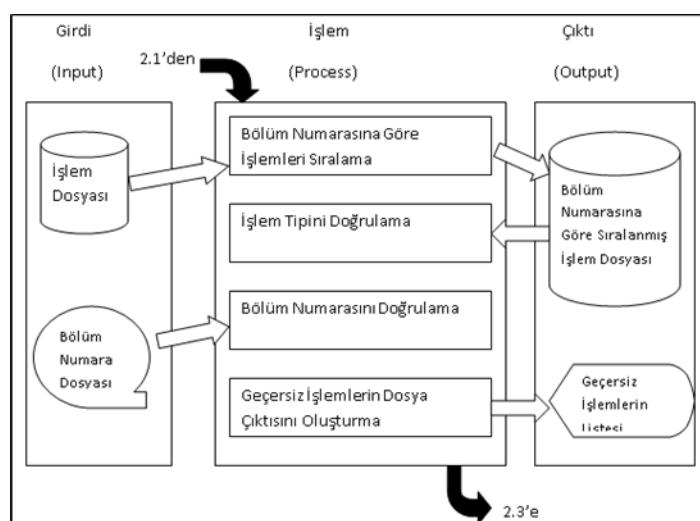


Genel IPO

- VTOC diyagramındaki her fonksiyon için IPO detaylı olarak tanımlanır.



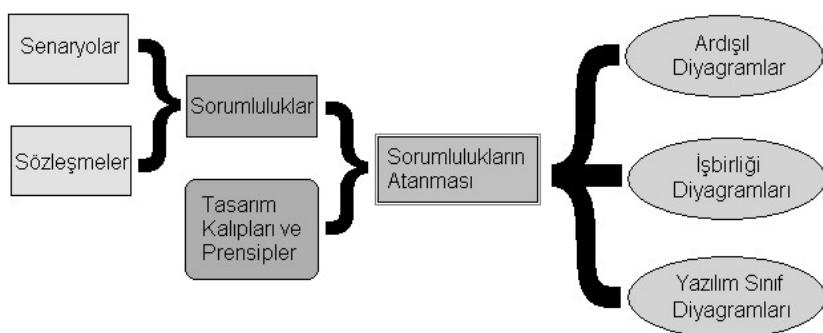
Ayrıntılı IPO



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

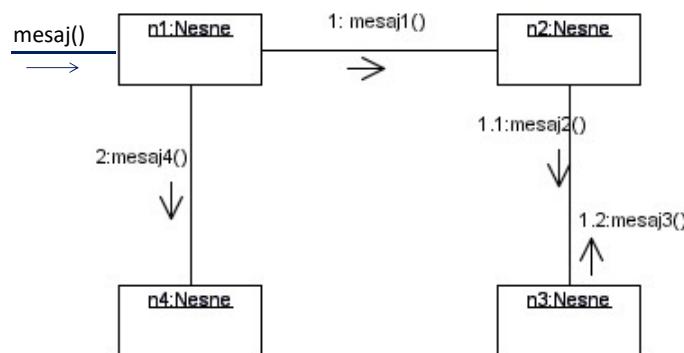
Nasıl Yapılır?



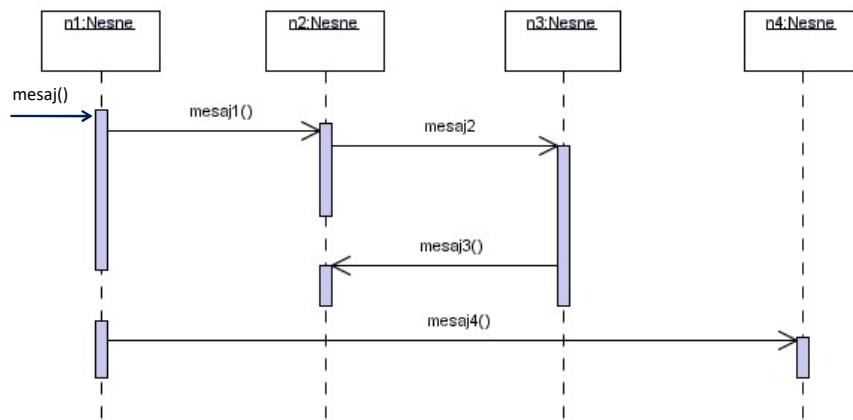
Etkileşim Diyagramları

- İşbirliği Diyagramları
 - Az yer kaplar
 - Dallanma paralellik ve iterasyonlar kolay gösterilir
 - Mesaj sırasını anlamak zor
- Ardışıl Diyagramlar
 - Fazla yer kaplar
 - Mesaj sırasını anlamak kolaydır

