

# Yazılım Mühendisliğine Giriş

2018 GÜZ

# Dört Temel Yazılım Geliştirme Metodolojisi

## I)Yapısal Analiz ve Tasarım 1960 lı yılların sonu 1970 li yıllar

- ❖Fonksiyonel ayrıştırma (functional decomposition) ve veri akış analizi (data-flow analysis) yazılım geliştirme araçlarının temelini oluşturur (modeling tools).

## II) Nesneye Yönelik Analiz ve Tasarım (Object-oriented analysis and design) 1980li ve 1990lı yıllar

- ❖Birleştirilmiş Modelleme Dili (Unified Modeling Language-(UML) yazılım geliştirme araçlarının temelidir.

## III)Çevik Yazılım Geliştirme (Agile Software Development ) 1990lı yılların sonu ve 2000li yıllar

- ❖Yazılım endüstrisinde yazılım ürünü geliştirilirken “lightweight” yaklaşım

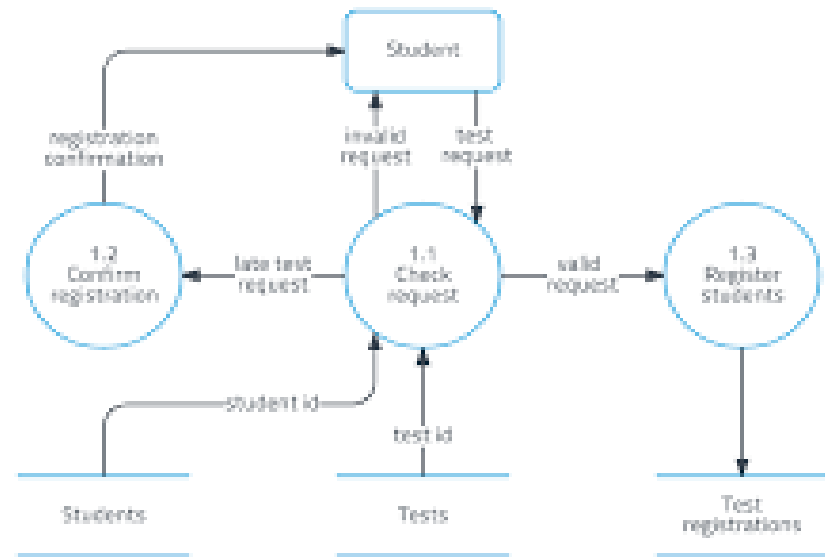
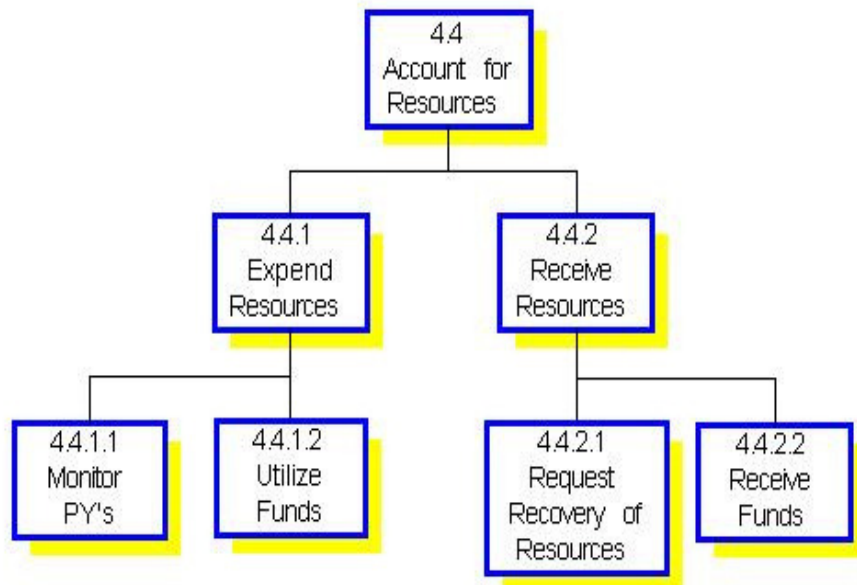
## □ Alana özel yazılım geliştirme (Aspect-Oriented Software Development ) 2000 li yıllar

- ❖ Diğer yazılım geliştirme metotları terkedilmemiştir; sadece çalışılan alanla ilgili olan metotlar kullanılır.

