

# YMH212 Yazılım Gereksinim Analizi

*Yazılım Gereksinim Doğrulama ve  
Geçerleme*

Dr. Öğr. Üyesi Rukiye SAVRAN KIZILTEPE

Yazılım Mühendisliği

2024-2025 Bahar Dönemi



# Ders İçeriği

- Giriş ve Temel Kavramlar
- Yazılım Yaşam Döngüsü
- Yazılım Geliştirme (Süreç) Modelleri
- Yazılım Geliştirmede Çevik Yöntemler
- Yazılım Gereksinimleri ve Türleri
- Gereksinim Toplama
- Gereksinim Analizi ve Dokümantasyon
- Gereksinim Modelleme ve Spesifikasyon
- Gereksinim Doğrulama ve Geçerleme
- Gereksinim Yönetimi
- Yazılım Tasarımı
- Proje Sunumları

# Gereksinim Standartlari

- **NASA Systems Engineering Handbook= NASA/SP-2007-6105**
  - Section 4.2 – Technical Requirements Definition
  - Section 6.2 – Requirements Management.
  - Appendix C – How to write a good Requirement
  - Appendix D – Requirements Verification Matrix
- **International Council of Systems Engineering (INCOSE)**
  - Systems Engineering Handbook, Version 3.1
- **ISO/IEC 15288 (IEEE STD 15288-2008)**
  - Systems and software engineering
  - System life cycle processes 6.4.1
    - Stakeholder Requirements Definition Process

