YMT 412-Yazılım Kalite Güvencesi ve Testi

Kalite Konseptleri

Bölüm-1

Dr. Muhammet BAYKARA

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

Ders Hakkında Temel Bilgilendirme

Ders Sorumlusu : Dr. Muhammet BAYKARA

Ders. Lab. Yardımcısı : Arş. Gör. Burak Alakuş

Ders Saatleri : 3 saat teori, 2 saat laboratuar

Dersler A406'da uygulamalar ise laboratuar ortamında yapılacaktır.

Devam / Devamsızlık Durumu : Teori : **%30** – Uygulama : **%20**

Dersin Amacı :

1. Yazılım doğrulama testi ve geçerleme tekniklerinin öğrenilmesi ve güncel test araçları ile yazılımların analizi.
2. Yazılım Kalite süreç standartlarının öğrenilmesi.

Dersin İşleniş Biçimi &Nasıl Geçerim

- Teorik anlatım
- Haftalık Ödevler
- Dönemlik araştırma ve uygulama projesi
  - Raporlama + Sunum
- Ara sınav, Final Sınavı ve Quizler
- Ders Geçme Notu = (Ara sınav(%33) + Haftalık Ödevler(%33) + Quizler(%34)) *0,4 + ((Final+Proje)/2)*0,6

Temel Kaynaklar

1)Ali Arifoğlu, Ali Doğru, Yazılım Mühendisliği, SAS Bilişim Yayınları, 2004.
2) Roger Pressman, Bruce Maxim, Software Engineering: A Practitioner's Approach, 8th edition, 2014
3) Jeff Tian, 2005,Software Quality Engineering: Testing, Quality Assurance, and Quantifiable, Wiley-IEEE Computer Society.
4) Braude, E.J. and Bernstein, M.E., Software Engineering – Modern Approaches, Wiley, 2011.
5) Hassan Gomaa, Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Cambridge Publications, 2011.
6) Software Quality Assurance: From Theory to Implementation by Daniel Galin, Addison-Wesley, 2004, ISBN: 0201709457.
7) Yazılım Test Rehberi, Asiye Bozkurt, Ahmet Adıgüzel, Adem Çüçen, Seçkin Yayıncılık, 2016.
8) Kaliteli Yazılım nasıl Geliştirilir, Hakan Atbaş, Pusula Yayıncılık, 2012.
9) Yazılım Proje Yönetimi, Ali Nizam, Papatya Yayıncılık, 2013.

İçindekiler

1. Kalite nedir?

- Kalite üzerine değişik bakış açıları ve değişik ekoller tarafından farklı tanımlar vardır. En basit tanımıyla kalite, müşteri isteklerine cevap verebilmektir. Genel bir tanım olarak ise bir ürün veya hizmetin belirlenen ihtiyaçları karşılama kabiliyetine dayanan özelliklerin toplamıdır.
- Örneğin, bir müşterinin satın almak istediği otomobilin performansı ile ilgili beklentisi sadece düşük yakıt tüketimi ise ve otomobil bunu sağlıyorsa, müşteri açısından otomobilin kaliteli olduğu söylenebilir.

user satisfaction = compliant product + good quality + delivery within budget and schedule

1. Kalite nedir?

Kalitenin aynı zamanda soyut ve karmaşık bir kavram olduğunu görürüz. Farklı ürün veya hizmetler için ya da farklı kişiler-üreticiler, tasarımcılar, yöneticiler, müşteriler, hatta farklı kalite guruları-açısından kalitenin algılanışı birbirinden farklıdır.

- Amaçlanan kullanıma uygunluktur. (Joseph M. Juran)
- Gereksinimlere uygunluktur. (Philip B. Crosby)
- Nesnenin tabiatında var olan özelliklerin gereksinimleri karşılama derecesidir. (ISO 9000)
- Bakımın sonucudur. (Robert Pirsig)

Kalite=Algılanan Performans/Beklentiler

1.1. Yazılım Kalitesi:

- Kullanım amaçlarına göre açıkça tanımlanmış işlev,
- Başarı gereksinimlerine uyum,
- Kullanıcı isteklerine yanıt verebilme,
- Standartlara sadık kalma,
- Yüksek güvenilirlik sağlama,
- Çeşitli teknik özelliklere sahip olma,
- Teslim sonrası destek olarak tanımlanabilir.

1.2. Kalitenin Önemi

Kara Pazartesi 1987: Hisselerin otomatik satış talimatlarını yanlış zamanda tetikleyen yazılım, gün içerisinde borsanın ilk açıldığı yer olan Hong Kong'dan başlayıp bir günde Dow Jones'u %23, S&P 500'ü %20 düşürerek hala kırılmamış bir rekora imza atmıştır. (1987)

Therac-25 tedavi cihazındaki hata: Cihazın Hata mesajı verdiği durumda sinyalin hastaya uygulanmamış olması gerekirken yüksek güçlü radyasyon hastalara uygulanmıştı. Operatörün işlemi birkaç kere tekrar etmesiyle hastalar çok büyük oranda radyasyona maruz kalmışlardır. Cihazdaki hata sonucu 1985-1987 yılları arasında 6 kişi hayatını kaybetti.

Patriot Füze Hatası (1991) : Tarafına atılan füzeleri vurmak için tasarlanan Patriot Füzeleri 1. Körfez savaşı sırasında yazılımındaki bir yuvarlama hatası nedeniyle hedeflerinin tamamını vurmaktan uzak kalmıştır.

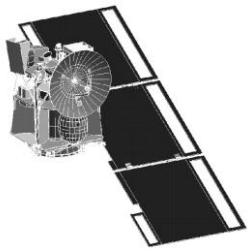
Mars Uydusu Mars'a Çakıldı (1998) : NASA'nın belirlediği standartların dışında (metrik) uzunluk ölçüsü kullanmayan bir yazılım taşeronu Mars'ın yörüngesinde dönmesi gereken 125 milyon$'lık Mars uydusunun gezegene çakılmasına sebep oldu.