# Fonksiyonel Ayırıştırma -Veri Akış Diyagramı Örnekleri

## Kavramsal Çözümleme (Conceptual Solution) Görsel Çözümleme (Visual Solution)

CONCEPTUAL LEVEL FUNCTION CHART

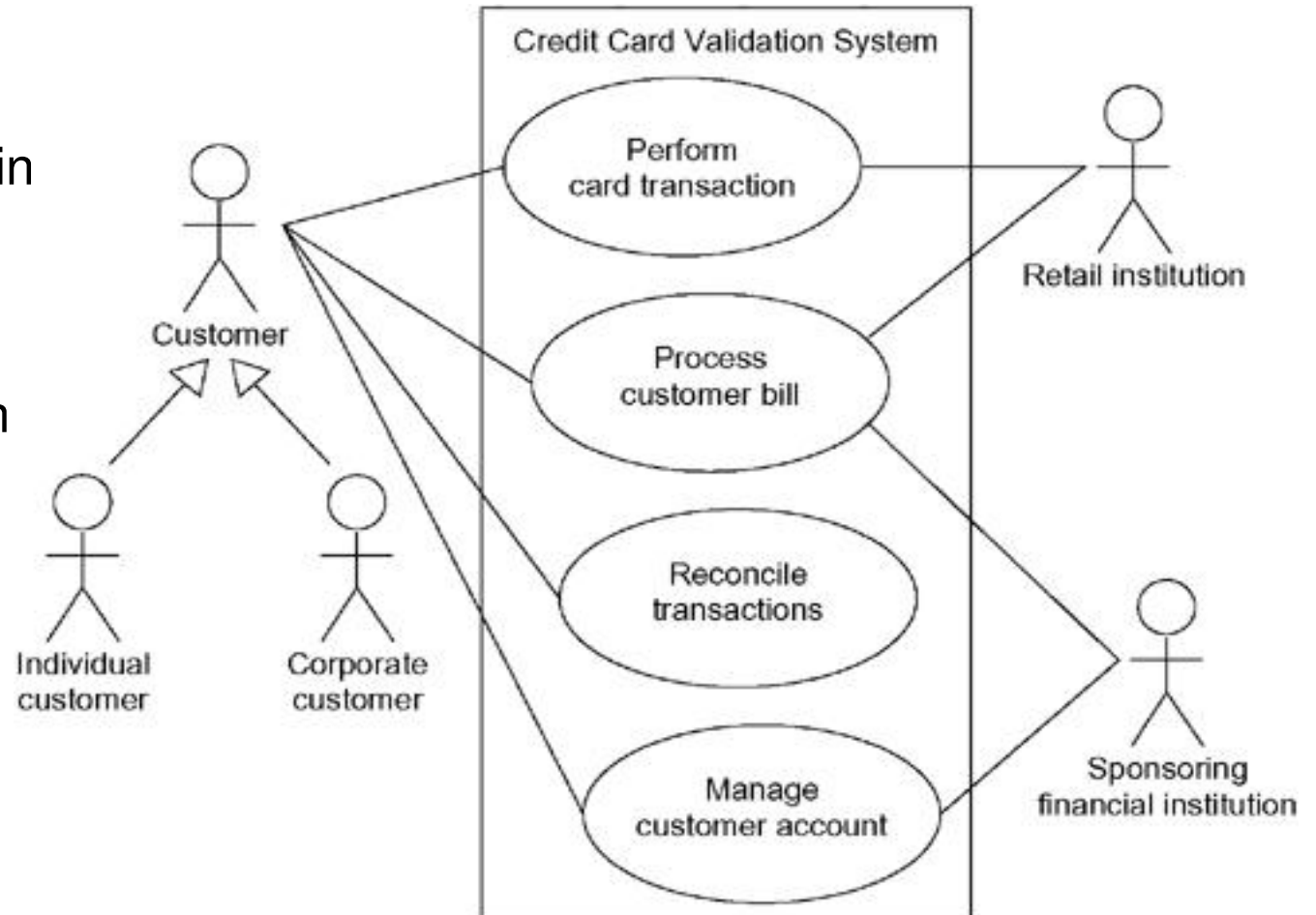


# UML (Unified Language Processing – Birleştirilmiş Modelleme Dili Örneği)

Kavramsal Çözümleme (Conceptual Solution)

Görsel Çözümleme (Visual Solution)

- ❑ Nesneye yönelik tasarımın analiz aşamasında müşterinin isterlerinin görsel çözümüdür.
- ❑ Bu aşamadaki problemin çözümünün amacı uygulanana modelin tasarım aşamasına geçişi kolaylaştırmaktır.



