

YAZILIM BAKIMI

Yazılım geliştirme amacı, kullanıcılarının gereksinimlerini karşılamak için bir ürün sağlamak olduğu kadar, bu ürünü sürdürülebilir, güvenli ve güvenilir kılmaktır.

Geliştirilmesi tamamlanan ve müşteriye teslim edilen yazılım, çeşitli sebeplerle sürekli değişim baskısı altında kalmakta ve bu baskı sonucunda çeşitli gereksinimler ortaya çıkmaktadır. Gereksinimler yazılım kullanıldığı sürece devam etmekte ve bu sebeple iyi planlanmış desteğe ihtiyaç duyulmaktadır. Bu desteğin karşılığı iyi planlanmış "Yazılım Bakımı"dır.

Yazılım Bakımı : Yazılımın hatalarını gidermek, performans ve diğer özelliklerini iyileştirmek ya da farklı bir ortama uyarlamak gibi gereksinimlerin bütünsellik içinde, maliyet etkin olarak karşılanması ve desteğin sağlanması olarak tanımlanabilir.

Yazılım bakımına gerek duyulmasının temel nedenleri aşağıdaki şekilde belirtilmektedir.

Hizmet sürekliliğini sağlamak: Sistemi çalışır ve erişilebilir durumda tutmak,

Zorunlu güncelleştirme desteğini sağlama: Yasalarda ve yazılımın çalıştığı platformda meydana gelen değişiklikler nedeniyle oluşan isteklerinin karşılanması,

Kullanıcı gereksinimlerindeki değişim isteklerini sağlamak: Kullanıcıların fonksiyonelliği artırma yönündeki istekleri ve gereksinimlerde meydana gelen değişimlerin karşılanması,

Gelecekte bakıma gereksinim duyma: Ticari ve finansal nedenlerden dolayı yazılım geliştirme kısa yoldan yapılması nedeniyle kodda veya veritabanında yeniden yapılandırma ve dokümantasyonda güncelleme gereksinimleri,

BAKIM TÜRLERİ

Bakım işleri yalnızca hataları düzeltmek olmayıp ürünün tesliminden sonra yapılması gereken çeşitli işleri de kapsamaktadır. ISO/IEC 14764-2006'ya göre 4 çeşit yazılım bakım türü tanımlanmaktadır.

- **Düzeltilici bakım**
- **Uyarlayıcı bakım**
- **İyileştirici bakım**
- **Önleyici bakım**

Düzeltilici Bakım :

Tespit edilen hataların kaynağını araştırıp hatayı ortadan kaldırmaya yönelik çalışmalara düzeltilici bakım adı verilir. Çok sık düzeltilici bakım ile karşılaşmamak için;

- Güçlü bir test uygulaması benimseyin,
- Yüksek kaliteli kod geliştirin,
- Tasarım şartnamesinin doğru uygulanmasına odaklanın,
- Sorunları önceden tahmin etme yeteneğinizi geliştirin.

Uyarlayıcı Bakım :

Yazılımın, yeni bir çalışma ortamına uyarlanmasıdır.

Teknoloji, yasalar, politikalar, kurallar, işletim sistemi vb. nedenlerle yazılımın çalıştığı platform değişiyor olabilir. Yazılımınızın yaşadığı ortamı değiştiriyorsanız, yazılımınızın diğer bölümlerinde de değişiklikleri tetikleyeceğini unutmayın.

Uyarlayıcı bakım işlemlerini hızlı bir şekilde yapmanız gerekir çünkü gecikme daha sonra doğru bir bakım yapamamanıza neden olabilir, bu da pahalı bir yaklaşımdır.

İyileştirici Bakım :

Sisteme yeni işlev ve özelliklerin eklenmesi, performansın artırılması gibi bakım çalışmalarıdır.

Örneğin;

- Hız optimizasyonu,
- Kullanıcı arayüzlerinde iyileştirme,
- Yazılım kullanılabilirliğindeki iyileştirmeler,
- Yazılım işlevselliğinin iyileştirilmesi,
- Yazılım performansında iyileşme.

