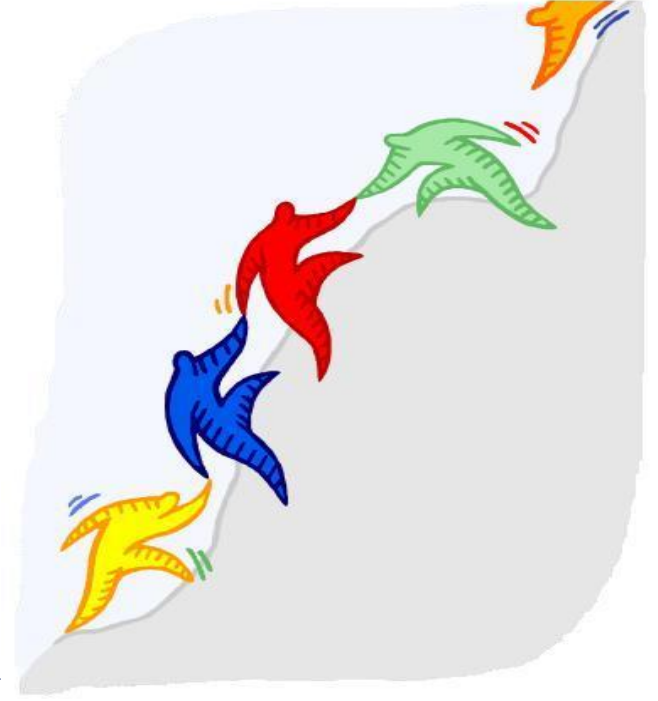


(Software Quality Assurance)



YAZILIM

KALİTE GÜVENCESİ

# İçerik

---

- ▶ Kalite Güvence
- ▶ Kalite Etkenleri
- ▶ Süreç Modelleri
- ▶ Kalite Sistem Standartları
- ▶ Gözden Geçirme
- ▶ Kalite Güvence Yöntemi, Testleri
- ▶ Kalite Güvence Ekibi ve Metrikleri
- ▶ Örnek Sorular

# Kaliteli Yazılım Nasıl olmalıdır?

---

Kaliteli yazılım, önceden belirlenmiş olan işlevsel gereksinimlere, yazılım geliştirme standartlarına ve bir yazılımdan beklenen bütün özelliklere tamamen uygun olan bir yazılımdır.

# Yazılımın Kalite Güvencesi

---

1. Yazılım Kalite Etkenleri
2. Süreç Modelleri
3. Kalite Sistem Standartları
4. Gözden Geçirme
5. Kalite Güvence Yöntemi
6. Kalite Güvence Testleri
7. Kalite Yönetim Yapısının Oluşturulması
8. Yazılımın Kalite Metrikleri(Ölçüm)

# 1.Yazılımın Kalite Etkenleri

---

- a) **Doğruluk:** Yazılımın belirtilmiş ihtiyaçlarını karşılaması.
- b) **Güvenilirlik:** Gerekli işlevi ne hassaslıkla yerine getireceği beklentisi.
- c) **Verimlilik:** İşlevin gerçekleştirilmesi için kullanılması gereken bilgisayar kaynakları ve kod miktarı.
- d) **Güvenlik(Bütünlük):** Yazılım ve bilgilerine istenmeyen insanlarca ulaşımın ne derece engellenebildiği.
- e) **Kullanılabilirlik:** Programın öğrenilmesi, çalıştırılması, girdi hazırlama ve çıktı yorumlama işlemlerinin kolaylık derecesi.
- f) **Hata bulma kolaylığı:** Hatanın yerini bulma ve düzeltme kolaylığı.

# 1.Yazılımın Kalite Etkenleri(devam)

---

- g) **Esneklik:** Yazılımda değişiklik yapma kolaylığı.
- h) **Test kolaylığı:** Yazılımın doğruluğunun sınanmasındaki doğruluk.
- i) **Taşınabilirlik:** Programın farklı donanımlarda ve yazılım sistemi ortamlarında kullanılması.
- j) **Tekrar kullanılabilirlik:** Yazılım tamamını ya da bir bölümünün farklı bir uygulamada kullanılabilmesi.
- k) **Birlikte çalışabilirlik:** Bir yazılım sisteminin diğerleri ile bağlantı sağlaması kolaylığı.