# UML Örneği Sınıf Diyagramı ve Implementasyon



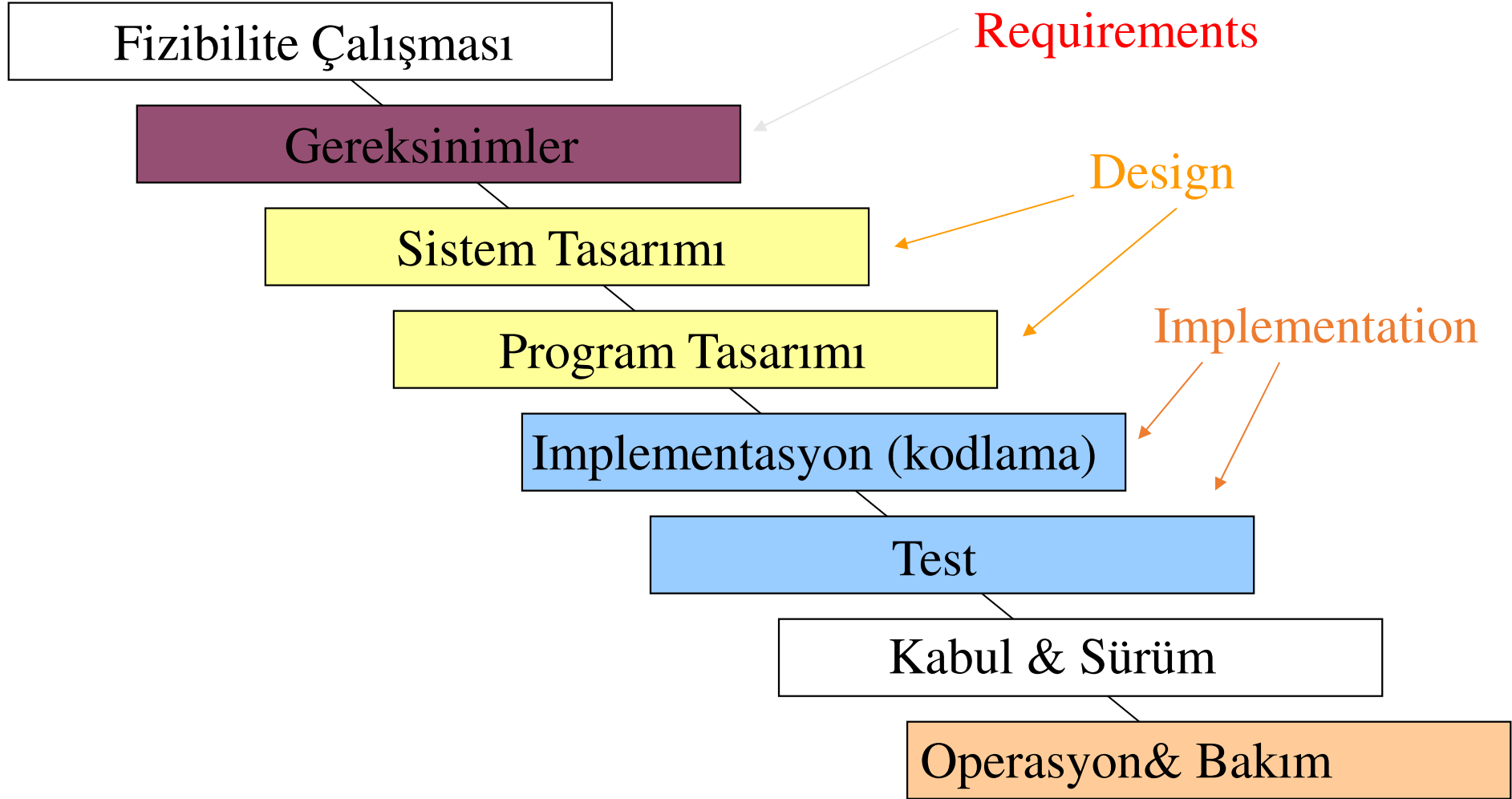
- ❑ Nesneye yönelik tasarımda analiz aşamasında müşteriye yakın çözüme ulaşılır.
- ❑ Nesneye yönelik tasarımın tasarım aşamasında ise sınıflar ve sınıflar arasındaki ilişkiler incelenir. Bu aşama implementasyona, yani kodlamaya geçişi gerçekleştirir.