Genellikle bir sisteme çalışmaya başladıktan sonra yeni bir işlev eklemek, aynı işlevin henüz geliştirme sürecindeyken eklenmesine göre çok daha maliyetlidir.

Önleyici Bakım :

Yazılımın gelecekte uygulanabilecek değişikliklere daha iyi adapte olması için bakılabilirliği ve güvenilebilirliği artırabilmek için alınabilecek tedbir niteliğindeki işlemlere önleyici bakım denir.

Örneğin;

- Belgelerin güncellenmesi: Belgeyi sistemin mevcut durumuna göre güncellemek.
- Kodu en iyi duruma getirme: Programların daha hızlı yürütülmesi veya depolama alanının verimli bir şekilde kullanılması için kodu değiştirme.
- Kodun yeniden yapılandırılması: Kaynak kodunu azaltarak programın yapısını dönüştürmek, kolayca anlaşılmasını sağlamak.

BAKIM AŞAMALARI

Bakım evresinde bulunan bir yazılım için bir bakım işi ortaya çıktığında yazılım geliştiricisi tarafından standart bir süreç izlenmelidir. Bakım aşamasını aşağıdaki gibi özetleyebiliriz.

- Problem Tanımlama Aşaması,
- Problem Analiz Aşaması,
- Tasarım Aşaması,
- Uygulama Aşaması,
- Sistem Test Aşaması,
- Kabul Test Aşaması,
- Teslimat Aşaması.

Problem Tanımlama Aşaması :

Bu aşama yazılımsal değişiklik talepleri değerlendirilir;

- Sorun veya değişiklik istekleri tanımlanır,
- Hangi bakım faaliyetine ait olduğu sınıflandırılır,
- Sistemden beklenen yeni düzenlemeler ve işlevlerle ilgili isterler tanımlanır,
- Öncelik durumu ve yazılım bakımı için gereken kaynakların tahmini yapılır,

Bu aşamanın çıktısı onaylanmış veya red edilmiş bir değişiklik isteğidir.

Problem Analiz Aşaması :

Bu aşamada, onaylanan her değişiklik isteğinin uygulanabilirliği ve kapsamı aşağıdakileri içeren bir analiz sonucunda belirlenir.

- Değişikliklerin etkisi,
- Alternatif çözümler,
- Dönüşüm gereksinimleri,
- Güvenlik,
- İnsan faktörleri,
- Zaman ve kaynak faktörleri,
- Test stratejisi.

Bu aşamanın çıktısı, değişiklik talebi için fizibilite raporu, güncellenmiş şartlar, test stratejisi, ayrıntılı analiz raporu ve ön değişiklik listesi gibi unsurları içerir.

Tasarım Aşaması :

Bu aşamada, yazılımda yapılacak değişiklikler tasarlanır. Tasarım işlemleri aşağıdaki adımları içerir;

- Etkilenen yazılım modüllerinin tanımlaması,
- Yazılım modülü dokümantasyonunun değiştirilmesi,
- Güvenlik ve güvenlik konularını içeren yeni tasarım için testlerin geliştirmesi,
- Regresyon (işlevselliğin bozulmadığının gösterilmesi) testlerinin oluşturulması,
- Güncellenmiş gereksinimlerin belgelenmesi,
- Değişikliklerin listesinin gözden geçirilmesi.

Tasarım aşamasında, aşağıdaki kontrol işlevleri yerine getirilir;

- Tasarımın standartlarına uygunluğu incelemesi yapmak,
- Yeni tasarım / gereksinimin belgelendiğini doğrulamak,
- Güvenlik ve güvenlik konularıyla birlikte yeni tasarımın dahil edildiğini doğrulamak,
- Test belgelerinin değiştirildiğini doğrulamak.

