

Yazılım Kalite Güvencesi ve İlişkili Kavramlar

İçerik

- Kalite Nedir ?
- Yazılım Kalitesi
- Yazılım Kalite Güvence ve Öğeleri
- ISO 12207 Standardı ve Yazılım Kalite Güvence
- Kalite Maliyeti

“Kalite” Nedir?

Deming, 1968

- “Ability to satisfy requirements”
→ İhtiyaçları karşılama yeteneğidir.

Crosby, 1979

- “Conformance to user requirements”
→ Kullanıcı gereksinimlerine uygunluktur.

Price, 1985

- “Doing right it the first time”
→ İlk seferde doğrusunu yapmaktır.

Juran, 1988

- “Fitness for intended use”
→ Amaçlanan kullanıma uygunluktur.

“Kalite” Nedir?

- ISO 9001:2000
 - “the degree to which a set of *inherent* characteristics fulfills *requirements*”
 - (Nesnenin) tabiatında var olan özelliklerin gereksinimleri karşılama derecesidir.
- National Institute of Standards and Technology, 2003
 - Baldrige National Quality Program
 - “Customer-driven quality”
 - Müşteri güdümlü kalite

Kalite Mitleri

- “Kalite soyut bir kavramdır ve ölçülemez”
 - Kalite, uygunsuzluğun maliyeti (hata maliyeti) ile ölçülebilir.
- “Bizim işimiz diğerlerinden farklı”,
“Daha iyisini yapmanın maliyetini şu an için karşılayamayız”
 - Kaliteyi baştan sağlamanın (hata önleme) maliyeti, sonradan sağlamaya çalışmanın (hata bulma ve düzeltme) maliyetinden daha azdır.
- “Kalite problemlerine en alt kademede çalışanlar sebep olur”
 - En alt kademede çalışanlar, daha üst kademede çalışanlardan daha az hata maliyetine sebep olur.
- “Kaliteyi, kurumların Kalite Bölümleri başlatır”
 - Kalite anlayışı üst yönetim seviyesinde başlar ve tüm çalışanlar tarafından benimsenmelidir.

Kalite Nasıl Sağlanır?

- Geleneksel anlayış: Hata ayıklama
 - “Kalite kontrol”
 - Bir ürün veya hizmetin tanımlanmış gereksinimleri karşılayıp karşılamadığının tespitinde kullanılan teknikler ve uygulanan faaliyetler
 - Ürün/hizmet üzerinden kaliteyi sağlama anlayışı
- Gelişmiş anlayış: Hata önleme
 - “Kalite güvence”
 - Bir ürün veya hizmetin tanımlanmış gereksinimleri karşılanmasını yeterli derecede güvence altına almak için gerekli olan, tüm planlanmış ve sistematik faaliyetler
 - Ürünü/hizmeti oluşturan sistem üzerinden kaliteyi sağlama anlayışı

Ürün Yoluyla Kalite

- Üretim hattını durdurmak yerine; kaza eseri olduğunu varsayarak hatalı ürünler birleştirilir.
 - Hataların giderilmesi sonraya bırakılır.
- Çalışanı hatayı tekrarlamamaya zorlar.
 - Çalışanların hataları saklamasına neden olur.
 - Yönetimin çalışanlara karşı tavır almasına sebep olur.

Süreç Yoluyla Kalite

- Hatalı ürün tespit edildiğinde; bunun ürünü geliştiren sürecin bir problemi olduğu varsayılarak üretim hattı durdurulur.
 - Hatanın kaynağı bulunur ve iyileştirilir.
- Çalışanları hataları bulmaya ve sistemi iyileştirmeye teşvik eder.
 - Yönetsel rollerde çalışanların görev almasını sağlar.
 - Alt kademedeki çalışanların görev almasını sağlar.