## Ürün izleme

**Bakım** (Sorun çıktığında düzeltebilir miyim?)

**Esneklik** (Değiştirebilir miyim?)

**Test edilebilirlik** (Test edebilir miyim?)

## Ürün uyumluluğu

**Taşımaabilirlik** (Başka bir bilgisayarda kullanabilir miyim?)

**Yeniden kullanılabilirlik** (Aynı yazılımı tekrar kullanabilir miyim?)

**İlişkilerlik** (Başka sistemle birlikte kullanabilir miyim?)

## YAZILIM KALİTE GÜVENCE YAPISI

**Doğruluk** (İstediğim işi mi yapıyor?)

**Güvenilirlik** (Her zaman doğru mu çalışıyor?)

**Verimlilik** (Donanımında en iyi şekilde mi çalışıyor?)

**Kullanılabilirlik** (Yazılım, kullanıcı için mi tasarlandı?)

Ürün işleyişi

Yazılım Mühendisliği

## 2.Süreç Modelleri

---

Yazılım geliştirmede uygulanan çeşitli süreç modelleri vardır.

Bunlar;

- ◆ **Şelale modeli**
- ◆ **Evrimsel teslim modeli**
- ◆ **Spiral model**
- ◆ **Aşama Noktasına Dayalı Geliştirme modeli**
- ◆ **Agile**



### 3. Kalite Sistem Standartları

---

- ▶ CMM(A.B.D)
- ▶ CMMI(A.B.D)
- ▶ SixSigma
- ▶ Trillium(Kanada)
- ▶ TickIT(İngiltere ve Norveç)
- ▶ SPICE(Avrupada Yaygın)
- ▶ ISO 9000 (İngiltere,Hindistan ve Avrupanın bazı ülkeleri )
- ▶ AQAP-I50/I60(NATO)
- Günümüzde yazılım ürünlerine yönelik bir çok standart bulunmaktadır.
- Bazı ülkelerin kendi resmi standart geliştirme veya uygulama örgütü bulunmaktadır, bazı ülkelerde ise en yaygın olanları kullanılmaktadır.

# 1. CMM(Capability Maturity Model)

---

(Yetenek Olgunluk Modeli)

- ▶ Etkin bir yazılım geliştirme sürecinin anahtar elemanlarını tanımlayan bir çerçeve model olarak, Amerika Savunma Bakanlığı'nın 1980'ler de ortaya çıkan bir yazılım geliştirme sürecidir.
- ▶ CMM, asıl olarak, olgun olmayan bir süreçten olgun ve disiplinli bir sürece giden evrimsel bir yol çizer.

# CMM'nin yapısında şunlar vardır;

---

- ▶ Olgunluk düzeyleri
- ▶ Anahtar süreç alanları
- ▶ Ortak özellikler
- ▶ Anahtar uygulamalar

