

# Understand the Problem

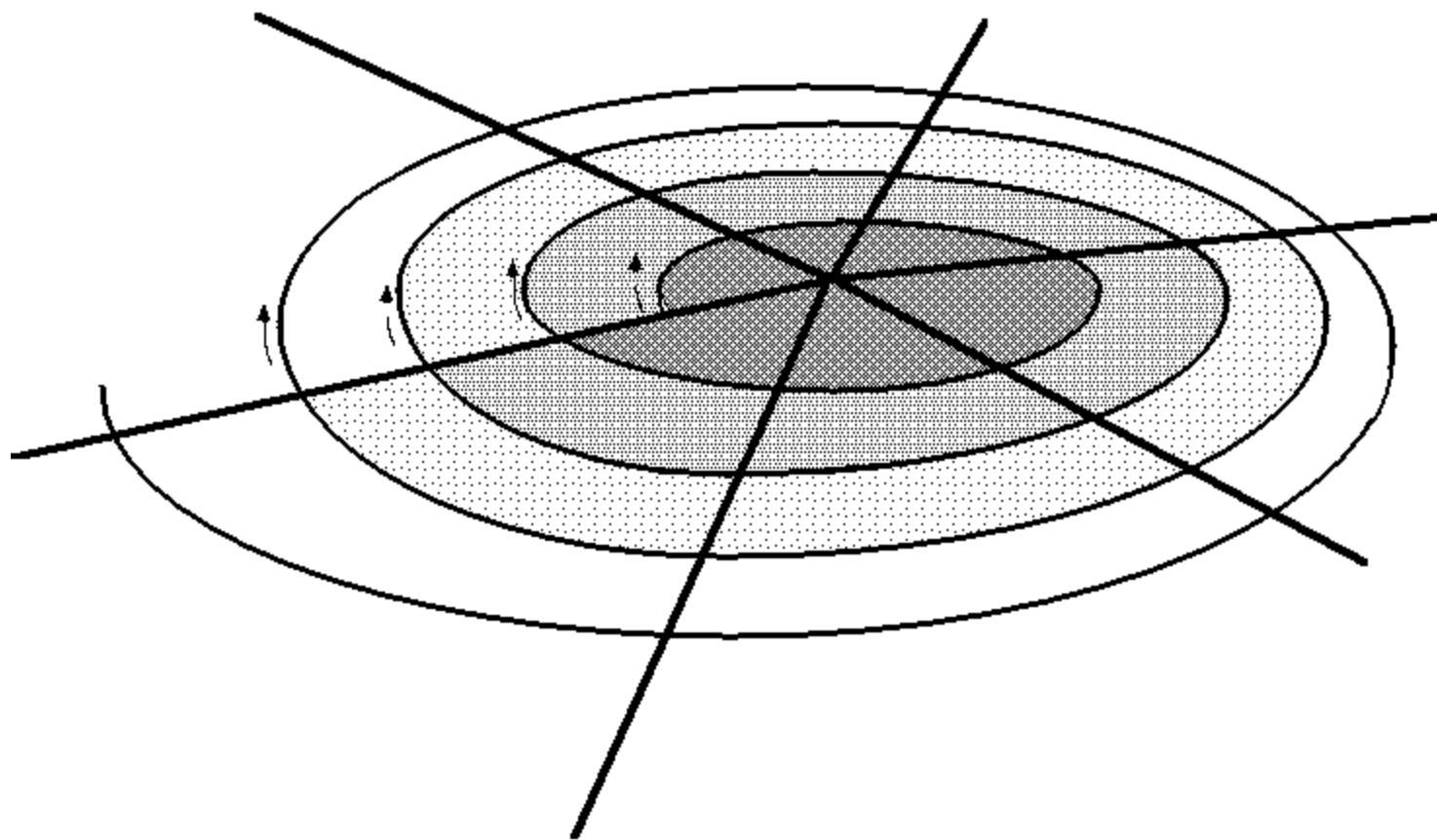
- *Who has a stake in the solution to the problem?* That is, who are the stakeholders?
- *What are the unknowns?* What data, functions, and features are required to properly solve the problem?
- *Can the problem be compartmentalized?* Is it possible to represent smaller problems that may be easier to understand?
- *Can the problem be represented graphically?*
- Can an analysis model be created

