**T.C.**

**FIRAT ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**Proje Dokümantasyonu**

Projedeki ifadeleri programda belirlenen kurallara uygun

Olup olmadığını kontrol etme

**Proje Ekibi**

**180541038-Hazal DİLER**

**2021 – ocak**

|  |
| --- |
|  |
| 1. **GİRİŞ** |
| * 1. Projenin Amacı   2. Projenin Kapsamı   3. Tanımlamalar ve Kısaltmalar |
| 1. **PROJE PLANI** |
| * 1. Giriş   2. Projenin Plan Kapsamı   3. Proje Zaman-İş Planı   4. Proje Ekip Yapısı   5. Önerilen Sistemin Teknik Tanımları   6. Kullanılan Özel Geliştirme Araçları ve Ortamları   7. Proje Standartları, Yöntem ve Metodolojiler   8. Kalite Sağlama Planı   9. Konfigürasyon Yönetim Planı   10. Kaynak Yönetim Planı   11. Eğitim Planı   12. Test Planı   13. Bakım Planı   14. Projede Kullanılan Yazılım/Donanım Araçlar |
| 1. **SİSTEM ÇÖZÜMLEME** |
| * 1. **Mevcut Sistem İncelemesi**      1. Örgüt Yapısı      2. İşlevsel Model      3. Veri Modeli      4. Varolan Yazılım/Donanım Kaynakları      5. Varolan Sistemin Değerlendirilmesi   2. **Gereksenen Sistemin Mantıksal Modeli**      1. Giriş      2. İşlevsel Model      3. Genel Bakış      4. Bilgi Sistemleri/Nesneler      5. Veri Modeli      6. Veri Sözlüğü      7. İşlevlerin Sıradüzeni      8. Başarım Gerekleri   3. **Arayüz (Modül) Gerekleri**      1. Yazılım Arayüzü      2. Kullanıcı Arayüzü      3. İletişim Arayüzü      4. Yönetim Arayüzü   4. **Belgeleme Gerekleri**      1. Geliştirme Sürecinin Belgelenmesi      2. Eğitim Belgeleri      3. Kullanıcı El Kitapları |
| 1. **SİSTEM TASARIMI** |
| * 1. **Genel Tasarım Bilgileri**       1. Genel Sistem Tanımı      2. Varsayımlar ve Kısıtlamalar      3. Sistem Mimarisi      4. Dış Arabirimler         1. Kullanıcı Arabirimleri         2. Veri Arabirimleri         3. Diğer Sistemlerle Arabirimler      5. Veri Modeli      6. Testler      7. Performans   2. **Veri Tasarımı**      1. Tablo tanımları      2. Tablo- İlişki Şemaları      3. Veri Tanımları      4. Değer Kümesi Tanımları   3. **Süreç Tasarımı**      1. Genel Tasarım      2. Modüller         1. XXX Modülü            1. İşlev            2. Kullanıcı Arabirimi            3. Modül Tanımı            4. Modül iç Tasarımı         2. YYY Modülü      3. Kullanıcı Profilleri      4. Entegrasyon ve Test Gereksinimleri   4. **Ortak Alt Sistemlerin Tasarımı**      1. Ortak Alt Sistemler      2. Modüller arası Ortak Veriler      3. Ortak Veriler İçin Veri Giriş ve Raporlama Modülleri      4. Güvenlik Altsistemi      5. Veri Dağıtım Altsistemi      6. Yedekleme ve Arşivleme İşlemleri |
| 1. **SİSTEM GERÇEKLEŞTİRİMİ** |
| * 1. Giriş   2. Yazılım Geliştirme Ortamları      1. Programlama Dilleri      2. Veri Tabanı Yönetim Sistemleri         1. VTYS Kullanımının Ek Yararları         2. Veri Modelleri         3. Şemalar         4. VTYS Mimarisi         5. Veritabanı Dilleri ve Arabirimleri         6. Veri Tabanı Sistem Ortamı         7. VTYS'nin Sınıflandırılması         8. Hazır Program Kütüphane Dosyaları         9. CASE Araç ve Ortamları   3. Kodlama Stili      1. Açıklama Satırları      2. Kod Biçimlemesi      3. Anlamlı İsimlendirme      4. Yapısal Programlama Yapıları   4. Program Karmaşıklığı      1. Programın Çizge Biçimine Dönüştürülmesi      2. McCabe Karmaşıklık Ölçütü Hesaplama   5. Olağan Dışı Durum Çözümleme      1. Olağandışı Durum Tanımları      2. Farklı Olağandışı Durum Çözümleme Yaklaşımları   6. Kod Gözden Geçirme      1. Gözden Geçirme Sürecinin Düzenlenmesi      2. Gözden Geçirme Sırasında Kullanılacak Sorular         1. Öbek Arayüzü         2. Giriş Açıklamaları         3. Veri Kullanımı         4. Öbeğin Düzenlenişi         5. Sunuş |
| 1. **DOĞRULAMA VE GEÇERLEME** |
| * 1. Giriş   2. Sınama Kavramları   3. Doğrulama ve Geçerleme Yaşam Döngüsü   4. Sınama Yöntemleri      1. Beyaz Kutu Sınaması      2. Temel Yollar Sınaması   5. Sınama ve Bütünleştirme Stratejileri      1. Yukarıdan Aşağı Sınama ve Bütünleştirme      2. Aşağıdan Yukarıya Sınama ve Bütünleştirme   6. Sınama Planlaması   7. Sınama Belirtimleri   8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri |
| 1. **BAKIM** |
| * 1. Giriş   2. Kurulum   3. Yerinde Destek Organizasyonu   4. Yazılım Bakımı      1. Tanım      2. Bakım Süreç Modeli |
| 1. SONUÇ |
| 1. KAYNAKLAR |