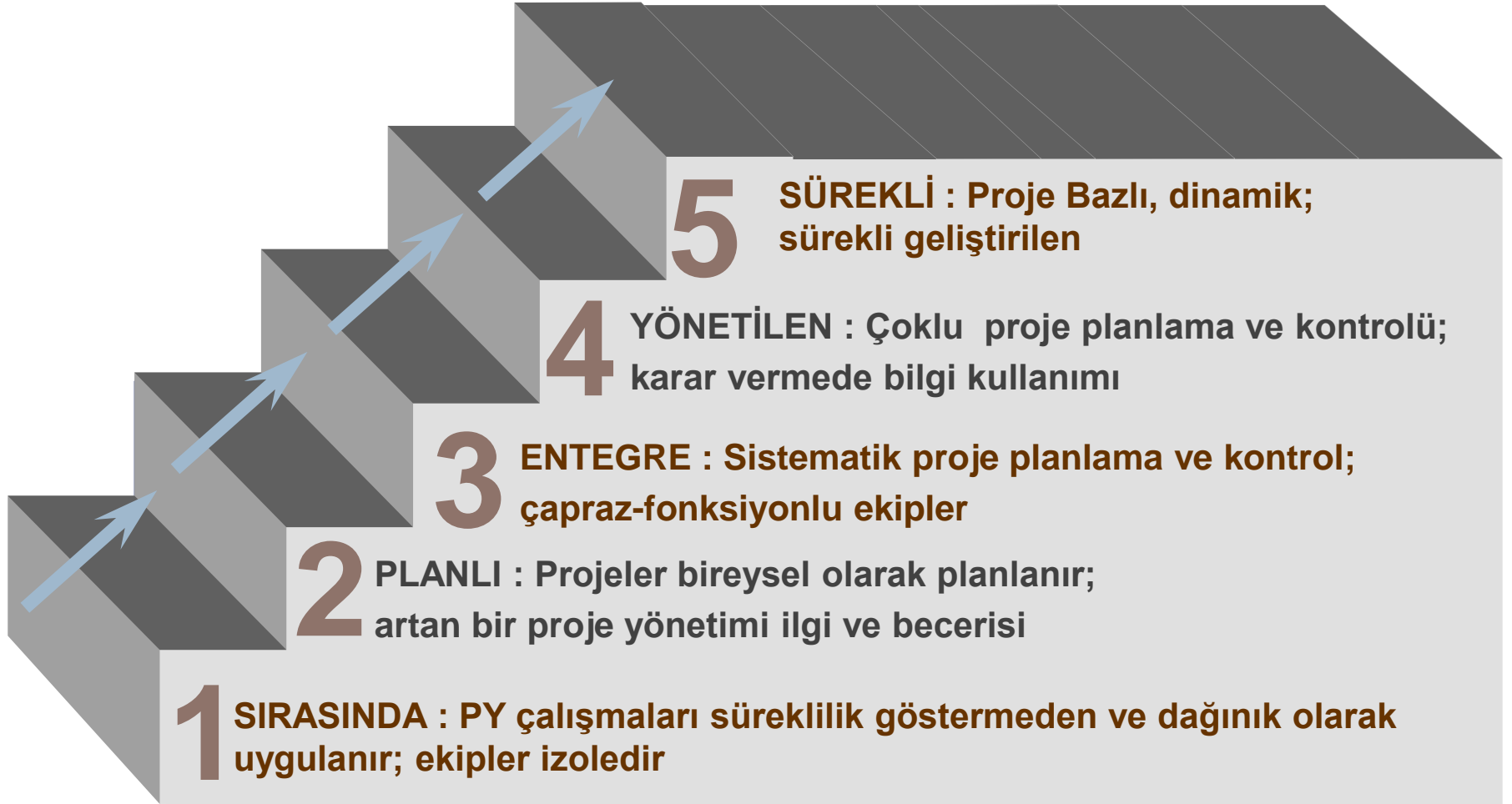
\*CMM ile yapılan değerlendirme belirli süreçlerde tekrarlanmak zorundadır.

# 1.1.CMMI

---

- ▶ Karşılaşılan sorunları çözmek üzere, var olan gelecekte var olacak modelleri birleştirecek bir yapı kurmak ve başlangıç için bir tümleşik modeller seti oluşturmak amacıyla başlatılmıştır. (Amerikan Savunma Bakanlığının desteğiyle CMMI (CMM Tümleştirme Projesi) oluşturulmuştur)
- ▶ Uzun süren çalışmalar ve yayınlanan değişikliklerle oluşturulan CMMI 2002’de yayınlanmıştır.
- ▶ CMMI, **sürekli** ve **basamaklı** olmak üzere iki gösterim şekli kullanmaktadır.
- ▶ Yetenek düzeyleri, bir örgütün her bir süreç alanında süreç iyileştirmede gösterdiği başarı için uygulanır.

# Capability Maturity Modeli



# Six Sigma - DMAIC



**“Müşteri ihtiyaçlarını kusursuz yakın karşılama hedefi”**

## **Define:**

Tanımla – Problemi/fırsatı, süreci, müşteri taleplerini tanımla

## **Measure:**

Ölç – Ölçüleri, toplamayı, birleştirmeyi, verileri göstermeyi tanımla

## **Analyze:**

Analiz et – Süreçlerde geliştirme fırsatı olabilecek alanları bul

## **Improve:**

Geliştir - Problemi giderici yada süreci geliştirici çözümler yada fikirler üret

## **Control:**

Kontrol Et – Geliştirme çalışmalarının dengeliliğini ve yeni sistemin sahiplenildiğini izle

**“organizasyonun daha fazla müşteri tatmini, karlılık ve rekabetçi pozisyon için kültürel değişim gayreti”**

# Six Sigma - DMAIC

Çok farklı alanlarda kanıtlanan Yararları vardır.