1.2. Kalitenin Önemi

Ariane 5 roket kazası, 4 Haziran 1996

- Ariane 5 roketi fırlatıldıktan 40 sn sonra parçalandı.
- Zarar yaklaşık yarım milyar dolardı.
- Tarihteki en pahalıya mâl olan yazılım hatalarından biridir!
- Ariane 4' te kullanılan bir modül düzgün test edilmeden yeniden kullanılmıştı ve Ariane 5'e uyum sağlamamıştı.
- Parçalanmanın sebebi bir yazılım hatasıydı. 64 bitlik ondalıklı sayı, 16 bit işaretli tam sayıya çevrilirken bulunan sonuç beklenenden büyük çıkıyordu.

1.2. Kalitenin Önemi



Mars Climate Orbiter hatası, 23 Eylül 1999

- Gezegenler arası ilk iklim uydusu olarak 1997'de fırlatılan Mars Orbiter, 1999'da kayboldu.
- Kazanın bir yazılımda İngiliz ölçü birimlerinin metrik sisteme yanlış çevrilmesinden kaynaklandığı belirtildi.
- NASA ekibi hesaplarında İngiliz ölçü birimini (inç, feet), projeye katılan diğer ekipse metrik (mm, cm, m) sistemi kullanmıştı.
- 125 milyon dolarlık uydu yörüngeye sabitlenmeye çalışırken Mars'a olması gerekenden fazla yaklaştı. Uydunun, Mars'ın atmosferinde imha olduğu düşünülüyor.

1.2. Kalitenin Önemi

İntel İşlemci Bölme Sorunu (1993) : İntel İşlemcilerin ondalık sayıların bölüm sonuçlarında 0.006'lık bir sapma ile hata yapması sattığı 5 milyon chip için 475milyon$'ına mâl oldu. Intel bu hatanın chip başına neye mâl olduğunu hesaplamaya çalışırken hatayı da düzeltmiştir.

(Toplam Maliyet / Chip Sayısı)

L.A. Havaalanı (2007) : Network kartındaki ufak bir hatadan dolayı 8 Saat boyunca kimse LA Havaalanı'ndan Amerika'ya giriş çıkış yapamadı.

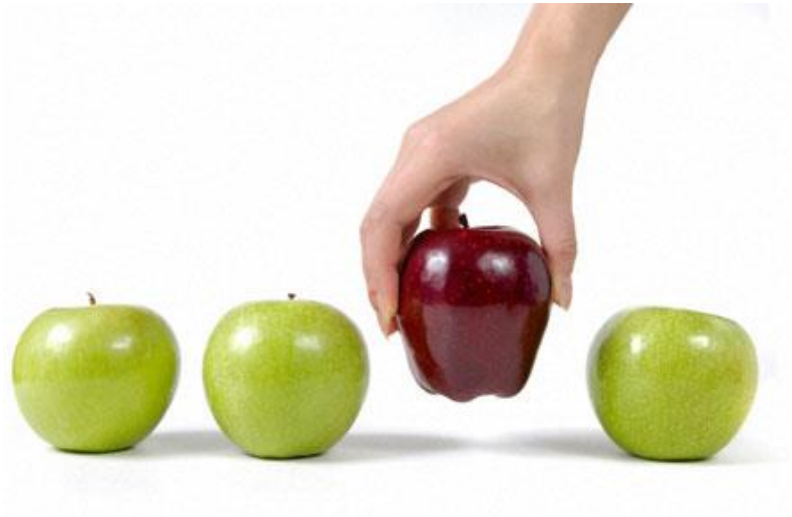
Toyota Prius (2014): Toyota 160.000 Prius Hybrid Aracını bir motorunu zamansız durduran yazılım hatasından dolayı geri çağırdı. Bu otomobil tarihindeki ilk büyük yazılım kökenli geri çağırmadır.

1.3. Kalite Örnekleri

Bakış Açısı: Kalite, ürün niteliklerine bakılarak ölçülür.

Yazılım: Yazılımın özelliklerini ölçeceğiz. Örneğin, hataların ortaya çıktığı ortalama süreye bakarak güvenilir olup olmadığına karar vereceğiz.

Elma: Uygun boyut ve şekle, iyi tada ve renge sahip olmadığına bakacağız.



1.3. Kalite Örnekleri

Bakış Açısı: Kalite kullanıma uygunluktur.

Yazılım: Kullanıcılara yazılımın görevlerini gerçekleştirip gerçekleştirmediğini soracağız.

Elma: Tarifimiz için uygun olup olmadığına bakacağız.

1.3. Kalite Örnekleri

Bakış Açısı: Kalite, iyi üretim sürecine ve tanımlanan isteklerin karşılanmasına dayanır. Testle, hata ve başarısızlıkların analiziyle ölçülür.

Yazılım: Önceden tanımlanmış yazılım geliştirme süreci kullanacağız. İşleyişi aksatacak, sorun yaratacak kusurlar içermemelidir.

Elma: Satın almadan önce organik olarak üretilip üretilmediğine ve üzerinde herhangi bir leke ve zararlı olup olmadığına bakacağız.

1.4. Kalite Kontrol ve Kalite Güvence

- Kalite Kontrol: Geliştirilen ya da üretilmiş bir ürünün kalitesini değerlendirmek için tasarlanmış aktiviteler kümesidir. Hata ayıklamaya yönelik, sistematik ve belirli bir anda yapılır.
- Kalite güvence: Ürünün nitelik için belirlenen özelliklerini karşılamak maksadıyla yeterli güveni sağlamak için gereken planlı etkinlikler kümesidir. Hata oluşmasını önlemeye yönelik, sistematik, zamana yayılmıştır.

1.5. Yazılımda Kaliteye Ulaşmak

- Kalite, iyi proje yönetiminin ve katı mühendislik kurallarının uygulanması sonucunda ortaya çıkar. Proje yönetimi ve kurallar, yazılım takımının yüksek kaliteye ulaşmasında yardımcı olmak için aşağıdaki maddeler çerçevesinde uygulanır.

