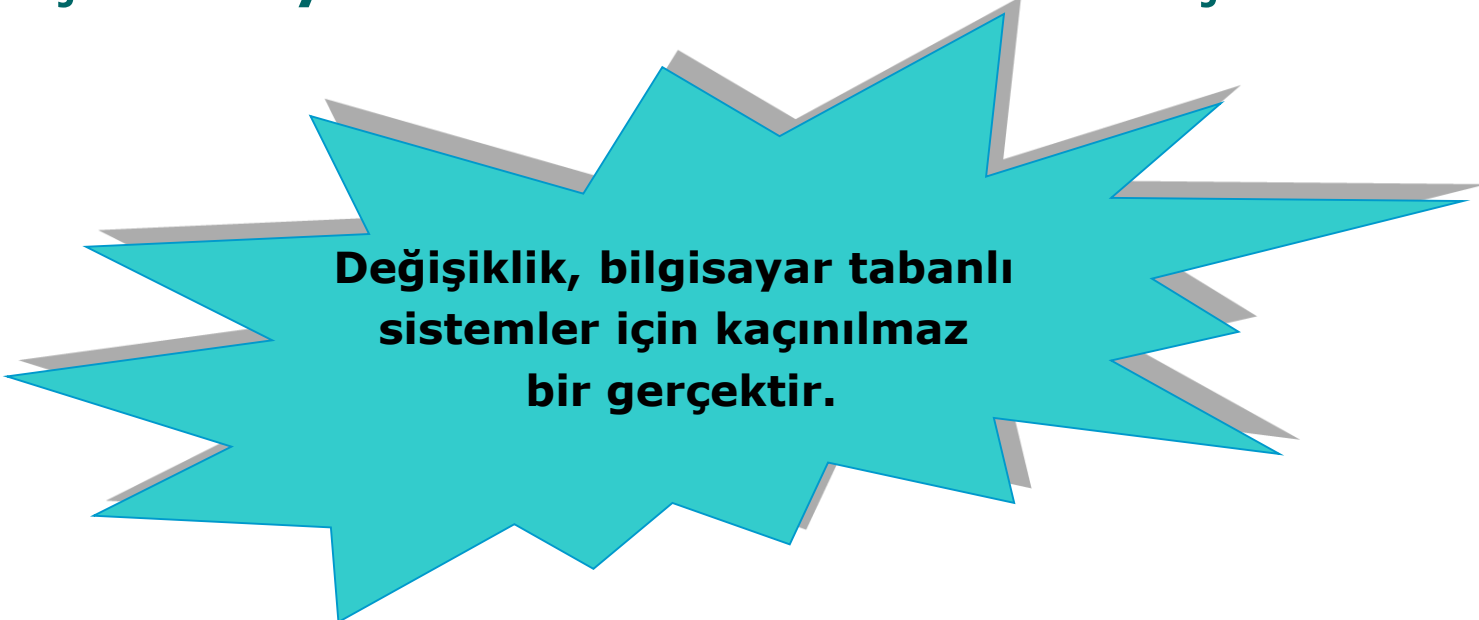




Zaman içinde ortaya çıkan yeni isterlerin karşılanması veya bulunan hataların giderilmesi çalışmaları yazılım bakımının temelini oluşturur.



Değişiklik, bilgisayar tabanlı sistemler için kaçınılmaz bir gerçektir.