# Gereksinim Mühendisliği (Requirements Engineering)

- *Bir sistem veya ürün için gereksinimlerin;*

- keşfedilmesi,



- analiz edilmesi,



- belgelenmesi,



- doğrulanması,



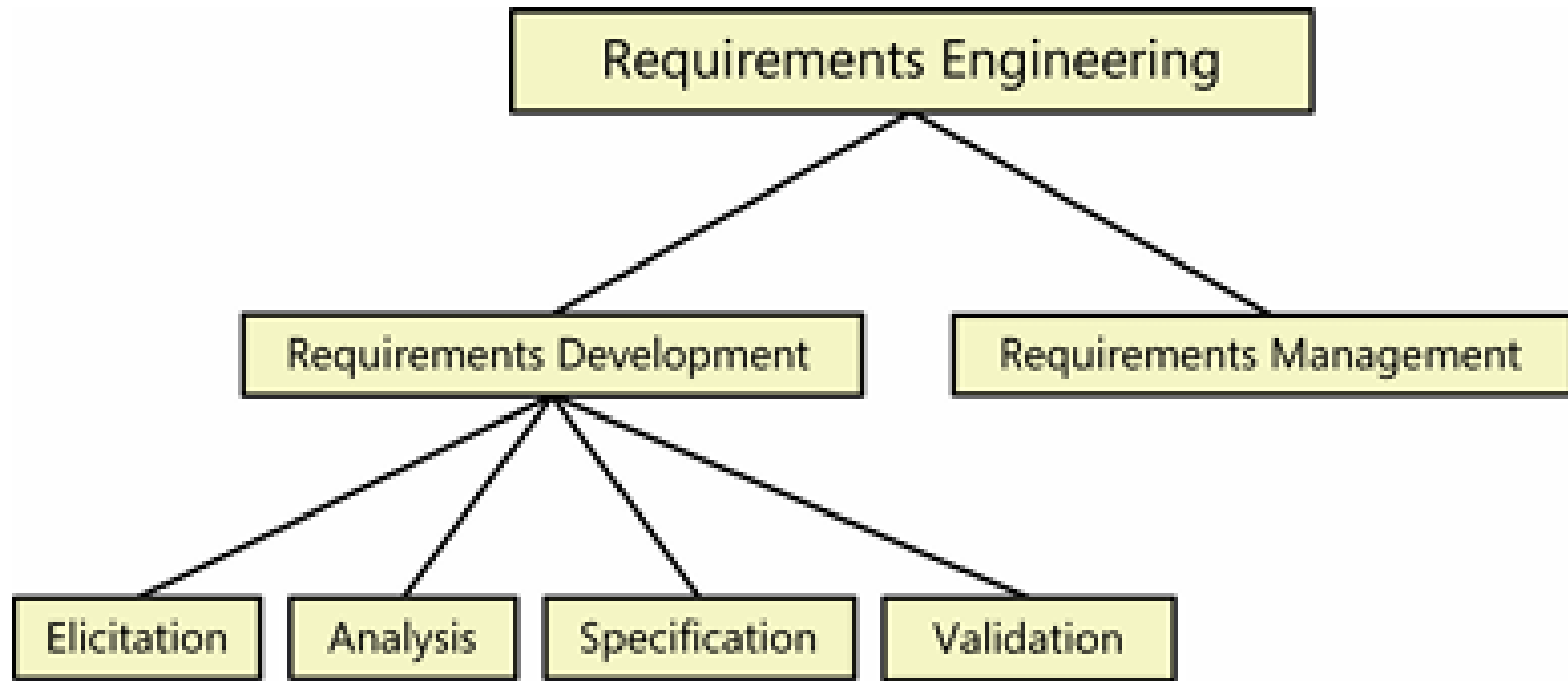
- yönetilmesi



*sürecidir.*

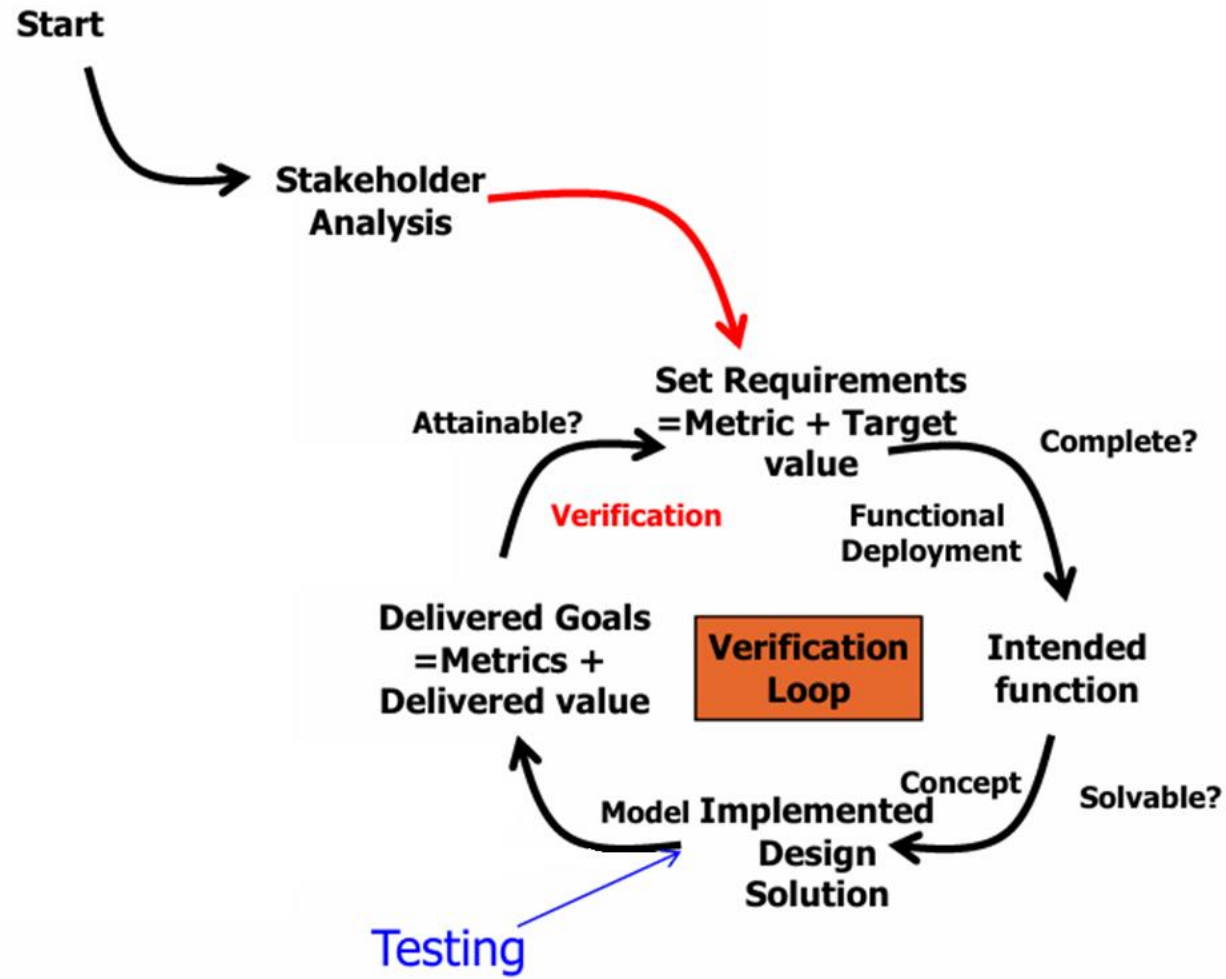
# Requirements Engineering Verification & Validation

- **NASA Systems Engineering Handbook= NASA/SP-2007-6105**
  - Section 5.3 – Product Verification (ss. 88-99)
  - Section 5.4 – Product Validation (ss. 99-105)
  - Appendix E – Creating Validation Plan ... (ss. 203-204)
  - Appendix I – Verification and Validation Plan Outline (ss. 216-222)
- **Software Engineering, Karl Wieggers and Joy Beatty.**
  - Chapter 17 Validating the requirements (ss. 331-350)