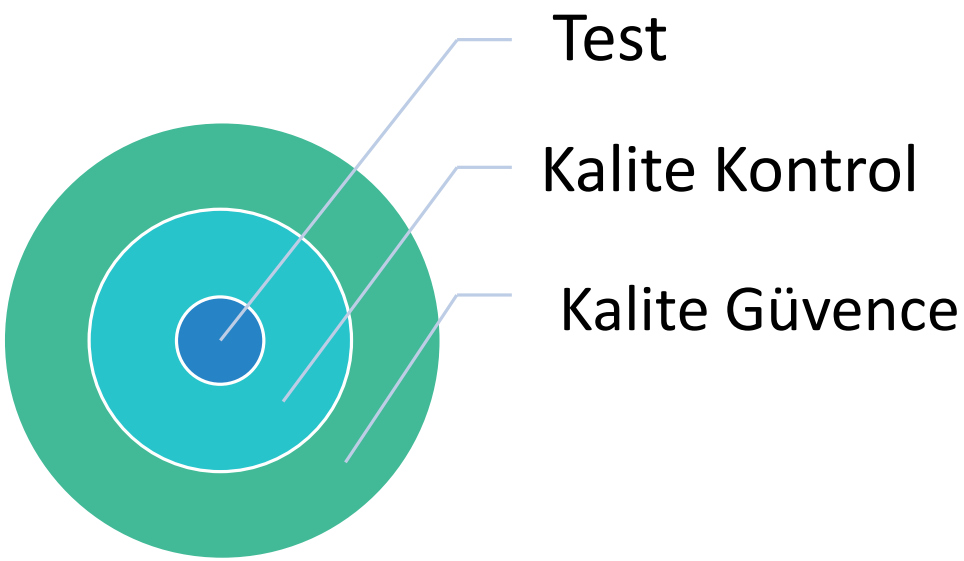
- Yazılım mühendisliği metotları
- Proje yönetim teknikleri
- Kalite kontrol eylemleri
- Yazılım kalite güvencesi

1.5. Yazılımda Kaliteye Ulaşmak

- Kalite güvence yönetimi sayesinde:
  - Geliştirilen yazılımın sonradan ortaya çıkan kusurları azaltılmış olur.
  - Test ve bakım aşamalarında daha az işgücü gerekir.
  - Yüksek güvenilirlik, müşteri memnuniyetini artırır.
  - Maliyet düşer.
  - Çalışanlar arasında kurulan iş disiplini sayesinde verimlilik artar.

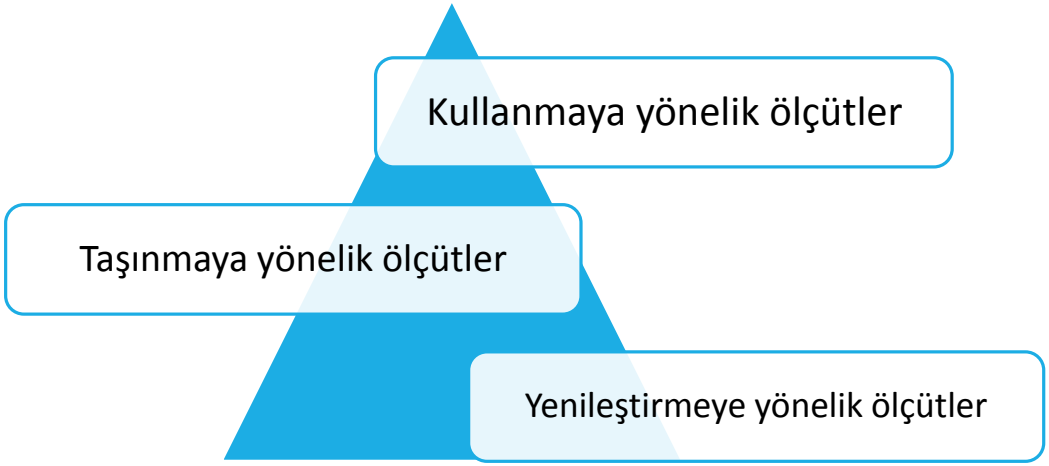
1.4. Kalite Kontrol ve Kalite Güvence

- Kalite kontrol ve test, kalite güvencenin alt elemanlarıdır.



2. Kalite Ölçütleri

- Bir yazılımın kalitesini belirlemek için kullanılan ölçütlerdir. Çeşitli gruplara ayrılır. Bu grupları şu şekilde sıralayabiliriz:



2.1. Kullanıma Yönelik Ölçütler

- Bir yazılım ürününün yalnızca kullanımını dikkate alırsak, kullanıcı hoşnutluğunu en iyi şekilde sağlayacak özellikleri listelemek gerekir:

İşlevsellik	**Doğruluk**	**Sağlamlık**
Güvenilirlik	**Verimli Çalışma**	**Korunmalı Olma**
Kullanım Kolaylığı	**Ekonomiklik**	**İşletim Sürekliliği**

2.1. Kullanıma Yönelik Ölçütler

- **İşlevsellik:** Yazılım kullanıcının tüm isterlerini karşılayabilmelidir. Ana işlevler yanında yardımcı işlevlerle zenginleştirilmiş olmalıdır.
- **Doğruluk:** Yazılımın öngörülen tüm işlevleri istenildiği şekilde, doğru ve yeterli hassaslıkta yerine getirebilmesidir.
- **Güvenilirlik:** Yazılım güncel bir sürüme sahip olmalı, sürekli ve hatasız çalışabilmeli, tüm işlevleri doğru yapmalı, hatalı girdilere ve kullanıcı yanlışlıklarına karşı korumalı olmalıdır.