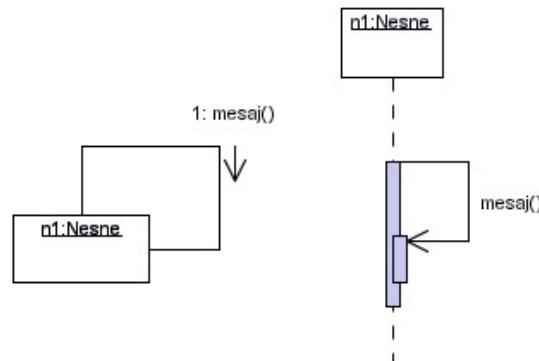
Mesaj Sıra Numaraları



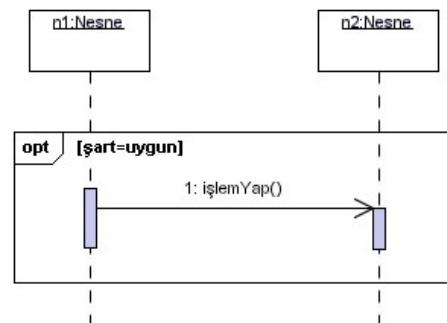
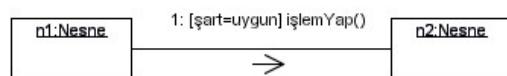
Mesaj Sıra Numaraları



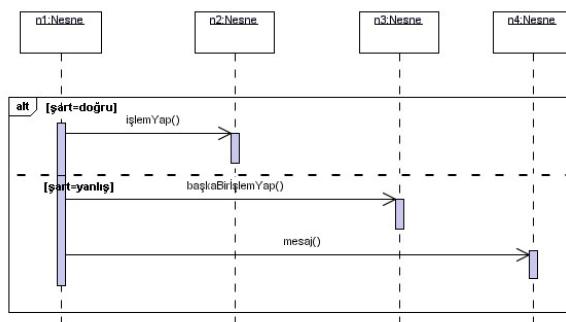
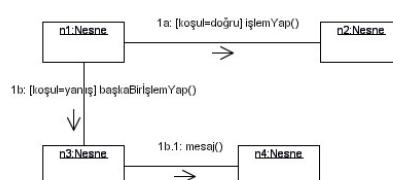
Kendine mesaj



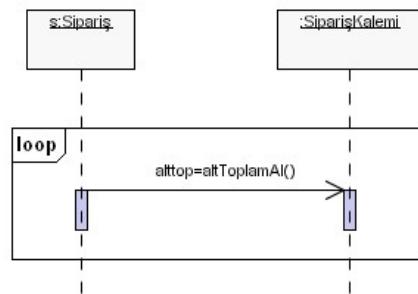
Koşullu mesajlar



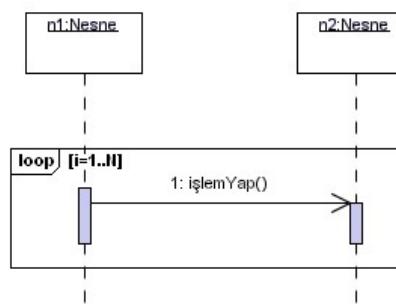
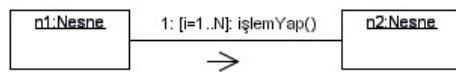
Karşılıklı dışlamalı mesajlar



Döngüler



Döngüler



Kullanım senaryolarının gerçekleştirmesi

- Senaryodaki durumların modellenerek gerçekleştirmesi
- Tasarım: yazılım sınıflarına metodların 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 gerçekleştirmesi

- Sorumlulukları yerine getirmek için metodlar oluşturulur
- Bir sorumluluğu erine getirmek için bir metod başka metodlarla işbirliği yapabilir
- İlk iterasyonda senaryoların ana akışları gerçekleştirilir
- İkinci iterasyonda alternatif akışlar ele alınır ve gerçekleştirir.
- 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.

“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 (özellik) 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 atmak gereklidir.