- ❑ Nesneye yönelik yöntemlerin çoğu, kod yazıldıkça doğruluğunu kontrol etmek üzere test işlemi gerçekleştirir.
- ❑ Bunun için de uygun data değerleri girilerek mini kodun tümüyle doğru çalışıp çalışmayacağına bakılır.

```
class Employer
{
    Employee[ ] employees;
    .....
}

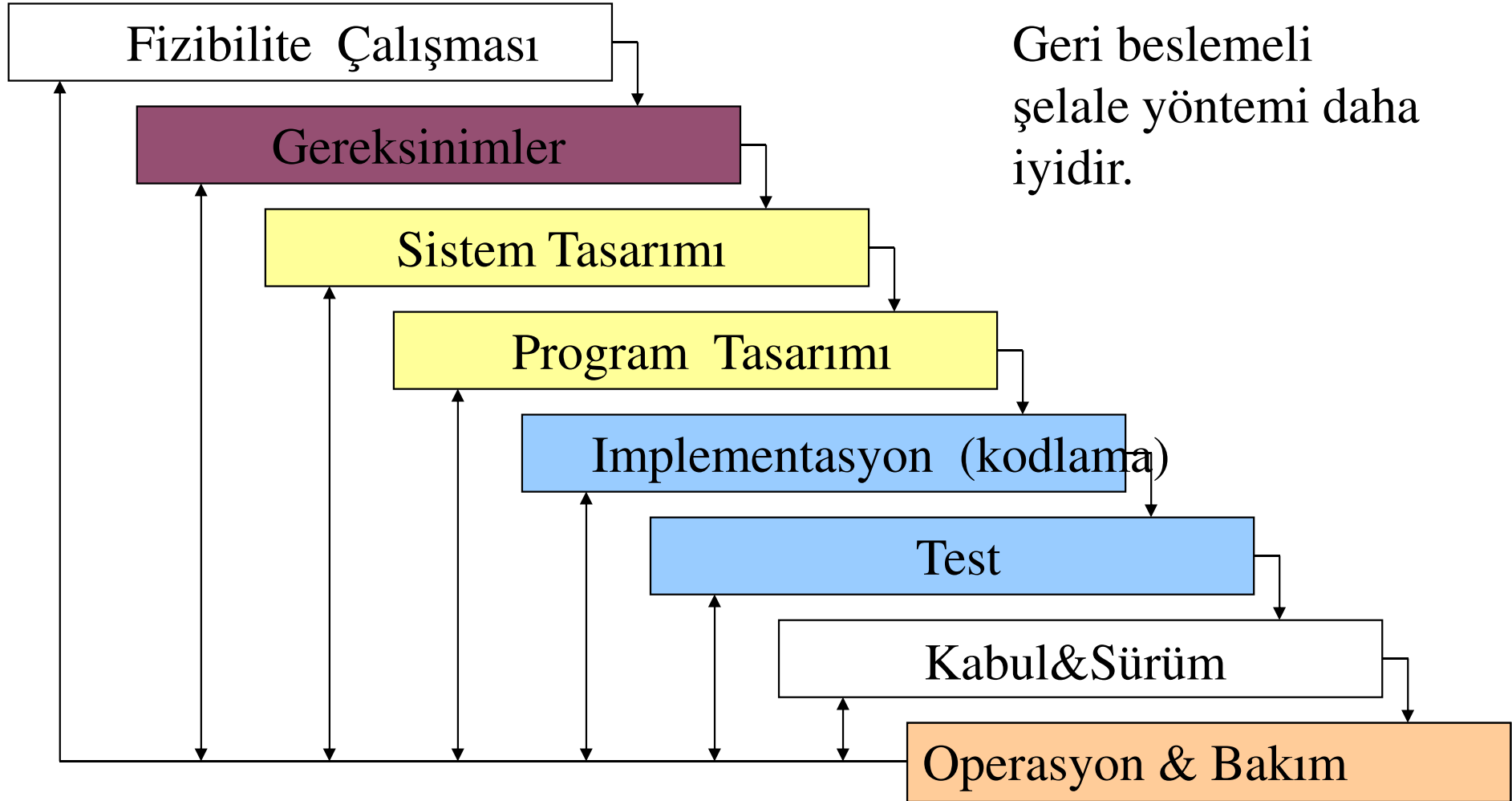
class Employee
{
    Employer[ ] employers;
    .....
}
```