- ✓Maliyetlerde azalma
- ✓Üretkenlikte artış
- ✓Pazar payında artış
- ✓Müşteri tatmininde artış
- ✓Döngü-süresinde azalma
- ✓Hata oranında azalma
- ✓Olumlu kültürel değişim
- ✓Ürün/hizmet geliştirme

şeklinde özetlenebilir.



## 2. SPICE (Software Process Improvement and Capability Etermination)

---

- ▶ **İki boyutlu bir model** olup içe dönük süreç iyileştirme ile içe ve dışa dönük yetenek belirleme amacı taşır. Birinci boyutta **süreçler**, ikinci boyutta **yetenek** düzeyleri vardır.
- ▶ Yazılım edinme, geliştirme, işletim, bakım ve destek için gerekli olan planlama, yönetim, icra, denetim ve iyileştirme aracı konularını kapsar.
- ▶ **Yetenek düzeyi**, daha ayrıntılı, tekrarlanabilir ve nesnel olarak notlandırma amacıyla, alt düzeyde, süreç öznitelikleri tanımlamıştır.



## 3.ISO 9000

---

- ▶ Uluslararası Standartlaştırma Örgütü tarafından yayınlanan ISO 9000 serisi özellikle Avrupa'da büyük bir ilgi görmüştür.
- ▶ ISO 9000 serisi standartlar çok çeşitlidir, ancak ana kaynaklar ele alındığında, yayınlanmış ve sürekli güncellenen bu nitelik sistemlerine ilişkin beş temel ISO standardı vardır:
  1. ISO 9000
  2. ISO 9001
  3. ISO 9002
  4. ISO 9003
  5. ISO 9004

# ISO 9001

---

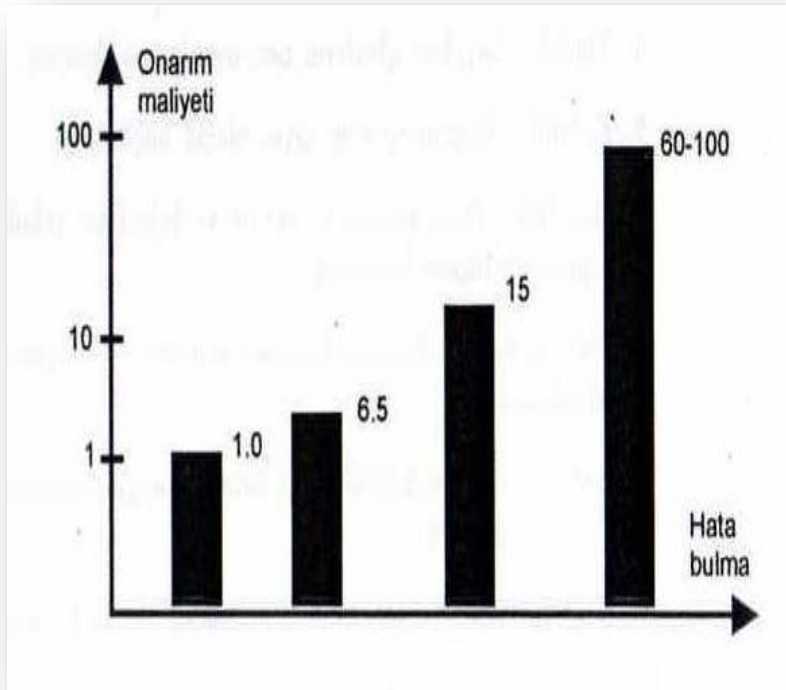
- ▶ Ancak bu standart bir süreç modelinden farklı olarak, dışarıya nitelik güvencesi vermeye yönelik belgelendirmeye önem verirler.
- ▶ Bir denetim yöntemine gereksinim duyan bu standartlar, firma düzeyi hakkında ayrıntı değil, genel bir bilgi verirler.