Kalite Kontrol ve Kalite Güvence

- Kalite kontrol
 - Ayıklamaya yönelik, sistematik, belirli bir anda
- Kalite güvence
 - Önlemeye yönelik, sistematik, zamana yayılmış
- Geliştirme sürecinin kalitesini iyileştirirsek; hem ürün kalitesini iyileştirmiş, hem de geliştirme maliyetini ve zamanını azaltmış oluruz.
 - “Kalite bedavadır”

Yazılım Kalitesi

- Geliştirdiğimiz yazılımın kalitesi, yazılımı nasıl geliştirdiğimize büyük ölçüde bağlıdır.
 - Yazılım geliştirme süreci, yazılımı nasıl geliştirdiğimizi tanımlar.
 - Örnek: Çağlayan modeli
 - Analiz → tasarım → gerçekleştirme → test → bakım
 - Yazılım geliştirme sürecinin temel aşamalarını tanımlar.
- Kaliteyi, yazılım geliştirme aşamaları boyunca yazılım ürününe yerleştirmek zorundayız.
 - Kaliteyi en sonunda sağlamaya çalışmak hem zordur, hem de maliyet çok yükselir.

Kaliteli Yazılım Ne Demektir?

- Kabullenilebilir hata seviyesinde,
- Gereksinimleri karşılayan,
- Amaçlanan kullanıma uygun,
- Zamanında tamamlanmış,
- Belirlenen bütçe sınırları içinde gerçekleştirilmiş,
- Standartlara uyumlu,
- Bakımı sağlanabilen yazılımdır.

