

# YAZILIM TEST MÜHENDİSLİĞİ

Dr. Öğr. Üyesi Hasan YETİŞ

# Yazılım Kalite Güvencesi

- Kalite kavramı temel olarak müşteri isteklerine maksimum düzeyde yanıt verebilmeyi içerir.
  - Yazılımlarda var olan hataların bulunması
  - düşük zaman-maliyet eşliğinde çözülmesi,
  - bakım ve yapılabirliği, istatistiksel süreç ve kontrolü,
  - test edilebilirliğin değerlendirilmesi,
  - tasarım hata tespiti gibi yazılım iyileştirme adımları

# Yazılım Kalite Güvencesi

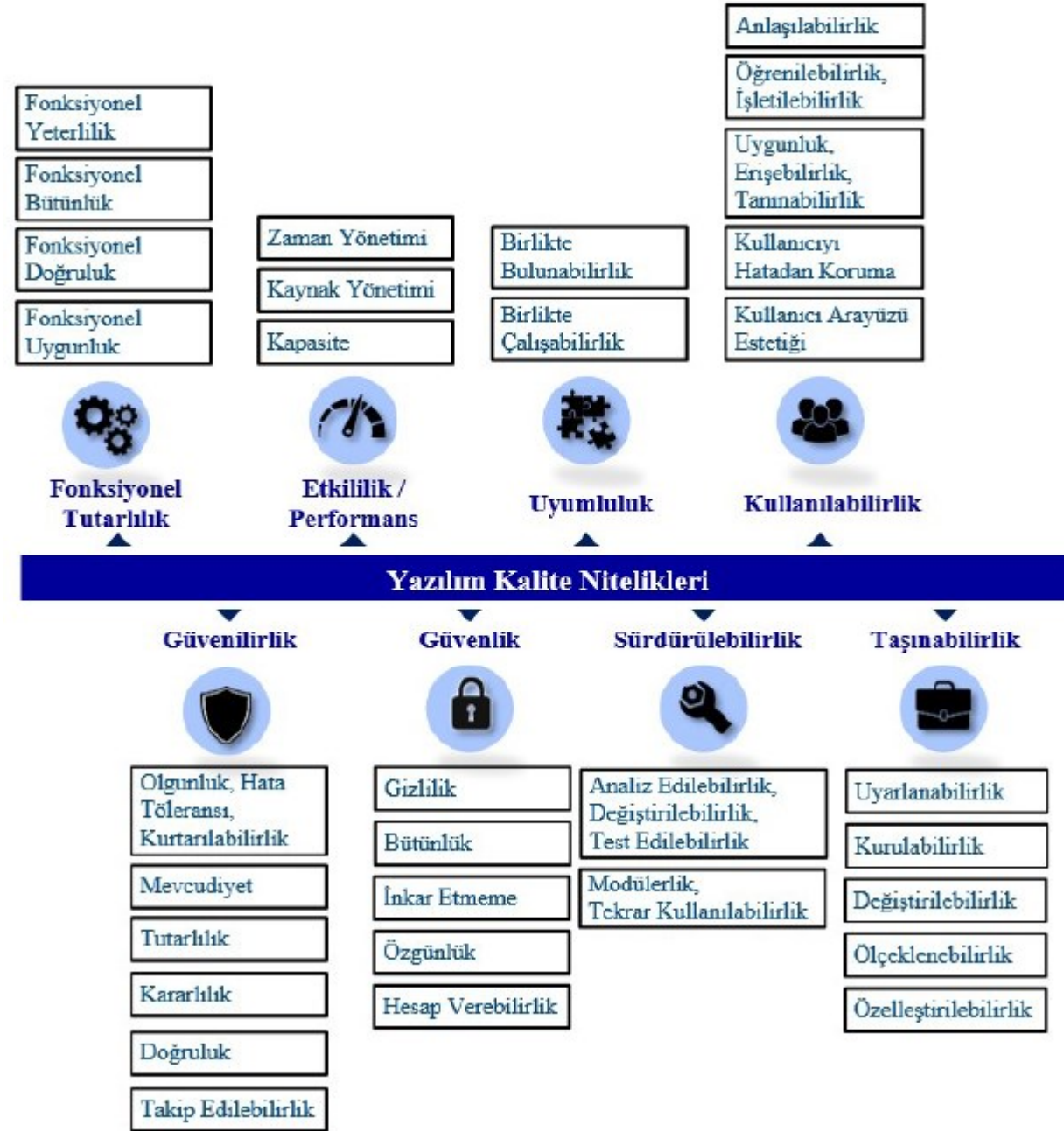
- Günümüzde kullanılan ve ihtiyaç duyulan yazılımlar, yüksek beklenti ve düşük maliyet gibi artan kullanıcı talepleri çerçevesinde şekillenmektedir.
- Projenin büyümesi iş yükünün artması anlamına gelmektedir.
- İş yükünün dağıtılması adına proje ekibinin büyütülmesi kaçınılmaz olmaktadır. Bu da projenin yönetilmesini zorlaştıran bir durumdur.
  - geliştirilen yazılımın maksimum verim ve doğrulama ile çalışması,
  - minimum düzeyde hata içermesi,
  - düşük maliyete sahip olması,
  - kodun kolay anlaşılabilirliği,
  - sistemin kolay kullanılabilirliği ve güncellenebilir olması