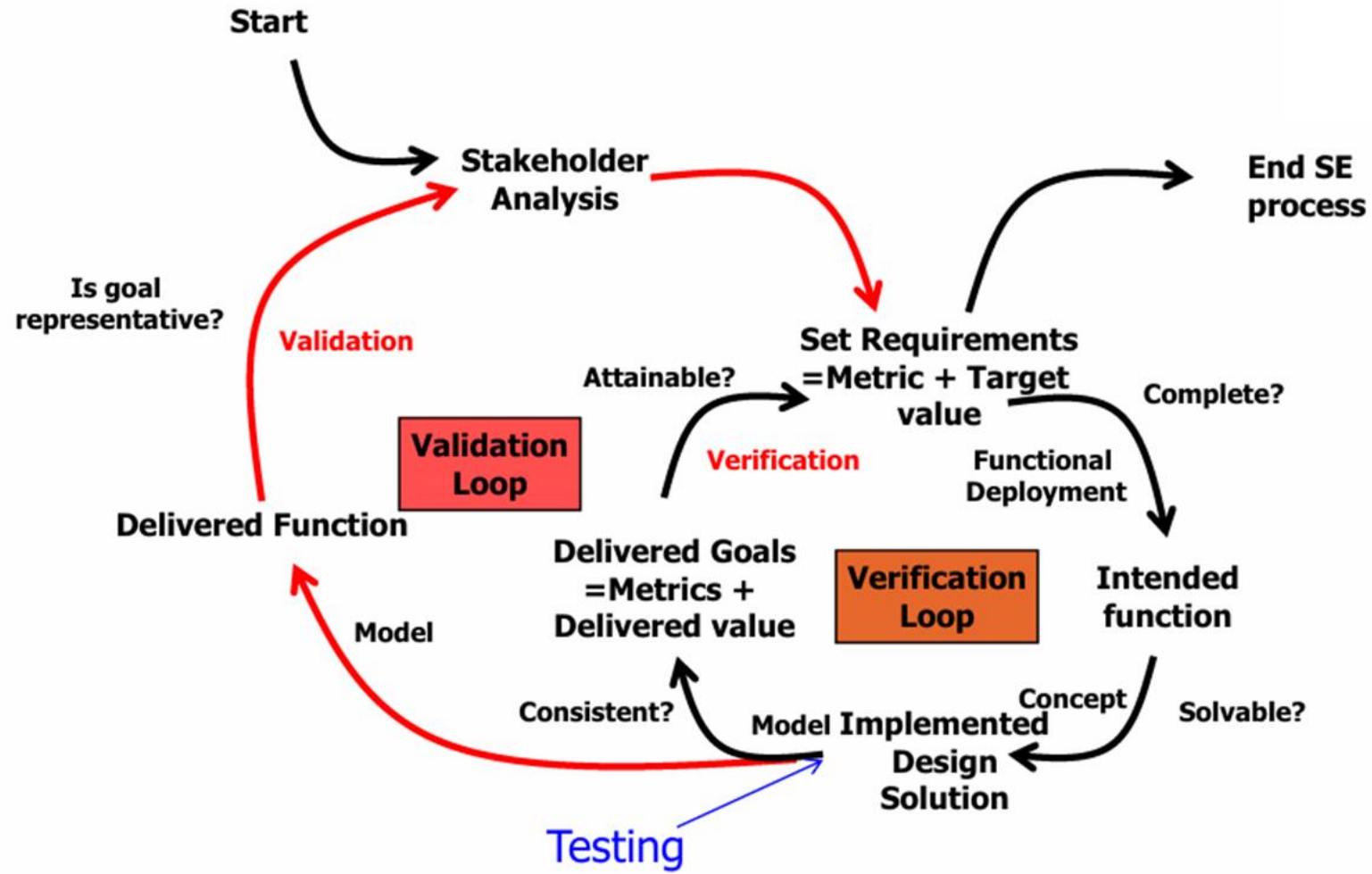


**FIGURE 1-4** Subdisciplines of software requirements engineering.

# Verification & Validation



# Verification & Validation





# Verification & Validation

- Verification and Validation is the process of investigating whether a software system satisfies specifications and standards and fulfills the required purpose.
- Barry Boehm described verification and validation as the following:
  - Verification: Are we building the product right?
  - Validation: Are we building the right product?

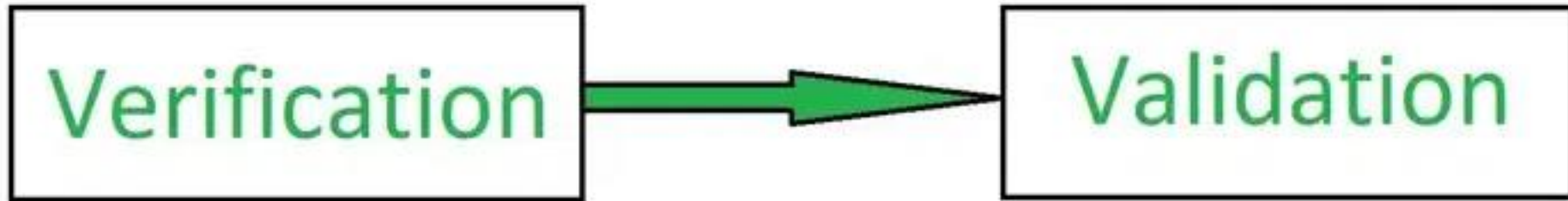
# Verification

- Verification is the process of checking that software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have.
- Verification is simply known as Static Testing.
- **Static Testing**
- Verification Testing is known as Static Testing and it can be simply termed as checking whether we are developing the right product or not and also whether our software is fulfilling the customer's requirement or not.
- Here are some of the activities that are involved in verification:
  - Inspections
  - Reviews
  - Walkthroughs
  - Desk-checking

# Verification

- Verification is the process of checking that software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have.
- Verification is simply known as Static Testing.
- **Dynamic Testing**
- Validation Testing is known as Dynamic Testing in which we examine whether we have developed the product right or not and also about the business needs of the client.
- Here are some of the activities that are involved in Validation:
  - Black Box Testing
  - White Box Testing
  - Unit Testing
  - Integration Testing

Verification is followed by Validation



# V&V Differences

## Verification (Doğrulama)

- Geliştirme süreci boyunca
- Ürünün doğru bir şekilde geliştirildiğini kontrol eder
- Yazılım geliştirilirken gerekli adımların takip edildiğinden emin olur
- Ürün geliştirme sürecindeki çıktıların SRS ile tutarlılığını kontrol eder
- Gereksinimlerin karşılandığını kontrol eder
- Genellikle geliştirme ortamında
- Bileşen/alt sistem odaklı

## Validation (Geçerleme)

- Entegrasyon boyunca ya da sonrasında
- Doğru ürünün geliştirildiğini kontrol eder
- Geliştirilen ürünün müşterinin gerçek ihtiyacını karşılayacağından emin olur
- Müşteri amaç ve istekleri doğrultusunda SRS kontrolü
- Müşterini niyetinin karşılandığını kontrol eder
- Genellikle gerçek ya da simülasyon ortamında
- Sistem odaklı
- Kabulü **GEÇ** -> **GEÇERLEME**

# Concept Question

Aşağıdaki aktiviteleri sınıflandırınız.