Senaryoların gerçeklenmesi

- Anlatılan kalıplara uygun olarak
 - Belirlenen tüm senaryolar
 - Tüm sözleşmeler

Gerçekleştirilir.

- Kavramsal sınıflardan yola çıkılarak yazılım sınıfları oluşturulur.
- Sorumluluklar, metodlarla gerçekleştirilir.
- Her bir işbirliği, tasarım sınıf diyagramında bir bağlantı olarak gösterilir.

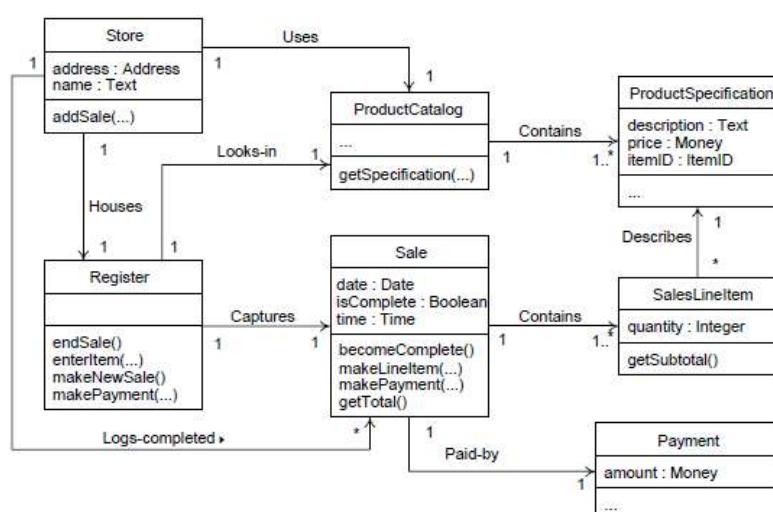
Başlangıç İşlemleri

- Sistem ilk çalışmaya başladığında yapılacaklar ayrı bir senaryo grubu olarak yazılabilir.
- Bu işleri tasarımın en son aşamasında belirlemek uyundur.
- Bir başlangıç nesnesi (initial domain object) belirlenir.
- Program çalışmaya başlayınca bu nesne yaratılır.
- Doğrudan içerdeği 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.

Tasarım Sınıf Diyagramı

- Senaryoları gerçeklerken çizilen etkileşim diyagramlarına paralel olarak, yazılım sınıflarından oluşan diyagram(lar) da çizilir.
- Yazılım sınıfları, özellikleri ve tipleri, metodların parametreleri ve erişim hakları büyük ölçüde belirtilir.
- İlişkiler ve bağımlılıklar yönlü olarak gösterilir. (Yeni ilişkiler de bulunabilir)

Örnek



Gelecek Ders

- Veri Modelleme
- Veri Yapısı ve Veri Tabanı Tasarımı

Sistem Analizi ve Tasarımı

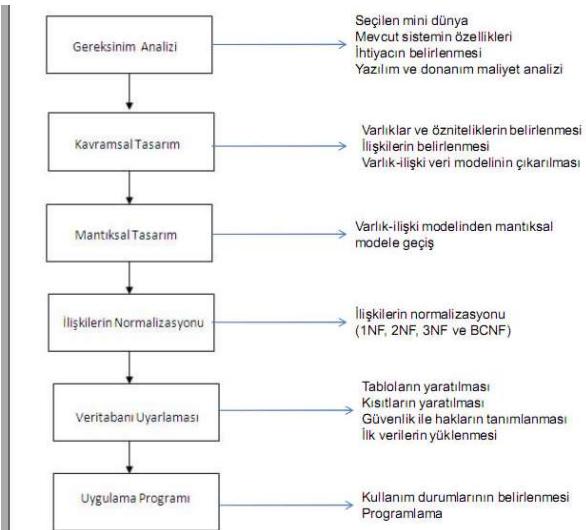
7.Ders

Göksel Biricik

Bu Derste

- Sistem Tasarımı
 - Veri Modelleme
 - Veri Yapısı ve Veri Tabanı Tasarımı

Veritabanı Tasarım Aşamaları



Varlık-İlişki Modeli

- Entity-Relationship Model
 - Kavramsal tasarımda veritabanında tutulacak verilerin daha üst seviyede gösterilmesi için
 - Kavramsal tasarım için en çok kullanılan ve en popüler model
 - Diğer modeller?
 - Sıradüzensel: Kayıt kütükleri
 - Yarı yapılmış veri modeli: XML
 - Nesneye dayalı model: Kavramsal nesneler
 - Yapılanmış bellek: NoSQL, MongoDB, AllegroGraph, ...