2.1. Kullanıma Yönelik Ölçütler

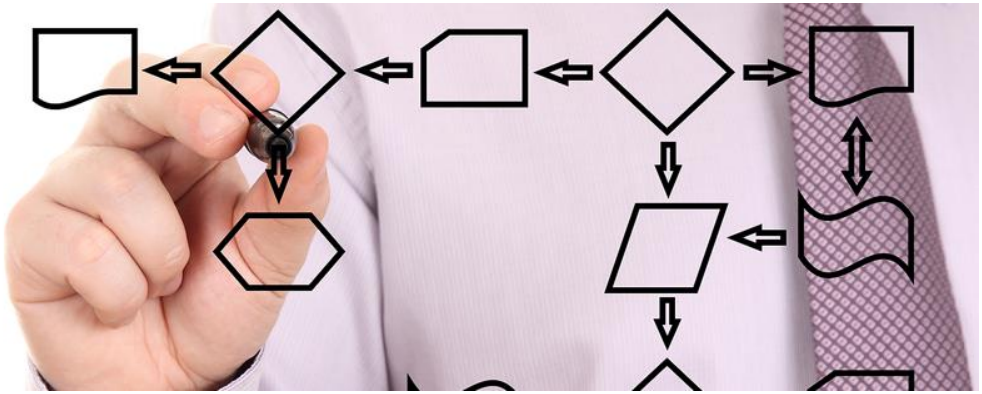
- **Sağlamlık:** Yazılım, normal olmayan çalışma koşullarında da güvenle çalışabilmeli, sisteme hata hoşgörülü bir yapı kazandırılmalıdır. Tasarım ve çözümleme aşamasında belirlenemeyen bazı koşullar sonradan ortaya çıksa da sistem işlevlerini yerine getirebilmelidir.
- **Verimli çalışma:** Yazılım işlevlerini yerine getirirken sistem öz kaynaklarını uygun şekilde kullanmalıdır. İşlemci, bellek, disk ve diğer kaynakların kullanımı etkinlikle yapılmalı, iletişim, grafik sergileme ve yazıcı çıktıları yüksek başarıma sahip olmalıdır.

2.1. Kullanıma Yönelik Ölçütler

- **Kullanım kolaylığı:** Yazılım insanların kullanım amaçlarına hizmet ettiğinden birinci öncelik her zaman insan unsuru olmalıdır. Üretilen yazılımı insanların rahatlıkla kullanabilmesi için gerekli kolaylıklar sağlanmalı, özellikle grafiksel kullanıcı ara yüzü düzenli, estetik ve kullanımı kolay olmalıdır.
- **Korunmalı Olma:** Yazılım yetkisiz kişilerin yapabileceği değişikliklere izin vermeyecek şekilde nesne kodunu, veri yapılarını, yapılandırma dosyalarını ve belgeleri elektronik ortamda koruma altına alabilmelidir.

2.1. Kullanıma Yönelik Ölçütler

- Ekonomiklik: Yazılıma uygun fiyata sahip olmak, sürekli kullanabilmek, eldeki donanım olanaklarını değerlendirebilmek kullanıcı için önemli ölçütlerdir.
- İşletim Sürekliliği: Yazılımın sürekli kapatma açma gerekmeden çok uzun süre aynı başarım ve doğruluk düzeyinde çalışabilmesidir.



2.2. Taşınmaya Yönelik Özellikler

- Geliştirilen yazılımın bir başka yerde kullanılabilmesi, bir başka ortama taşınması ya da başka bir geliştirici grubuna aktarılması gerektiğinde zorluklarla karşılaşılmaması için yazılımı geliştirirken dikkate alınması gereken özellikler şu şekildedir:

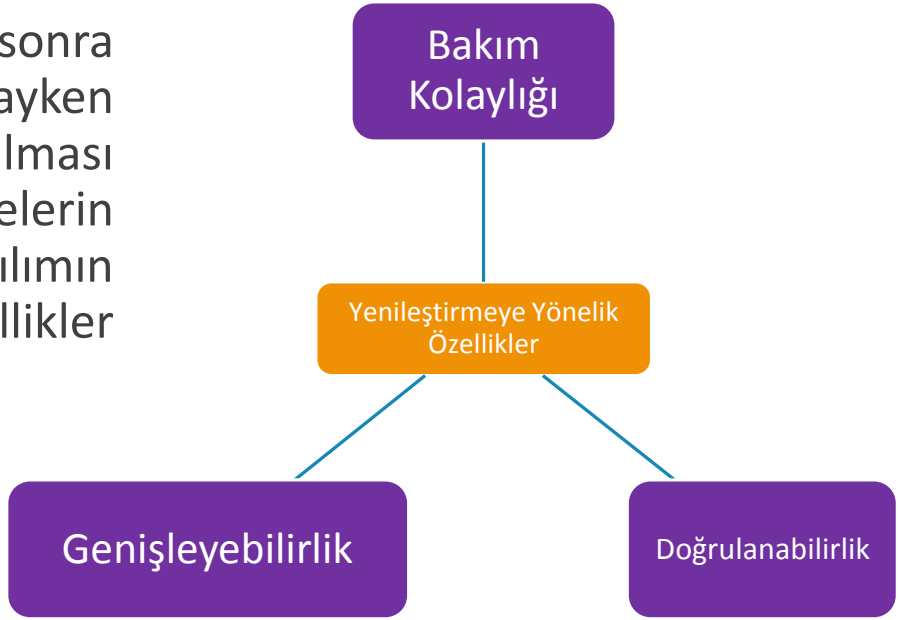
Tekrar Kullanılabilirlik	Uyumluluk	Taşınabilirlik

2.2. Taşınmaya Yönelik Özellikler

- **Taşınabilirlik:** Yazılımlar farklı bilgisayar donanım ve yazılımının kullanıldığı ortamlara aktarılabilir olmalıdır. Yeni işletim sistemi sürümlerinin kullanılması, bilgisayar değiştirilmesi gibi durumlarda, daha önce geliştirilmiş olan yazılım en az çaba ile tekrar kullanılabilmelidir.
- **Uyumluluk:** Bir yazılım ürünü daha önce üretilmiş olan veya beraber çalışan diğer ürünlerle tam uyumlu olmalıdır. Birbiriyle etkileşen sistemler ortak özelliklere sahip olarak yaratılmalıdır. Bunun için dosya biçimleri, veri yapıları, kullanıcı ara yüzleri üzerinde belirli bir standart oluşturulmalı, tüm yazılımlar bunlara uygun olarak tasarlanmalı ve geliştirilmelidir.
- **Tekrar kullanılabilirlik:** Yeni geliştirilen yazılım daha sonra kısmen veya tamamen yeni bir uygulamada kullanılabilmelidir. Tasarım ve gerçekleştirim bu amaca hizmet edecek şekilde olmalıdır.