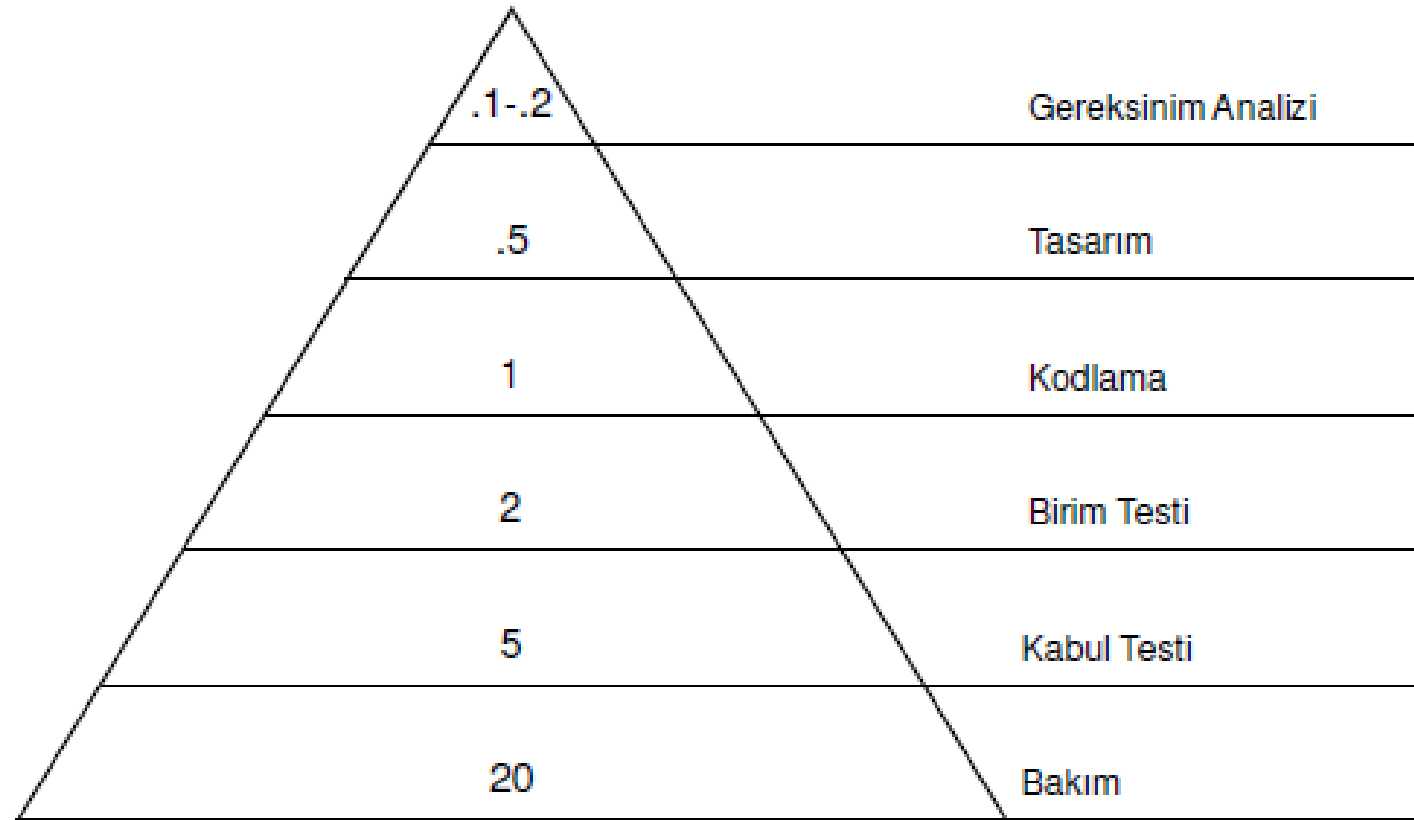
Yazılımda Kalitesizliğin Sorunları

- Müşteri tarafındaki sorunlar:
 - Gereksinimlerin sağlanamayışı
 - Kolay anlaşılabilir ve kullanılabilir olmaması
 - İstenilen zamanda bakım yapılabilir olmaması
 - Eğitim desteğinin yetersiz olması
- Yazılım firması tarafındaki sorunlar:
 - Geciken ya da bitmeyen projeler
 - Yüksek maliyet
 - Çalışanların tatminsizliğı
 - Firmaya olan güven kaybı

Yazılımda Kalite Neden Gerekli?

- Tecrübeli bir mühendis her 7-10 satırda bir hata üretmektedir.
 - Orta ölçekli bir projede binlerce satıra karşılık gelir.
- Hataların çoğunun test aşamasında düzeltilmesi gerekmektedir.
- Testler uzadıkça maliyet artmakta, teslimat gecikmekte, ürün kalitesi düşmekte ve bakım maliyeti geliştirme maliyetini aşmaktadır.