	Verification (Doğrulama)	Validation (Geçerleme)	No Sure (Emin değilim)
Yeni bir aracın Alaska'daki kar koşullarında test edilmesi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Laboratuvarda ön çarpışma testi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Yeni bir oyuncuğun kreşte test edilmesi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Titreşim masasında uydu titreşim testi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Google Glass'ın 1000 pilot kullanıcı ile alan testi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Telefon uygulama güncellemesinin, belirli cihazlar ve işletim sistemleriyle uyumlu olup olmadığını kontrol	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

# Concept Question

	Verification (Doğrulama)	Validation (Geçerleme)	No Sure (Emin değilim)
Yeni bir aracın Alaska'daki kar koşullarında test edilmesi	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Laboratuvarda ön çarpışma testi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Yeni bir oyuncağın kreşte test edilmesi	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Titreşim masasında uydu titreşim testi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Google Glass'ın 1000 pilot kullanıcı ile alan testi	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

# Concept Question

	Verification (Doğrulama)	Validation (Geçerleme)	No Sure (Emin değilim)
Telefon uygulama güncellemesinin, belirli cihazlar ve işletim sistemleriyle uyumlu olup olmadığını kontrol etmek	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Banka işlem platformunun, belirli güvenlik protokolleri ve standartlara uygun olup olmadığını kontrol etmek	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tıbbi cihaz yazılımının, klinik ortamlarda gerçek hastalar üzerinde yapılan testler sonucunda beklenen tıbbi sonuçları doğrulamak için klinik denemeler yapmak.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>



# V&V

- *Requirements verification* is the process of confirming that system requirements include all the necessary elements of well-written requirements.
- *Requirements validation* involves ensuring that the written requirements align with the stakeholders' expectations and needs.
- *Verification* focuses on checking whether the requirements are complete, accurate, and consistent.
- *Validation* is concerned with determining whether the requirements adequately define and describe the intended system objectives and functionalities.
- **Nasa Handbook – Appendix C.4. (pp. 198-200) & Appendix D (pp. 201-202)**

# Verification Techniques

- Functional requirements are typically verified through testing and demonstration.
- Non-functional requirements can often be verified through observation.

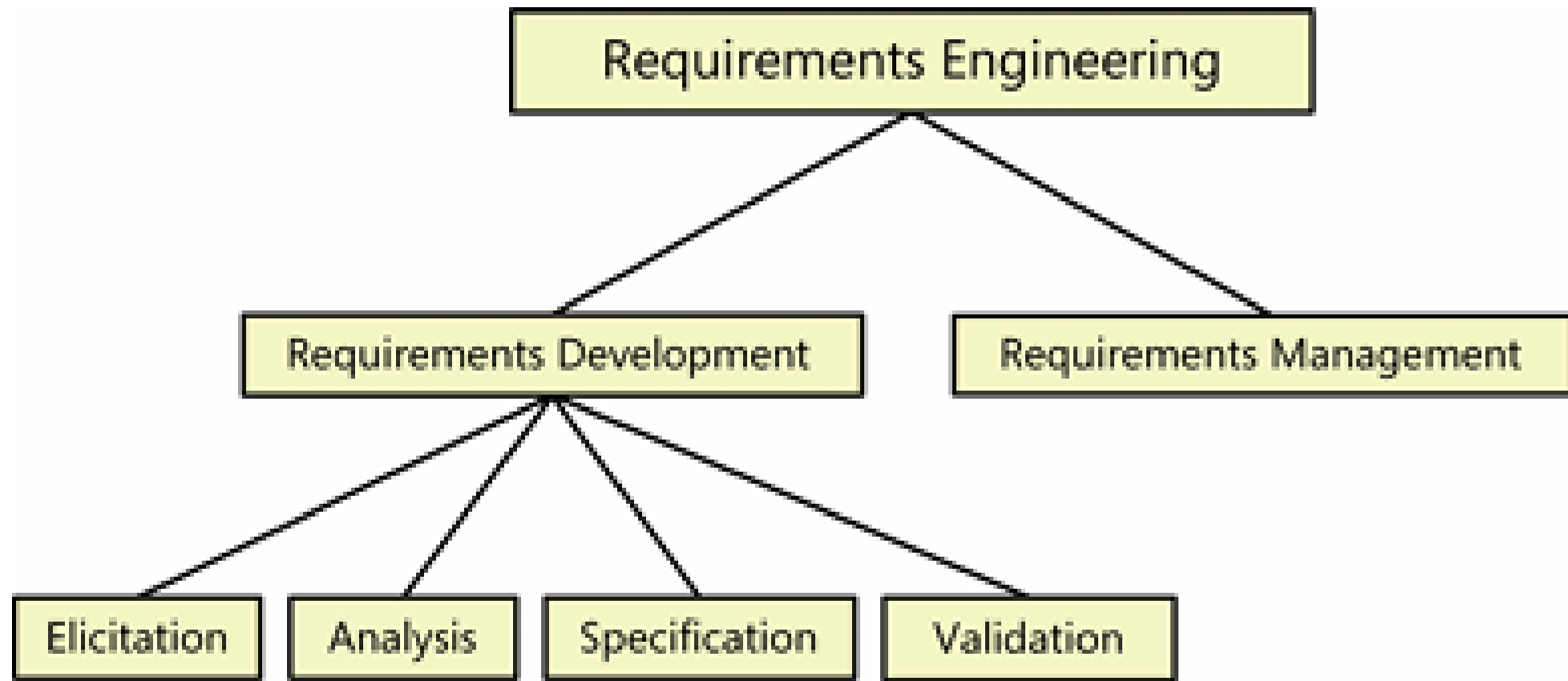
# V&V

- Why is Requirements Verification Important?
  - Identifying problems related to requirements.
  - Early detection of errors caused by misunderstandings.
  - Reducing errors in requirements to enhance reliability.
  - Mitigating risks (e.g., acceptance testing, time, budget, etc.).
- When Are Requirements Verified?