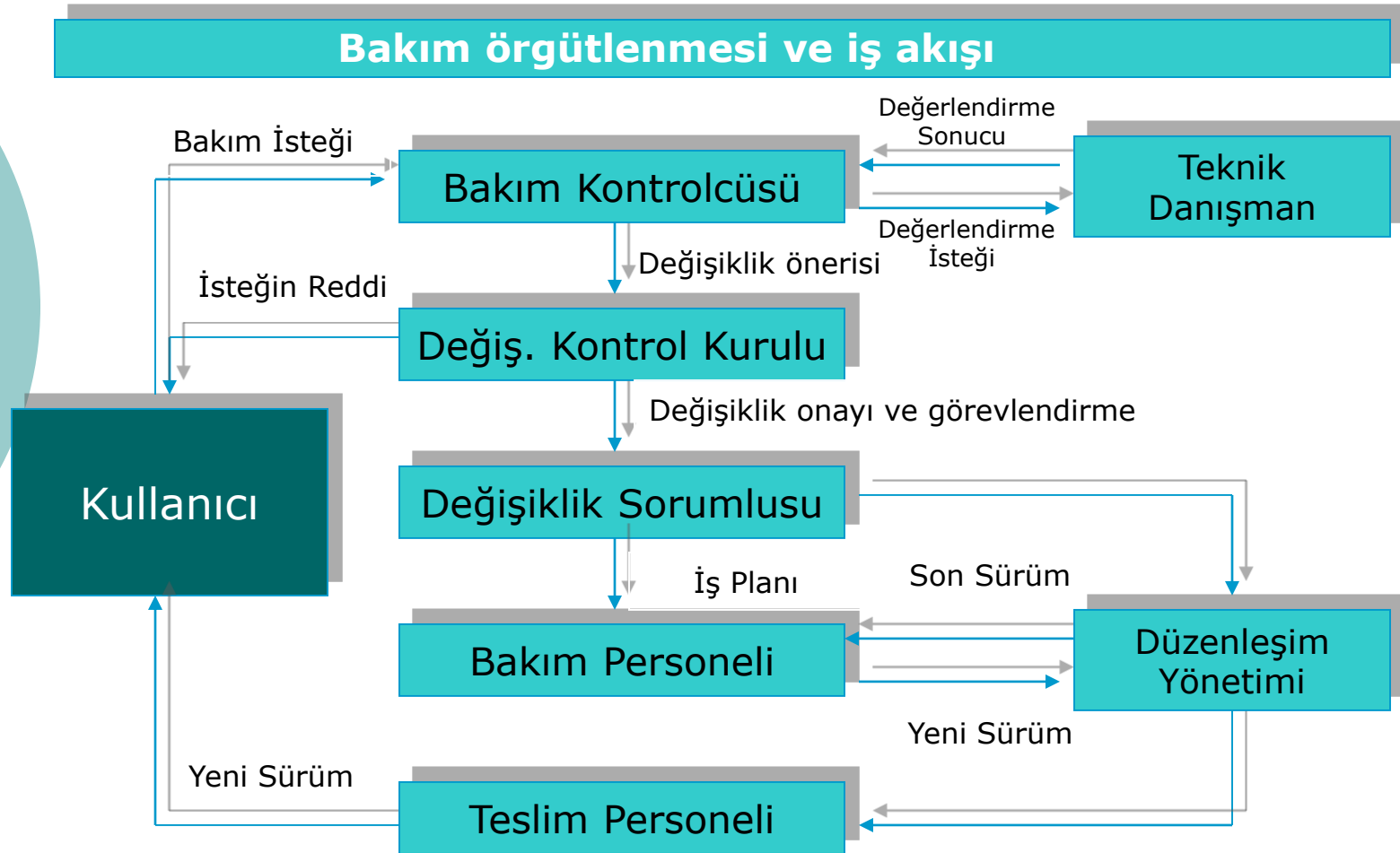
Değişiklikleri değerlendirmek , denetim altında tutmak ve uygulamak için bir düzen geliştirilmeli ve ona uyulmalıdır.



Bakım Türleri : Sadece hata düzeltimi değil teslim sonrası yapılacak işleri de kapsar.

