

YAZILIM TEST MÜHENDİSLİĞİ

Dr. Öğr. Üyesi Hasan YETİŞ

- Yazılım testi, yazılımın daha önce tanımlanmış **teknik ve işlevsel gereksinimleri karşılayıp karşılamadığının** ve **yazılımın beklendiği gibi çalışıp çalışmadığının** kontrolüdür.
- Yazılım test süreci de temel olarak **elde edilen ürünün beklenen kalitede olduğunu belirlemek**, değilse istenilen kaliteye ulaştırılmasını sağlamayı amaçlayan bir süreçtir

- Yazılım, kullanıcı istekleri doğrultusunda, bir bilgisayar sistemini komutlarla tasarlanan uygulamalar yardımıyla yönetme işlemidir.
- Üretilen yazılımda amaç, **optimum maliyet ve süre ile yüksek kaliteli bir ürün oluşturabilmektir.**
- Yazılımın kalitesi, **sistemin beklentileri karşılama düzeyinin yüksekliği ve minimum zaman ve maliyet ile ölçülür.** Bunun yanında sistemin kalite ölçütlerinden bir diğeri ise **minimum hataya** sahip olmasıdır.

HATA

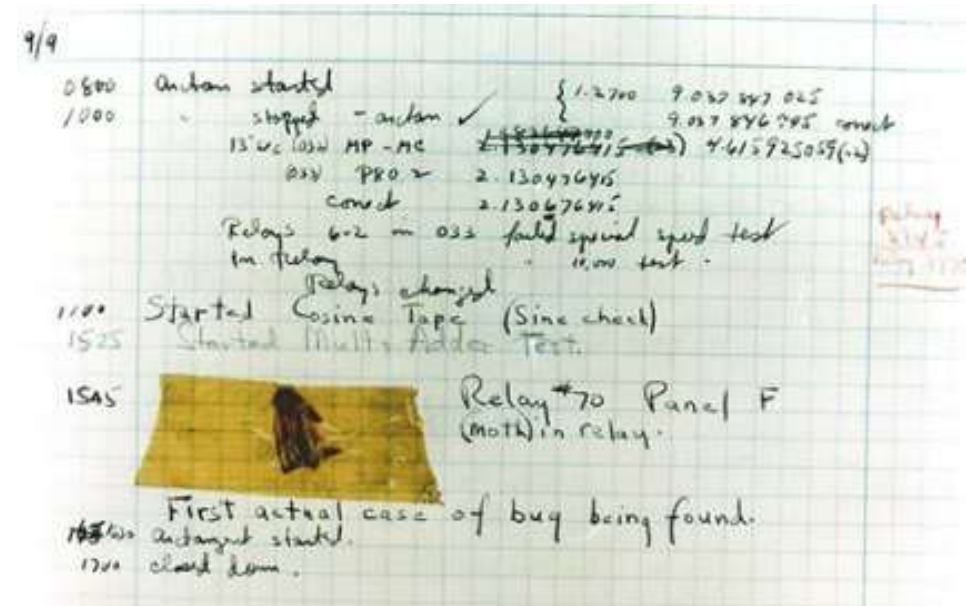
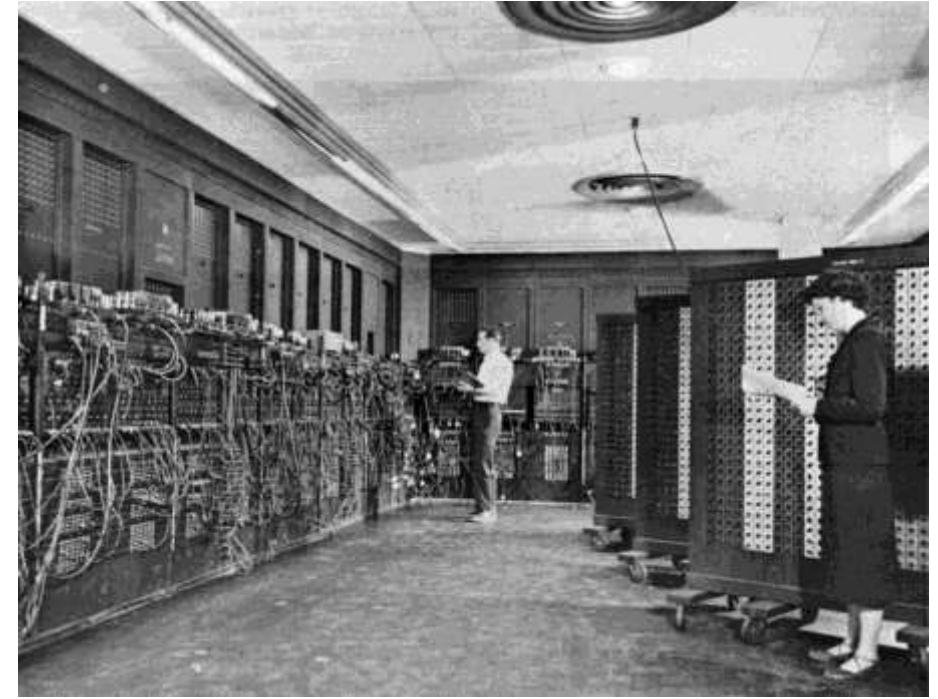
KALİTE

TEST

- Mükemmel yazılım yoktur ve bir yazılım asla %100 test edilmez.
- Hatasız yazılımın mümkün olmadığı göz önüne alınırsa, geliştirilen yazılımın **minimum hataya sahip olması** yazılımın kalite oranını gösterir.
- Hatalı yazılımlar sonucu meydana gelen maddi ve manevi zararların tarihte pek çok örneği bulunmaktadır.

Bug Vakası

- 9 Eylül 1947 tarihinde tarihe geçen ilk bilgisayar bugu, gerçek bir böceğin bir bilgisayarın içine girmesiyle ortaya çıktı.
- O tarihlerde odaları kaplayan bilgisayarlar böcekler için yaşam alanı oldukları için bazı böcekler devrelere dokunup sistemin kısa devre yapmasına ve programların hatalı işlemelerine sebep olmuşlar.
- Her ne kadar o tarihe kadar bilgisayarlardaki bug terimi daha önceden kullanılmış olsa da Grace Hopper'ın bunu loglaması terimin yaygınlaşmasına sebep oldu.