# Yazılım Geliştirme Modellerine Genel Bakış

# Sıralı Yazılım Geliştirme (Sequential Development) Şelale Yöntemi ( Waterfall Model)



# Deđiştirilmiř řelale Yöntemi (Modified Waterfall Model)

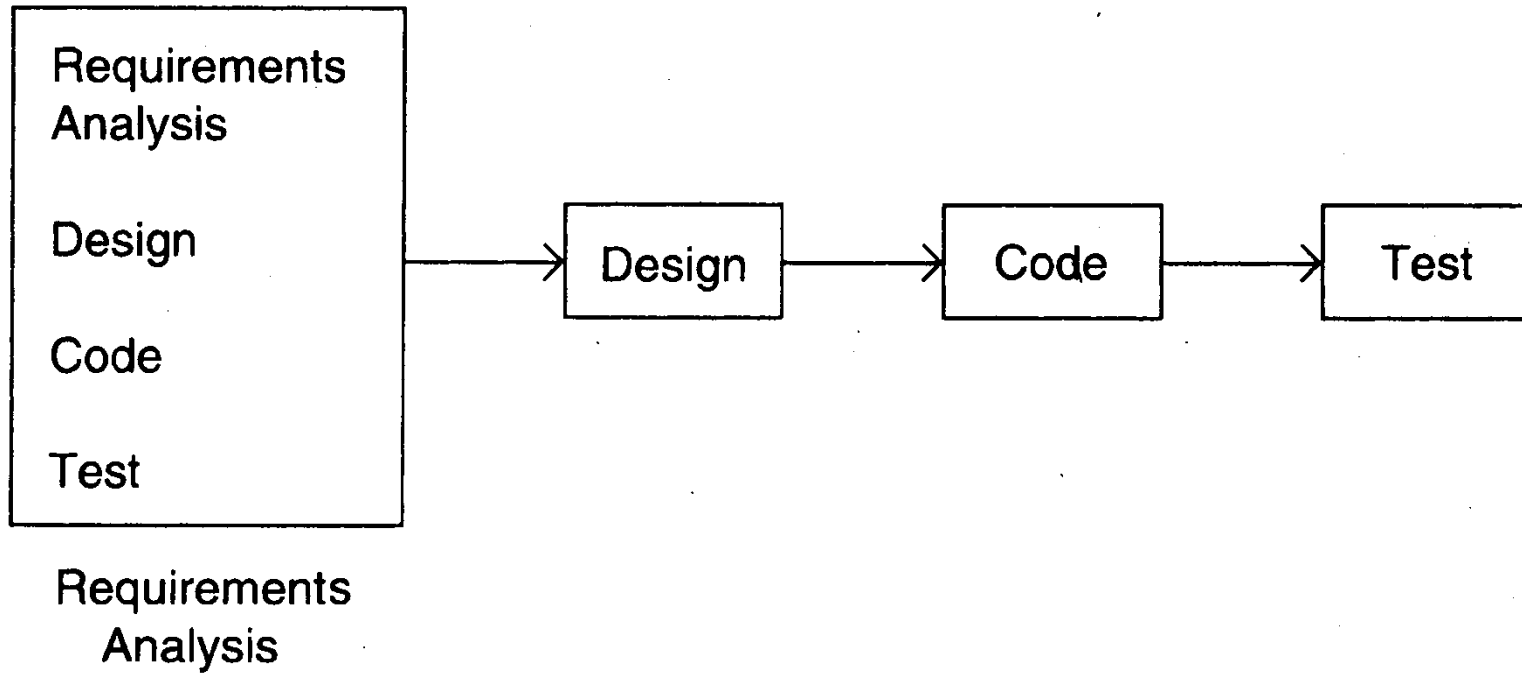




# Prototipleme Süreç Modeli

- ❑ Prototipleme modeli şelale yönteminde **isterlerin** belirlenmesindeki **sınırlamaları** (limitleri) hafifletir.
- ❑ Mevcut durumda bilinen gereksinimlere göre prototip olarak problem çözümüne devam edilir.
  - ❖ Oysa şelale yönteminde analiz aşamasından tasarım aşamasına geçmeden isterler dondurulmak zorundadır.
- ❑ Prototipleme tasarım, kodlama ve test aşamaları ile devam eder.
  - ❖ Her bir aşama formal olarak ve bütünüyle gerçekleştirilir.