2.3. Yenileştirmeye Yönelik Özellikler

- Yazılımın bir kez üretildikten sonra kullanıma sunulması, kullanımdayken de değişikliğe gereksinim duyulması doğaldır. Bu tür yenileştirmelerin kolaylıkla yapılabilmesi için yazılımın sahip olması gereken özellikler şunlardır:



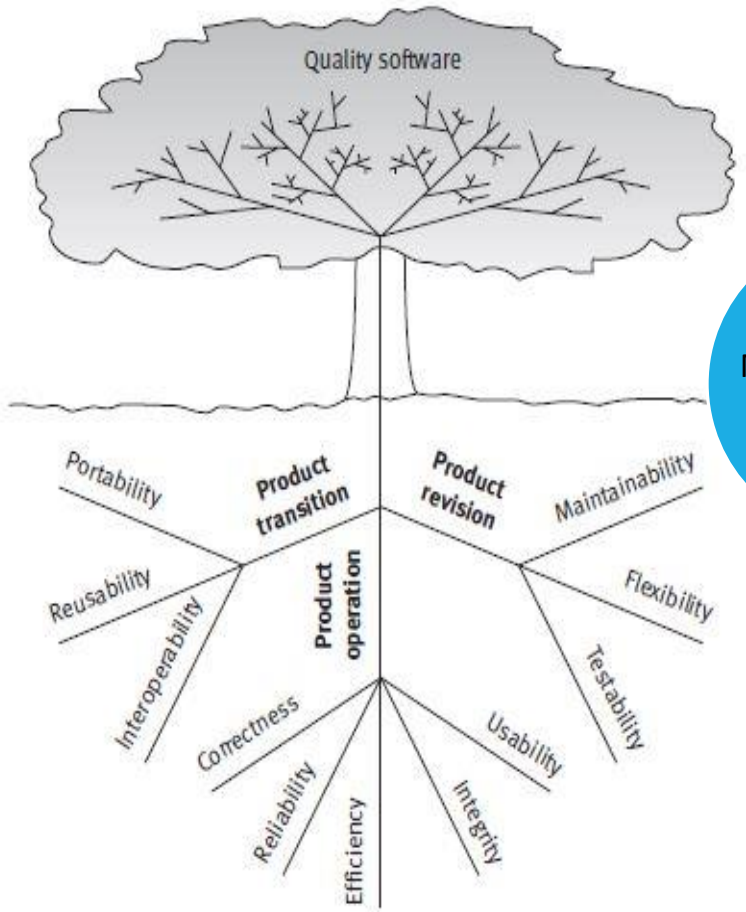
2.3. Yenileştirmeye Yönelik Özellikler

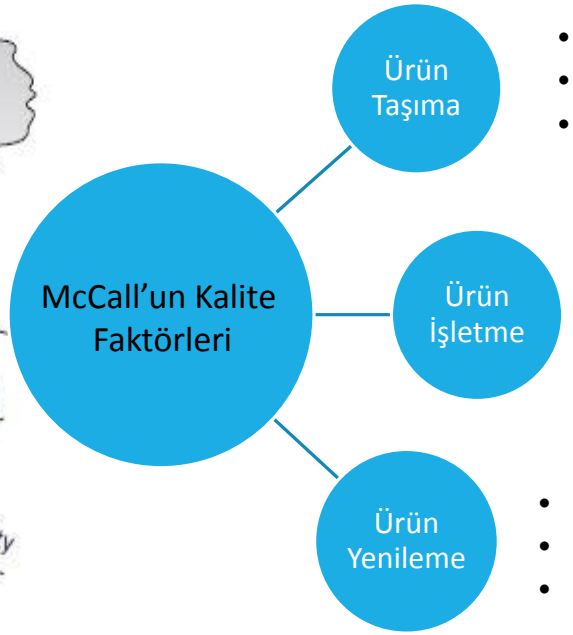
- **Bakım kolaylığı:** Başka bir yazılım geliştirici kişi ya da grup tarafından yazılımın bakımının yapılabilmesi için kaynak kodun anlaşılabilir bir şekilde yazılmış olması, iyi belgelendirilmesi, sorun çözümlemesinin ve testinin kolay olması gereklidir.
- **Doğrulanabilirlik:** Yazılımın doğru çalıştığının test edebilme kolaylığıdır. Testi mümkün olmayan yazılımın gerektiği zaman doğru çalışması da garanti edilemez.
- **Genişleyebilirlik:** Bir yazılım her zaman için kullanıcı isteklerine göre yeniden uyarlanabilir bir özelliğe sahip olmalıdır. Ancak yazlımlar büyüdükçe bunu sağlamak güçleşir. Sistemde gerekli değişiklikleri uygun bir şekilde ve kolayca yapabilmek, ona yeni işlevler ekleyebilmek için tasarım ve gerçekleştirimin uygun şekilde yapılması gereklidir.