# Safehome

- Every software project is precipitated by some
- business need—
  - the need to correct a defect in an existing application;
  - the need to adapt a ‘legacy system’ to a
- These slides are designed to accompany *Software Engineering: A Practitioner’s Approach, 7/e*
- (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman. 24
- changing business environment;
  - the need to extend the functions and features of an
- existing application, or
  - the need to create a new product, service, or system.

# Process Assessment and Improvement

- **Standard CMMI Assessment Method for Process Improvement (SCAMPI)** — provides a five step process assessment model that incorporates five phases: initiating, diagnosing, establishing, acting and learning.
- **CMM-Based Appraisal for Internal Process Improvement (CBAIPI)**—provides a diagnostic technique for assessing the relative maturity of a software organization; uses the SEI CMM as the basis for the assessment
- **[SPICE—The SPICE (ISO/IEC15504)** standard defines a set of requirements for software process assessment. The intent of the standard is to assist organizations in developing an objective evaluation of the efficacy of any defined software process. [
- **ISO 9001:2000 for Software**—a generic standard that applies to any organization that wants to improve the overall quality of the products, systems, or services that it provides. Therefore, the standard is directly applicable to software organizations and companies.



## **HIZLI UYGULAMA GELİŞTİRME (RAD: Rapid Application Development)**

- Kısa geliştirme çevrimleri üzerinde duran artımsal bir model.
- Ön koşullar:
  - Uygulamanın yaklaşık/ortalama 3 aylık bölümlere ayrılabilmesi,
  - Yeterli sayıda bölümün eşzamanlı ilerlemesinin sağlanabilmesi,
  - Yazılımın bileşenlerden kurulabilmesi.
- Eksiler:
  - Büyük ölçekli çalışmalarda yeterli sayıda bölümü eşzamanlı ilerletebilecek sayıda çalışanın bulunamaması.
  - Çalışanlar hıza uyum sağlayabilmelidirler.
  - Yüksek teknik risklere uygun değil.