# Prototipleme Süreç Modeli



❑ Gereksinimler değişime şelale yöntemine göre daha fazla açıktır.

❖ Yine de gereksinimlerin her aşamada değiştirilmesi mümkün değildir.

❑ Prototipleme modeli tamamı bilinmeyen gereksinimler için çözüm için projenin başlangıcında seçilmiş olabilir.

Yazılım ve Bilgi  
Teknolojilerinde  
Big Bang Boom Teorisi\*

Copyright © 2014 The Standish Group International, Inc.\*

# Büyük Projelerde Big Bang Teorisi

- ❑ Big Bang Teorisi önceden geliştirilmiş büyük yazılım ve bilgi teknolojileri projelerinden yararlanarak yeni ürünlerin geliştirilmesi ilkesinden yola çıkar .
  - ❖ Böylece aynı amaca hizmet eden birçok projenin katılımcıları tarafından uluslararası düzeyde ortak olarak hizmet göreceği şekilde kullanılması sağlanmış olacaktır.
- ❑ Çalışan bir yazılım ürününün tüm **katılımcılarının (iştirakçiler-stakeholders)** işbirliği ile ortaya çıkması ve çalışması çok önemlidir.
- ❑ Big Bang kavramının en temel özelliği projenin tüm fonksiyonelliği ile tamamlanmış olarak **belirli bir tarihte teslim edilmiş** olmasıdır.

Kısaca:

- ❑ Big Bang teorisinin temel hedefi geliştirilmesi hedeflenen bir büyük projede de kullanılarak **geliştirilecek yeni projeye ışık tutmasının** sağlanmasıdır.

# Proje İştirakçilerinin (Katılımcılar-Stakeholders) Önemi

- ❑ Projeye dahil bir **iştirakçi (stakeholder)** projede bulunan herhangi bir kişi (şahıs) ya da organizasyondur.
- ❑ Bu kişi ya da organizasyonun projenin çalışmasının devam etmesine ve tamamlanmasına **pozitif ya da negatif etkisi** olabilir.
- ❑ Proje yöneticisi ve proje geliştirme takımı projenin doğal katılımcılarıdır.

Bu tanım **Project Management Institute (PMI)** tarafından yapılmıştır ve 2013 yılından itibaren de bir **projenin çıktısı (outcome)** olarak kabul edilmektedir.

# Bir Projenin Farklı Katılımcıları (İştirakçileri)

- Proje yöneticisi
- Proje takımı
- Proje sponsoru
- Projeyi gerçekleştiren organizasyon
- Ortaklar (Partners)
- Müşteri (Client)
- Diğerleri
  - ❖ Bunlar **proje çıktılarında etkilenen her şey** olarak alınır.