2.4. Garvin'in Kalite Ölçütleri

Performans kalitesi	Dayanıklılık
Özellik kalitesi	Kullanılabilirlik
Güvenilirlik	Estetiklik
Uygunluk	Algılama

2.4. McCall' un Kalite Faktörleri





- Taşınabilirlik
- Yeniden kullanılabilirlik
- Birlikte çalışabilirlik

- Doğruluk
- Güvenilirlik
- Verimlilik
- Bütünlük
- Kullanılabilirlik

- Bakım kolaylığı
- Esneklik
- Test edilebilirlik

2.4. ISO 9126 Kalite Faktörleri

ISO 9126 standardı, bilgisayar yazılımları için kalite özellikleri tanımlama amacıyla geliştirilmiştir. Bu standart, 6 adet kalite özelliği tanımlamaktadır. Bunlar:

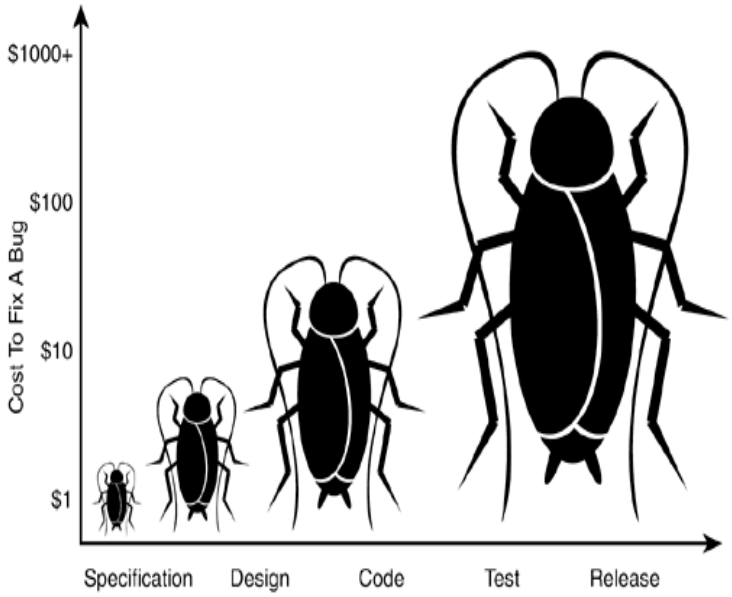
3. Kalitenin Maliyeti

- Kalite ve maliyet ilişkisinde şu sözleri sıkça duyarız. ‘Kalitenin önemli olduğunu biliyoruz. Fakat zaman ve para bakımından oldukça maliyetlidir. Gerçekten kaliteli bir yazılım istiyorsak çok fazla para ve zaman harcamamız gerekir’.

- Bu argüman geçerli bir bahane olabilir. Kalitenin maliyetli olduğu konusunda aksi bir düşünce yoktur. Fakat kalite eksikliği de maliyetlidir. Kullanıcılar ve üreticiler bu konuda hemfikirdir.

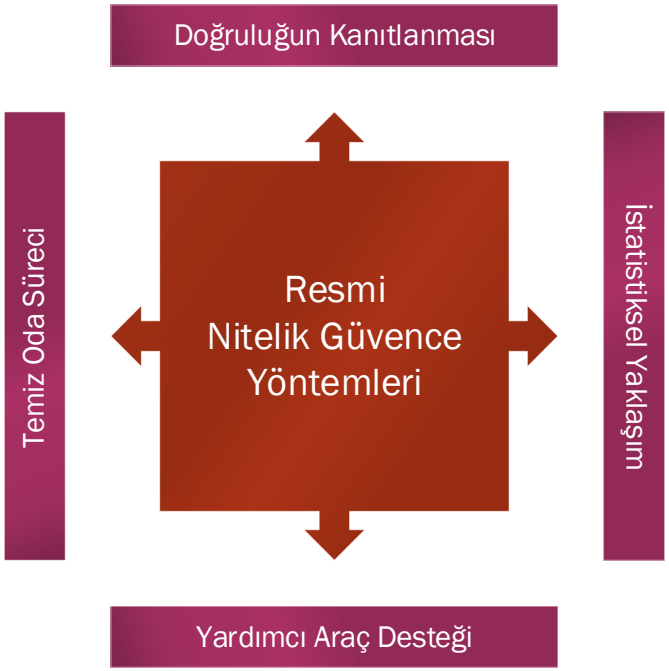
3. Kalitenin Maliyeti

- Asıl soru şudur: Hangi maliyet bizi daha fazla endişelendirmelidir?
- Bunu cevaplayabilmek için kaliteye ulaşmanın maliyeti ve düşük kaliteli yazılımın maliyetini anlayabilmemiz gerekir. Hataların ortaya çıktığı aşama ile bize getirdiği maliyetin arasında doğru bir orantı vardır. Bu yüzden kaliteli yazılım üretmenin maliyeti, düşük kaliteli yazılım üretmenin maliyetinden daha düşüktür.



Hata Giderme Maliyet Oranları

4. Resmi Kalite Güvence Yöntemleri



- Nitelik güvence her yazılım geliştirici bireyin sorumlu olduğu bir görev alanıdır. Bu görevin belirli kurallara göre yürütülebilmesi için çeşitli resmi yöntemler ve yaklaşımlar ortaya atılmıştır. Bunların en önemlileri yandaki şekilde verilmiştir.

4.1. Doğruluğun Kanıtlanması

- Bilgisayar yazılımlarının çoğu algoritmalardan oluşur. Bir yazılımın doğruluğunu kanıtlamanın yöntemlerinden biri, kullanılan programlama dili ile yazılan kodun algoritmalara uygun olduğunu göstermektir.
- Yazılımın doğruluğunun matematiksel olarak kanıtlanması oldukça pahalı bir yöntemdir. Ancak, otomatik kod üreten yazılım araçlarının geliştirilmesinde bu yaklaşımın da kullanımında yarar vardır.

4.2. İstatistiksel Yaklaşım

- Yazılım kusurları toplanır ve sınıflandırılır.
- Her kusur izlenerek neden kaynaklandığı bulunur ve bu bilgiler de kaydedilir.
- Neden olan ortak kusurlar bulunur.
- Bulunan kusurlar düzeltilir.

- Kusurlara neden olan noktaların belirlenmesinden sonra her bir kusura ilişkin istatistiksel veriler toplanır ve bir sıraya konur. Buna göre, yeni bir hata ortaya çıktığında çözümün nerede aranması gerektiği konusunda bir fikir oluşur.