Yazılım Hatalarını Düzeltme Maliyeti



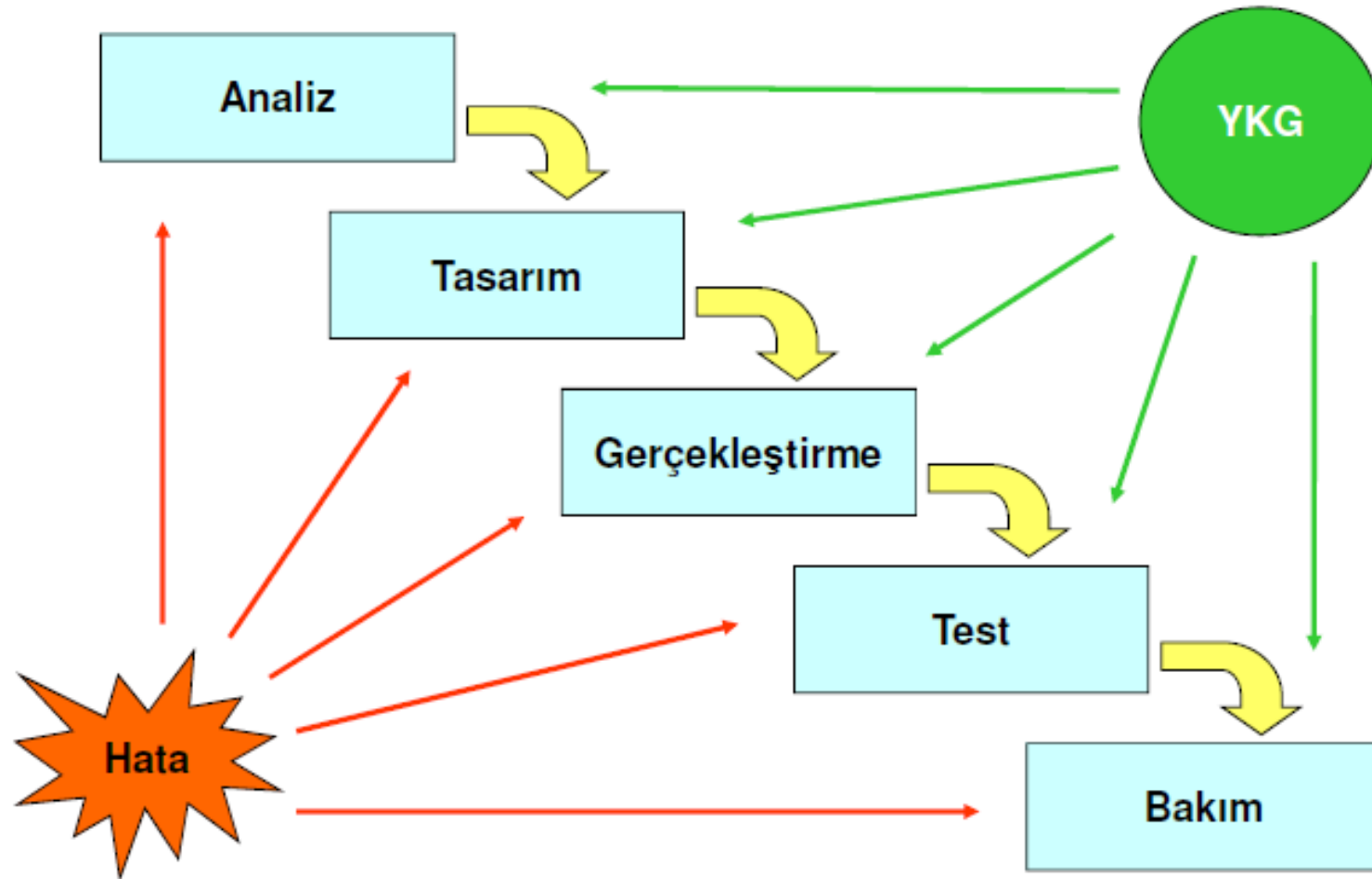
Karşılaştırmalı Hata Düzeltme Maliyeti Oranı

Yazılım Kalite Güvence ve Öğeleri

Yazılım Kalite Güvence

- Yazılımın geliştirmesi ve bakımı boyunca kaliteyi sağlamanın maliyet-etkin yoludur.
- Kalitenin yazılım ürününe yerleştirilmesi için, proje yaşam döngüsü boyunca üretilen ürünlerinin ve uygulanan süreçlerinin, tanımlı gereksinimlere uyduğunu güvence altına almayı hedefler.
- Proje yaşam döngüsü boyunca çeşitli etkinlikleri planlayarak, işleterek ve izleyerek hataların erken tespit edilmesine hizmet eder.

Yazılım Kalite Güvence



Yazılım Kalite Güvence

- Yazılım Kalite Güvence (YKG), yazılım ürünün istenilen kalitede olmasını güvence altına alan ve geliştirme süreci boyunca uygulanan etkinlikleri içerir.
 - “Software quality assurance”
- Doğrulama ve Geçerleme (D&G), yazılım geliştirme sürecinin her aşamasındaki çıktıların, istenilen özelliklere uygunluğunu kontrol etmeye yarayan etkinliklerdir.
 - “Verification and validation”
- Test, bir ürünlerdeki hataların ortaya çıkarılması etkinliğidir.

D&G, YKG'nin bir parçasıdır ve Test'i içerir.