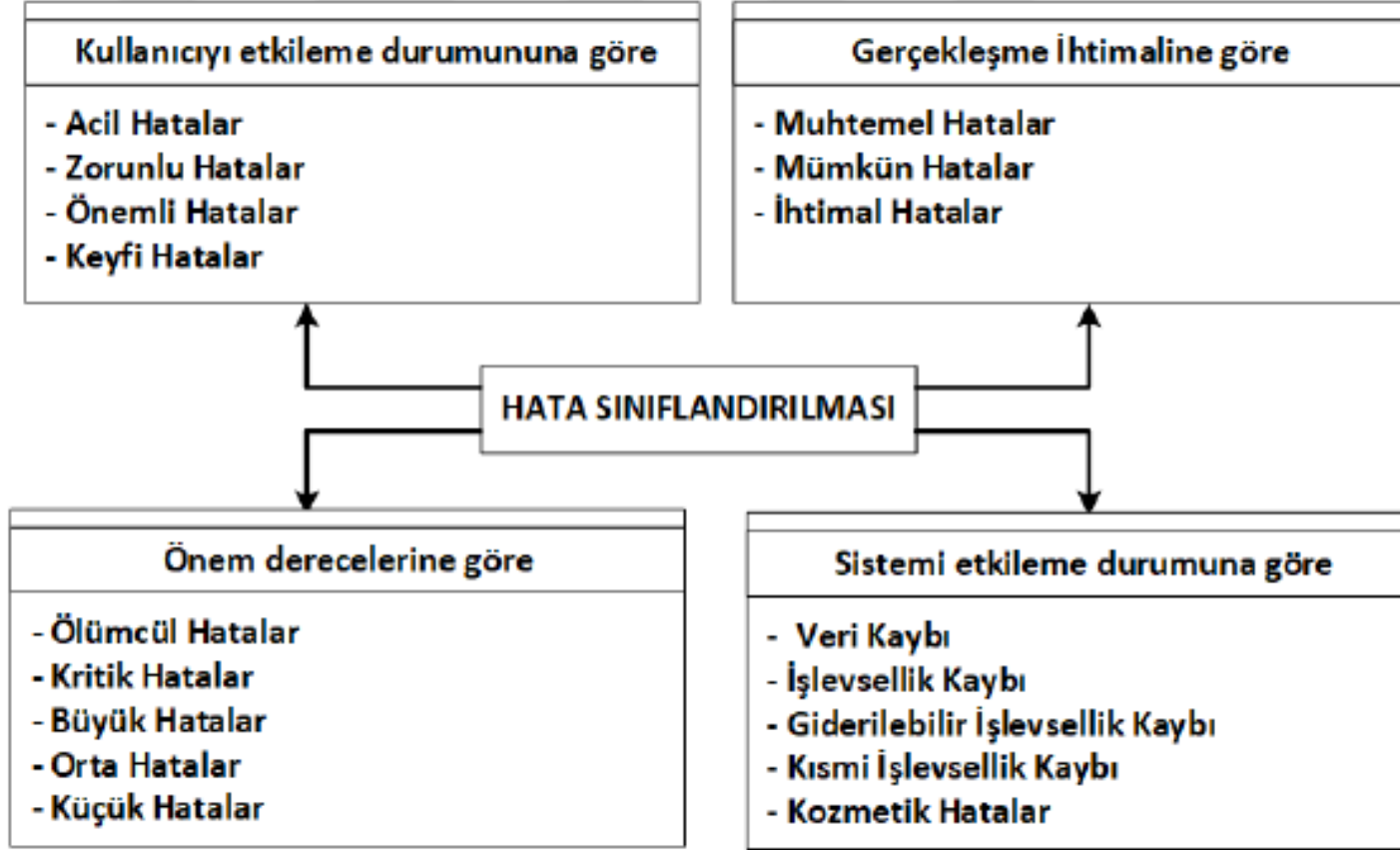
- *Patriot Füze Hatası:* 1991 yılında yaşanan Körfez Savaşı'nda, Amerika tarafından İsrail'e yerleştirilen Patriot füzelerinden biri, Irak'tan gelen Scud füzesini imha edemedi ve Amerikan askerlerine sahip bir barakaya isabet etti. Bu olay, 29 Amerikan askerinin ölümüne neden oldu. Daha sonra gerçekleştirilen incelemelerde, Patriot füzelerinde zaman hesaplaması için kullanılan 24 bitlik değişkende oluşan bir zaman hatası fark edilmiştir. Bu hata nedeniyle füze 500 metrelik bir sapma oluşmuştur [15].

- *Pentium İşlemci Sorunu:* 1993 yılında Intel Pentium işlemci sorunu yaşanmıştır. Bilgisayar yongasında bulunan bir hata nedeniyle belirli aralıkta yer alan sayıların bölünmesi hatalı sonuçlar vermiştir. İşlem sonucu 0.00006'lık bir hata meydana gelmiş ve durum çok sayıda kullanıcıyı etkilemiştir. Piyasada bulunan çok sayıda yonga nedeniyle Intel şikayette bulunan tüm kullanıcıların yongasını değiştirme kararı almış ve bu durum Intel firmasına tam olarak 475 milyon dolarlık bir zarara sebep olmuştur [16].

- *Arienne-5 Füze Facias:* NASA, 4 Haziran 1996'da fırlatılması planlanan Arienne 5 uzay aracını kodlarken, Arienne 4 roketinin kodlarını kopyalayarak bir hata yaptığının farkında değildi. O gün fırlatma için geri sayım yapıldı ve roketin motorları ateşlenerek kalkış başladı. Hızlanarak yoluna 37 saniye boyunca devam eden Arienne 5 roketi o saniyeden sonra yanlış yöne doğru 90 derece dönmeye başladı. Bu durum, roketin ordu. 1arındaki en panan yazının natarından biri arasına giren bu kazanın sebebi, yazılımda oluşan bir bugdı. Bu bug, milyarlarca potansiyel değeri temsil edebilen 64 bit değişkeni, yaklaşık 65 bin değeri alabilen 16 bit değişkene sığdırmaya çalışmasına sebep oldu. Roketler hızlandıkça kodlarda ortaya çıkan bu bug işleri sarpa sardırı ve roket patladı [17].