# Gereksinim Yönetimi

## Requirement Management

- **NASA Systems Engineering Handbook= NASA/SP-2007-6105**
  - Chapter 6.2 – Requirements Management (ss. 111-117)
  - Appendix D – Requirements Verification Matrix
  - Appendix I – Verification and Validation Plan Outline (ss. 216-222)
- **Software Engineering, Ian Sommerville.**
  - Chapter 4.7 Requirements Management (ss. 111-117)
  - Chapter 5 System Modelling
  - Chapter 6 Architectural Design



**FIGURE 1-4** Subdisciplines of software requirements engineering.

# SMART Gereksinimler

- **Specific(Spesifik - Özgül):** "Bazı", "birkaç", "birçok" gibi belirsiz ifadelerden kaçınin. Açıklığı sağlayın.
- **Measurable (Ölçülebilir):** Göstergeler nicel olmalı. "Ne kadar?" "Kaç tane?" gibi sorular sorun.
- **Attainable (Ulaşılabilir):** Gereksinimi karşılayacak kişinin kim olduğunu belirleyin ve bu kişinin gereksinimi karşılayıp karşılayamayacağını doğrulayın. Önceki projelerden parçaları yeniden kullanın.
- **Reasonable (Makul):** "Bu değerli mi?" "Zamanlama doğru mu?" gibi soruları sorarak gereksinimin mantıklılığını değerlendirin.
- **Traceable (İzlenebilir):** Gereksinimler, kaynaklarını, iş gereksinimlerini, bağımlılıklarını, diğer gereksinimlerle olan ilişkilerini ve önemini içermelidir.
- Bu ölçütler, gereksinimlerinizi **daha etkin bir şekilde yönetmenize** ve proje hedeflerinize ulaşmanıza yardımcı olur.

# Gereksinim Tanımı

Projede her bir gereksinimi tanımlarken, aşağıdaki bilgileri detaylı bir şekilde belgeleyeceğiz:

- 1. Gereksinim Adı:** Gereksinimin tanımlanması için kullanılacak benzersiz isim veya tanımlayıcı.
- 2. Gereksinim Tanımı:** Gereksinimin neyi kapsadığını açık ve anlaşılır bir biçimde açıklayan detaylı bir açıklama.
- 3. Bu Gereksinimi Önemseyen Paydaş:** Bu gereksinimin kim veya hangi gruplar için önemli olduğunu belirten bilgi.
- 4. Önem Ölçütü:** Gereksinimin varlığının veya yokluğunun proje üzerindeki etkisinin nicel bir değerlendirilmesi. Bu, gereksinimin ne kadar kritik olduğunu belirlememize yardımcı olur.
- 5. Kabul Kriterleri:** Gereksinimin başarıyla karşılandığını nasıl anlayacağımızı tanımlayan, niceliksel olarak ifade edilen ölçütler. Bu kriterler, gereksinimin tam olarak ne zaman ve nasıl yerine getirildiğini belirtir.

# Requirements Checking

- **Validity:** Does the system provide the functions which best support the customer's needs?
- **Consistency:** Are there any requirements conflicts?
- **Completeness:** Are all functions required by the customer included?
- **Realism:** Can the requirements be implemented given available budget and technology
- **Verifiability:** Can the requirements be checked?



# Gereksinimlerin Temeli (Requirements Baseline)

- **Baseline:** Bir ürün çıktısı için taahhüt edilmiş gereksinimler seti incelenmiş, onaylanmış ve anlaşmaya varılmıştır.
- Onaylanmış temel gereksinimlerdir.
- Temel gereksinimler onaylandığında:
  - Resmi değişiklik kontrolü başlar.
  - Yöneticiler zamanlama taahhütlerinde bulunur.
  - Yöneticiler gerekli personel ve bütçeyi belirler.

# Gereksinim Yönetimi Uygulamaları

- Gereksinimlerin temelini oluşturmak (requirement baseline)
- Gereksinim dokümanlarının versiyonlarını yönetmek
- Değişiklik kontrol sürecini benimsemek ve uygulamak
- Gereksinim değişiklik etki analizi yapmak
- Gereksinim özniteliklerini saklamak
- Her gereksinimin durumunu izlemek
- Gereksinimleri tasarımlara, kodlara ve testlere yansıtmak
- Gereksinimleri bir gereksinim yönetimi aracında saklamak

# Gereksinim Yönetimi

- Gereksinim yönetimi faaliyetleri, tüm paydaş beklentilerinin, müşteri gereksinimlerinin ve teknik ürün gereksinimlerinin en alt düzey ürün bileşeni gereksinimlerine kadar yönetilmesini kapsar.
- Gereksinim Yönetimi Süreci kullanım amacı:
  - WBS' nin tüm seviyelerinde gereksinimleri tanımlamak, kontrol etmek, ayrıştırmak ve tahsis etmek.
  - İki yönlü izlenebilirlik sağlamak.
  - Sistem ürünlerinin yaşam döngüsü boyunca belirlenen gereksinim temel çizgilerindeki değişiklikleri yönetmek.

Bu süreç, projenin her aşamasında gereksinimlerin doğru bir şekilde yönetilmesini sağlayarak, projenin başarılı bir şekilde tamamlanmasına katkıda bulunur. Ayrıca, tüm gereksinimlerin ve beklentilerin açıkça tanımlandığından ve uygun bir şekilde takip edildiğinden emin olunmasına yardımcı olur.