Bu aşamanın çıktısı; ayrıntılı değişiklik analizi, değiştirilmiş test planları, doğrulanmış gereksinimler ve uygulamadaki kısıtlamalar ve riskleri içeren bir listedir. Çıktı, uygulama planının yanı sıra kullanıcı gereksinimlerinin detaylı analizine dair bilgi sağlar. Bu plan, uygulama aşamasında uygulanacak görevleri listeler. Tasarım test planları güncellenmiş tasarıma göre hazırlanır.

Uygulama Aşaması :

Bu aşamada, yazılım kodunda gerçek değişiklikler yapılır, mevcut yazılımın özelliklerini destekleyen yeni özellikler eklenir ve değiştirilen yazılım yüklenir. Bu aşamada aşağıdaki alt başlıklar uygulanır.

- **Kodlama ve ünite testi:** Tasarım aşamasında tanımlanan kullanıcı gereksinimleri, yazılım kodu şeklinde uygulanır. Yazılım kodu yazıldıktan sonra, tek tek modüller üzerinde birim testi gerçekleştirilir.
- **Bütünleştirme:** Yazılım kodu yazıldığında ve tüm modüller test edildiğinde, değiştirilen yazılım modülleri mevcut sistemle bütünleştirilir. Yazılım sisteminin bütünleştirilmesinden sonra bütünleşme testi ve regresyon testi yapılır. Bu testler, değiştirilmiş yazılım sisteminin işlevselliği, performansı, kullanılabilirliği ve güvenliği üzerindeki etkiyi belirlemek için yapılır. Yazılım sisteminin arızası gibi ciddi hatalara yol açan etkiler, hata kaynağını belirledikten ve anladıktan sonra giderilebilecekleri kaydedilir.
- **Risk analizi ve gözden geçirme:** Uygulama aşamasının tamamlanmasından sonra periyodik olarak yapılmaktadır. Yaygın olarak ortaya çıkan riskler, veri arızası ve yedekleme sağlanamamasıdır.
- **Test hazırlığı incelemesi:** Bu inceleme, sistem testlerinin standartlarına uygun olup olmadığını değerlendirmek için yapılır.

Bu aşamada yapılan kontrollerde;

- Standartlarına uygun olarak kodun yazılım denetimleri,
- Ünite ve bütünleşme testlerinin yapılmış ve belgelendirilmiş olması,
- Test planı ve test senaryoları gibi test belgelerinin güncellenmesi veya oluşturulması,
- Yeni yazılımın kalite kontrolü altında doğrulaması,
- Eğitim ve teknik dokümantasyondaki değişikliklerin yapıldığının doğrulanması, sağlanmış olur.

Bu aşamanın çıktısı, değişiklik isteğine, güncellenmiş tasarım dokümanlarına, güncellenmiş test dokümanlarına, güncellenmiş kullanıcı dokümanlarına ve test hazırlık inceleme raporuna göre güncellenmiş/değiştirilmiş yeni yazılımdır.

Sistem Test Aşaması :

Bu aşamada, bakım aktivitesinin bir sonucu olarak yazılımda yeni hataların ortaya çıkmamasını sağlamak için değiştirilmiş sistemde regresyon testi gerçekleştirilir. Giriş niteliği, güncellenmiş yazılım belgelerini, test hazırlığı inceleme raporunu ve güncellenmiş sistemi içerir.

İşlem özniteliğinde, tam bütünleşmiş sistemde, sistem işlevsel testi, arabirim testi ve regresyon testi gibi çeşitli testler gerçekleştirilir. Kontrol niteliğindeyken, sistem testi gerçekleştirilir. Sistem testi ayrı bir test fonksiyonu yardımı ile gerçekleştirilir. Test fonksiyonu, sistem testinin çıktısını kaydetmek ve sistem testi için test planında

oluřturulan kriterin durumunu belirlemek için gereklidir. Bu bilgi, kabul testi olan bir sonraki ařamaya geçmek için çok önemlidir.

Bu ařamanın çıktısı, bütünüřtirilip test edilmiş yazılımın test raporudur.