- *Honda Yazılım Hatası:* 2011 yılında Honda'ya ait araçlarda hatalı yazılım sonucu bazı özellikler yanlış zamanda aktif edilmiştir. Yazılımda bulunan hataların, aracın kontrol modülü ile diğer parçalar arasındaki çalışma zamanı uyumsuzluğundan oluşabileceği açıklanmıştır. Bu durumun geri görüş kamerasının çalışmasını engellemesi, sinyallerin yanlış yanması veya sileceklerin yanlış zamanlarda çalışması gibi sorunlara neden olabileceği belirtilmiş, bunun üzerine Honda firması, 2.5 milyon aracını geri çağırarak zorunda kalmıştır [18].

- *Boeing Yazılım Hatası:* Dünyanın en büyük sivil uçak üreticilerinden biri olan ve önceki yıllarda ölümcül kazalarla adını duyuran Boeing firması, 2020 yılı içerisinde Boeing 737 Max tipi yolcu uçaklarında iki farklı yazılımsal hatanın keşfedildiğini duyurdu. 2018 yılının Ekim ayında Endonezya'da, 2019 yılının Mart ayında ise Etiyopya'da yüzlerce kişinin hayatını kaybetmesine sebep olan 737 Max tipi uçakların uçuşu, Etiyopya'daki kazanın ardından birçok ülkede geçici olarak durdurulmuştu. Boeing firması, bu uçakların, uçuşlara yeniden başlaması için gerekli çalışmaları hızla sürdürürken yaptığı açıklamalarda, 737 Max tipi uçaklarda iki farklı yazılımsal hatanın daha fark edildiği yönündedir. Bu hatalardan birinin, bilgisayarın uçuşunu kontrol eden mikroişlemciadaki varsayımsal hatalarla ilişkili olduğu açıklandı. Hata sonucu, uçağın pilotun kontrolünü kaybetmesine neden olabileceği belirtildi. Bunun yanında başka bir hatanın otomatik pilot işlevinin devre dışı bırakılmasına neden olabileceği belirtildi.



Şekil 2.2. Yazılım hatalarının sınıflandırılması

Sistemi Etkileme Durumuna Göre Hatalar

- *Veri kaybı:* Hatanın, sistemin veri tabanına erişebildiği ve veri tabanında bulunan verileri sildiği hata çeşididir.
- *İşlevsellik kaybı:* İşlevsellik kaybı, sistemde bulunan hatanın sistemin işlevselliğini bloke ettiği durumları ifade eder.
- *Giderilebilir işlevsellik kaybı:* Hatanın, sistemde bulunan herhangi bir işlevi bloke ettiği durumları kapsar. Bu tür hatalarda meydana gelen bloke durumu, başka bir işlevsel metot kullanılarak giderilebilir ve bu hata türleri sistemin çalışmasını aksatmaz.
- *Kısmi işlevsellik kaybı:* Sistemin, önemli olmayan işlevlerinin bloke edildiği hata türüdür.
- *Kozmetik hatalar:* Kullanıcı isteklerine göre tasarlanan arayüz ve grafiklerin, kullanıcı kriterlerine uygun olmayan görüntü veya uyarı mesajları içermesi durumudur. Bu tür hatalar, sistemin işlevselliğini etkilemez.

Kullanıcıyı Etkileme Durumuna Göre Hatalar

- *Acil hatalar:* Sistemin çalışmasını durduran veya çalışmayı yanlış sonuçlandıran, performansı büyük ölçüde etkileyen ve düzeltilmesi aciliyet gerektiren hatalardır. Bu tür hatalar, büyük oranda maddi ve manevi zarara yol açabilir.
- *Zorunlu hatalar:* Bu tür hatalar, test aşamasında fark edilir. Bu kategorideki hataların çözülmeden sistemin kullanıcıya teslim edilmemesi gerekir.
- *Önemli hatalar:* Sistemin, yazılım yaşam döngüsünün planlama aşamasında belirlenen kriterlere uymaması sonucu kullanıcı tepkilerine neden olabilecek hatalar ve kusurlardır.
- *Keyfi hatalar:* Yeterli zaman kalması durumunda düzeltilmesi sisteme olumlu etki eden hatalardır.

Önem Derecelerine Göre Hatalar

- *Ölümcül hatalar:* Yazılımın işlevselliğinde meydana getirdiği kesinti nedeniyle testlerin devamını engelleyen hatalardır. Bu hatalar, çözülmediği takdirde büyük ölçüde maddi ve manevi zarara neden olabilir.
- *Kritik hatalar:* Bu tür hataların varlığında testler aksamaya uğramazlar. Ancak bu hatalar, ileri aşamalarda sistemi kesintiye uğratabilir. Bu hatalar çözülmeden sistemin, kullanıcıya teslim edilmemesi önerilir.
- *Büyük hatalar:* Bu tür hatalar ürünün kullanıcıya sunulmasına engel teşkil etmez. Ancak sistemin ileri aşamalarında bu hatadan kaynaklanan zararın telafisi ve buna bağlı olarak maliyeti zor olabilir.
- *Orta hatalar:* Bu hatalar, test işleminin devamını etkilemez. Sistem bu şekilde kullanıcıya sunulabilir ve bulunan hatalar sonucu meydana gelen zararların telafisi mümkündür.
- *Küçük hatalar:* Sistemin çalışmasını, test sürecini önemli derecede etkilemeyecek olan bu hataların varlığında, sistem kullanıcıya bu şekilde sunulabilir.