Varlık-İlişki Sembollereri



Varlık Sınıfı



Öznitelik



Birincil Anahtar



Bağıntı

Varlık

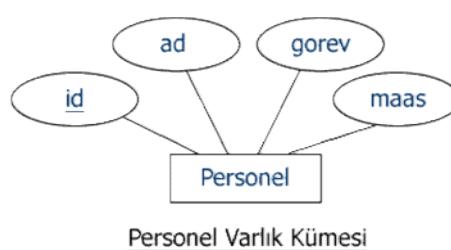
- Modelin en temel üyesidir.
- Var olan ve benzerlerinden ayrıt edilebilen her şey varlıktır.
 - Ör: Kitap, öğrenci, araba birer varlıktır.
- Modelin içerisinde varlık kümesi dikdörtgen ile gösterilir.
- Veritabanı olarak düşünülürse her bir tablo bir varlık kümesidir.

Nitelik

- Varlıkların her bir özelliği bir nitelik olarak ifade edilir.
 - Ör: öğrenci adı ve numarası öğrenci varlığının nitelikleridir.
- Modelin içerisinde nitelikler oval ile gösterilir ve içerisinde niteliğin ismi yazılır.
- Veritabanı olarak düşünülürse tablonun her bir sütunu bir niteliği gösterir.

Anahtar Nitelik

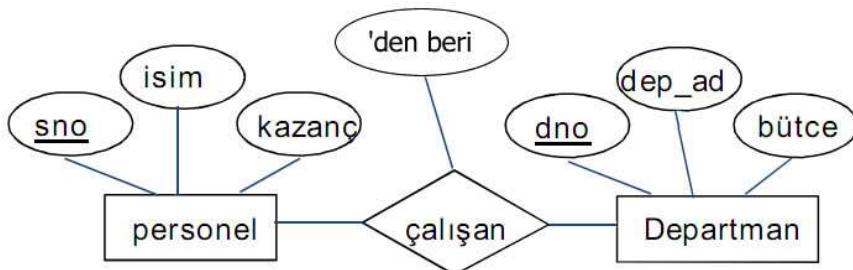
- Bir niteliğin değeri her bir varlık için farklıysa bu nitelik anahtar nitelik olarak belirlenir.
- Şemada niteliğin altı çizilerek gösterilir.



İlişki

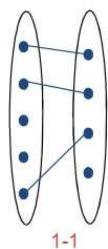
- Farklı varlıklar arasındaki ilişkileri ifade eder.
 - Ör: öğrenci ve dersler ayrı varlık kümeleridir ama öğrenciler ders almak zorunda olduğu için iki varlık arasında ders alma ilişkisi vardır.
- Model içerisinde ilişkiler baklava dilimi ile gösterilir ve içerisine ilişkinin adı yazılır.
- Tablolar arasında kullanılan ilişkiler 1-1, 1-n, n-1, n-m ile gösterilir.

Varlık-İlişki Örneği



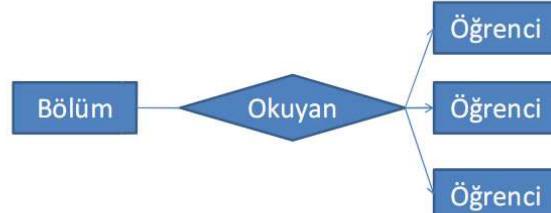
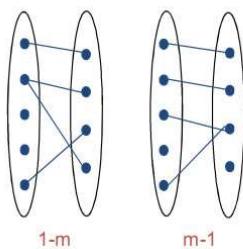
İlişki Tipleri – Birebir İlişki

- **Bire-bir İlişki** : Herhangi bir varlık kümesindeki her varlık diğer varlık kümesinin en çok bir varlığı ile ilintilidir.



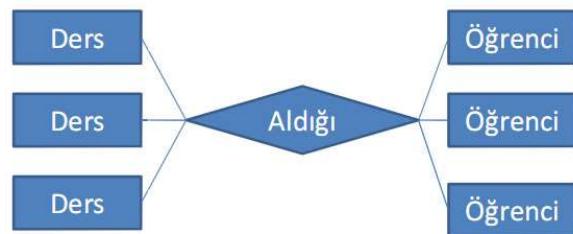
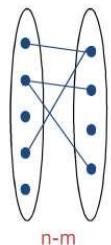
İlişki Tipleri – Bire-çok İlişki

- **Bire-çok ilişki** : İlk kümedeki her varlık diğer kümelerin en çok bir varlığına ilintidir.