4.3. Temiz Oda Süreci

- Nitelikli yazılım geliştirmenin bir tekniği de temiz oda sürecidir. Endüstride çeşitli alanlarda (bkz. clean room engineering) kullanılan bu teknikte ilke olarak geliştirme ortamının her türlü zararlı etmenden arındırıldığı kabul edilir.

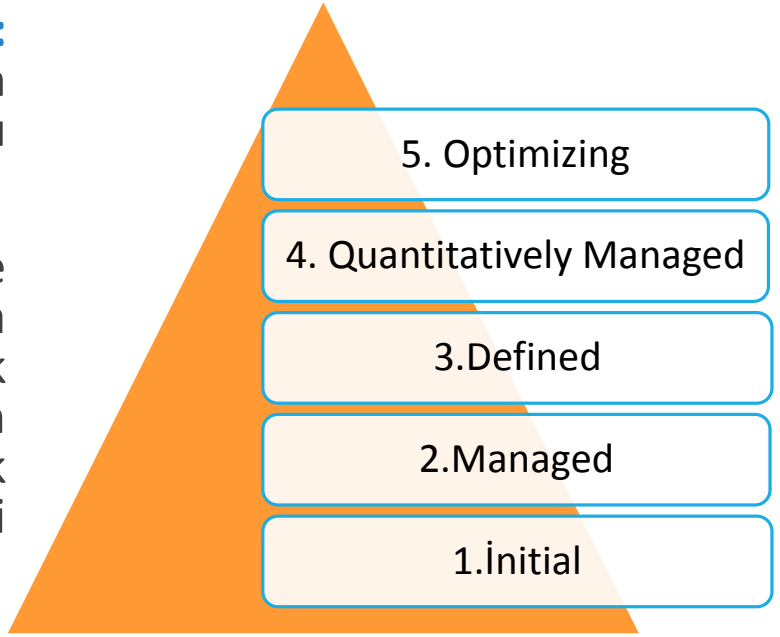
Amaç hatanın oluşabileceği ortamı ortadan kaldırmaktır. Fakat bunun için matematik modellerinin çok iyi geliştirilmiş ve oturtulmuş olması gerekir. Bu nedenle temiz oda süreci daha çok, küçük ölçekli yazılım projelerinde kullanılmaktadır.

4.4. Yardımcı Araç Desteği

- Günümüzde pek çok yardımcı araç nitelikli kod üretimi ve üretilen kodun sınanması amacıyla kullanılmaktadır. Sıkça kullanılan yazılım modüllerinin ortak hale dönüştürülmesi ve kalıpsal bir yapıya sokulması hem hataları azaltır hem de geliştirme süresini düşürür.
- Bunun yanında, verilen parametrelere göre aynı tür kod üreten araçlar da bulunmaktadır. Genel olarak nitelik güvence ekibi tarafından benimsenmiş araçlarla yapılan yazılım geliştirme sonucu elde edilen ürün yüksek nitelikli kabul edilir.

5. Nitelik Sistem Standartları

- **Yetenek Olgunluk Modeli- (CMM) :** Bir süreç modeli olup, örgütlerin yazılım süreçlerinin olgunluğunu değerlendirme modelidir.
- Birçok büyük yazılım ihalesinde CMM Seviyesinin 3 yada yukarısına sahip olunması ön koşul olarak karşımıza çıkmaktadır. Yazılım şirketleri de iyi elemanları çekmek için çoğu zaman CMM seviyelerini belirterek ilan vermektedirler.



5.Nitelik Sistem Standartları

CMM

1. Başlangıç: Başarı sadece bireylere bağlıdır. Proje denetimi için resmi yordamlar bulunsa da bunların düzenli olarak kullanılmasını sağlayacak bir örgütsel düzenek yoktur. Örgüt başarılı bir yazılım geliştirebilir, ancak yazılım öznitelikleri ve yazılım süreci kestirilemez. Herhangi bir kriz anında işi bırakmak olasıdır.

2. Yönetilen: Yazılı olmayan ve kısmen tutarlı süreçler vardır. Nitelik güvencesi ve düzenleşim denetim yordamları oluşturulmuştur. Projenin başarısı yöneticilerin bireysel özendirme çalışmalarının sonucudur.

3. Tanımlı: Firma kültürü yazılı hale gelmiştir. Bir örgüt kendine ait süreçlerle tanımlanır. Resmi yordamları vardır ve tanımlanmış süreçler bütün projelerde izlenmektedir.

4. Nicel Olarak Yönetilebilen: Tanımlı süreçler ölçülmekte, başarım göstergeleri değerlendirilmektedir. Niceliksel veri toplulukları tanımlanmış, süreç ve ürün ölçütleri toplanmış ve süreç iyileştirme çalışmasına geçilmiştir. Başarımı kestirmek mümkündür.

5. En iyilenen: Kurumsallaşma gerçekleşmiştir. Örgüt süreç iyileştirmeyi sürekli bir hedef haline getirmiştir. Süreç iyileştirme için öz kaynak ayrılmaktadır. Geri beslemelerin sistematik bir şekilde değerlendirilmesine başlanmıştır.

5.Nitelik Sistem Standartları

- **Trillium:** Teknolojik olgunluk içeren ve iyileşmeyi bu olgunlukla düzenleyen yol haritası yaklaşımı getiren Trillium modeli, Kanada iletişim sektörü tarafından geliştirilmiştir. Yol haritası kavramı, ürün geliştirme süreci içerisinde bir örgüt alanına, bir gereksinime ya da bir ögeye odaklanan uygulamalar seti olarak tanımlanır.

Trillium Yetenek Alanları			
Örgüt Süreç Niteliği	İnsan Kaynakları Yönetimi	Süreç	Yönetim
Nitelik	Sistem Geliştirme Uygulamaları	Geliştirme Ortamı	Müşteri Desteği

5. Nitelik Sistem Standartları

- **Spice:** İki boyutlu bir model olup içe dönük süreç iyileştirme ile dışa dönük yetenek belirleme amacını 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.