Gerçekleşme İhtimaline Göre Hatalar

- *Muhtemel hatalar:* Sistem işleyişini aksatacağına, kullanıcıya sunulan çıktı ile beklenen çıktının farklı olacağına veya sistemi bloke edeceğine kesin gözüyle bakılan hatalardır. Bu hataların düzeltilmemesi durumunda sistem işleyişini sağlıklı bir şekilde yerine getiremez. Böylelikle sistem bu hatalar çözümlenmeden kullanıcıya teslim edilemez.
- *Mümkün hatalar:* Bu tür hataların, sistem işleyişini etkilemeleri olasıdır. İşleyişte aksaklığa neden olabilecekleri gibi, herhangi bir kesintiye neden olmayabilirler. Kaliteli yazılım, hata oranının en az olduğu yazılımdır. Bundan dolayı, bu tür hataların çözümlenmesi yazılım kalitesini olumlu etkiler.
- *İhtimal hatalar:* Sistemi etkileme ihtimali düşük olan hatalardır. Yazılımın işlevselliğini etkilemezler. Ancak her hata gibi bu hatalarında çözümlenmesi yazılımın kalitesini artırır.

Ref.	Hatanın adı	Tarih	Hatanın nedeni	Hatanın türü	Sonuç
[20, 21]	Mariner-1 Uzay Roketi	22 Temmuz 1962	Kağıda doğru yazılan formülün bilgisayara yanlış yazılması	SE-İşlevsellik Kaybı, ME-Acil Hatalar, ÖD- Ölümcül Hatalar	Yörüngeden ayrılan roket yok edilmiş, NASA'ya 80 milyon dolar kayıp olmuştur.
[22, 23, 24]	Hartford Colesium Binası	1978	Çoklu değişkene sahip matematiksel formülün hesaplanmasındaki işlem hatası	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Ölümcül Hatalar	CAD yazılımı ile tasarlanan Hartford Colesium binası çökmüş, 90 milyon dolar olan bir zarara neden olmuştur.
[21, 25, 26]	Sovyet Gaz Hattı	1982	Programdaki güvenlik açığı	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Ölümcül Hatalar	CIA ajanlarının geliştirdiği böcek, tarihin en büyük nükleer olmayan patlaması gerçekleşmiştir.
[20, 21]	Therac-25 Tedavi Cihazı	1985-1987	Hastalarda kullanılan radyasyon ışınlarını dengeleyen plakaların yazılım kaynaklı devreye girmemesi	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Ölümcül Hatalar	6 hastadan 3'ü aşırı radyasyona maruz kalması nedeniyle hayatını kaybetmiştir.

[21] [27] [28]	Berkeley Unix Sisteminde Tampon Bellek Taşması	1988	Tampon bellek yetersizliği	SE: İşlevsellik Kaybı, ME: Zorunlu Hatalar, ÖD: Kritik Hatalar	Bir grup siyah şapkalı bilgisayar korsanı tarafından 2000-6000 arası bilgisayara sızılmıştır.
[15] [29]	Patriot Füze Hatası	25 Şubat 1991	Zaman hesaplamasında kullanılan 24 bitlik sabit noktalı değişkende bulunan hata	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Ölümcül Hatalar	Hata nedeniyle 29 Amerikan askeri hayatını kaybetmiştir.
[16] [20] [30]	Pentium İşlemci Sorunu	1993	Tam sayı aralığında bir değere sahip olması gereken matematiksel formülün yanlış yazılması	SE: İşlevsellik Kaybı, ME: Zorunlu Hatalar, ÖD: Kritik Hatalar	Firma, 475 milyon dolar zarara uğramıştır.
[21] [28]	Kerberos Rastgele Sayı Üretme Hatası	1988-1996	Programında kullanılan rastgele sayı üreticisinin uygun bir parametre ile beslenmemesi	SE: İşlevsellik Kaybı, ME: Zorunlu Hatalar, ÖE: Kritik Hatalar	Kerberos firmasına ait yetkili bilgisayarlardan bir kaçına izinsiz giriş sağlanmıştır.

[21] [30]	Bilgisayar Ağı Kontrol Komutları Hatası	1995-1996	IP numaralarının yanlış yazılması ve doğru yazılmayan bir "Ping" komutunun internet üzerinde herhangi bir yerden gönderilmesi	SE: İşlevsellik Kaybı, ME: Zorunlu Hatalar, ÖD: Kritik Hatalar	Bir çok bilgisayar mavi ekran hatası vermiştir. En çok etkilenen işletim sistemi Windows'tur.
[17] [21] [31]	Arienne-5 Füze Faciası	4 Haziran 1996	Aritmetiksel işlemlerde 64 bit kullanılması gerekirken, önceki sürümden kalan 16 bitlik kullanımının devam etmesi	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Ölümcül Hatalar	Fırlatıldıktan 40 saniye sonra havada infilak eden füze, 500 milyon dolarlık bir zarara neden olarak en pahalı yazılım hataları arasında yer almıştır.
[29] [30] [31] [32]	Mars Climate Orbiter Hatası	23 Eylül 1999	Uzunluk ölçümlerinde kullanılan İngiliz ölçü birimleri ile metrik değerleri arasındaki uyumsuzluk	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Kritik Hatalar	Hatanın NASA'ya maliyeti 125 milyon dolardır.

[21] [28]	National Cancer Institue	2000	Yazılım kaynaklı hata nedeniyle hastalara yanlış dozda radyasyon verilmesi	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Ölümcül Hatalar	8 hasta hayatını kaybetmiştir ve bilinen 20'den fazla kişi ciddi sağlık sorunları yaşamıştır.
[20] [28] [29] [33]	Mercy Hastanesi Ölüm İlanları	2002	Hastane sistemlerindeki veri tabanı kayıtlarının düzensizliği	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Kritik Hatalar	Veri tabanında bulunan hasta kayıtlarındaki 8500 kişi ölü ilan edilmiştir.
[20] [29]	Michigan Erken Tahliye Hatası	Ekim 2005	Veri tabanında bulunan verilerin hatalı sorgulanması	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Kritik Hatalar	Michigan Düzeltme Bakanlığı'na ait bilgisayar programlarındaki aksaklık nedeniyle 239 mahkum, 39 ile 161 gün öncesinden tahliye edilmiştir.
[18] [20] [28]	Honda Yazılım Hatası	2011	Hatalı algoritma sonucu araç özelliklerinin yanlış zamanlarda aktif edilmesi	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Kritik Hatalar	Firma 2.5 milyon aracını geri çağırmak zorunda kalmıştır.