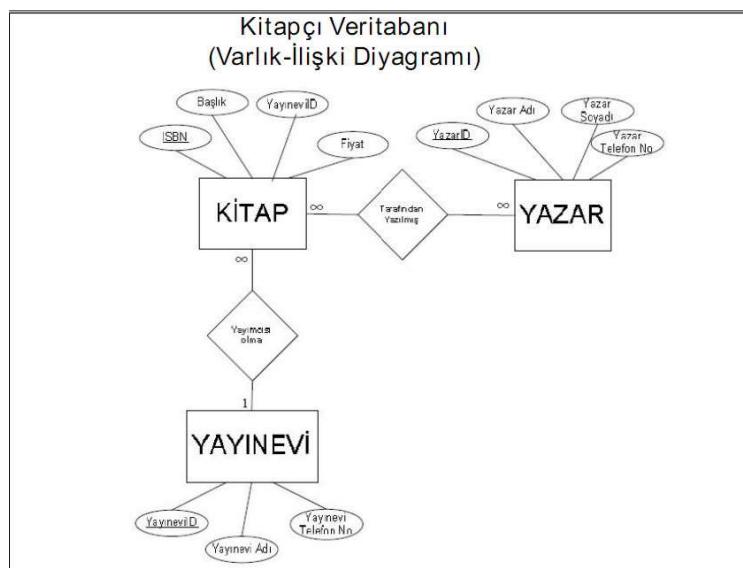
İlişki Tipleri – Çoka-çok İlişki

- **Çoka-çok ilişki :** Bir ilişkide herhangi bir kümede bulunan varlıklardan herbiri diğer kümede bulunan birçok varlıkla ilişkilidir.



Birincil anahtarları A ve B tablolarının yabancı anahtarlarından oluşan, bağlantı tablosu diye adlandırılan üçüncü bir tablo tanımlayarak oluşturulur.

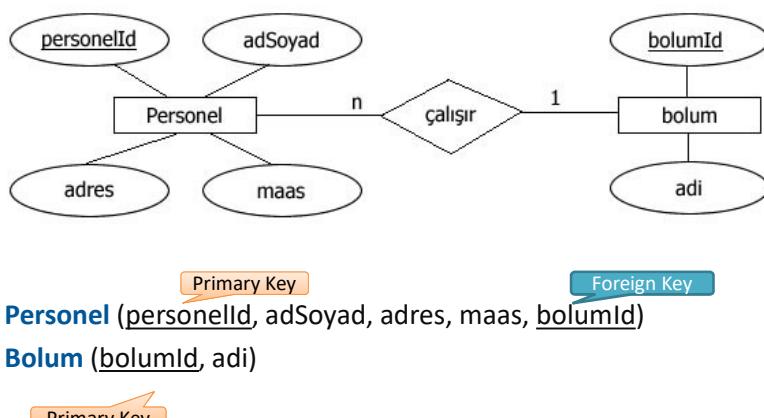
Örnek Varlık-İlişki Diyagramı



Veri Modelinin Gerçeklenmesi

- Varlık kümeleri tablolara dönüştürülür.
- Nitelikler tablonun sütunlarına dönüştürülür.
- Modelde oluşturulan ilişkilerin durumuna göre tabloların ilişkileri ve doğal olarak da anahtar sütunları belirlenir.
- 1-1 ilişkide bir varlık kümelerinin birincil anahtarı diğer varlık kümelerinin yabancı anahtarı olarak belirlenir. Hangisinin birincil hangisinin yabancı anahtar olacağına tablonun içereceği bilgilere göre karar verilir.
- 1-n ilişkinin n tarafındaki tabloya 1 tarafından tablonun birincil anahtar sütunu yabancı anahtar olarak eklenir.