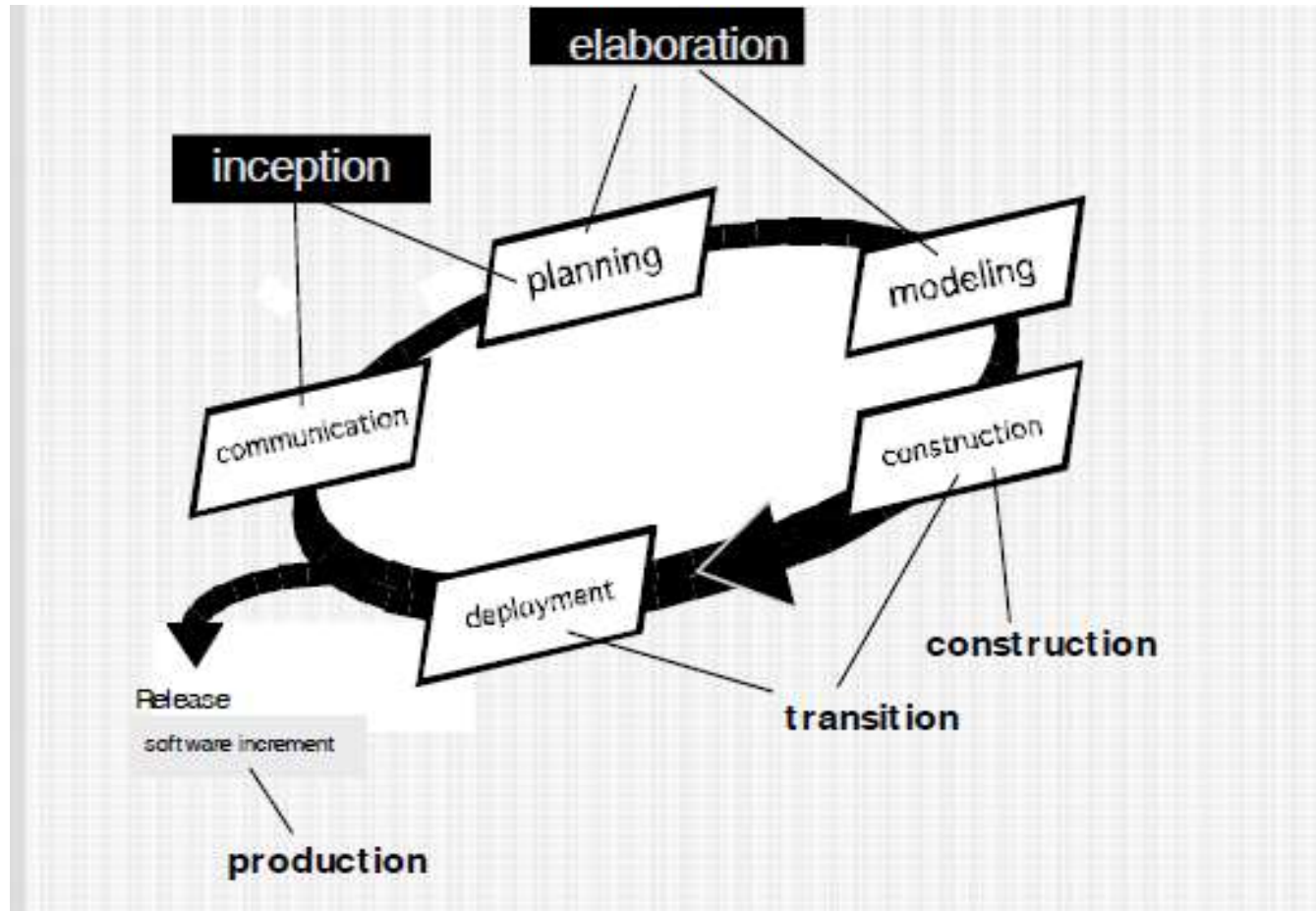
## **YAZILIM GELİŞTİRME SÜREÇLERİ**

- Artılar:
  - Bu sürece uygun yazılım projelerinde verimliliğin artması.
- Sonuç:
  - Prototip geliştirmede kullanılması veya ana fikirlerinin diğer süreçlere uygulanması yerinde olacaktır.

## **BİLEŞEN TABANLI (Component Based) UYGULAMA GELİŞTİRME**

- Uygulamanın hazır yazılım bileşenlerinden oluşturulmasını öngörür.
- Aşamaları:
  - Konu alanı mühendisliği (Domain Engineering)
  - Aday bileşenlerin sınıflandırılması ve seçilmesi (Qualification)
  - Seçilen bileşenlerin kendi yazılımımıza uyarlanması (Adaptation)
  - Bileşenlerin bir araya getirilmesi (Composition)
- Eksiler:
  - Uygun bileşenlerin bulunması gerekliliği (her zaman bulunmaz)
  - Bileşenlerin uyarlanması gerekliliği (göründüğü kadar kolay olmayabilir)
- Artılar:
  - Yeniden kullanımın özendirilmesi (azalan giderler?)
- Sonuçlar:
  - Özellikle hızlı uygulama geliştirme olmak üzere, ana fikirleri çeşitli süreçlere uygulanabilir.