[30, 34, 35]	Apple Maps Yön Hatası	2012	Farklı veri tabanlarındaki bilgilerin entegrasyonunda oluşan karmaşıklık	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Büyük Hatalar	Apple ile Google Maps arasındaki harita(maps uygulaması) rekabet sonucu firmanın itibar kaybetmiştir.
[36, 37, 38]	Knight Capital 2012 Group Finans Hatası		Yanlış algoritma kullanılması	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Kritik Hatalar	Firma, 30 dakika gibi kısa bir sürede 440 milyon dolar maddi kayıp yaşamıştır.
[21, 28]	Yanlış Jüri Daveti	2012	Veri tabanında gerçekleşen hatalı sorgulama	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Büyük Hatalar	Kaliforniya'da gerçekleşen olay sonucu, 1200 kişiye jüri davetiyesi gönderilmiş, tüm eyaletler arası yollarda trafik nedeniyle saatlerce aksaklıklar yaşanmıştır.
[39]	iCloud Güvenlik Açığı	Eylül 2014	Yazılım sisteminde var olan güvenlik açığı	SE: Veri Kaybı, ME: Zorunlu Hatalar, ÖD: Büyük Hatalar	Güvenlik açığı nedeniyle ünlüler dahil olmak pek çok insanın kişisel bilgileri ele geçirilmiş, bu olay üzerine Apple CEO'su Tim COOK, kullanıcı güvenini tazelemek için iCloud güvenlik özelliklerini güçlendirme sözü vermiştir.

[28] [40] [41]	Toyota Prius Aracı	2015	Motor kontrol ünitesindeki bir yazılımın aşırı ısınma ve güç kaybına neden olması	SE: İşlevsellik Kaybı, ME: Acil Hatalar, ÖD: Kritik Hatalar	Firma, 2014 yılında 2 milyon, 2015 yılında ise 625 bin aracını geri çağırmış, bu hata nedeniyle aşırı ısınan araçlar trafik sorunlarına neden olmuştur.
[42]	Google+ Güvenlik Açığı	2018-2019	Yazılım sisteminde var olan bir güvenlik açığı	SE: Veri Kaybı, ME: Zorunlu Hatalar, ÖD: Büyük Hatalar	Yer alan güvenlik açığı nedeniyle yaklaşık 500.000 Google+ kullanıcısının kişisel bilgileri izin olarak ele geçirildi. 2018'de farkedilen hata düzeltildi.
[43]	Ethereum 300 Milyon Dolarlık Hata	2017	Yazılım sisteminde var olan güvenlik açığı	SE: Veri Kaybı, ME: Zorunlu Hatalar, ÖD: Büyük Hatalar	Yer alan güvenlik açığı, fonların bir süreliğine kilitlenmesine ve 300 Milyon \$'lık kripto paranın kaybına neden olmuştur.
[44]	Tesla Otomatik Frenleme Yazılım Hatası	23 Ekim 2021	Kendi kendine sürüş Beta yazılım sürümünde otomatik fren sisteminin zamansız aktifleşmesi durumunda arkadan çarpma ve araçtaki kişilerin yaralanmasına ihtimalleri	SE: Giderilebilir İşlevsellik Kaybı, ME: Zorunlu Hatalar, ÖD: Kritik Hatalar	Firma, bu hatadan etkilenen araçları geri çağırarak için Güvenlik Geri Çağırma Raporu yayınlamıştır. Aralarında Model S, Model Y, Model X ve Model 3'ünde bulunduğu 12.000'e yakın aracı geri çağırmıştır.

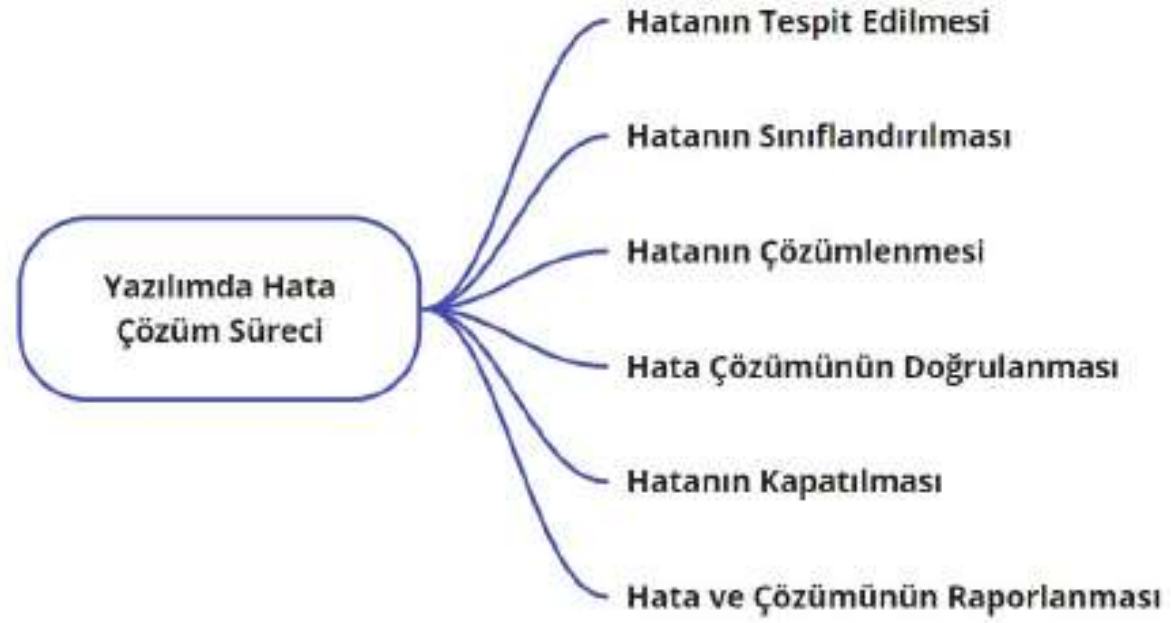
- IEEE Standart 610.12 (IEEE,1990) failure, fault ve error terimlerini şu şekilde tanımlamıştır:
- Error: Kodlayıcı kaynaklı, doğru olmayan bir sonuç elde edilmesi.
- Failure: Sistemin veya bir parçasının gerekli fonksiyonu yeterli performansta yerine getirememesi.
- Fault: Bir yazılım içerisindeki doğru olmayan adım, işlem veya veri tanımı.