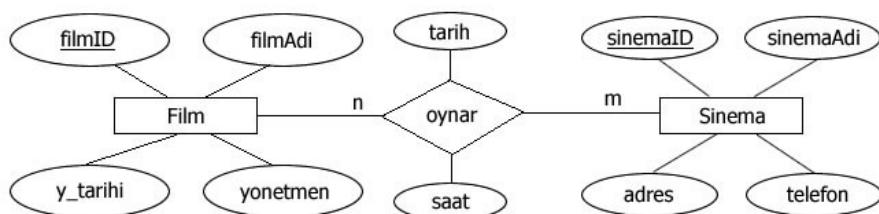
Örnek



Çoka-Çok İlişkilerin Gerçeklenmesi

- Varlık kümeleri tablolara dönüştürülür.
- Oluşturulan ilişki isminde tablo oluşturulur.
- Nitelikler tabloların sütunlarına dönüştürülür.
Tanımlayıcı nitelikler ilişkiden oluşturulan tabloya sütun olarak eklenir.
- İlişkiyi oluşturan tabloların birincil anahtarları ilişkiyi oluşturan tabloya yabancı anahtar olarak eklenir.
- İlişkiden oluşturulan tablonun birincil anahtarı oluşturulan yabancı anahtarların birleşiminden oluşur.
Eğer, bu şekilde oluşturulan birincil anahtar ihtiyaçlara cevap vermiyorsa yeni bir sütun eklenerek birincil anahtar yapılır.

Örnek



Film (filmID, filmAdi, y_tarihi, yonetmen)

Sinema (sinemaID, sinemaAdi, adres, telefon)

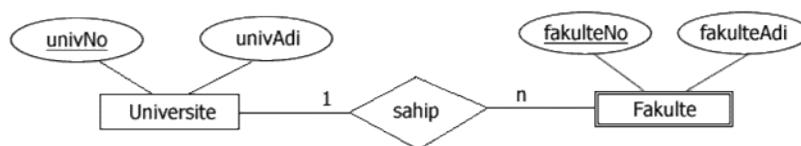
Oynar (oynarID, filmID, sinemaID, tarih, saat)

Zayıf Varlık Kümeleri

- Mevcutluğu diğer varlık türüne bağlı olan varlık türüdür.
- Eğer bir varlık kümelerinin niteliklerinin tümü alınsa bile bir anahtar oluşturamayorsa buna zayıf varlık kümeli denir.
- Ör: Üniversite-fakülte ilişkisinde, bir fakülte üniversite olmadan olamayacağı için ve aynı fakülte isminde başka üniversitelerde fakülte olabileceği için fakülte varlık kümeli zayıf varlık kümeleridir.



Zayıf Varlık Kümelerinin Gerçeklenmesi



Universite (univNo, univAdi)

Fakulte (univNo, fakulteNo, fakulteAdi)

Zayıf varlık kümeleri
çift çizgili dikdörtgen
ile gösterilir.

Gelecek Ders

- Kodlama Teknikleri
- Yazılım Kalitesi
- Test Teknikleri
- Yeni Sisteme Geçiş
- Bakım

Sistem Analizi ve Tasarımı

8.Ders

Göksel Biricik

Bu Derste

- Kodlama
- Yazılım Kalitesi
- Test Teknikleri
- Yeni Sisteme Geçiş
- Bakım

Kodlama

- Tasarım tamamlanıp diyagram, algoritma ve sözde kodlar oluşturulduktan sonra yapılır.
- Kodlama spesifikasyonlara uymalı, kolay okunabilmeli, düzenlenebilmeli, test edilebilmeli, değiştirilebilmelidir.
- Belirlenen kodlama standartlarına uyulmalıdır.
 - Açıklama satırları kullanılmalıdır.
 - Kod yazım deseni kullanılmalıdır.
- Anlamlı isimlendirmeler yapılmalıdır.

Programlama Dili ve Ortamı

- Gereksinimlere uygun olacak dil(ler) fizibilitede seçilip analiz aşamasında kesinleştirilmiştir.
- Dil Seçiminde dikkat edilmesi gereken faktörler
 - Genel uygulama alanı
 - Algoritma ve veri yapılarının karmaşıklığı
 - Yazılımın kullanılacağı ortam
 - Uygulama koşulları
 - Personelin bilgi düzeyi
 - Yapılacak yatırım miktarı
 - Müşteri koşulları

Yazılım Kalitesi

- Kalite: işlevsel gereksinimlere, geliştirme standartlarına ve beklenen tüm özelliklere tamamen uygun yazılım
- Kalite Faktörleri:
 - Doğruluk
 - Güvenilirlik
 - Verimlilik
 - Güvenlik
 - Kullanışlılık
 - Hata bulma kolaylığı
 - Esneklik
 - Sınama kolaylığı
 - Taşınabilirlik
 - Tekrar kullanılabilirlik
 - Bağlanabilirlik