# Gereksinim Yönetimi

- Gereksinim yönetimi, gereksinim mühendisliği süreci ve sistem geliştirme sırasında değişen gereksinimleri yönetme sürecidir.
- Bir sistem geliştirilmekteyken ve kullanıma sunulduktan sonra yeni gereksinimler ortaya çıkar.
- Bireysel gereksinimleri takip etmeniz ve gereksinim değişikliklerinin etkisini değerlendirebilmeniz için bağımlı gereksinimler arasında bağlantılar kurmanız gerekir.
- Değişiklik teklifleri yapma ve bunları sistem gereksinimleriyle bağlantılandırma için resmi bir süreç oluşturmanız gerekmektedir.

# Gereksinim Yönetim Planı

- Proje paydaşları nasıl tanımlanacak?
- Gereksinimlerin önceliklerini kim belirleyecek?
- Gereksinim yönetiminden kim sorumludur?
- İzlenebilirlik nasıl geliştirilecek ve kurulacak?
- Değişiklikler nasıl izlenecek?

# Gereksinim Yönetim Planı

Gereksinim yönetiminin gerektirdiği detay düzeyini belirler. Gereksinim yönetimi kararları:

- **Gereksinim Tanımlama:** Her gereksinim, diğer gereksinimlerle karşılaştırılabilmesi için benzersiz bir şekilde tanımlanmalıdır.
  - **Değişiklik Yönetimi Süreci:** Bu, değişikliklerin etkisini ve maliyetini değerlendiren bir dizi aktivitedir.
  - **İzlenebilirlik Politikaları:** Bu politikalar, her gereksinim arasındaki ve gereksinimler ile sistem tasarımı arasındaki ilişkilerin kaydedilmesi gerektiğini tanımlar.
  - **Araç Desteği:** Kullanılabilecek araçlar, özel gereksinim yönetimi sistemlerinden elektronik tablolara ve basit veritabanı sistemlerine kadar değişebilir.
- 
- EK: Requirements Management Plan

# Gereksinim İzlenebilirlik Matrisi (Req. Traceability Matrix)

Gereksinim İzlenebilirlik Matrisi, proje gereksinimlerinin başlangıçtan sonuca kadar izlenmesini sağlayan bir araçtır. Bu matris, projenin her aşamasında gereksinimlerin nasıl ele alındığını, hangi çıktılarına bağlandığını ve durumlarının ne olduğunu görsel bir şekilde gösterir.

Gereksinim İzlenebilirlik Matrisi kullanımı, projelerin yönetiminde büyük bir şeffaflık sağlar ve proje ekibinin tüm gereksinimleri uygun şekilde yönetip yönetmediğini kontrol etme imkanı verir. Bu da projenin başarılı bir şekilde tamamlanmasına olanak tanır.

# Gereksinim İzlenebilirlik Matrisi (Req. Traceability Matrix)

RTM'nin içerdiği temel özellikler:

- 1. Gereksinim Tanımları:** Her bir gereksinimin net bir şekilde tanımlanması, neyin başarılması gerektiğini açıkça belirtir.
- 2. Kaynak Bilgisi:** Gereksinim kaynağını (örneğin, bir paydaş, bir pazar araştırması veya bir iş kuralı) belirtir.
- 3. Proje Hedefleri ile İlişkisi:** Her gereksinimin proje hedeflerine nasıl katkıda bulunduğunu gösterir.
- 4. WBS ile İlişkisi:** Gereksinimin projenin hangi bölümünde ele alındığını belirtir.
- 5. Doğrulama ve Geçerlilik:** Her bir gereksinimin doğrulama ve geçerlilik durumunu takip eder, böylece projenin gereksinimlerin karşılanıp karşılanmadığını gözlemlemek mümkün olur.
- 6. Durum Güncellemeleri:** Gereksinimlerin mevcut durumunu (aktif, tamamlandı, beklemekte vs.) içerir.
- 7. Sorumluluklar:** Hangi takım üyesinin hangi gereksinimi takip ettiğini ve raporlamasını gösterir.