Kabul Test Ařaması :

Bu ařamada, kullanıcı tarafından tam bütünüřmüş sistemde veya kullanıcı tarafından belirtilen üçüncü bir kiři tarafından kabul testi gerçekteřtirilir. Amaç, hataları tespit etmek ve yazılım özelliklerinin deęişiklik isteęinde belirtilen gereksinimlere uygun olduęunu doęrulamaktır. Girdi nitelięi tamamen bütünüřmüş sistem, kabul testi planları, kabul testi durumları ve kabul testi prosedürlerini içerir.

Bu ařama işlemsel olarak;

- Kabul testi gerçekteřtirmek,
- Performansını ve çalıřmasını belirlemek için sistemi test etmek,

Kontrol özellięi olarak ise ;

- Kabul testlerini yürütmek
- İşlevsel denetim yapmak
- İşlevsel Yapılandırma Denetimi için test sonuçlarını raporlamak,
- Kabul testi dokümantasyonunu kontrol etmek,

prosedürlerini içerir.

Kabul testi ařamasının çıktısı; test sonuçları ve kabul testi raporudur.

Teslimat Ařaması :

Bu ařamada, deęiřtirilmiş (veya yeni) yazılım sistemi kullanıcıya iletilir. Bu yazılım sistemde tamamen test edilmiş ve kabul edilmiş bir versiyonunu içerir.

Ek olarak, kullanıcılara, kılavuzun ve yazılımın donanım spesifikasyonu ile birlikte kullanımını açıklayan yardım dosyalarından oluřan uygun bir dokümantasyon saęlanır.

Teslimat ařamasının işlemleri ařaęıdaki adımları içerir.

- Fiziksel Yapılandırma Denetimi Yapma,
- Kullanıcıları bilgilendirme,
- Sistem yedeklemesi için hazırlık yapılması,
- Deęiřtirilmiş sistemi kurmak ve kullanıcılara eğitim verilmesi,

Kontrol özellięi ařaęıdaki işlevleri yerine getirir.

- "Fiziksel Kontrol Denetlemesi" raporunun düzenlenmesi ve belgelenmesi,
- Kullanıcıların sistem malzemelerine erişmesine izin verilmesi,
- Tam bir "Sürüm Açıklama Belgesi" saęlanması,
- Teslimatın içerięinin yapısal olarak kontrol edilmesi,

BAKIM FAALİYETLERİ

Yazılım bakım faaliyetleri yazılım yaşam döngüsünün en son yapılan faaliyeti olması açısından tüm diğer faaliyetler ile ilişkili, çeşitli uzmanlık alanlarını içeren ve müşteriye de içine alacak şekilde bir takım çalışmasını gerektiren, karmaşık faaliyetler bütünüdür.

Yönetim, iletişim, yazılım ve bilgisayar bilimlerine ait bilgilerin kullanıldığı, görev ve süreç tabanlı ilişkilerin ve rollerin bulunduğu bir alandır.

Bakım faaliyetleri;

- Bakımın ve yazılım konfigürasyonunun yönetilmesi ile ilgili;
yönetim,
- Yazılımındaki değişikliklerin sınanması, ürünün sürümlendirilmesine yönelik;
denetim,
- Değişimlerin ürünün bütününe zarar vermediğinden emin olunması ile ilgili;
kalite
- Değişimi gerektiren etkilerin belirlemesi ve analiz edilmesine yönelik;
araştırma,

alt faaliyetlerini içermektedir

Bakım ile ilgili gereksinimler, çevrimiçi yardım dokümantasyonları, dinamik kütüphaneler, güvenlik, diğer sistemlerle olan iletişim ve ilişkiler, veri toplama araçları ve toplanan verilerin saklanmasına yönelik yapılar, bakımın kaynağını oluşturmaktadır

Ayrıca yukarıdaki faaliyetleri tamamlayan, bakım sisteminin yürütülmesine yardımcı olan bakımcı, kullanıcı ve operatör gibi insan kaynaklarının da yönetimin de bakım faaliyetleri içerisindedir.

İyi bakım iyi bakımcılarla yapılır.