1. **Düzeltilici bakım** (Bir Kısım yazılım kusuru ancak kullanım sırasında ortaya çıkar. Kusurun kaynaklanma sebebini araştırmaya ve gidermeye yönelik işlere denir.)
2. **Uyarlayıcı bakım** (Hızlı değişim ve teknolojik gelişmeler nedeniyle yazılımın yeni donanım , işletim sistemlerine , bunların yeni sürümlerine , yeni uç birimlere göre uyarlanması , sürümün yükseltilmesine güncelleştirilmesi işlerine uyarlayıcı bakım denir.)
3. **İyileştirici Bakım** (Yazılım geliştirip test edilerek başarılı bir şekilde kullanıcıya sunulduktan sonra , yeni işlevler eklemek , var olanlara başarımlı ve verimi artırıcı düzenlemeler yapmak iyileştirici bakım işleri denir.)
4. **Önleyici.** (Yazılımın gelecekte uygulanabilecek değişikliklere daha iyi bir temel oluşturması bakım özelliğinin ve güvenilirliğinin artırılması için ön tedbir niteliğindeki işlemler önleyici bakım işleri olarak tanımlanır.)

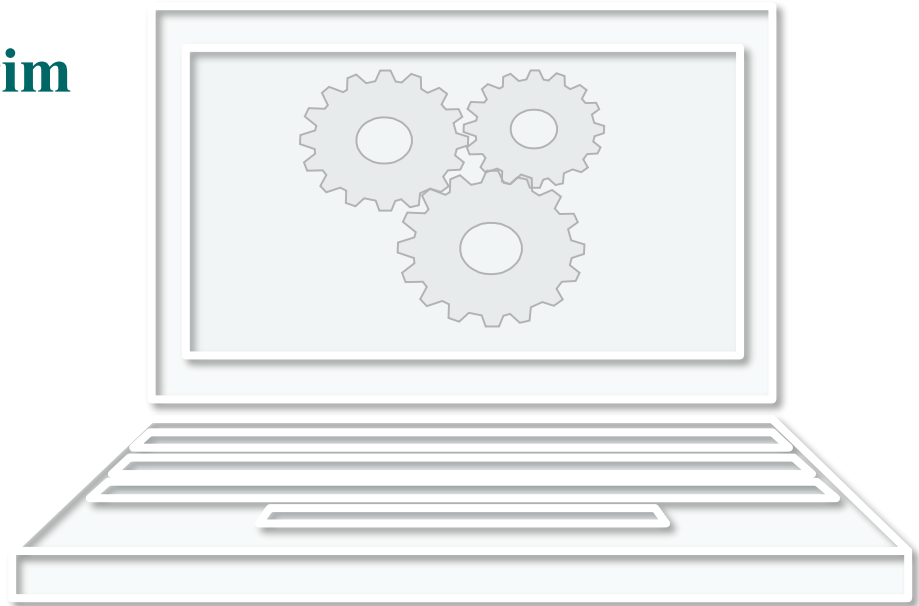
Genel olarak bakım işlemleri de belli bir düzen ve plan dahilinde yapılması gereklidir.



Bakım Aşamaları :

Yazılım geliştirme evresine benzer bu evrede standart bir süreç izlenmelidir.

1. **İsterlerin çözümlenmesi**
2. **Tasarım**
3. **Gerçekleştirim**
4. **Test**
5. **Teslim**



Raporlama : Kullanıcı , yazılımda ne tür değişiklik arzu ettiğini açıklayan bir yazılı öneri vermelidir.

Kullanıcının isteğini karşılamak üzere tespit edilmiş, iyileştirme amacıyla verilen bir değişiklik öneri raporu hazırlanmalıdır.

Herhangi bir yazılım veya değişiklik raporunu en az şu bilgileri içermelidir.