# Örnek

- NASA Handbook Appendix D & Appendix I

# Gereksinim Yönetimi Süreci

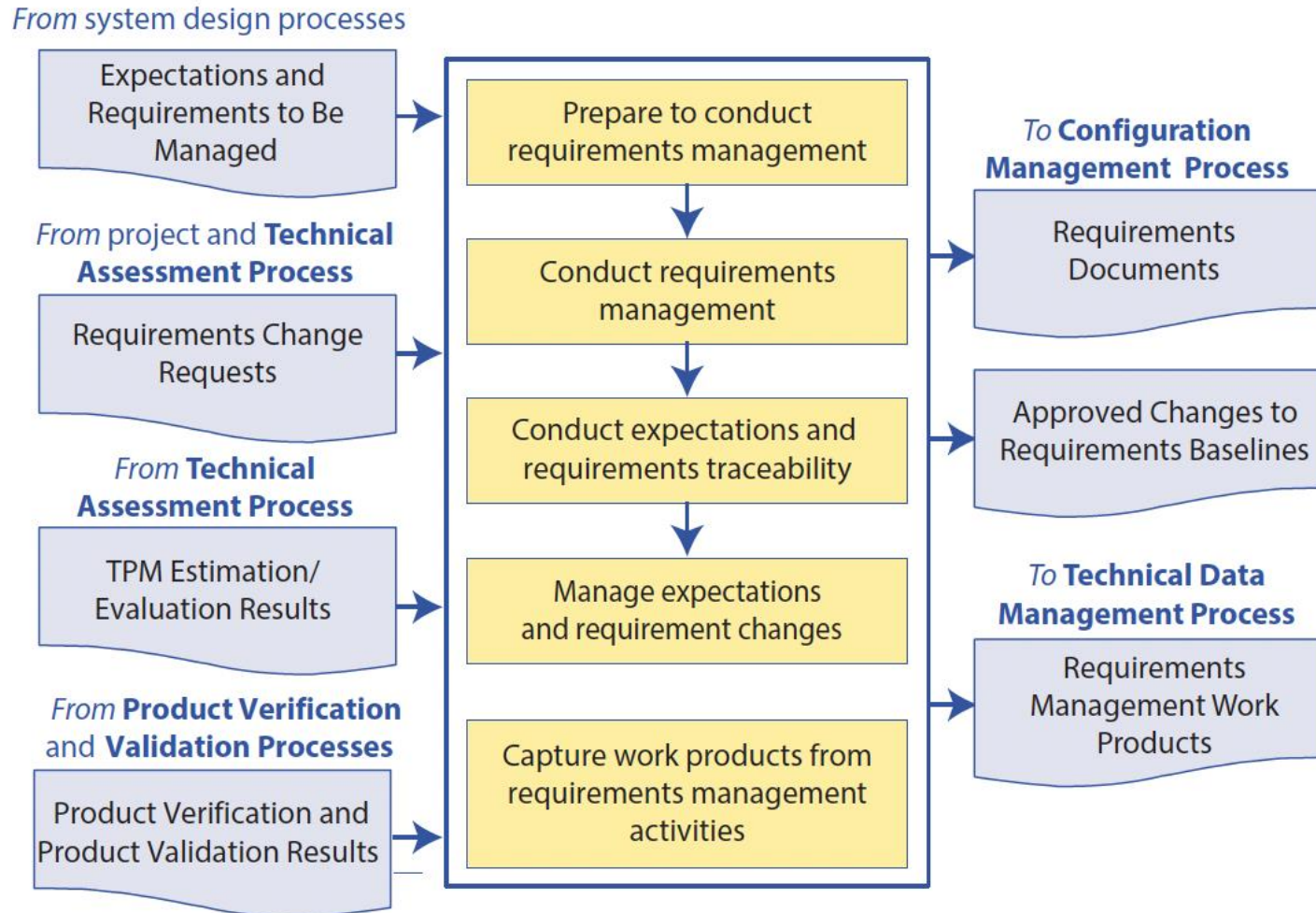


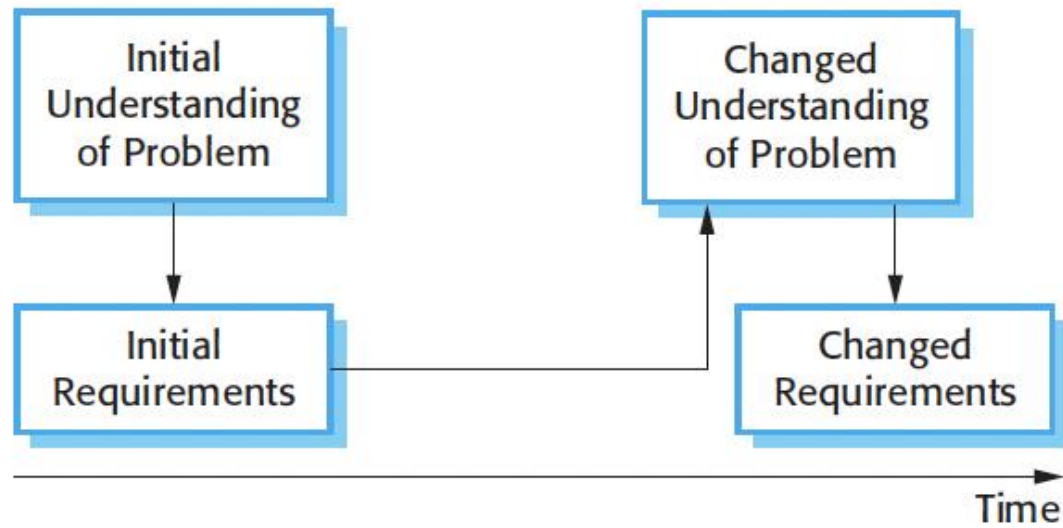
FIGURE 6.2-1 Requirements Management Process

# Gereksinim Değişiklikleri

- Sistem kurulduktan sonra işletme ve teknik çevre her zaman değişir. Yeni donanımlar tanıtılabilir, iş öncelikleri değişebilir ve sistem tarafından mutlaka uyulması gereken yeni yasalar ve düzenlemeler tanıtılabilir.
- Bir sistem için ödeme yapan kişiler ile bu sistemi kullananlar genellikle aynı kişiler değildir.
- Sistem müşterileri, organizasyonel ve bütçesel kısıtlamalar nedeniyle gereksinimleri belirler. Bunlar, son kullanıcı gereksinimleriyle çelişebilir ve teslimattan sonra, sistem hedeflerine ulaşabilmesi için kullanıcı desteği için yeni özellikler eklenmek zorunda kalabilir.
- Büyük sistemler genellikle çeşitli kullanıcı topluluklarına sahiptir ve birçok kullanıcının çelişkili veya zıt gereksinimleri ve öncelikleri olabilir.
- Sistem gereksinimleri bu çelişkiler arasında bir uzlaşa sağlar.