# Yazılım Kalite Güvencesi

- Yazılım projelerinin uzman olmayan tecrübesiz kişiler tarafından yönetilmesi sonucunda aşağıdaki aksamalar ortaya çıkmaktadır.
  - Proje bütçesi denkleştirilmeyebilir.
  - İhtiyaç duyulan yazılımsal ve donanımsal kaynakların temin edilemeyebilir.
  - Proje için ihtiyaç duyulan nitelikli ve uzman personel belirlenemeyebilir.
  - Sonradan tespit edilen büyük hatalar büyük maddi kayıplara neden olur.
  - Değişen ve gelişen kullanıcı istekleri doğrultusunda yazılım güncellemeleri mümkün olmayabilir. Sonuç olarak geliştirilen yazılımın kullanım ömrü kısa olur.
  - Yazılım projelerinde kullanılan teknoloji ve yöntemler yanlış seçilebilir.

# Kalite Nitelikleri

- Yazılımda kalite, yazılımı çeşitli yönleriyle inceleyerek, her yönüyle ortaya çıkan ürünün belirli standartlara sahip olmasını amaçlar.



# Fonksiyonel Tutarlılık

- Yazılımın içerdiği tüm fonksiyonellerin incelenmesi ve belirli standartlara göre incelenerek kontrol edilmesidir.
  - *Fonksiyonel Yeterlilik:* İhtiyaç analizinde belirlenen tüm durumlar göz önüne alınarak, yazılım için gerekli **tüm fonksiyonlar belirlenir**. Geliştirilen yazılımın da bu belirlenen tüm fonksiyonları içermesidir.
  - *Fonksiyonel Bütünlük:* Büyük yazılım projeleri farklı geliştirici gruplar tarafından küçük modüller halinde geliştirilip entegre edilir. Birleştirme sürecindeki tüm modüllerin **birbirleriyle uyumlu** ve problemsiz çalışmasıdır.
  - *Fonksiyonel Doğruluk:* İhtiyaç analizinde yer alan fonksiyonların **doğru çalışması** için yapılan testlerdir.
  - *Fonksiyonel Uygunluk:* Yazılıma ait fonksiyonların **kullanıcı isteklerine uygunluğu** test edilir. Eksik özelliklere sahip veya kullanıcı tarafından istenmeyen özellikleri barındıran yazılımlar uygunluk bakımından yetersiz kalmaktadır.

# Performans

- Yazılımın kullandığı kaynak miktarı, çalışma süresi ve etkili bir şekilde kaç kişiye kadar hizmet verebileceği gibi faktörler performans adı altında incelenmektedir
  - *Zaman Yönetimi:* Yazılımda yer alan fonksiyonların çalışma sırası ve süresi gibi etmenleri kapsar. Bu kapsamda **paralel çalışabilecek** fonksiyonların veya sırayla çalışması gereken fonksiyonların belirlenmesi önemlidir. Ayrıca yazılımın kısa sürede çalışması ve kullanıcıya kısa sürede cevap vermesi tercih edilir.
  - *Kaynak Yönetimi:* Yazılımların daha geniş kitleler tarafından kullanılabilmesi için önemlidir. Kaynak yönetimi, yazılımın kullanmış olduğu sabit disk ve bellek(RAM) gibi bilgisayar bileşenlerinin **daha etkili kullanılmasını** amaçlar. Yazılımlar, daha az kaynak ihtiyacı gerektirdiğinde, çok daha fazla kullanıcı tarafından tercih edilecektir.
  - *Kapasite:* Geliştirilen uygulama **çok kullanıcı tarafından** kullanıldığında bile problemsiz bir şekilde çalışmaya devam etmesi gerekir. Yük testleri ile bu başarımlar ölçülebilir.