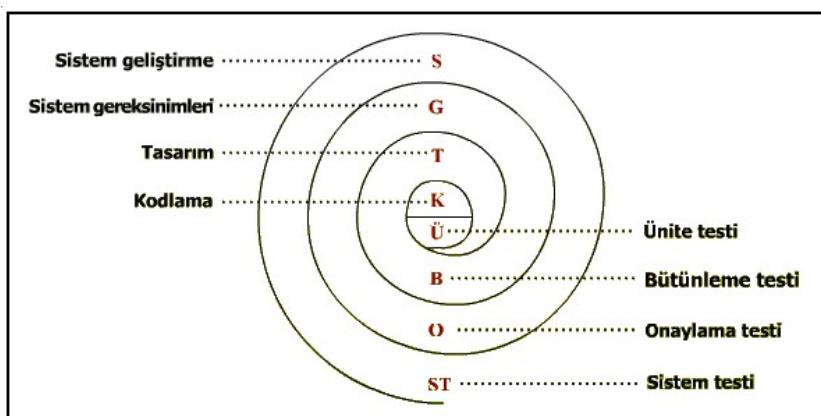
Yazılımda Kalitenin Sağlanması

- Planlama aşamasında kalite kontrolü yöntem ve araçları belirlenmeli
- Geliştirme sürecinde durak noktalarında yapılanlar gözden geçirilmeli
- Kaynak program sınanmalı
- Gerçekleştirilmiş projenin gereksinimleri karşılaması gözden geçirilmeli
- Bunlar plan dahilinde yapılmalı:
 - Yazılım inceleme planı
 - Kaynak programı sınama planı
 - Kabul muayene planı

Yazılımın Test Edilmesi

- Hataları bulmak için yapılır.
- **Fonksiyonel Test:** Girdi-İşlem-Sonuç üçlüsünün doğruluğu test edilir. Uç değer analizi de yapılır.
- **Performans Testi:** Yanıt süreleri, iç-dış bellek kullanımları, iletişim hızları vs. ölçülür. Darboğazlar belirlenir.
- **Dayanıklılık Testi:** Aşırı yüklenme, iletişim darboğazı, kullanıcı yüklenmesi gibi durumlarda sistemin tepkisi ölçülür.
- **Yapısal Test:** Sistemin iç işletimi sınanır. Alt programların mantıksal çalışma yolları denetlenir.

Test Adımları



Test Adımları

- **Ünite Testi:** Her modüle ayrı uygulanarak kodun doğruluğu test edilir.
- **Bütünleme Testi:** Modüllerini bağlayarak sistemin oluşturulması sırasında yapılır. Bütün olarak ya da arttırmalı yapılabilir.
- **Onaylama Testi:** Gereksinimleri karşılama derecesi test edilir.
- **Sistem Testi:** Sistemin bütün öğeleri hep birlikte test edilir. Hatalar için düzeltme, güvenlik, dayanıklılık, performans testleri yapılır.

Yeni Sisteme Geçiş

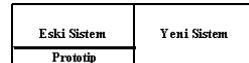
- **Geçiş Adımları:**
 - Ön tasarımda belirlenen donanımın kurulumu
 - Ağ yapısının oluşturulması
 - Kayıtların yeni sisteme aktarılması
 - **Eğitim**
 - Sistemle ilgilenenler
 - Uç kullanıcılar
 - Üst yönetim
 - Devreye alma

Geçiş Yöntemleri

- Doğrudan geçiş



- Prototip geçiş



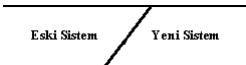
- Paralel geçiş



- Dağıtılmış geçiş



- Dereceli geçiş



Bakım

- Yazılım yaşayan bir süreçtir, evrimleşir.
- Kullanıma geçişten sonra yazılımdaki değişikliklere bakım denir.
- Zaman içinde ihtiyaçlar değişir. → İyileştirici bakım
- Altyapı değişimi olabilir. → Uyarlayıcı bakım
- Testlerde fark edilmeyen hatalar olabilir.
→ Düzeltici bakım

Gelecek Dersler

- Proje Sunumları