- Her boyutta firmaya hitap eder.

- **TickIT:** Değerlendirme en az iki aşamalı bir süreçtir. İlkinde örgütün nitelik sistemi, standarda göre değerlendirilir. İkincisinde, örgütün pratikte gerçekten kendi nitelik sistemine ve standarda uyumlu çalışıp çalışmadığı denetlenir, nitelik sisteminin etkinlik derecesine bakılır.

Nitelik Sistem Standartları

➢**ISO 9000:** ISO tarafından yayınlanan nitelik sistemi ve güvencesi standartları büyük ilgi görmüştür. Beş temel ISO standardı vardır:

ISO-9000	**Nitelik yönetimi ve nitelik güvence standartları seçim ve kullanım rehberi**
ISO-9001	Nitelik Sistemleri- Tasarım, geliştirme, üretim, tesis ve servis için nitelik güvence
ISO-9002	Üretim ve tesis için nitelik güvence
ISO-9003	Nitelik Sistemleri- Son muayene ve testlerde nitelik güvence
ISO-9004	Nitelik yönetimi ve nitelik sistemleri öğeleri

5. Nitelik Sistem Standartları

REPUBLIC OF TURKEY
MINISTRY OF NATIONAL DEFENCE
INDUSTRIAL QUALITY ASSURANCE LEVEL
CERTIFICATE

ANKARA
JUNE 2015

CERTIFICATE NO : 2015/13

COMPANY NAME : ALTAY KOLLEKTİF ŞİRKETİ M.MURAD DURAL VE ORTAĞI Kızılırmak Cad.Nu.:2 Bakanlıklar 06640 Ankara

The auditing carried out to confirm that your company's quality assurance system complies with AQAP-160 requirements, has been completed.

It is certified that your company's quality assurance system explained in Quality Manual KEK/.2 dated 09 February 2015 complies with AQAP-160 requirements for production mentioned below.

"Information Technologies Services, Software Design and Development."

This certificate is valid until 26 May 2018 by 17.00 o'clock.

Ministry of National Defence of the Republic of Turkey's auditors will continuously monitor and audit effectiveness and application of the quality assurance system.

This certificate will be subject to cancel if changes and renewals to be made in the system defined in quality manual presented to the Ministry of National Defence are not submitted to and approved by Ministry of National Defence previously or in the case that results of periodical reviews are negative.

Şaban UMUT
Brigadier General, TUAF
Deputy Undersecretary

- **AQAP-150/160:** Askeri amaçlı olarak geliştirilen ürünlerin ortak özellikler arasında ileri teknoloji kullanımı, sürekli artan karmaşıklık düzeyleri, yüksek maliyet gerektirmeleri sayılabilir. Bu ortak özellikler neticesinde Nato'ya üye ülkeler arasında asgari amaçlı malzeme, parça araç ve gerecin temininde bir seri standart prosedür olarak tanımlanmıştır.

- SEI CMMI
- ISO 9001:2008
- NATO AQAP-160

5. Nitelik Sistem Standartları

ISO 9001	CMM	SPICE
2 düzeyli	5 düzeyli	2 boyutlu, çok düzeyli
Belgeleme	Yetenek belirleme ve belgeleme	İyileştirme ve yetenek belirleme
Kısa, soyut, genel amaçlı belgeler	Uzunca, daha somut, yazılıma özgü	Ayrıntılı, somut, yazılıma özgü
Denetleme, hata arayan, kanıt isteyen olumsuz bir tutum	Değerlendirme, somut ölçekler, uzun ve ayrıntılı	Değerlendirme, gerçek arayan olumlu bir tutum, işbirlikçi, saydam
Kapsamlı değerlendirme	Kapsam değerlendirme	Küçük çapta yada kapsamlı
Tüm örgüt tek not alır.	Tüm örgüt için tek tek değerlendirme yapılır.	Her süreç, her proje ve her grup için ayrı yapılabilir.
Rehber değil, çok sınırlı	Rehber olma niteliği var, iyileştirme yolu var	Rehber olma niteliği var, esnek, amaca göre iyileştirme yapılabilir.
Basit	Kolay	Zor, model bilgisi gerekli

Çalışma Soruları

1. «Kaliteli Yazılım» ne demektir? Tartışınız.

2. Nitelik sistem standartları kaç tanedir? Yazıp açıklayınız.

3. Kalitenin Yazılım Mühendisliği'ndeki önemini anlatmak için çeşitli örnekler veriniz.

4. Bir ürünün yeniden kullanılabilirliği ne demektir? Örnek vererek açıklayınız.

5. Kalitenin hangi aşamalarda daha maliyetli olduğunu, nasıl en düşük maliyetle en kaliteli ürün üreteceğimizi düşünüp tartışınız.

6. Temiz oda sürecinin gerçekleştirilmesiyle ilgili somut örnekler veriniz.

7. Kalite güvence ile kalite kontrol arasındaki farkı açıklayınız.

8. Resmi kalite güvence yöntemleri kaç tanedir? Açıklayınız.

9. Nitelik Sistem Standartlarının kıyaslamasını yapınız.

10. Üretim veya hizmette kaliteye ulaşma yöntemi nasıl sağlanır?

Kaynaklar

[1] Software Engineering A Practitioner's Approach (7th Edition), Roger Pressman, 2013

[2] Yazılım Mühendisliği(2. Baskı), M.Erhan Sarıdoğan,2008

[3] Yazılım Test Mühendisliği (1. Baskı), Rifat Çölkesen, 2010

[4] https://s3-eu-west-1.amazonaws.com/kurapov/file/166.pdf

[5] http://www.softwaretestinggenius.com/download/bgstpadmini.pdf

[6] http://www.dtajans.com/kobi-blog/kalite-nedir-kalitenin-boyutlari-nelerdir