# Uyumluluk

- Uyumluluk yazılım açısından iki farklı şeyi ifade edebilir. Birincisi, diğer yazılımlar ile çalışabilirlik, ikincisi ise sürüm farklarından kaynaklanan problemlerin minimuma indirgenmesidir.
  - *Birlikte Bulunabilirlik:* Farklı yazılımların bilgisayarda **aynı anda bulunabilmesidir**. Günümüzde işletim sistemleri birlikte çalışabilirlik durumlarını kontrol ettiğinden dolayı, bu tarz problemlerle daha az karşılaşılmaktadır.
  - *Birlikte Çalışabilirlik:* Farklı yazılımların **aynı anda çalışabilmesidir**. İşletim sistemleri kaynak kontrolünü elinde bulundurduğu için bu kapsamdaki problemler azalmaktadır. Ancak yine de ortak kaynak kullanımını (statik soket numarası gibi) gerektirecek şekilde geliştirilen yazılımlarda bu tarz problemler ortaya çıkmaktadır. Birlikte çalışabilirlik **farklı sürümlerin** aynı anda çalışabilirliğini de ifade etmektedir. Yeni versiyon çıktığında, eski versiyona sahip bilgisayarlarda da yazılımın bozulmadan kullanılabilmesi gerekir.



# Kullanılabilirlik

- Yazılımın kullanıcı tarafından kolay kullanılabilmesini amaçlayan kriterler bu başlık altında yer alacaktır.
  - *Anlaşılabilirlik: Arayüzün basit, sade ve anlaşılabilir* olması gerekmektedir. Bu kapsamda menülerin uygun yerlere yerleştirilmesi, gerekli açıklamaların yapılması ve kullanılacak ikonların anlaşılır olmasına dikkat edilmelidir.
  - *Öğrenilebilirlik-İşletilebilirlik: Ortaya çıkan arayüzler kolay öğrenilebilir* olmalıdır. Arayüzler, kullanıcı alışkanlıklarına cevap verir nitelikte olmalıdır. Bunun için alışılmışın dışında yerlere menüler eklenmesi gibi anormal konumlandırmalardan uzak durulmalıdır.
  - *Uygunluk-Erişilebilirlik-Tanınabilirlik: Ortaya çıkan yazılımların kullanıcılara uygun şekilde dizayn edilmesi gerekmektedir. Kullanıcıların engelli bireyler olması durumunda, yazılımın erişebilir olması büyük önem taşımaktadır.*

# Kullanılabilirlik

- *Kullanıcıyı Hatadan Koruma:* Yazılımda kodlama zamanında belli olmayan ancak **çalışma zamanında ortaya çıkan hatalar** yer almaktadır. Bu hatalar işletim sisteminin o an kaynakları başka bir uygulama için ayırması, işletim sisteminin dosya **okuma yazma işlemine** izin vermemesi veya kullanıcının **beklenmeyen bir değer girmesi** gibi çeşitli sebeplerden kaynaklanabilir. Yazılımın bu tarz hataları tolere ederek çökme yaşamadan çalışmaya devam etmesi gerekmektedir.
- *Kullanıcı Arayüzü Estetiği:* Her ne kadar çoğu zaman göz ardı edilse de kullanıcı arayüzü estetiği kullanıcıların yazılımı kullanmasını belirleyen önemli etkenlerden biridir. Bu noktada **renk seçimi, görsel tasarım** gibi işlemler uygun bir şekilde yapılarak, yazılımın her yerinde ortak bir arayüz izlenimi uyandırılmalıdır.

# Güvenilirlik

- Güvenilirlik yazılımın kullanılması açısından göz önünde tutulması gereken önemli bir faktördür. Güvenilirlik konusu, güvenlik konusundan tamamen farklıdır. Geliştirici firmanın kullanıcıya sağladığı olanaklar, satış sonrası destek, garanti süresi güvenilirlik kriterlerini ön plana çıkarmaktadır.
  - *Olgunluk-Hata Toleransı-Kurtarılabirlik*: Yazılımın olası çökmelere ve hatalara karşı **veri kaybının önlenmesidir**. Geliştirilen yazılımların otomatik kaydetme veya kayıt tutma(**loglama**) gibi özelliklere sahip olmalı, olası hatanın geri alınmasını sağlanmalıdır.
  - *Mevcudiyet*: Bir yazılıma karşı duyulan güvenilirliğin artması için o yazılımın yayınlanmasının **sürekli ve kullanılabilir** olması gerekmektedir. Bir süre sonra yayından kaldırılan veya çalışamaz duruma gelen yazılımlar güvenilir değildir.