## 4. Gözden Geçirme

---

- ▶ Gözden geçirme işlemlerinin bir çok yöntemi vardır. Resmi gözden geçirme kalite açısından önemli bir kavram olmuştur.
- ▶ Resmi gözden geçirme yazılım geliştiriciler tarafından gerçekleştirilir. 5 katılımcı ile yapılır ve genelde 2 saat sürer.
- ▶ Resmi gözden geçirmeler; hata bulmak ve ihtiyaçların karşılandığı, standartlara uyulduğu, tutarlı yaklaşımlar uygulandığının denetimi için yapılır.
- ▶ Kişiler değil **ürün** sorgulanır.

## 4. Gözden Geçirme



Hataların bulunma evresi ve giderilme maliyeti

- ▶ Geliştirme sırasında hataların olacağı beklenir ve olabildiğince erken bulunması ve giderilmesi istenir.
- ▶ Hataların geç bulunması onların giderilmesini çok pahalılaştırmaktadır.

## 5. Kalite Güvence Yöntemi

---

- ▶ Her projede mutlaka belirli bir Kalite Güvence Yönetimi bulunmalıdır ve bir **Yazılım Kalite Güvence Planı** (Software Quality Assurance Plan) hazırlanmalıdır.

Bu plan standartlara uygun olmalıdır.

- ▶ **Yazılım Nitelik Güvence Planı**, geliştirme etkinlikleri boyunca nitelik yönetimine ait ne tür işlerin yapılacağını, hangi standartların kullanılacağını, gözden geçirme ve denetimlerin ne şekilde yapılacağını ve diğer destek etkinliklerini tanımlayan bir yol haritası gibidir.
- ▶ Kalite güvence etkinliklerini yürütmek üzere proje grubundan bir **Kalite Güvence Yöneticisi** bulunur.

## 6. Kalite Güvence Testleri

---

**Birim Testi:** Programın doğruluğunu ölçmek için kullanılır.

**Tamamlama Testi:** Sistemin tüm parçalarının birbiriyle düzgün haberleştiğini ve veri akışının doğru yapıldığını test eder.