# Unified Process



# Unified Process

## Inception phase

Vision document  
Initial use-case model  
Initial project glossary  
Initial business case  
Initial risk assessment.  
Project plan,  
phases and iterations.  
Business model,  
if necessary.  
One or more prototypes

## Elaboration phase

Use-case model  
Supplementary requirements  
including non-functional  
Analysis model  
Software architecture  
Description.  
Executable architectural  
prototype.  
Preliminary design model  
Revised risk list  
Project plan including  
iteration plan  
adapted workflows  
milestones  
technical work products  
Preliminary user manual

## Construction phase

Design model  
Software components  
Integrated software  
increment  
Test plan and procedure  
Test cases  
Support documentation  
user manuals  
installation manuals  
description of current  
increment

## Transition phase

Delivered software increment  
Beta test reports  
General user feedback



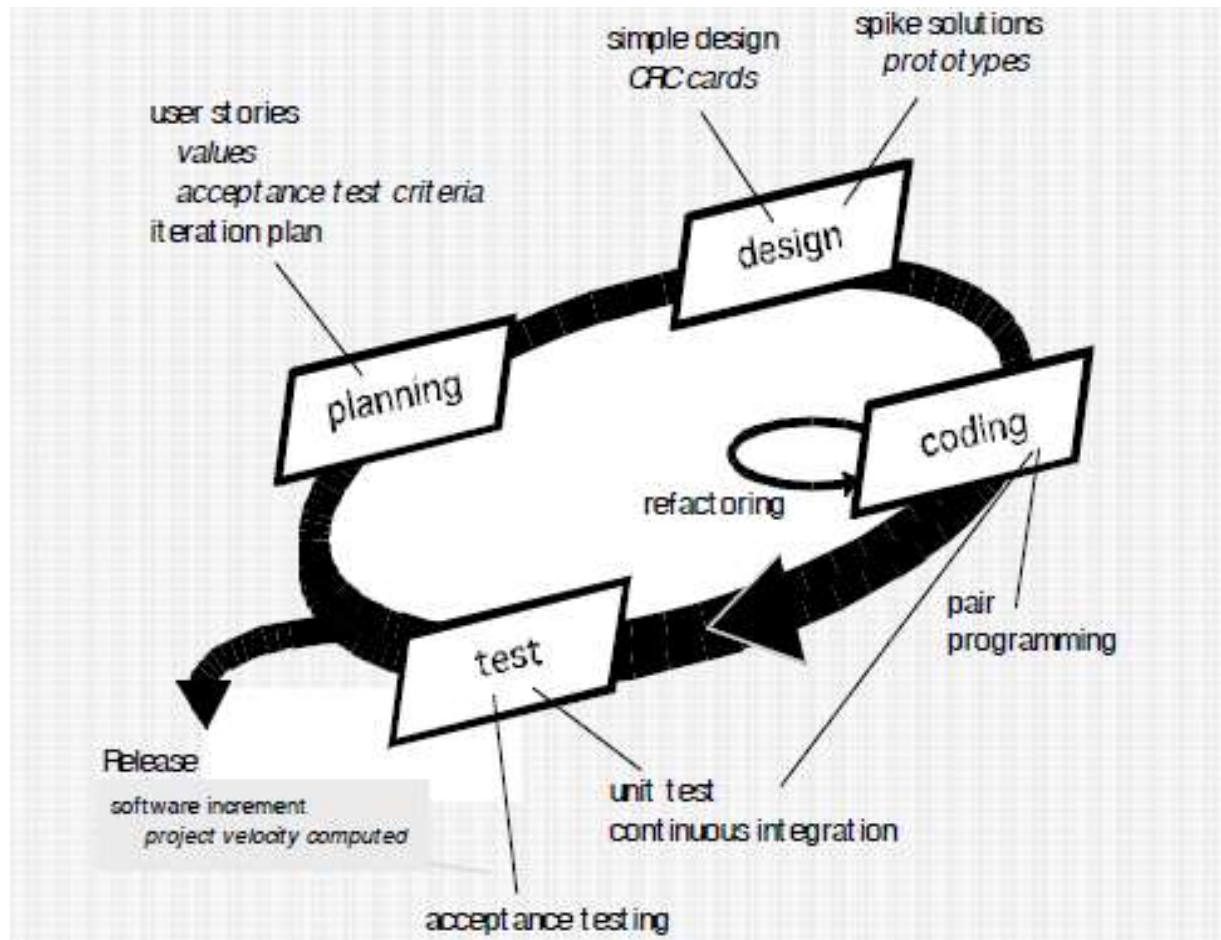
# Agility Principles

- 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- 3. Deliver working software frequently,
- 4. Business people and developers must work together daily throughout the project.
- 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation

# Agility Principles

- 7. Working software is the primary measure of progress.
- 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 9. Continuous attention to technical excellence and good design enhances agility.
- 10. Simplicity – the art of maximizing the amount of work not done – is essential.
- 11. The best architectures, requirements, and designs emerge from self-organizing teams.
- 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior

# Extreme programming



# CRC

CLASS  
RESPONSIBILITY  
COLLABORATION

## YAZILIM GELİŞTİRME SÜREÇLERİ

### ÇEVİK (Agile) SÜREÇLER

Örnek CRC kartı:

Sınıf adları

<b>Sınıf: Satış</b>	
Kasada yapılan ödemeyi simgeleyen sınıf.	
<b>Üst Sınıf(lar):</b> Yok	
<b>Alt Sınıf(lar):</b> Yok	
<b>Sorumluluk:</b>	<b>İşbirlikçi:</b>
Satışın yapıldığı tarih ve saati saklamak	
Yapılan ödeme tutarını saklamak	Ödeme
Satılan malların listesine erişim	Mal

# Scrum ve Diğer Modeller

Kriter	Waterfall	Spiral	İterative	Scrum
Tanımlı süreçler	Gerekli	Gerekli	Gerekli	Planlama ve kapatma aşamasında
Final ürün	Planlamada tanımlanmış	Planlamada tanımlanmış	Proje esnasında şekilleniyor	Proje esnasında şekilleniyor
Proje maliyeti	Planlama esnasında hesaplanıyor	Kısmen değişken	Proje esnasında şekilleniyor	Proje esnasında şekilleniyor
Tamamlanma tarihi	Planlama esnasında belirleniyor	Kısmen değişken	Proje esnasında şekilleniyor	Proje esnasında şekilleniyor
Dış çevreye duyarlılık	Sadece planlamada	Planlama öncelikli	İterasyon sonlarında	Her zaman
Takım esnekliği,yaratıcılığı	Limitli,fazla yaratıcılık ve esneklik desteklenmiyor	Limitli,fazla yaratıcılık ve esneklik desteklenmiyor	Limitli,fazla yaratıcılık ve esneklik desteklenmiyor	İterasyonlar boyunca yaratıcılıkta ve esneklikte limit yok
Bilirkişi desteği(Harici)	Proje eğitimi esnasında	Proje eğitimi esnasında	Proje eğitimi esnasında	Gerekirse proje içinde takıma bile dahil edilebiliyor
Başarı olasılığı	Düşük	Orta düşüklükte	Orta Seviye	Yüksek

# Requirements Engineering

- Inception—ask a set of questions that establish ...
  - basic understanding of the problem
  - the people who want a solution
  - the nature of the solution that is desired, and
  - the effectiveness of preliminary communication and collaboration between the customer and the developer
- Elicitation—elicit requirements from all stakeholders
- Elaboration—create an analysis model that identifies data, function and behavioral requirements
- Negotiation—agree on a deliverable system that is realistic for

## Aşırı Programlama (XP)

- Adımlar: Planlama – Tasarım – Kodlama – Sınama Artımsal Ürün
- Kodlama:
  - Önce birim sınamaları hazırlanır.
  - Programcı tarafından yapılan, sınıfların (NYP'de; yapısal'da fonksiyonlar, vb.'lerin) temel işlevselliklerini sınama amaçlı kod.
  - Sadece sınavı geçmeye yarayan kod yazılır (KISS).
  - Çift kişi ile kodlama:
    - Bir programcı eldeki sorunu çözerken diğeri çözümün genel tasarıma uygunluğunu gözetir ve kodlamanın takımın karar verdiği ölçütlere (kalite, vb.) uygunluğunu denetler.
- Sınama:
  - Birim sınamalarının otomatik çalıştırılması.
  - Müşterinin artımsal ürünü denemesi.



