

YAZILIM MÜHENDİSLİĞİNE GİRİŞ : TANIM, ETİK OLUŞUMU VE İYİ BİR YAZILIM



Yazılım nedir?

Yazılım, başta bilgisayar olmak üzere çeşitli dijital cihazlarda çalışan, belirli görevleri yerine getirmek üzere kodlanmış talimatlar bütünüdür. Bir yazılım; kullanıcıların belirli bir işlevi yerine getirmesi, bilgiye ulaşması veya işlemleri gerçekleştirmesi için gerekli olan adımları içerir. Yazılım, bilgisayarlara yada cihazlara tam olarak ne yapmaları gerektiğini söyler ve onları birer “çözüm aracı” haline getirir.

Yazılım Krizi Nedir?

Yazılım krizi, 1960'ların sonunda yazılım geliştirme projelerinin çoğunun başarısızlıkla sonuçlanması ya da beklenen bütçenin ve sürenin çok üstünde tamamlanması ile tanımlanan bir dönemdir. Dönemde bilgisayar kullanımları genişledikçe yazılımlar daha karmaşık hale geldi, döneme göre uyulan düzen karmaşıklığa yetersiz geldi. Süreçler iyi tanımlanmadığı için hatalar ortaya çıkıyor ve yazılımcılar bu hataları hızlıca çözemiyordu. Bu kriz, yazılım endüstrisinde büyük bir farkındalık yaratmış ve yazılım mühendisliğinin doğmasına öncülük etmiştir.

Krizi çözmeye yönelik çabalar, 1968 yılında Almanya'nın Garmisch kasabasında gerçekleştirilen NATO Yazılım Mühendisliği Konferansı'nda başlamıştır. Bu konferans, yazılım krizini çözmek için yazılım mühendisliğinin gerekliliğini ortaya koydu ve yazılımın daha disiplinli bir şekilde geliştirilmesi için bir dizi öneri sundu.

Yazılım krizinin çözülmesi için yazılım mühendisliği disiplininin gelişmesi, çeşitli yazılım geliştirme metodolojilerinin (ör. Şelale, Agile) ve programlama dillerinin (ör. yüksek seviyeli diller) ortaya çıkması sağlandı.



1968 yılında NATO Yazılım Mühendisliği Konferansından bir görüntü

Yazılım Mühendisliği nedir?

Yazılım mühendisliği, yazılım geliştirme sürecinin tüm aşamalarını kapsayan bir mühendislik dalıdır. Bu disiplin, yazılımın tasarımından bakıma kadar olan süreçlerde mühendislik ilkelerini, bilimsel yöntemleri ve matematiksel

hesaplamaları kullanarak çözümler üretir. Yazılım mühendisliği, yalnızca yazılımın kodlanması ile ilgili değil, aynı zamanda tüm geliştirme süreçlerinin yönetilmesi, planlanması, test edilmesi ve bakımı ile ilgilidir.

Yazılım mühendisliğinin amacı, daha sağlam, güvenli ve kullanıcı dostu yazılımlar geliştirmek ve bu yazılımların sürdürülebilirliğini sağlamaktır. Bu süreç, yazılım projelerinde daha verimli ve hatasız sonuçlar elde edilmesine olanak tanır.

Yazılım Mühendisliği Etiği

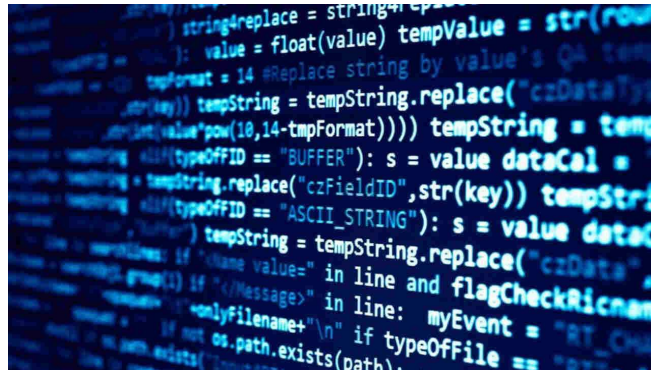
1990'ların ortalarında İş Etiği kavramının önemi artmaktaydı. Bununla birlikte yazılım mühendisliği alanında da etik kavramı ile ilgili araştırmalar hızla başlanmıştır. Yazılım mühendisliği alanında yapılan önemli çalışmalardan birisi 1996 yılında yapılan "Yazılım Mühendisliğinde Profesyonellik" başlığıyla yayımlanan rapordur. Bu tarihten sonra da 1998 yılında IEEE Computer Society ve ACM tarafından "Yazılım Mühendisliği Etik Kuralları ve Profesyonel Uygulaması" raporu yayınlanmıştır. Bu raporlarla birlikte yazılım mühendisliğinde daha net bir etik kavramı ortaya çıkmıştır.