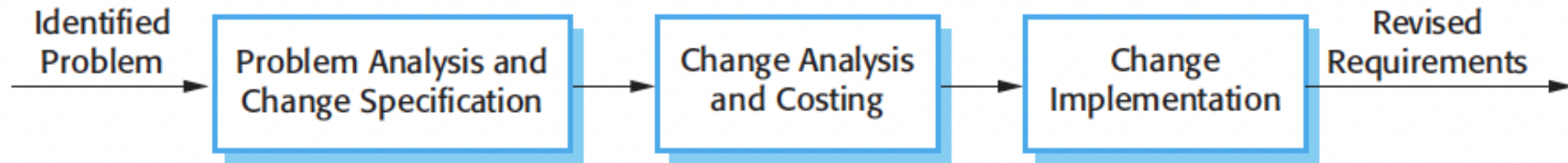
# Gereksinim Değişiklikleri

**Figure 4.17**  
Requirements  
evolution



# Gereksinim Değişiklikleri

- Gereksinim değişikliklerinin etkili yönetimi, değişikliğin onaylanması ve uygulanmasından önce önerilen **değişikliklerin etkisini değerlendiren bir süreç** gerektirir. Bu normalde Konfigürasyon Yönetim Sürecinin kullanılmasıyla gerçekleştirilir. KYS'nin bu işlevi yerine getirebilmesi için, bir temel yapılandırma belgelenmeli (baseline) ve mevcut durum üzerindeki etkileri değerlendirmek için araçlar kullanılmalıdır.



# Gereksinim Değişiklikleri

Gereksinim değişikliğinin kabul edilip edilmemesine karar verme süreci şu aşamalardan oluşur:

- **Problem Analizi ve Değişiklik Önerisi:** Bu aşamada, problem veya değişiklik önerisi geçerliliğinin kontrol edilmesi için analiz edilir. Bu analiz, değişiklik talep eden kişiye geri bildirim olarak verilir. Talep eden kişi, daha spesifik bir gereksinim değişikliği önerisiyle yanıt verebilir veya talebi geri çekmeye karar verebilir.
- **Değişiklik Analizi ve Maliyetlendirme:** Önerilen değişikliğin etkisi, izlenebilirlik bilgileri ve sistem gereksinimlerine ilişkin genel bilgi kullanılarak değerlendirilir. Bu analiz tamamlandıktan sonra, gereksinim değişikliği ile devam edilip edilmeyeceğine karar verilir.
- **Değişiklik Uygulaması:** Gereksinimler dokümanı ve gerekli olduğunda sistem tasarımı ve uygulaması değiştirilir. İdeal olarak, dokümanın değişikliklerin kolayca uygulanabileceği şekilde düzenlenmiş olması gerekir.

# Gereksinim Değişiklikleri

Değişimin etkilerini analiz etmek için kullanılan tipik araçlar şunlardır:

- **Performans Marjları:** Ana sistem performans marjları üzerindeki etkileri analiz eder.
- **CM Konu Değerlendiricileri Listesi:** Uygun personelin değişiklikleri değerlendirmesini sağlar.
- **Risk Sistemi ve Tehditler Listesi:** Proje risklerini ve ilişkili maliyetleri tanımlar, bu da azaltma stratejilerinin önceliklendirilmesine yardımcı olur.
- **Dokümantasyon Güncellemeleri:** Değişiklik paketinin bir parçası olarak dokümantasyonun güncellenmesini sağlar ve bu güncellemeleri takip eder.

# Gereksinim Değişiklikleri

- Requirements Creep!
- Projede gereksinimlerin genellikle istenmeyen bir şekilde artışına, bu da artan karmaşıklık ve maliyete yol açar.
- **Önleme Stratejileri:**
  - **ConOps:** Tüm paydaşlar tarafından kabul edilmiştir.
  - **Gereksinimlerin erken tanımlanması:** Başlangıçta açık olmayanlar dahil, tüm potansiyel gereksinimleri tanımlayın.
  - **Sıkı değişiklik değerlendirme süreci:** Konfigürasyon Yönetimi
  - **Resmi Değişiklik talebi kanalları:** Kimlerin değişiklikleri resmi inceleme için sunabileceğini kontrol eder.
  - **Etki değerlendirme:** Her gereksinim değişikliğinin işlevselliğini ve sistem genelindeki etkisini değerlendirir.
  - **Kaynak bütçeleri:** Değişikliklerin mevcut bütçeler içinde karşılanıp karşılanamayacağını belirleyin.



# Management Tool Örnek Videolar

IBM DOORS:

- <https://www.youtube.com/watch?v=blk33ca0oWQ&t>

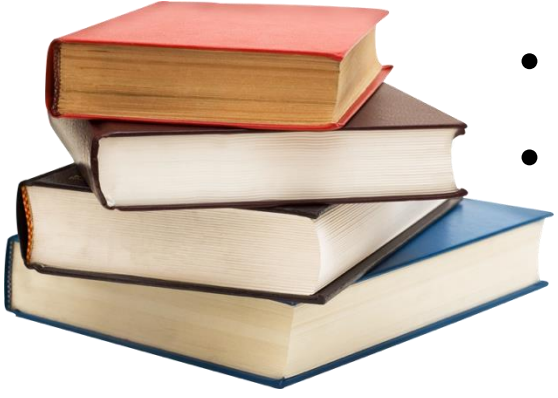
Cameo Systems Modeler:

- <https://www.youtube.com/watch?v=4mwdROAKIY4>

# Key points

- You can use a range of techniques for requirements elicitation including interviews, scenarios, use-cases and ethnography.
- Requirements validation is the process of checking the requirements for validity, consistency, completeness, realism and verifiability.
- Business, organizational and technical changes inevitably lead to changes to the requirements for a software system. Requirements management is the process of managing and controlling these changes.

# Kaynaklar



- Fundamentals of System Engineering, Prof. Oliver L. De Weck, MIT.
- Software Engineering, Ivan Marsic, Ruthers University.
- Software Engineering, Karl Wieggers and Joy Beatty.
- Prof. Dr. Recep Erġiyit, Ankara Üniversitesi Enformatik Bölümü, Tezsiz Yüksek Lisans Programı, Yazılım Mühendisliği Açık Kaynak Ders Notları.

# Gelecek Hafta

## Yazılım Tasarımı