## Scrum:

- Adımlar: Görev Listesi – Koşu – İşlev Gösterimi
- Görev Listesi = Kullanıcı öyküleri.
- Önceliklendirilmiştir.
- Koşu:
  - Görev listesinin maddelerinden biri seçilir ve önceden belirlenmiş kısa bir süre içerisinde (Ör. 1-4 hafta) gerçekleşir.
  - Koşu süresince ekibin her gün yaptığı kısa (Ör. 15dk) toplantılar:
  - Proje lideri yönetir.
  - Cevaplanmaya çalışılan üç ana soru:
    - Son toplantıdan bu yana ne yaptınız?
    - Karşılaştığınız engeller nelerdir?
    - Yarınki toplantıda neleri başarmayı hedefliyorsunuz?
  - İşlev Gösterimi: Müşterinin en yeni işlevi veya o ana dek gerçekleşen tüm işlevleri sınaması.

## GEREKSİNİM MÜHENDİSLİĞİ

- Üzerinde çalışılmaya başlanacak projenin amaçlarını, boyutlarını ve etkilerini belirlemeye yönelik çalışmalardır.
  - Genel amaçlı proje yönetimi faaliyetleri arasında yer alan yapılabilirlik (feasibility) çalışmasına bir girdi olarak düşünülebilir.
- Müşteri ne istediğini bilmez mi? Gereksinimler zaten belli değil mi?
  - Çoğunlukla müşterinin kafasında sadece genel bir fikir vardır.
  - Yoruma açık ve ayrıntıları kesin çizgilerle belirlenmemiş gereksinimler projenin başarısızlığına davetiye çıkarır.
  - Kesin belirlenmiş gereksinimler bile zaman içerisinde değişebilir.
- Değişler:
  - Şeytan ayrıntıda gizlidir.
  - Yanlış veya eksik işi yapan mükemmel yazılım değil, doğru işi yapan iyi çözüm gereklidir.
  - SONUÇ: Gereksinim mühendisliği gerekli bir etkinliktir.

## GEREKSİNİM MÜHENDİSLİĞİ

- Gereksinimler, SRS (Software Requirements Specification) belgesi altında toplanır.
- SRS raporuna yönelik çeşitli standartlar mevcuttur.
  - IEEE 830-1998: SRS odaklı
  - ISO/IEC/IEEE 29148:2011 Systems and Software Engineering – Life Cycle Processes – Requirements Engineering
  - Ders kapsamında, gereksinim mühendisliği ve analiz bölümlerinde bu belgede yer alabilecek artefact'ların en yaygınları üzerinde çalışacağız.
  - Yazılım geliştirme sırasında kod dışında ortaya çıkardığımız her türlü metin ve şemaya "artefact" denilmektedir.

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Gereksinim mühendisliğinin genel adımları:
  - Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- Gereksinim mühendisliği adımları gerçekleştirilecek yazılımın doğasına ve kullanılan sürece göre düzenlenmelidir.
- Gereksinim mühendisliği adımları süresince yazılım ekibi ve müşteri birlikte çalışmalıdır.
  - Müşterinin bir ekibinin, yazılım geliştirme sürecinin mümkün olduğunca çok adımının bir parçası olması yararlıdır.