**2.GİRİŞ**

**2.1 PROJENİN AMACI**

* Kaynakların kaynak verme standartına uygunluğunu ölçme
* Kaynak, standartauygun değilse hatalar, raporlar bölümüne ekleyecek ve düzeltilmesini isteyecek. Program çalışmayacak
* Hatalar düzeltildikten sonra “istatistiki bilgi” tutulmalı
* Dokümanda kaç şekil var
* Çift tırnak arasında kaç tane kelime var
* Önsöz kısmında teşekkür var mı
* İçindekler kısmı gerçekten 3. Bölüm 71. Sayfada mı ?
* Şekillerin ve sayfaların o sayfada olup olmadığının kontrolü.
* Doküman sayısı 50’den büyükse rapor et
* Denklem numaralarının bölüm numaralarıyla ilişkilendirilmiş bir şekilde olup olmadığını kontrol et.

**2.2 PROJENİN KAPSAMI**

* Projenin kapsamı; kendi PDF dosyalarında yapılan hataları kolay bir şekilde bu program ile taratıp düzeltmek isteyen kullanıcılara kolaylık sağlar. Bu yüzden proje kapsamı geniş kitleleri ilgilendirir.

**3.PROJE PLANI**

Yazdığımız bu pdf’leri belirlenen kurallara göre tarayan program; web tarayıcısı üzerinden erişimi sağlanır. Böylelikle program maliyeti çok azdır.ayrıca bulut üzerinden erişim sağlandığı için depolama sıkıntısı da yaşanmaz.

**3.1 PROJE KAPSAMI**

Bu program yazım kurallarında sorun yaşayan insanlar için yapılmıştır web tarayıcısı üzerinden sınırsız erişim sağlayanır böylelikle depolama sıkıntısı ortadan kalkmış olacaktır. Aynı zamanda PDF’de bulunan kural dışı durumları o anda düzeltme şansı sunar.