# PMBOK

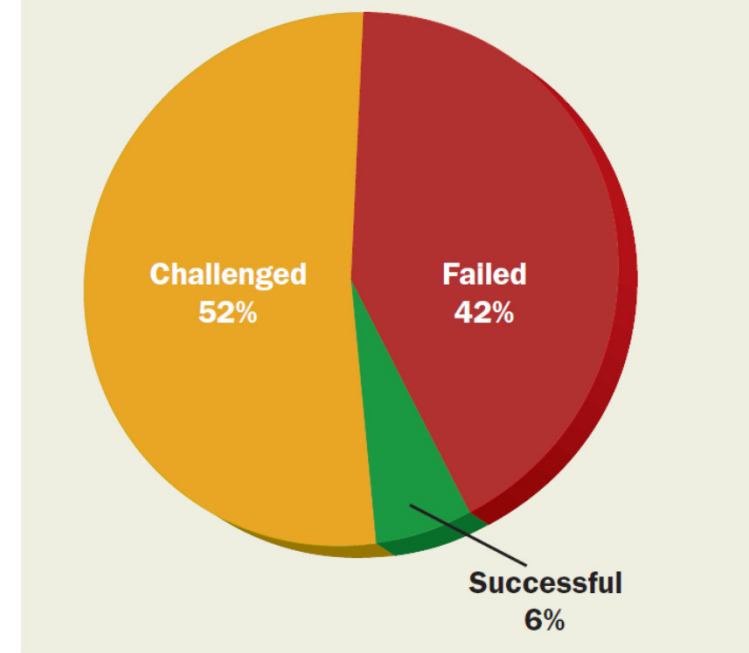
## Project Management Body of Knowledge

- ❑ PMI (Project Management Institute) gönüllülerden oluşan bir topluluktur ve endüstri standartlarını belirlemeyi amaçlar
- ❑ «A Guide to the Project Management Body of Knowledge» isimli kılavuz, American National Standards Institute (ANSI) tarafından da tanınmıştır.
- ❑ 2013 yılında adapte edilerek **PMBOK 5** versiyonu olarak proje yönetimi proseslerine uyarlanmıştır.
- ❑ Proje yönetimi kılavuzu (PMBOK ) ise proje yönetimi ile ilgili bir dizi standart tanımlar.

# Büyük Yazılım Projeleri (2003-2012)

CHAOS\* veri tabanında bulunan çok büyük yazılım projelerindeki göre değerlendirmeler:

- ❑ **Başarılı (successful)** projeler zamanında teslim edilen , maliyetine uygun ve implementasyonu gereksinimleri sağlayan projelerdir.
- ❑ **Problemlili (Challenged)** Projeler maliyetinin üzerinde sonuçlanan, geç teslim edilen ve/veya gereksinimlere cevap vermeyen projelerdir.
- ❑ **Başarısız (failed)** projeler ya proje tamamlanmadan iptal edilmiş ya da implementasyonu sonunda kullanılmayan projelerdir.



**CHAOS:** Standish Group'un veri tabanıdır. Standish Group bir araştırma kurumudur. Uluslararası ölçekteki IT (bilgi teknolojiler) projelerini izler ve sonuçları ile ilgili değerlendirme raporları yayınlar.



# Big Bang Teorisi ile Geliştirilen bir Proje

## Örneđi: NPAC

NPAC (Number Portability Administration Center) isimli proje önerisi ile ilgili yapılan çalışmalar sonunda Big Bang Teorisinin kullanılmasına karar verilmiştir.

- ❑ Projede uluslararası düzeyde sabit ve mobil telefon kullanıcılarının yerlerini (konumlarını) ve operatörlerini değiştirdiklerinde telefon numaralarını korumalarını sağlayan bir sistem geliştirilecektir.
- ❑ Önerilen projenin CHAOS veri tabanındaki diğer projelerle karşılaştırılarak geliştirilmesi hedeflenmektedir. Bu nedenle Big Bang Teorisi kullanılıyor olarak yorumlanmaktadır.

# Big Bang Teorisi ile Geliştirilen bir Proje Örneği: NPAC

- Yeni proje, **önceki proje Neustar Federal Communications Commission (FCC)** tarafından belirlenen kurallara göre geliştirilmiştir .
  - ❖ Yararlanılacak mevcut projenin bu standartlara uygun olarak geliştirilmiş olması, yeni geliştirilecek proje için önemli bir yapı taşı teşkil edecektir.
- Yeni projede, **NPAC**, teslim edilecek yazılımın, yürütülecek hizmetlerin ve operasyonların tümünün **Big Bang Teorisi** içerisinde yapılması hedeflenmiştir.
  - ❖ Çünkü iştirakçiler mevcut işleyen yapı olan Neustar sistemi düzeyinde hizmet ve fonksiyonellik talep etmektedirler.

# Yazılım Projelerinde Standish Grup Katkısı

## Örneđi : NPAC

□ Önerilen yeni **NPAC sisteminin geliřtiricisi** ve tüm çevre birimlerinin

- ❖ projenin büyüklüğü,
- ❖ karmaşıklığı,
- ❖ farklı tipteki iřtirakçilerin olması ,
- ❖ birbirleri ile bağlantılı pek çok alt sistemler,
- ❖ yoğun test işleme olan gereksinim,
- ❖ projenin tamamlanma tarihindeki sıkıntılar

gibi pek çok kavramda yararlanılacak mevcut sistem ile benzerlikleri bulunmaktadır.