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- 
- Başlangıç:
    - Yazılım projesinin **ilk aşamalarının** başlatılıp başlatılmamasına karar verilen adımdır.
    - Müşterinin bir yazılım projesi başlatılmasını düşünmesine neden olan olaylar:
      - Yeni bir iş gereksiniminin belirlenmesi.
      - Mevcut iş süreçlerinde güçlüklerle karşılaşılması.
      - Müşterinin üst düzey karar vericileri ve astları arasında geçen kısa bir sözlü konuşma veya toplantı ile bile bir proje başlayabilir.
    - Bir uygulama yazılımı söz konusu ise:
      - Yeni bir pazarın veya hizmetin farkına varılması,
      - Yazılım şirketinin üst düzey karar vericileri ve teknik ekibinin sözlü konuşması ile yeni bir yazılım projesi başlatılabilir.

# Requirements Engineering

- Specification—can be any one (or more) of the following:
  - A written document
  - A set of models
  - A formal mathematical
  - A collection of user scenarios (use-cases)
  - A prototype
- Validation—a review mechanism that looks for errors in content or interpretation
  - areas where clarification may be required
  - missing information
  - inconsistencies (a major problem when large products or systems are engineered)
  - conflicting or unrealistic (unachievable) requirements.
- Requirements management

# Use Case

- A collection of user scenarios that describe the thread of usage of a system
- Each scenario is described from the point-of-view of an “actor”—a person or device that interacts with the software in some way
- Each scenario answers the following questions:
  - Who is the primary actor, the secondary actor (s)?
  - What are the actor’s goals?
  - What preconditions should exist before the story begins?
  - What main tasks or functions are performed by the actor?
  - What extensions might be considered as the story is described?
  - What variations in the actor’s interaction are possible?
  - What system information will the actor acquire, produce, or change?
  - Will the actor have to inform the system about changes in the external environment?
  - What information does the actor desire from the system?
  - Does the actor wish to be informed about unexpected changes?

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Tanımlama (Specification)
  - Bilgi Toplama (Elicitation)
  - Doğrulama (Validation)
  - İşleme (Elaboration)
  - Yönetim (Management)
  - Pazarlık (Negotiation)
- 
- Başlangıç aşamasında paydaşlar belirlenmelidir.
    - Paydaş: Gerçeklenecek sistemden doğrudan veya dolaylı olarak yararlanabilecek ve etkilenebilecek herkes.
      - Her paydaş sisteme farklı bir açıdan bakar.
      - Projenin başarısı veya başarısızlığı paydaşları farklı şekillerde etkiler.
      - Paydaşlara sorulacak sorularla belirlenmesi gerekenler:
        - Paydaşların bakış açıları,
        - Paydaşları etkileyebilecek nedenler,
        - Söz konusu etkilerin sonuçları.



## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- 
- Bilgi toplama aşamasının genel ilkeleri:
    - Gereksinimler hakkında ayrıntılı bilgiler, tüm paydaşların etkin katılımı ile elde edilmelidir.
    - Tüm paydaşların katıldığı toplantılar yapılmalıdır.
    - Toplantılara hazırlık ve katılım kuralları belirlenmelidir.
    - Gündem belirlenmelidir: Önemli konuları atlamayacak kadar sıkı, yaratıcılığı önlemeyecek kadar açık olmalıdır.
    - Düzeni sağlayacak ve tikanıklıkları çözecek bir oturum başkanı seçilir.
      - Her ne kadar bu deneyimle keskinleştirilecek bir sanat olsa da, yararlı kaynaklardan ön çalışma yapılabilir.
      - Örneğin: H.M. Robert, et.al., "Robert's Rules of Order - Newly Revised", Hachette Book Group, Sep. 2020 (12th ed).

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- 
- İşleme:
    - Bilgi toplama aşamasında toplanan 'ham' bilgilerin 'işlenmesi'.
    - Son kullanıcının ve diğer paydaşların yazılımla nasıl etkileşimde bulunacağını belirlenmesi ve ayrıntılandırılmasını amaçlar.
    - Etkileşimler, kullanım senaryoları ile gösterilir (ileride anlatılacak).
    - İşleme kimi bilgilerin genişletilmesi, kimi bilgilerin özetlenmesi şeklinde gerçekleşir.
    - Gereksinimlerin sınıflandırılması
      - Normal gereksinimler
      - Beklenen gereksinimler: Çok temel gereksinimleri kullanıcı belirtmeyebilir. Bunların da elde edilmesi gereklidir.
      - Heveslendirici gereksinimler: Müşteri beklentilerinin ötesinde ve varlığında müşteriye sevindirecek özellikler.