# Güvenilirlik

- *Tutarlılık:* Tutarlılık zamandan ve mekandan bağımsız olarak **aynı girdiler için aynı çıktının** üretilmesi anlamına gelmektedir.
- *Kararlılık:* Bir yazılımın **en az hata** ile problemsiz çalışma durumudur. Sürekli çöken yazılımlar veya sürekli hata veren yazılımlar Beta test aşamasında anlayışla karşılanabilirken, kararlı sürümlerinde tahammül edilemez olmaktadır.
- *Doğruluk:* **Yanlış sonuç üreten** veya kullanıcıyı yanlış yönlendiren yazılımlar kullanıcıya güven vermezler. Doğruluk kavramı da düzenli gerçekleştirilen testler ile sağlanabilir.
- *Takip Edilebilirlik:* Yapılan işlemlerin **log kaydının** alınması takip edilebilirlik açısından önemlidir. İlerde karşılaşılabilecek olası hata durumunda veya bir problemde bu loglar yardımı ile hangi kullanıcının hangi işlemi ne zaman gerçekleştirdiği gibi bilgilere ulaşılabilir. Bu sayede sistemin geri alınabilirliği ve kurtarılabilirliği arttırılmış olur.

# Güvenlik

- Kullanıcı bilgilerinin korunması, sistemin açık ve şeffaf bir şekilde çalışması gibi faktörler güvenlik başlığı altında incelenebilir.
  - *Gizlilik:* Kullanıcı verilerinin kullanıcı isteği ve **onayı dışında başka kişilerle paylaşılmamasıdır**. Ayrıca bilgilerin korsan saldırılara karşı korunması ve olası saldırılar karşısında sistemin durumu gibi konular gizlilik kapsamı altında ele alınmalıdır.
  - *Bütünlük:* Yazılım sistemlerinin bir bütün halinde güvenli olmasını amaçlar. Yazılımda bütünlük, kavramı, **sistemde bulunan verilerin ve sistemde dönen işlemlerin** birbirleriyle uyumlu olması şeklinde tanımlanabilir.

# Güvenlik

- *İnkâr Etmeme*: Yazılımın kullanıcıya karşı **şeffaf olmasını** gerektirir. Sistemde meydana gelen kritik değişiklikler ya da hatalar kullanıcı bilgisine sunulmalıdır.
- *Özgünlük*: Bu kavram, yazılım geliştiricilerin mümkün olduğunca **farklı yaklaşımlar** kullanarak proje gerçekleştirmeleridir. Örnek verilecek olursa; hazır şablonlar kullanmak yerine, yazılım geliştiricilerin kendine özgü şablon veya desen geliştirmeleridir.
- *Hesap Verebilirlik*: Yazılım sisteminde meydana gelen değişikliklerin kayıt altına alınması ve gerektiği zamanda **sistemi ihlal eden davranışların kimin tarafından gerçekleştirildiğinin tespit edilmesi** gerekir. Takip edilebilirlik maddesi ile ilişkilidir

# Sürdürülebilirlik

- Yazılımların kullanıma açılmasının ardından kullanıcı istekleri doğrultusunda güncellenebilme özelliğinin devam etmesidir.
  - *Analiz Edilebilirlik-Değiştirilebilirlik-Test Edilebilirlik:* Yazılımların sürdürülebilir olması için yapılacak **güncellemelerin sisteme kolayca entegre** edilebilmesi gerekir. Bu sebeple yazılımın dinamik alt yapıya sahip olması önemlidir.
  - *Modülerlik – Tekrar Kullanılabilirlik:* Farklı ekipler tarafından geliştirilen modüler yazılımların farklı **projelerde veya uygulamalara entegre edilebilmesi** veya modüllerin yeniden kullanılmasıdır. Ayrıca modüler yazılım altyapısı sayesinde yazılan kodların (fonksiyonların) başka işleri gerçekleştirmek için ortak olarak kullanılabilmesi mümkündür.