YKG > D&G > Test

Yazılım Kalite Güvence: Yöntemler

- **Statik yöntemler:** Kodu çalıştırmadan yapılır.
 - Örnek: Gereksinim ve tasarım belgeleri ile kodun denetimi veya gözden geçirilmesi
 - İnceleme ("inspection")
 - Gözden geçirme ("review")
 - Denetleme ("audit")
- **Dinamik yöntemler:** Kodu çalıştırarak yapılır.
 - Örnek: Ürünün veya bileşenlerinin gerçeğe yakın testi
 - Birim test ("unit test")
 - Tümlleştirme test ("integration test")
 - Sistem test ("system test" / "functional test" / "qualification test")
 - Kabul test ("acceptance test")

Doğrulama ve Geçerleme

- Doğrulama (“verification”): Müşterinin istediği ürünü, doğru şekilde geliştirmeyi güvence altına alır.
 - Mühendislik adımlarının doğru uygulanıp uygulanmadığının kontrolüdür.
 - Her aşamada, bir önceki aşamada yapmayı taahhüt ettiklerimizi yaptık mı? Bir önceki belgede tanımladıklarımıza göre uyumsuzluk veya eksik var mı?
 - “*Are we building the product right?*”
- Geçerleme (“validation”): Müşterinin istediği, (doğru) ürünü geliştirmeyi güvence altına alır.
 - Mühendislik ürününün müşteri ihtiyaçlarına uyup uymadığının kontrolüdür.
 - Her aşamanın çıktısı müşteri gereksinimlerini karşılıyor mu?
 - “*Are we building the right product?*”

Test

- Bir programın davranışını; beklenen davranışa uymadığı durumları bulma amacı ile, sonsuz bir küme içinden sınırlı sayıda seçilen test vakalarını kullanarak, dinamik yöntemlerle sınaama işlemidir.
[SWEBOK 2004].
 - Beklenen : Tanımlanmış gereksinimlere uyan
 - Sınırlı : Yeterli sayıda
 - Seçilmiş : Uygun test vakaları
 - Dinamik : Kod çalıştırılarak

Amaç: Hata Önleme veya Bulma

- YKG'nin amacı hataları bulmaktan çok önlemektir.
 - En uygun yazılım geliştirme ortamının kurulması
 - Gerekli süreçlerin ve talimatların tanımlanması
 - Süreç uygulamalarının etkinliğini ölçme yöntemlerinin belirlenmesi
 - Uyulacak standartların ve kullanılacak şablonların tanımlanması
 - ...
- D&G ve Test hata bulmaya odaklanır.
 - Denetleme, Gözden Geçirme, İnceleme (statik)
 - Birim, Tümlleştirme, Sistem, ve Kabul Test (dinamik)

ISO 12207 Standardı ve Yazılım Kalite Güvence

ISO/IEC 12207-2008 (IEEE Std 12207-2008)

- Sistem ve yazılım mühendisliği –
Yazılım yaşam döngüsü süreçleri
- 1995'den bu yana sektörde yaygın olarak kullanılmaktadır.
 - 2008 revizyonu, 1995 tarihli standart ile standardın 2002 ve 2004 yıllarında yayınlanan iki ekini birleştirmiştir.
- Kurumun veya projenin ihtiyaçlarına göre uyarlanabilir.
 - Standartta tanımlanan uyarlama kuralları ve önerileri dikkate alınmalıdır.
 - Tam uyumluluk / Seçerek uyarlama

ISO/IEC 12207 Kapsamı

- Daha büyük bir sistemin parçası olarak veya tek başına yazılım ürünü veya hizmeti için, bir grup yaşam döngüsü süreci ile bunların altındaki etkinlikleri ve görevleri tanımlar.
- Yazılım ürününün veya hizmetinin satın alınması, tedariği, geliştirilmesi, işletilmesi, bakımı ve elden çıkarılması boyunca referans alınabilir.
- ISO/IEC 15288 standardı ile yapı, terimler ilişkili süreçler açılarından tam olarak uyumludur.
 - ISO/IEC 15288-2008: Sistem ve yazılım mühendisliği – Sistem yaşam döngüsü süreçleri