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- 
- Pazarlık:
    - Müşteriler sınırlı insan, zaman ve bütçe kaynakları çerçevesinde karşılanamayacak aşırı isteklerde bulunabilir.
    - Paydaşlar gereksinimleri farklı önem düzeylerinde görebilir.
    - Farklı paydaşların gereksinimleri birbiri ile çelişebilir.
    - Pazarlık sonucunda tüm paydaşların razı olacağı bir gereksinimler listesi elde edilir.

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- 
- Tanımlama:
    - Gereksinimler tanımlama aşamasında, pazarlık sonucu üzerinde uzlaşılan haliyle kağıda dökülür.
    - Tanımlama araçları:
      - Konuşma dili ile yazılmış belgeler
      - Kullanıcı senaryoları: Görülecek
      - Kullanım şemaları: Görülecek
      - Formel modeller (Matematiksel gösterim, işlenilmeyecek)
      - Bir ön ürün
    - Birden fazla tanımlama aracı birlikte kullanılabilir.

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- 
- Doğrulama:
    - Tanımlanmış gereksinimlerin tutarsızlıklara karşı sağlaması yapılır.
      - Gereksinimler açıkça ve yoruma yer bırakmayacak şekilde tanımlanmış mı?
      - Birbiri ile çelişen gereksinimler var mı?
      - Gereksinimlerde hatalar ve eksikler var mı?
      - Eksik gereksinimler var mı?
      - Gerçekçi olmayan gereksinimler var mı?
      - ...
    - Doğrulama yapma için önerilen temel yol teknik değerlendirmedir (Formal technical review, sınama teknikleri arasında anlatılacak).

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
  - Bilgi Toplama (Elicitation)
  - İşleme (Elaboration)
  - Pazarlık (Negotiation)
  - Tanımlama (Specification)
  - Doğrulama (Validation)
  - Yönetim (Management)
- 
- Yönetim:
    - Yazılım geliştirme süreci içerisinde gereksinimlerde değişiklikler olabilir:
      - Yeni gereksinimler eklenmesi
      - Mevcut gereksinimlerden bazılarının geçerliliğini yitirmesi
      - Gereksinimlerin önem sıralamasının değişmesi
      - Hatalı kestirimlerden dolayı bazı gereksinimlerden vazgeçilmesi
    - Gereksinimlerde ne tür değişikliklerin nasıl ve hangi şartlarla yapılabileceği, resmi bir sözleşme ile önceden belirlenebilir.
    - Gereksinimlerde değişiklikler müşteri ile karşılıklı anlaşma ile yapılmalıdır.

## GEREKSİNİM MÜHENDİSLİĞİ ADIMLARI

- Başlangıç (Inception)
- Bilgi Toplama (Elicitation)
- İşleme (Elaboration)
- Pazarlık (Negotiation)
- Tanımlama (Specification)
- Doğrulama (Validation)
- Yönetim (Management)
- Yönetim (devam):
  - Yazılım geliştirme süreci içerisinde gereksinimlerin gerçekleşmesinin (ve varsa gereksinimlerdeki değişikliklerin) izlenmesi gerekir.
  - İzleme tablolar aracılığı ile yapılır:
    - İzlenebilirlik tabloları (Traceability table).

	B1	B2	B3	...
G1	✓		✓	
G2		✓		
G3	✓			
...				

- G1,2,...: Gereksinimler
- B1,2,...: Sisteme çeşitli bakış açıları
  - Modüller, Paketler, Sınıflar, vb.