Bakımı yapan kişiler en az bakımın yönetimi, süreçlerin analizi ve bakımın sonuçlandırılması kadar önemlidir ve yazılım bakımına ait insan kaynakları üç temel alanda sınıflandırılmaktadır .

- Bakımcı,
- Bakım Yöneticisi,
- Tedarikçi,

BAKIM İSTEĞİ

Bir yazılım projesinde nitelikli bir bakım aşamaları oluşturabilmesi için aşağıdaki durumlara dikkat edilmesi gerekir.

- Bakım istekleri her zaman resmi bir deęişim denetim süreci kapsamında ele alınmalı,
- Bakım süreci hızlı bir şekilde yerine getirilmeli,
- Bakım istekleri için isabetli ihtiyaç analizi yapılmalı(işgücü, maliyet, süre vb),
- Bakım isteklerinin durumu sistematik olarak incelenmeli,
- Verimi artırıcı bakımı kolaylaştırıcı araçlar kullanılmalı,

NİCELİKSEL ÖLÇÜMLER

Bakım işlemlerinin özelliklerinin dikkate alarak bazı ölçevler ortaya koymak ve niceliksel ölçümler yapmak mümkündür. Bu ölçevler şu şekildedir.

- Sorunun belirlenip rapor edilmesi,
- Sorun bildirimlerinin dönemsel yoğunluğu,
- Sorunun çözülme süresi,
- Deęişikliğin tanımlaması için gereken süre,
- Düzeltme için geçen süre,
- Toplam süre,

BAKIMIN SORUNLARI

Bakım sorunlarını şu şekilde ifade edebiliriz.

- Yazılımın sürümü artıkça bakım zorlaşır.
- Yazılımın geliştirildięi süreci aynen takip etmek zaman ve iş gücü bakımından zahmetlidir.
- Bir başkasının yazdığı kodu anlamak genellikle zaman alır
- Belgelendirmeler yetersiz ve eksik olabilir
- Yazılımların çoğunun üzerinde deęişiklik yapabilecek şekilde tasarlanmadıkları için geliştirmek ve modül eklemek güçtür.

BAKIMIN YAN ETKİLERİ

Yazılım bakımı sürecinde dikkatli olmak gerekir. Test edilmiş ve kullanıma açılmış bir üründe yapılan küçük bir deęişiklik iyi çözülmezse, etkileri takip edilmezse ciddi sıkıntılara yol açar. Genellikle kod içinde, verilerde, belgelemede ve başarımda ortaya çıkan istenmeyen sonuçlara bakımın yan etkileri denmektedir. Yan etkileri şu şekilde sıralayabiliriz

- Kodlamanın etkilenmesi
- Verilerin etkilenmesi
- Başarımın etkilenmesi

SONUÇ

Tasarımdan elden çıkarmaya kadar geçen süreçte ; düzeltilebilir (correctable), adapte edilebilir (adaptable), iyileştirilebilir (perfectible) ve taşınabilir (migratable) bir yazılım ürünü tasarlamak ve gerekli bakım desteğini sağlamak tüm yazılımcıların ve bu işle uğraşan firmaların misyonu olmalıdır.

KAYNAKLAR

ISO/IEC 12207 Systems and Software Engineering — Software Life Cycle Processes

ISO/IEC 14764 Information Technology — Software Maintenance

IEEE Std. 1219-1998 Standart for Software Maintenance

Swebok V3.0 Guide to the Software Engineering Body of Knowledge

Yazılım Bakım Rehberi Tübitak – Bilgem

Software Maintenance Thomas M. Pigoski

Hasan KARABAĞLI



Dz.K.K.ılığnda 24 yıl Elektronik Astsubayı olarak görev yaptıktan sonra, sırasıyla Tesla Elektronik , Globecomm Systems(USA) ve Thales Group (FR) firmalarında saha mühendisi olarak çalışmıştır .
2008 yılından itibaren çalıştığı MilSOFT'da Entegre Lojistik Destek Uzmanı olarak görev yapmaktadır.