**Fonksiyonel Testi:** Fonksiyonel test, sistemin her elemanının gereksinimleri karşılamakta olduğunu gösterir.

## 6. Kalite Güvence Testleri (devam)

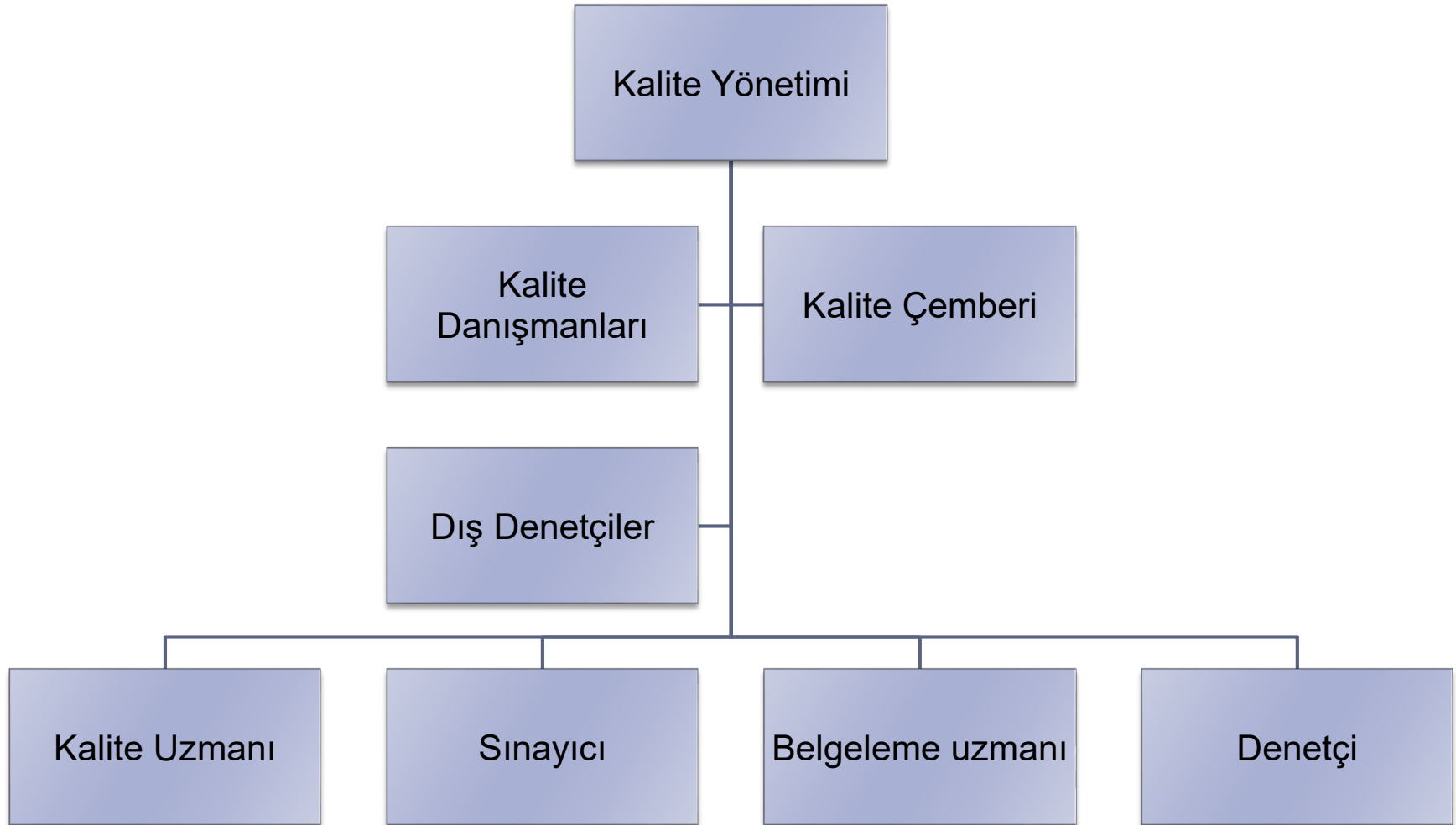
---

**Gerileme Testi:** Gerileme testi, her yeni versiyon yayınlandıktan sonra, yeni versiyonun eski versiyonlar üzerinde kötü etki bırakıp bırakmadığını ölçmeye yarar. Gerileme testi, projede devamlı bir artmalı ilerleme olduğunu gösterir.

**Güvenlik Testi:** Yazılımın, gerek iç ve gerekse dış kaynaklı yetkisiz erişimlere, kötü amaçlı kullanımlara karşı korunması ya da güvenliğini test etmek için yapılır

## 7. Kalite Ekip Yapısı

---





# Kalite Ekibinin Oluşturulması

---

- ▶ **Kalite Yöneticisi:** Tercihen endüstri mühendisliği ya da işletme bölümü mezunu kişiler. Projede gerekli olacak tüm bilişim sistem ve işlemleri için kalite politikaları oluşturmak ve ekibi bu konuda yönlendirmek.
- ▶ **Kalite Danışmanları:** Kalite uygulamaları konusunda teorik ve pratik bilgi sahibi kişiler. Kalite yöneticisine, proje genelinde yardımcı olmak.
- ▶ **Dış Denetçiler:** Kalite konusunda deneyimli, kurum dışı kişiler. Bu kişilerin görüş, öneri ve yönlendirmelerinden yararlanmak.

# Kalite Ekibinin Oluşturulması

---

- ▶ **Kalite Uzmanları:** Geliştirme ortamına ilişkin bilgi sahibi. Kalite güvence standartlarını hazırlamak ve uygulamak.
- ▶ **Belgeleme Uzmanı:** Masaüstü yayıncılık paketlerine ilişkin bilgi ve deneyim sahibi kişiler. Projeyi belgeleme işleriyle uğraşırlar.
- ▶ **Sinama Uzmanı:** Alfa sinama işlerinden sorumludurlar.
- ▶ **Denetçiler:** İstenilen kalite düzeyinde yürütülüp yürütülmediğini denetlerler.

# Kalite Ekibinin Oluşturulması

---

- ▶ **Kalite Çemberi:** Yeterli sayıda kalite uzmanı, oluşturacak standart ya da kurala ilişkin işi yapacak kişiler ve proje yönetiminden katılacak kişilerden oluşturulur.

Yapılacak projeye ilişkin kullanılacak standartların belirlenmesi, iş yönergelerinin oluşturulmasıyla ilgilenirler.