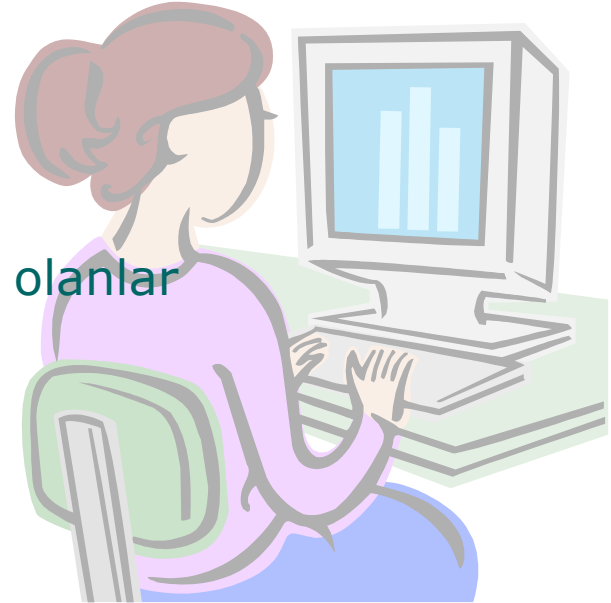
1. **Sistem veya alt sistem adı, öge adı**
2. **Sorunun açık tanımı**
3. **Hatalı duruma düşülmesine neden olan giriş verileri**
4. **Birimin içinde bulunduğu hatalı durum bilgisi**
5. **Varsa raporlama sisteminin yarattığı kayıtlar(log data)**
6. **Sorunu rapor eden kişiye ait erişim bilgileri**
7. **Sorunun rapor numarası**
8. **Sorunu daha ayrıntılı inceleyen mühendislere ait erişim bilgileri**
9. **Raporu inceleyen makamların imzaları ve tarihler**
10. **Karar(inceleme sonunda oluşturulur)**

Teknik olarak her türlü değişiklik istemini gerçekleştirmek mümkündür .

Ancak, önemli olan, bunu en düşük maliyetle , en kısa zamanda, doğru olarak ve yazılımın niteliklerini bozmadan yapabilmektir.

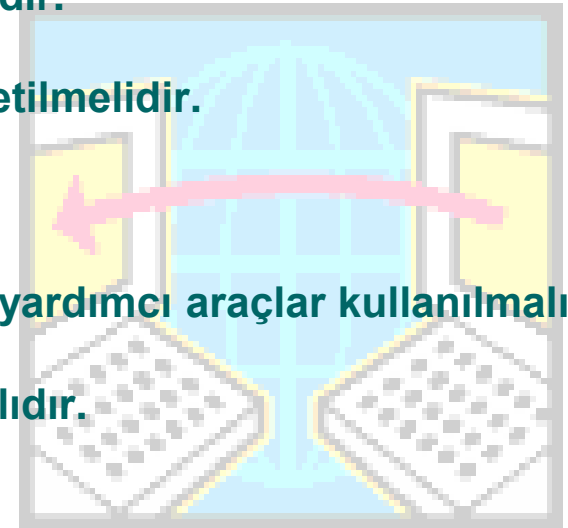
Denetim Etmenleri :

- Geliştirme Ortamına Bağlı olanlar
- Personele bağlı olanlar
- Müşteriye Bağlı olanlar



Bakım Niteliği : Yazılım nitelikli bir bakım evresine sahip olmalıdır.

1. **Bakım istekleri her zaman resmi bir değişim denetim süreci kapsamında ele alınmalıdır.**
2. **Yeterince hızlı olmalıdır.**
3. **İşgücü gereksinimi, süre ve maliyet kestirimi yapılabilmelidir. Öz kaynaklar etkin bir şekilde planlanmalıdır.**
4. **Bakım istekleri sistematik olarak izlenmelidir.**
5. **İstekler resmi olarak düzenlenmeli ve yönetilmelidir.**
6. **Standartlara uyulmalıdır.**
7. **Verimliliği artırıcı ve bakımı kolaylaştırıcı yardımcı araçlar kullanılmalıdır.**
8. **Uygun nitelikte teknik personel kullanılmalıdır.**



Niceliksel Ölçümler :

Bakım kolaylığı , nitelik ve güvenilirlik gibi ölçülmesi oldukça zor olan bir özelliktir.