ISO/IEC 12207 Kısıtları

- Herhangi bir yaşam döngüsü modelinin, geliştirme yaklaşımının veya yönteminin kullanılmasını öngörmez.
- Her sürecin beklenen çıktıları üreterek amacına ulaşması için uygulanacak etkinlikleri ve görevleri tanımlar.
 - Süreci, yöntem ve teknikleri içerecek şekilde detaylandırmaz.
 - Sürecin amacına ulaşması için neler yapılması gerektiğini tanımlar.
 - Süreç kapsamındaki etkinlikler ve görevler doğal bir sırada tanımlanmış olsa da bunlar, sürecin nasıl uygulanacağı konusunda yönlendirme vermez.
- Bilgi ürünlerinin (belgelerin) adı, biçimi, içeriği ve materyali ile ilgili detayları tanımlamaz.

Kalite Maliyeti

Kalite Maliyeti (“Cost of Quality”)

- Ürünler ve onları üreten süreçler mükemmel olduğunda ortadan kalkacak toplam maliyettir.
- Üretim ve hizmet sektörlerinde yaygın olarak kullanılmaktadır.
- Süreçteki zayıf halkaların tespit edilerek kalite maliyetinin düşürülmesini hedefler.

Kalite Maliyeti Kategorileri

- Uygunsuzluk maliyeti (“cost of non-conformance”)
 - Kaliteyi sağlamayı başaramadığımızda ödediğimiz bedeldir.
 - “Failure costs”: hatanın sebep olduğu maliyet
 - “Internal failures”: teslimattan önceki
 - “External failures”: teslimattan sonraki
- Uygunluğu sağlamanın maliyeti (“cost of conformance”)
 - Kaliteyi sağlamak için ödediğimiz bedeldir.
 - “Appraisal costs”: kalitesizliğin tespit etmenin maliyeti
 - “Prevention costs”: kalitesizliği önlenmenin maliyeti

Hata Maliyeti (“Failure Costs”)

- Ürünün sağlaması beklenen özelliklerle ilişkilidir.
 - Müşteri ihtiyaçlarını karşılayamama
- İç ve dış hata maliyetleri toplanarak hesaplanır.
 - İç hata maliyeti: tekrar çalışma (“rework”), tekrar test (“re-test”)
 - Dış hata maliyeti: müşteriden dönen ürün, müşteri şikayetleri
- Kalite maliyetinin düşürülmesinde anahtar kategoridir.

Hata Bulma Maliyeti (“Appraisal Costs”)

- Ürün veya hizmetin, kalite standartlarına ve performans gereksinimlerine uyduğunu ölçmek, değerlendirmek veya denetlemek ile ilgilidir.
- Hata giderme etkinlikleri:
 - Gözden geçirmeler, ürün denetlemeler, beta testler

Hata Önleme Maliyeti (“Prevention Costs”)

- Bir ürün veya hizmetin hata maliyetini (“failure costs”) ve hata giderme maliyetini (“appraisal costs”) en aza indirmek üzere tasarlanmış etkinliklerle ilişkilidir.
 - Hata önleme yaklaşımlarının keşfi, uygulaması ve iyileştirilmesi için yatırım gerektirir.
- Hata önleme etkinlikleri:
 - Eğitim, araç ve yöntemler, kalite planlama, süreç yönetimi

Donanım vs. Yazılım Kalite Maliyeti

- Donanım kalite maliyeti dağılımı, tasarım sonrasında yoğunlaşır.
 - Tasarım : %10
 - Yıpranma : %60
 - Üretim : %30
- Yazılım kalite maliyeti, geliştirme aşamasında yoğunlaşır.
 - Geliştirme : %99
 - Yıpranma : %0
 - Üretim : %1