## 8. Yazılım Kalite Metrikleri (Ölçüm)

---

**Süreç Metrikler:** Yazılım geliştirme sürecinin ve ortamının kullanım derecelerini ölçmeye yarar.

**Boyut Metrikleri:** Projenin büyüklüğüne ilişkin durumun önceden yapılan planla uyuşmakta olduğunun gözlemlenmesidir.

## 8. Yazılım Kalite Metrikleri (Ölçüm)

---

**Zamanlama Metrikleri:** Projenin takvime uygun yürütülmekte olduğunun gözlemlenmesine yardımcı olur.

**Maliyet ve Kaynak Metrikleri:** Verilerle proje bütçesi sürekli karşılaştırılarak yazılım geliştirme veya bakım masrafları hesaplanır.

## 8. Yazılım Kalite Metrikleri (Ölçüm)

---

**Ürün Nitelik Metrikleri:** Ürünün doğru ve kusursuz olarak üretildiğinden emin olmak için yapılır.

**Bakım ve Okunabilirlik Metrikleri:** Yazılım bakım ve okunabilirliğini ölçmek için yapılır.

## 8. Yazılım Kalite Metrikleri (Ölçüm)

---

**Üretkenlik Metrikleri:** Maliyete doğrudan etkisi olduğu için her gelişme projesi için önemli bir etken olan üretkenlik durumu tüm proje boyunca dikkatle takip edilmelidir. Bu amaçla her aşamadaki genel üretkenliği ölçebilecek metrikler tanımlanmalıdır.

Bunlar;

- **Kazanılmış değer hesaplamaları**
- **Bir aşamada ortaya çıkan hataların harcanan kişi-zaman cinsinden emeğe oranı**
- **Bir aşamada elde edilen boyut metriklerinin emeğe oranı**

# Kalite Güvence için Sorular



# Sistem Mühendisleri İçin Sorular

Kontrol Sorusu	Karar	Notlar
Önemli işlevler belirgin, anlaşılır ve sınırlandırılmış bir şekilde tanımlanmış mıdır?		
Sistemi oluşturan altsistemlerin ve ana öğelerin işlevleri, bunların birbirleriyle olan arayüzleri tanımlanmış mıdır?		
Başarım ölçütleri ve nasıl ölçüleceği belirlenmiş midir?		
Her öğe için tasarım kısıtlamaları belirli midir?		
Sistemi gerçekleştirmek üzere seçilen yol ve geliştirme yöntemi tanımlanmış mıdır, bu konuda diğer alternatif seçenekler göz önüne alınmış mıdır?		
Seçilen gerçekleştirim çözümünü teknik olarak uygulamak mümkün müdür?		
Sistemin doğrulama ve geçerlemesinin yapılabilmesi için yöntemler belirlenmiş midir?		
Sistemin öğeleri arasında herhangi bir tutarsızlık var mıdır?		

# Yazılım Proje Planlaması İçin Sorular

Kontrol Sorusu	Karar	Notlar
Yazılım geliştirme standartları belirlenmiş midir?		
Geliştirilecek yazılımın kapsamı belirlenmiş midir?		
Kullanılan terimler ve isimlendirme açık ve anlaşılır mıdır?		
Amaca ulaşmak için planlanmış özkaynaklar uygun ve yeterli midir?		
Tüm riskler belirlenmiş ve yapılacaklar tanımlanmış mıdır?		
Paralel geliştirme olanakları var mıdır, kaynaklar buna uygun olarak dağıtılmış mıdır?		
İş yükü tüm kaynaklara eşit şekilde dağıtılmış mıdır?		
Maliyet ve etkinlik çözümlemesi için kullanılmış dayanaklar geçerli midir?		
Daha önceki deneyimlere göre ve toplanmış verilere dayanarak özkaynak planlaması yapılmış mıdır?		
Öngörülen bitiş tarihleri ve maliyetler gerçekçi midir?		
Yapılan zaman programı tutarlı mıdır?		
Metrik toplama, nitelik güvence ve düzenleşim yönetim etkinlikleri yeteri kadar iyi tanımlanmış mıdır?		