1. Sorunun belirlenip rapor edilme süresi.
2. Sorun bildirimlerinin dönemsel yoğunluğu
3. İdari işlemlerde gecikme süresi
4. Sorunun gecikme süresi
5. Değişikliğin tanımlanması için gereken süre
6. Sorun bildirimlerinin müşterilere göre dağılımı
7. Sorun bildirimlerinin hata türlerine göre dağılımı
8. Sorun bildirimlerinin modüle göre dağılımı
9. Düzeltme için geçen zaman
10. Yerel test süresi
11. Kullanım yerindeki test süresi
12. Toplam süre
13. Toplam işgücü ve maliyet



Bakım Sorunları : Bakım yapılabilme özelliğinin eksikliği, bir disiplin içinde geliştirilmiş olmaması sorunların asıl kaynağıdır.

1. Çok fazla sürüm ortaya çıkması , bakımıda o derece güçleştirir.
2. Yazılım geliştirme sürecini aynen takip etmek zaman ve işgücü açısından çoğu zaman olanaksızdır.
3. Bir başkasının yazdığı kodu anlamak genellikle çok zaman alır. Eğer belgelendirme yanında kod içindeki açıklamalar da yetersiz ise çok ciddi sorunlar ortaya çıkar.
4. Belgelendirme yetersiz, eksik , hatalı olabilir yada hiç olmayabilir. Bu durumda yalnızca kodu okuyup anlamak gerekecektir.
5. İşlevsel olarak modüllere ayrılmamış yazılımda değişiklik yapmak yeniden geliştirmekten daha zordur.

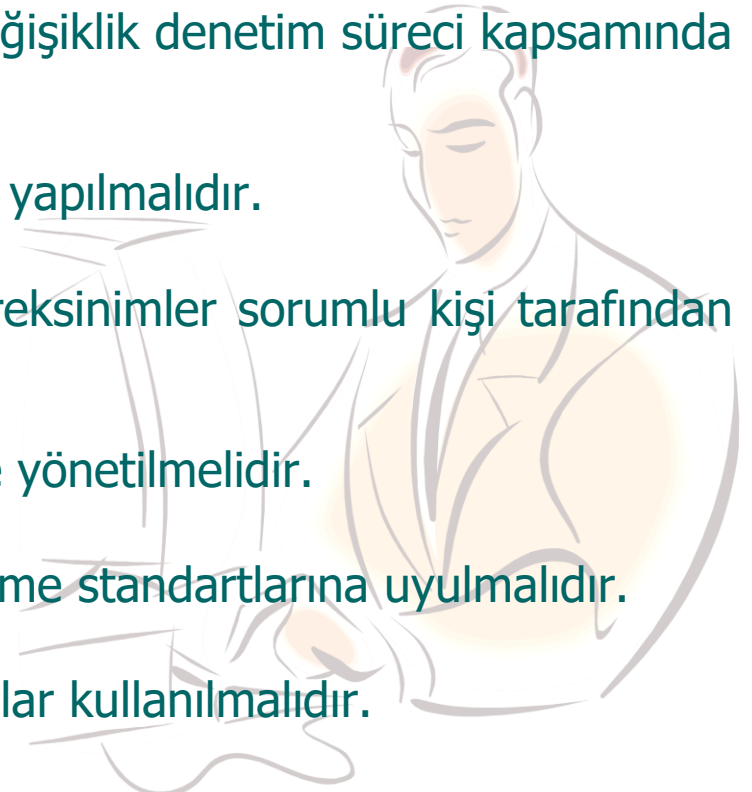
Bakımı kolaylaştırıcı bazı önlemler :

1. Bileşen bazlı geliştirme
2. Tasarımı bakım kolaylığı açısından gözden geçirme
3. Kaynak kodu bakım kolaylığı açısından gözden geçirme
4. Kurumsal kodlama standartlarına uyum
5. Ürünü geliştiren personeli bakım aşamasında da görevlendirme
6. Birim ve tümleştirme testleri için standart araçlar kullanma

Geliştirici İçin Kurallar :

Nitelikli yazılım için uygulanması gereken kurallar;

1. İstekler sistematik olarak izlenebilmelidir.
2. İstekler her zaman resmi bir değişiklik denetim süreci kapsamında ele alınmalıdır.
3. Süre ve maliyet kestirimi doğru yapılmalıdır.
4. Müşterinin ilettiği sorun ve gereksinimler sorumlu kişi tarafından toplanmalıdır.
5. Bakım projeleri tanımlanmalı ve yönetilmelidir.
6. Geliştirme sürecinin belgelendirme standartlarına uyulmalıdır.
7. Verimliliğe yardımcı olacak araçlar kullanılmalıdır.
8. Uygun nitelikte teknik personel kullanılmalıdır.