**3.2 PROJE İŞ-ZAMAN ÇİZELGESİ**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **5. hafta** | **6.**  **hafta** | **7.**  **hafta** | **8.**  **hafta** | **9.**  **hafta** | **10.**  **hafta** | **11.**  **hafta** | **12.**  **hafta** | **13.**  **hafta** | **14.**  **hafta** |
| Proje teklifi | ✚ |  |  |  |  |  |  |  |  |  |
| Proje planı |  | ◕ | ✚ |  |  |  |  |  |  |  |
| Analiz |  |  | ◕ | ✚ |  |  |  |  |  |  |
| Sistem çözümleme |  |  |  |  | ◕ | ✚ |  |  |  |  |
| Kullanıcı arayüz tasarımı |  |  |  |  |  | ◕ | ✚ |  |  |  |
| Gerçekleştirim |  |  |  |  |  |  |  | ◕ | ✚ |  |
| Test |  |  |  |  |  |  |  |  | ✚ |  |
| **sunum** |  |  |  |  |  |  |  |  |  | **✚** |

**3.3 Proje Ekip Yapısı**

* Proje ekibi tek kişiden oluşmaktadır. Bütün yazılım yaşam döngüsünü tek kişi gerçekleştirir.
  1. **Kullanılan Özel Geliştirme Araçları ve Ortamları**
* Programın kodlarının yazımı NetBeans ortamında java dili ile gerçekleştirilecektir.

**3.6 Test Planı**

* Programı kullanabilmek için Gmail adresleri ile kayıt olma koşulu sunulacak. Program şikayet almadan haftada bir test edilecektir. Test kodlamayı yapan tarafından gerçekleştirilecektir. Hataya göre iyileştirme yapılıp program mükemmel bir hale getirilecektir.

**3.7 Bakım Planı**

Hata olmasını beklemeden site haftada bir test edilerek bakım uygulanacak. ve site daha kullanışlı bir hala gelmesi için kullanıcı yorumları da gözden geçirilecek

**4. SİSTEM ÇÖZÜMLEME**

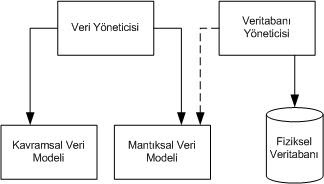
**4.1 Mevcut Sistemlerin İncelenmesi**

Mevcut sistemlerin incelenmesinde Adobe Acrobat göz önünde bulundurularak yapılmıştır ancak Adobe Acrobat yaptığımız uyguma ile bire bir uyuşmamaktır fakat benzerlik olarak yapacağımız uygulamaya benzeyen en uygun program olduğu için bu uygulama göz önüne alınmıştır.

**4.2 Varolan yazılım ve donanım kaynakları**

* Elektronik Sertifika Sağlayıcılar,
* Use case diyagramı,
* UML,
* Java programlama dili,
* Ana bilgisayar,
* kullanıcı bilgisayarı.

**4.3 Veri Modelleri**



* Fiziksel Veri Modelinde verilerin Mysql de tablolar içindeki alanlarda saklanacağı ve birbirleriyle ilişki içinde olduğunu söyleyebiliriz. Kavramsal Veri
* Üçlü şema mimarisinde görülen yapıların, kullanıcı gereksinimlerinden yola çıkılarak aşamalı bir şekilde fiziksel olarak gerçekleştirilmesidir.

* 1. Gereksinimlerin belirlenmesi - Veri tipleri - Veri grupları - Veriler ile ilgili kurallar - Veriler üzerinde yapılması gereken işlemler

* 2. Kavramsal Model Kullanıcıdan elde edilecek gereksinimler ile ilgili bir analiz çalışmasının yapılması ve birbiriyle bağlantılı verilerin gruplanarak bir düzenleme içinde modellenmesi gerekmektedir. Bu modeli grafiksel olarak varlık bağıntı seçenekleri ile gösteririz.

* 3. Mantıksal Model Veri tabanı tasarımlarımızın ilişkisel veritabanı modelinde tablolar ile ifade edilebilmesi için yapılması gereken dönüşümü içerir.

* 4. Fiziksel Model Fiziksel olarak sistemin kurulması sağlanır. Kullanılacak VTYS ile ilgili ilk temas burada kurulur.