Günümüzde kabul edilebilir davranışların standartlarının yasalara bağlı olmadığı fakat daha zayıf mesleki sorumluluk kavramına bağlı olduğu alanlar vardır. Bunların bazıları:

1. Gizlilik
2. Yetelilik
3. Fikri Mülkiyet hakların
4. Bilgisayarın kötüye kullanımı

Yazılım üretimindeki temel sorunlar

- Gereksinimlerin hatalı tanımlanması,
- Müşteri-geliştirici arasındaki iletişim başarısızlığı,
- Yazılım gereksinimlerinden kasıtlı sapmalar,
- Mantıksal tasarım hataları,
- Kodlama hataları,
- Kodlama standartlarına uymama,
- Test sürecinin eksiklikleri,
- Kullanıcı arayüzü ve prosedür hataları,
- Dokümantasyon hataları.



İyi bir yazılım nedir?

İyi bir yazılım, yalnızca işlevsel olması değil, aynı zamanda kullanıcı beklentilerini karşılayan, güvenilir, sürdürülebilir, verimli ve bakımı kolay olan bir yazılımdır. İyi bir yazılımın kullanıcı dostu olması, verimli çalışması ve teknik gereksinimlere uygun olarak optimize edilmiş olması beklenir. Ayrıca uzun vadede gerekli güncellemelerin yapılabilmesi, hataların giderilebilmesi ve kullanıcıların ihtiyaçlarına adapte olabilmesi de önemli özellikleridir.

İyi bir yazılımın özellikleri nedir?

Kabul edilebilirlik : Yazılım, hedeflenen kullanıcı kitlesi tarafından anlaşılabilir, kullanıcı dostu ve kullanım amacına uygun olmalıdır. Aynı zamanda kullanıcıların alışkın olduğu sistemlerle uyumlu olması, kabul edilebilirliğini artırır.

Güvenilirlik ve güvenlik : Güvenilir bir yazılım, kullanıcıya veya sisteme fiziksel ya da ekonomik zarar verecek hatalara yol açmamalıdır. Aynı zamanda, kötü niyetli kullanıcıların sisteme erişimini engelleyerek, yetkisiz erişimlere ve saldırılara karşı koruma sağlar.

Verimlilik : Yazılım, sistem kaynaklarını (örneğin hafıza, işlem gücü) gereksiz yere tüketmemelidir. Verimli yazılım, sorunlara hızlı yanıt verilebilirlik, işlem hızı ve bellek kullanımı açısından optimize edilmiş olmalıdır.

Bakım kolaylığı : Yazılım, zaman içinde kullanıcı ihtiyaçlarına göre değiştirilebilmeli ve iş gereksinimlerine göre güncellenebilmelidir. Yazılımın esnek olması ve uyum sağlama yeteneği, değişen iş dünyasında bu özelliği önemli kılar.

Durum Çalışmaları

Yazılım mühendisliği kavramını örnekle açıklamak için dört farklı tür sistemden örnek vermek gerekirse;

1. Gömülü Sistemler

Gömülü sistemler, belirli bir işlevi yerine getirmek üzere tasarlanmış, donanım ve yazılımın birleşimi olan sistemlerdir. Bu sistemler, genellikle düşük enerji tüketimi, yüksek güvenilirlik ve gerçek zamanlı performans gereksinimlerine sahiptir.

Örnek: Otomobilin Fren Sistemi

Bir otomobilin fren sistemi, yüksek güvenlik gereksinimleri taşıyan kritik bir gömülü sistem örneğidir. Bu sistem, aracın hızını, yol koşullarını ve diğer çevresel faktörleri sürekli olarak izler ve frenleme işlemini anında gerçekleştirir. Burada yazılım mühendisliği, doğru sensörlerin kullanılması, sistemin güvenliği ve hata toleransının sağlanması açısından önemlidir.

- **Yazılım Mühendisliği Bağlantısı:** Gömülü yazılım mühendisleri, bu tür sistemlerde hata payını minimize etmek için titiz bir yazılım geliştirme süreci izler. Gerçek zamanlı işlem yapabilen yazılımlar geliştirilir ve sistemin güvenliği ön planda tutulur.

2. Bilgi Sistemleri

Bilgi sistemleri, verilerin toplanması, işlenmesi ve depolanmasını sağlayan yazılım sistemleridir. İşletmelerin karar verme süreçlerinde kritik rol oynar ve genellikle büyük veri analizleri yapar.

Örnek: Bir Hastane Yönetim Sistemi

Bir hastane yönetim sistemi, hasta bilgilerini toplar, tedavi geçmişini izler, doktor ve hemşirelerin programlarını organize eder, tıbbi stokları yönetir ve hastaların ödemelerini takip eder. Bu sistem, aynı zamanda güvenli bir şekilde hasta bilgilerini saklamak zorundadır.

- **Yazılım Mühendisliği Bağlantısı:** Bilgi sistemleri geliştiren yazılım mühendisleri, veritabanı yönetiminden güvenli veri transferine kadar geniş bir beceri setine sahip olmalıdır. Ayrıca, kullanıcı dostu arayüzler tasarlamak, yüksek performanslı veri sorgulamaları yapmak ve kritik verilerin korunmasını sağlamak için etkileşimli bir tasarım süreci gerekir.

3. Sensör Tabanlı Veri Toplama Sistemleri