- Yazılım testlerinde esas hedef, hataların neden olabileceği maddi veya manevi zararların boyutunu en aza indirmek, hatta mümkünse ortadan kaldırmaktır.
- Yazılım geliştirmenin erken aşamalarında fark edilen hataların düzeltilmesi zaman ve maliyet açısından daha avantajlıdır.
- Yazılım hatası, sistemin işlevini yerine getirmesine engel olan her türlü durum şeklinde tanımlanabilir

- Bir yazılımı %100 oranında test etmeye çalışmak hem maddi gerekçelerle hem de pratik olarak mümkün olmadığından, yazılımın beklendiği gibi çalıştığını gösterebilecek sınırlı sayıda ancak özenle seçilmiş bir test kümesinin belirlenmesi ve bu test kümesi üzerinden test yapılması daha mantıklıdır.
- Yazılım testi başka bir ifadeyle bir yazılımın doğrulanması ve geçerlenmesi süreci olarak da tanımlanabilir

Testin amaçları

- Müşteriye sunmadan önce ürün kalitesinden emin olmak,
- Yeniden çalışma ve geliştirme için masrafları azaltmak,
- Geliştirme işleminin erken aşamalarında hataları saptayarak ileri aşamalara yayılmasını önlemek, böylece zaman ve maliyetten tasarruf sağlamak,
- Müşteri memnuniyetini arttırmak ve izleyen siparişler için zemin hazırlamak.



Şekil 2.1. Yazılım hata çözüm süreci adımları

- *Hatanın Tespit Edilmesi:* Geliştirilmiş yazılım veya programlardaki hata tespitinde önemli olan nokta, yazılımların son kullanıcıya teslim edilmeden önce hatanın fark edilmesidir.
- Yazılım hatalarının yazılım geliştirme yaşam döngüsünün erken aşamalarında tespit edilmesi veya ortaya çıkarılması çok önemlidir.
- Hataların erken giderilmesi, sistem için belirlenen zaman ve maliyetin daha asgari seviyede olmasını sağlayacaktır.

- *Hatanın Sınıflandırılması:* Hatanın sınıflandırılması ile yazılım geliştiricilerin görevleri, öncelik sıralamasına göre belirlenir.
- Böylelikle birden fazla hatanın ortaya çıkması durumunda kritik boyuttaki hataların çözülmesi öncelik kazanır.

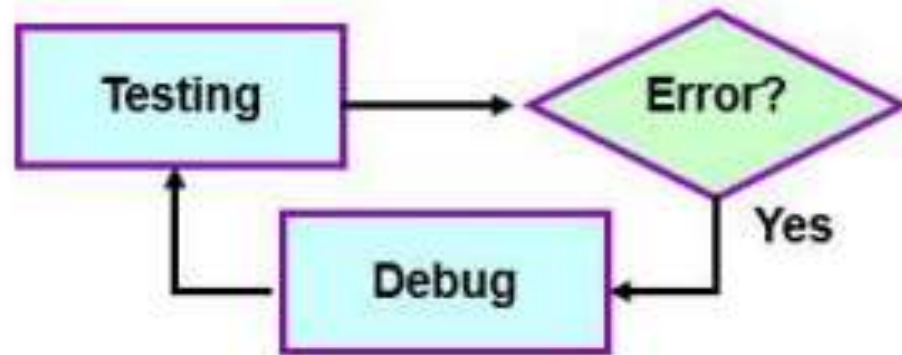
- *Hatanın Çözömlenmesi:* Hata çözümlerne adımları, kusurlar hakkında geliştiricilerin bilgilendirilmesi ile başlar.
- Daha sonra geliştiriciler, hata sınıflandırılması ile öncelikli olarak düzeltilecek hataları planlar ve çözümlü için gerekli çalışmaları başlatırlar.
- Son olarak test yöneticisine, geliştiriciler tarafından çözümlenen hataya ait bir çözüm raporu sunulur.

- *Hata Çözümünün Doğrulanması:* Geliştiricilerin, hatanın çözümünü test ekibine bildirmesini kapsayan aşamadır.
- Test ekibi tarafından test işlemi tekrar gerçekleştirilir, hatalı kısım incelenir ve çözümün yeni hatalara sebep olmadığı belirtilir.

- *Hatanın Kapatılması:* Hatanın kapatılması aşaması, hatanın çözümünden sonra gerçekleştirilir.
- Çözülen ve çözümü doğrulanan hatanın durumunun kapalı olması, hatanın çözümlendiğini belirtir.
- Hatanın aktiflik durumunun kapatılması, tekrar kontrol edilmesini önlemek için gereklidir

- *Hata ve Çözümünün Raporlanması:* Hatanın ve çözümün raporlanması aşamasında rapor içeriği, hata yönetim süreci adımları, hatanın durumu ve sınıflandırılması, hatanın çözülmesi ve doğrulanması hakkında detaylı bilgileri barındırır.
- Test yöneticileri tarafından hazırlanan hata raporu, yönetim ekibine gönderilir. Hata yönetim süreci ve çözümün doğrulanması için yönetim ekibinden geri bildirim alınır.
- Hatanın çözümü raporda, sonraki adımlarda hatanın yeniden ortaya çıkma ihtimaline karşı detaylıca belirtilmelidir.
- Ayrıca hata çözümünün, sistemin diğer modüllerini etkileme ihtimali karşısında da hatanın raporlandırılmasından faydalanılır.

- Hata ayıklama başarılı bir test sürecinin sonucudur. Şöyle ki, bir test sonucunda hatalar bulunduğu zaman hataların giderilme işlemine hata ayıklama(debugging) denir.