Bu yapı **Big Bang** olarak adlandırılır.

# Alternatif Proje Yaklaşımı

- ❑ **İteratif döngü** önerilmektedir.
- ❑ Küçük gruplar fonksiyonelliği gerçekleştirir ve onların geri bildirimini (feedback) daha fazla fonksiyonellik gerçekleştirmek üzere kullanılır.
- ❑ **İteratif geliştirme** bir dizi küçük projeden meydana gelir.
- ❑ 90 lı yılların başlarında Standish Group **iteratif ürün** geliştirme yöntemlerini yayınladıktan sonra bu yöntem **Scrum** gibi pek çok çevik (agile) metodolojilerin kaynağı olmuştur.
- ❑ Amazon, eBay, WebEx ve Google ürünlerini ve organizasyonlarını oluşturmak üzere iteratif yöntem kullanmışlardır.
- ❑ **Yeni firmalar** (start-up companies) dünya ölçeğinde «minimal kabul edilebilir ürün» politikası ile çalışmaya başlarlar.

# NPAC Ürününe adım adım erişim

- ❑ **Neustar ilk NPAC** ürününü geliştirdiğinde bu proje Big Bang tipinde bir proje idi. Aslında yeni proje
  - ❖ kapsam olarak (uygulama alanı) oldukça küçüktü
  - ❖ **Neustar** pek çok problemler içeriyordu
- ❑ **17 yıl sonra Neustar** geliştirdiği yeni sistemi (fonksiyonelliği ) endüstrinin tüm gereksinimlerini karşılayacak şekilde tüm hizmet düzeylerinde başarı ile çalıştırdı
- ❑ Çok yoğun bir iteratif geliştirme yapısı içerisinde telekom endüstrisi katma değerini çok arttırdı.
- ❑ **Gelecekte** önerilecek yeni bir NPAC üreticisi ,doğru bir Big Bang implementasyonu ile sonuçlanmış olan, bu süreçleri takip edecektir.

# Yeni NPAC Projesinin Çözümü

- ❑ Mevcut NPAC sistemini yeniden inşa etmek için CHAOS veri tabanındaki büyük projeler karşısında değerlendiriliyor.
- ❑ Tablo, veri tabanındaki benzer 100 den daha fazla projenin sonuçlarını yansıtıyor.
- ❑ Bu tabloda en yüksek başarısızlık oranı elde ediliyor.
- ❑ Bu sonuçlar **yeni bir NPAC projesi ile ilgili tahminlerdir.**



Projenin başarısının da projenin büyüklüğü (size), karmaşıklığı (iştirakçilerin sayıları ile birlikte yüzlerce özellik karmaşıklığı belirler), kullanılan metodoloji, çalışanların becerileri, kullanılan araçlar ve teslimat (delivery) önemlidir.

# Projenin Başarısında Ürünün Dağıtımının Önemi

**Yeni üreticiler** geliştirdikleri ürünün fonksiyonelliğini yansıtmakta **deneyimsiz olabilir**.

Müşteri ile iletişime geçmekte yetersiz kalabilir.

**Big Bang** dağıtım (delivery) ilkeleri bu problemi çözecektir.

# Projenin Başarısında Endüstrinin Önemi

Yeni projenin hangi endüstriye ait olduğu proje başarısını etkiler.

Endüstri çevresel (environmental) bir faktördür .

Ayrıca:

- ❑ Projenin karmaşıklığı, geliştirme tipi, uygulama alanı başarı oranını istatistik sonuçlarına göre bir miktar düşürebilir.
- ❑ Örneğin %6 bir başarı yüzdesi elde edilmiş ise bu oran %4 e düşebilir.



# NPAC Projesi ile İlgili Zaman Aşımaları

- ❑ Bu istatistiki değerlere göre Standish Grup ortalama iki katı sürede tamamlanacağını öngörmektedir.
- ❑ Diğer bir ifade ile; projenin geliştirilmesi için iki yıl öngörülmüş ise dört yılda tamamlanacağı beklentisidir.
- ❑ Zaman ve maliyetin fazla aşımını önlemek için bazı süreçlerden vazgeçilebilir.