# Yazılım isterler Çözümlemesi

## Gözden Geçirmesi İçin Sorular

Kontrol Sorusu	Karar	Notlar
Sistem bilgi işleme alanındaki çözümleme hassas ve doğru olarak yapılmış ve tamamlanmış mıdır?		
Yazılım ile çözülecek problemin tümü kapsanmış ve daha küçük parçalara ayrılmış mıdır?		
İç ve dış arayüzlerin tamamı tanımlanmış mıdır?		
Veri modeli tüm isterlere ait veri nesnelerini ve özelliklerini kapsamakta mıdır?		
Tüm isterleri sistem düzeyine kadar izleyebilmek mümkün mü?		
Bir prototip üretilmekte midir?		
Belirlenen kısıtlara göre sistem başarımı karşılanabilir durumda mıdır?		
İsterler planlanan zaman çizelgesine, kaynaklara ve bütçeye göre tutarlı mıdır?		
İsterlerin test edilmesi mümkün müdür?		
Doğrulama ve geçerleme yöntemleri belirli midir?		



# Ön Tasarım Gözden Geçirmek İçin Sorular

Kontrol Sorusu	Karar	Notlar
Yazılım isterlerinin tümü yazılım mimarisine yansıtılmış mıdır?		
Modül yapıları oluşturulmuş, bağımsızlık sağlanmış mıdır?		
Sistem dışı ve modüller arası arayüzler belirlenmiş midir?		
Temel veri yapıları tutarlı mıdır? Bu yapılar isterleri karşılamaya yeterli midir?		
Bakım unsurları dikkate alınmış mıdır?		
Nitelik etmenleri dikkate alınmış mıdır?		
Tüm isterler mevcut teknoloji ile karşılanabilmekte midir?		

# Kritik Tasarım Gözden Geçirmesi İçin Sorular

Kontrol Sorusu	Karar	Notlar
Modüllerin iç yapıları seçilmiş olan tasarım yöntemine uygun mudur?		
Algoritmalar arzu edilen işlevleri yerine getirebilmekte midirler?		
Hata yakalama, rapor etme ve kotarma yöntemleri belirlenmiş midir?		
Yerel veri yapıları uygun şekilde tanımlanmış mıdır?		
Programlama diline uygun yapılar kullanılmış mıdır?		
Ara katman özgü yapılar kurallarına uygun şekilde kullanılmış mıdır?		
Tasarımın ayrıntıları kullanılan programlama dili için yeterli midir?		
Başarıma yönelik tedbirler alınmış mıdır?		
Bakım unsurları dikkate alınmış mıdır?		
Nitelik unsurları dikkate alınmış mıdır?		
Yazılım Tasarım Tanımlaması içinde yer alması gereken bilgiler tutarlı mıdır?		
Tasarımı gerçekleştirmek teknik olarak mümkün müdür?		

# Yazılım Bakımı Soruları

Kontrol Sorusu	Karar	Notlar
Yazılım kabulü için gerekli kabul gözden geçirmesi yapılmış mıdır?		
Kaynak kod tekrar üretilebilecek şekilde saklanmış mıdır?		
Tüm belgelerin tüm sürümleri saklanmakta mıdır?		
Yazılım bakımı, başka bir grup tarafından yapılacaksa, yazılım aktarımı uygun şekilde gerçekleştirilmiş midir?		
Herhangi bir değişiklik isteği için neler yapılması gerektiği belirtilmiş midir?		
Değişiklikler uygun şekilde belgelendirilmekte midir?		
Değişiklik Kontrol Kurulu tanımlı mıdır?		
Yapılan değişikliklere göre yeni sabitlemeler yapılmakta mıdır?		

# KAYNAKLAR

---

- ▶ M.Erhan SARIDOĞAN-Yazılım Mühendisliği
- ▶ Ali ARİFOĞLU & Ali DOĞRU-Yazılım Mühendisliği
- ▶ Oya KALIPSIZ & Ayşe BUHARALI & Göksel BİRİCİK-Sistem Analizi ve Tasarımı
- ▶ Roger S. Pressman-Software Engineering
- ▶ <http://www.csharpnedir.com/makalegoster.asp?MId=201>
- ▶ <http://bornova.ege.edu.tr/~sengonca/Tanil/02.doc>
- ▶ [http://www.bilgiyonetimi.org/cm/pages/mkl\\_gos.php?nt=55](http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=55)