3 çeşit hata tespit stratejisi vardır

- Brute Force: Yürütme anındaki davranışlar izlenir, yazılım biriminin çeşitli noktalarına ekrana veya bir dosyaya o an akışın neresinde olduğunu, genel durumunu veya bir değişkenin değerini yazan deyimler eklenir.
- Backtracking: Kodun okunarak geri izlenmesi esasına dayanır. Hatanın olduğu yerden itibaren geriye doğru gidilerek kod incelenir; hata yaratan deyim yada kusurlu akış mantığı aranır.
- Cause elimination: Tümevarım veya tümdengelim yöntemlerine dayanarak elde edilen verilere göre hatanın nedeni araştırılır. Ortaya konan varsayımları doğrulayıcı ya da çürütücü testler tasarlanır. İlk testler olumlu sonuç verirse, daha ayrıntılı verilerle testlere devam edilerek hatanın tam yerinin saptanmasına çalışılır.

DOĞRULAMA

- Ürünü doğru mu üretiyoruz?
- Sistemin hatasız ve iyi bir mühendislik ürünü olup olmadığını ölçer.
- Geliştiriciler veya QA ekibi tarafından gerçekleştirilir.
- Doğrulama aşamasında bulunan hataların maliyeti daha azdır.

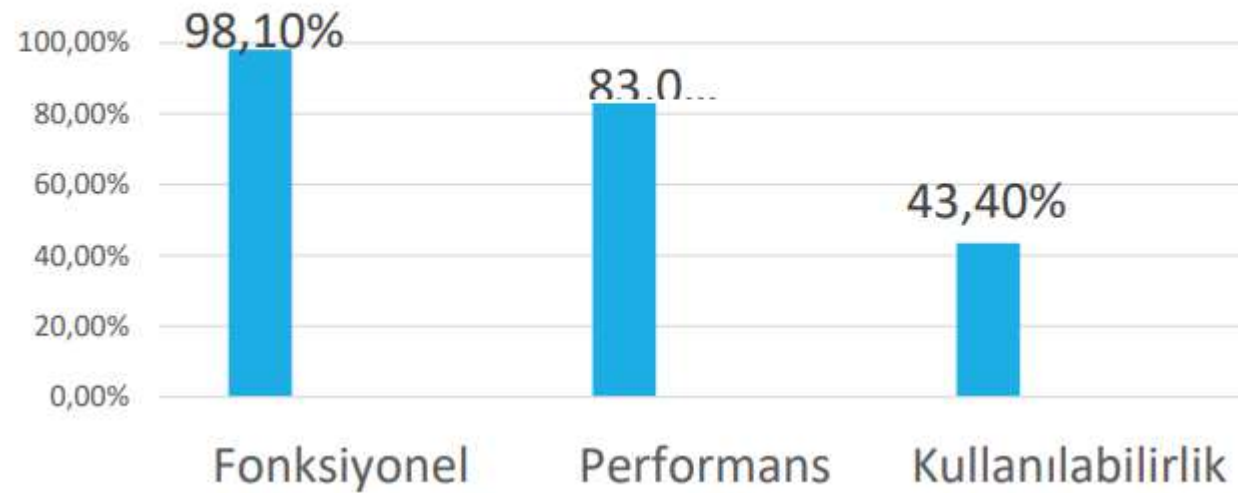
GEÇERLEME

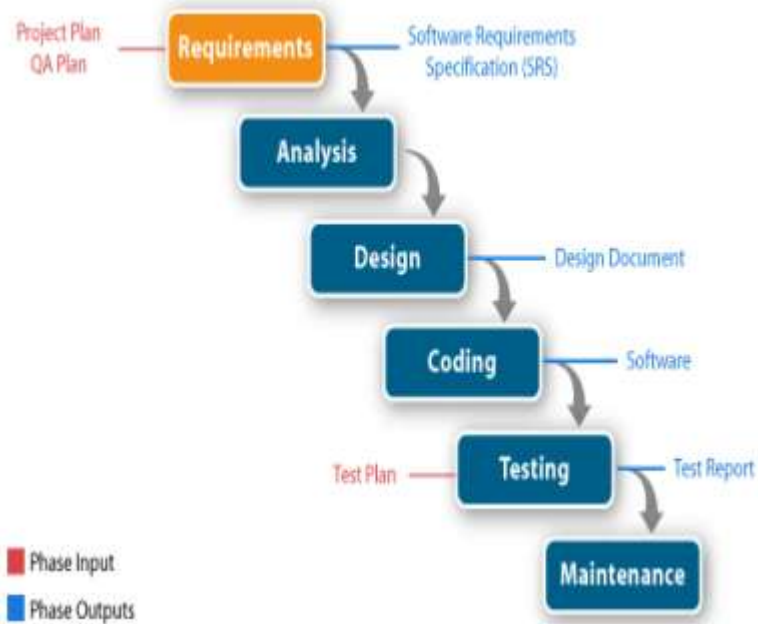
- Doğru ürünü mü üretiyoruz?
- Sistemin kullanıcı gereksinimlerine uygunluğu ölçer.
- Test ekibi tarafından gerçekleştirilir.
- Geçerleme aşamasında bulunan hataların maliyeti daha fazladır.