# Taşınabilirlik

- Ortaya çıkan yazılımın farklı ortamlarda rahat bir şekilde kullanılması ve kullanıcı isteklerine cevap verebilmesidir. Yazılımın taşınabilir olması yazılım altyapısının esnek olmasıyla mümkündür.
  - *Uyarlanabilirlik:* Geliştirilen yazılımın benzer modüllerde olması ve **farklı ihtiyaçlara karşılık verebilmesidir**. Örneğin kafe yazılımının, restoran sipariş yönetimi yazılımı olarak uyarlanması söylenebilir.
  - *Kurulabilirlik:* Yazılım farklı bilgisayarlara pratik bir şekilde **kurulabilir** olmalıdır.
  - *Değiştirilebilirlik:* Sürdürülebilirlik başlığı altında da geçen bu madde, yazılımdaki **değişikliklerin ve güncellemelerin pratik bir şekilde** yapılabilmesidir. Bu özellik sayesinde yazılımın uyarlanabilir olması da artar.



# Taşınabilirlik

- *Ölçeklendirilebilirlik:* Yazılımın **dinamik bir altyapıya** sahip olmasını gerektirir. Örneğin restoran otomasyonu için masa sayısının kullanıcı tarafından belirlenmesi ölçeklendirilebilirliğe örnek olarak gösterilebilir. Yazılımı kullanacak kişi sayısının belirlenmesi de bir ölçeklendirme örneğidir.
- *Özelleştirilebilirlik:* Geliştirilen yazılımlar, kullanıcı talepleri doğrultusunda **özelleştirilebilir** olmalıdır. Farklı restoranların kendi menülerini özelleştirmelerine izin verilmesi yazılımın özelleştirilebilirliğidir.

# ISO/IEC-9126 Kalite Standardı



# ISO/IEC-9126 Kalite Standardı



- *Fonksiyonellik(İşlevsellik)*, ISO-9126 kalite ölçütlerine göre yazılım sisteminin belirlenen koşullardaki ihtiyaç ve işlevleri yerine getirebilme yeteneğidir.
- Yani yazılım kullanıcının tüm isterlerini karşılayabilmelidir.

# ISO/IEC-9126 Kalite Standardı



- *Güvenilirlik*, yazılım sisteminin belirlenen koşullara göre kullanımı gerçekleştirildiğinde ortaya çıkan performansın belirli bir sürdürülebilirliğe sahip olmasıdır.
- Yani, yazılımın kullanıcı tarafından güncelleme, ihtiyaçlara cevap verebilme gibi desteklerin devam etmesidir. Yazılım güncel bir sürüme sahip olmalı, sürekli ve hatasız çalışabilmeli, tüm işlevleri doğru yapmalı, hatalı girdilere ve kullanıcı yanlışlıklarına karşı korumalı olmalıdır.

# ISO/IEC-9126 Kalite Standardı



- *Verimlilik*, yazılım işlevlerini yerine getirirken sistem öz kaynaklarını en uygun şekilde kullanmaktır. İşlemci, bellek, disk ve diğer kaynakların kullanımı en uygun şekilde yapılmalı, ileişim, grafik sergileme ve yazıcı çıktıları yüksek başarıma sahip olmalıdır.

# ISO/IEC-9126 Kalite Standardı



- *Taşınabilirlik*, yazılımlar farklı donanım ve işletim sistemine sahip bilgisayarların kullanıldığı ortamlara aktarılabilir olmalıdır. Yeni işletim sistemi sürümlerinin kullanılması, bilgisayar değiştirilmesi gibi durumlarda, daha önce geliştirilmiş olan yazılım en az çaba ile tekrar kullanılabilmelidir.

# ISO/IEC-9126 Kalite Standardı



- *Kullanılabilirlik*, yazılım sisteminin kullanıcı tarafından anlaşılması, hızlı öğrenilmesi ve kullanım kolaylığı olarak belirtilebilir.
- Üretilen yazılımı insanların rahatlıkla kullanabilmesi için gerekli kolaylıklar sağlanmalı, özellikle grafiksel kullanıcı arayüzü düzenli, estetik ve kullanımı kolay olmalıdır.



# ISO/IEC-9126 Kalite Standardı



- *Bakım kolaylığı*, başka bir yazılım geliştirici kişi ya da grup tarafından yazılım bakımının yapılabilmesi için kaynak kodun anlaşılabilir bir şekilde yazılmış olması, iyi belgelendirilmesi, sorun çözümlemesinin ve testinin kolay olması gereklidir.