Test edilmiş ve kullanıma sunulmuş bir üründe yapılacak küçük bir değişiklik iyi çözümlenemez ve etkileri takip edilmez ise daha büyük problemlere yol açabilir.

- **Kodlamanın Etkilenmesi**

(Hata yapmaya açık olan yapılanmadır...)

- **Verilerin Etkilenmesi**

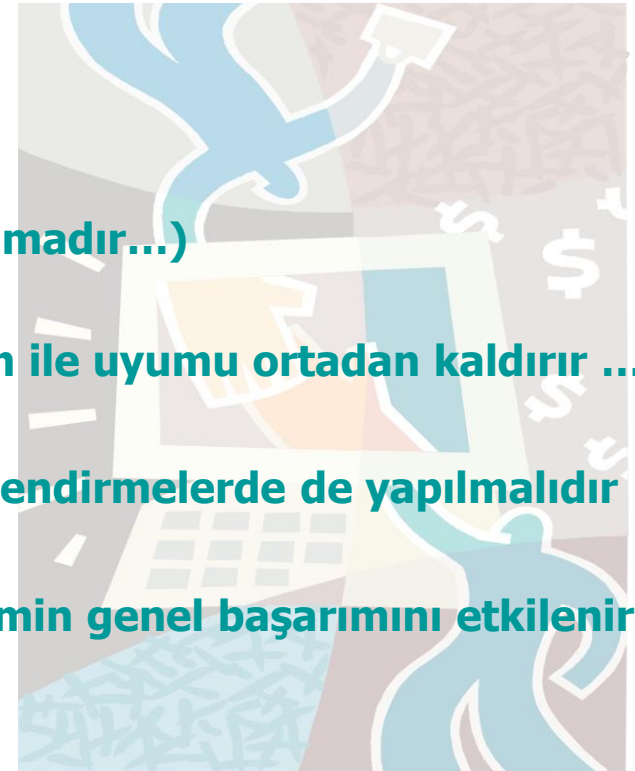
(Veri yapısının değişmesi tasarım ile uyumu ortadan kaldırır ...)

- **Belgelendirmenin Etkilenmesi**

(Kod içindeki değişiklikler belgelendirmelerde de yapılmalıdır ...)

- **Başarımın Etkilenmesi**

(işlevsellik bozulacağından sistemin genel başarımını etkilenir...)



Belgelendirilmemiş yazılımlar için uygulanacak bakım yöntemleri ;

1. Aynı kod üzerinde değişiklik yapılabilir
2. Kod başka ortamlara taşınabilir
3. Tersine mühendislik yapılabilir
4. Yeniden yapılanma

Temel yöntemler ;

1. **Her bir değişiklik isteği ayrı ayrı ele alıp tasarım ve kaynak kod içinde gereken değişiklikleri yapmak.**
2. **Köklü değişiklikler için yazılımın derinliklerini anlamaya çalışmak.**
3. **Değişiklik gerektiren kısımlarını yeniden tasarlamak.**
4. **Yazılımın tümünü tekrar tasarlamak ,kodlamak ve testlerden geçirmek.**

Dikkate alınması gereken riskler ;

1. **Geliştirme ve test ortamının bulunması**
2. **Personel devamlılığı**
3. **Personel deneyimi**
4. **Eski teknolojinin getirebileceği zorluklar**
5. **Müşteri hoşnutsuzluğu**
6. **Eski yazılıma değişiklik uygulamanın zorluğu**

Yazılım Bakımı