Turkish Testing Board

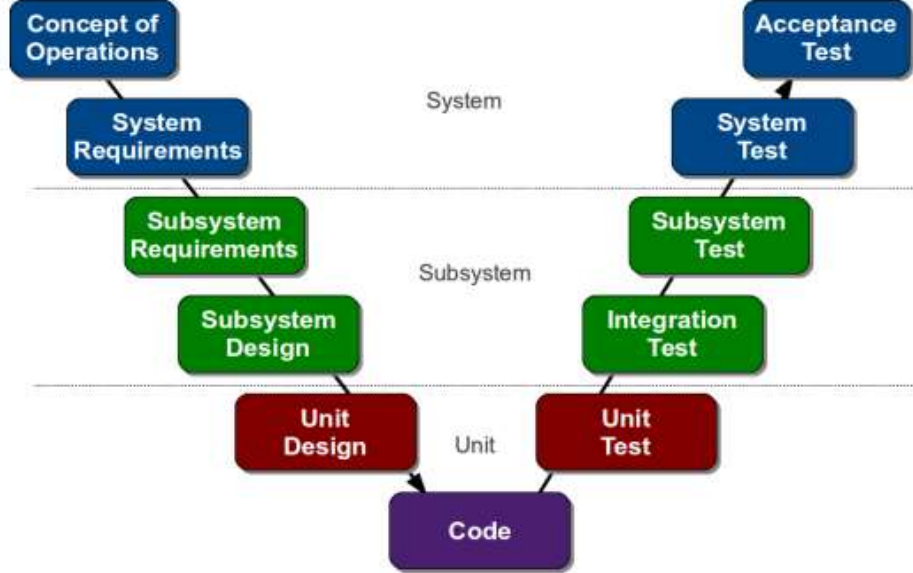
- TTB 2006 yılında yazılım test ve kalite alanında dünyanın en saygın gönüllü organizasyonu olan ISTQB'a (International Software Testing Qualifications Board – (www.istqb.org) bağlı olarak kurulmuştur. Bu tarihten itibaren , Türkiye'deki bilişim profesyonellerinin yazılım testi alanında ISTQB standartlarında eğitilmesi ve sertifikalanması amacıyla çalışmalarına başlamıştır.
 - Türkiye Bilişim Sektörünün uluslararası pazarlarda rekabet edebilmesi için sektörün yazılım testi ve kalitesi konusunda bilgilendirilmesi
 - ISTQB sınavlarının yapılarak sınavı kazanan adayların sertifikasyonu
 - Yazılım testi konusunda uluslararası kabul görmüş içeriğin Türkçeleştirilmesi

➤ En çok gerçekleştirdiğiniz test türleri nelerdir?

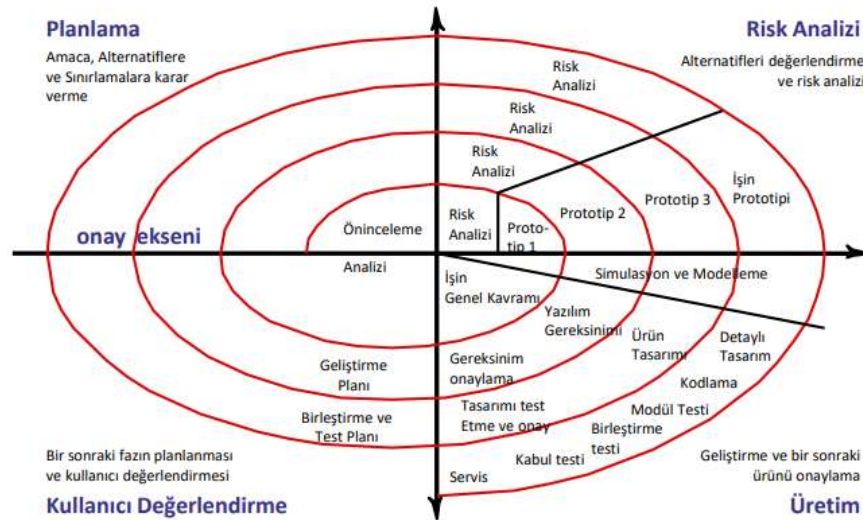


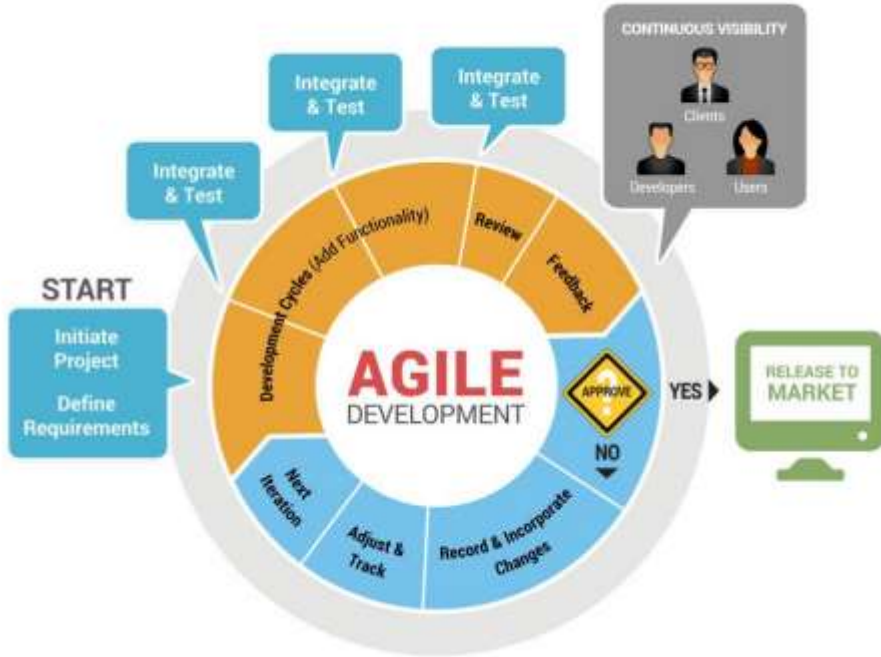


- Şelale Modeli: Yazılım süreci lineardır, bir sonraki safhaya geçebilmek için bir önceki safhada yer alan aktivitelerin tamamlanmış olması gerekir. Yani test aşamasına gelebilmek için diğer aşamalar tamamlanmalıdır. Hatalar sadece 5. aşamada giderilebildiğinden yazılımın maliyetini artırır ve başarısını azaltır



- V modeli: Test işlemlerinin ne zaman yapılacağını ön plana çıkarır. Bir önceki şekle bakacak olursak, sol kanat üretim etkinliklerini, sağ kanat da test etkinliklerini gösterir. Bu modelde geliştirme ve test paralel şekilde yapılır. Her aşama sonunda test edilecek ürün, test grubu tarafından sınanır, onay verildikçe bir sonraki aşamaya geçilir.





- Agile: Geleneksel yaklaşımın tersine, test profesyonelleri, yazılım geliştirme yaşam döngüsünün en başından itibaren sürece dahil olurlar.

TDD: Test Driven Development yöntemiyle kodlama yaparken genelde şu şekilde bir yol izlenmektedir.

1. Tek satır kod yazmadan kodun testini yaz.
2. Testi çalıştır ve testin geçemediğini gör.
3. Testi geçecek en basit kodu yaz ve testin geçtiğini gör.
4. Kodu düzenle
5. Başa dön