**4.4 Kullanıcı Arayüzü**

* Projede kullanıcının arayüzü tasarlanırken herhangi bir şekilde renkler seçilerek tarafsız rahat büyük puntolu yazılı bir arayüz tasarlanacaktır.

**4.5 İletişim Arayüzü**

* İletişimle ilgili olarak sitede bir şikayet bölümü olacak ve bu şikayet bölümü sayesinde programcı ile kullanıcı arasında iletişim sağlanacak.

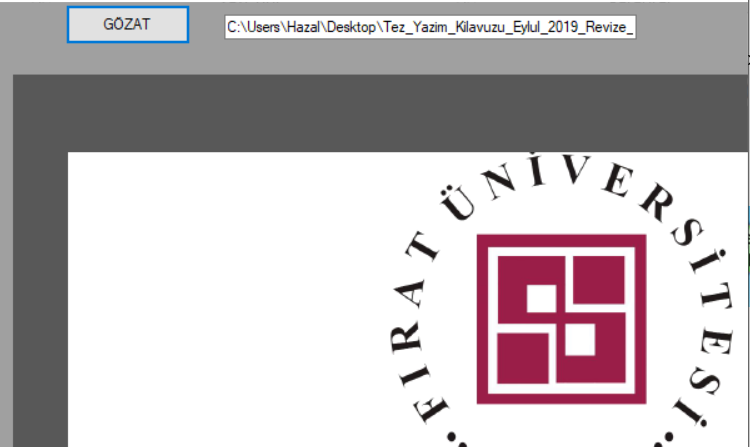
**4.6 Yönetim Arayüzü**

* Herhangi bir yönetimsel arayüz bulunmamaktadır.

1. **SİSTEM TASARIMI**

**\*GENEL TASARIM BİLGİLERİ**





**4.1 Testler**

Genel hatlarıyla testlerimiz iki aşamada gerçekleştirilecek. Bilinen adıyla pilot bölge uygulaması yapılacak.

**Alfa Aşaması**: Sistemin geliştirildiği yerde kullanıcıların gelerek katkıda bulunması sistemi test etmesi ile yapılacak.

**Beta Aşaması**: Kullanıcı, geliştirilen sistemi kendi yerleşkesinde, bir gözetmen eşliğinde yapılacak.

**4.2 Performans**

Sistemin performansını etkileyen faktörlerin test verileri değerlendirilecek Sistemin Tasarıma Uygunluk Performansı; Tasarımı yapılan sistemin stabilitesi ve işleyiş performansı değerlendirilecek. Veri Yapısının Sistemle Performansı; Veri yapısının sistemle stabilitesi ve çalışma zamanındaki uyumluluk düzeyindeki performansı değerlendirilecek.

**TABLO İLİŞKİ ŞEMALARI**

**4.3 Kullanıcı Profilleri**

Kullanıcı profili sadece kullanıcılardan oluşur herhangi bir yöneticiye gerek yoktur.

**4.4 Ortak Alt Sistemlerin Tasarımı**

**4.4.1 Ortak Alt Sistemler**

Kullanılacak tek ortak sistem devletin mevcut sistemidir bu sistemle müşteri numarası bazında kişinin tüm bilgisi alınacaktır.

**4.4.2 Modüller arası Ortak Veriler**

Modüller arasında ilişkili veri modelinde olduğu üzere ortak veriler mevcuttur. Bunlar müşteri no, Görevlendirme, Görev ve sonuç birbirleriyle ilişkili ortak verilerdir.

**4.4.3 Ortak Veriler İçin Veri Giriş ve Raporlama**

Modülleri Ortak veriler için böyle bir modül kullanılmadı .

**4.4.4 Güvenlik Altsistemi**

Yazılım sistemlerinin güvenilirliğe ilişkin nicelikleri, kullanıcıların gereksinimlerini karşılayacak şekilde ortaya koymak ve güvenilirliğin hesaplanmasına yönelik verileri toplama, istatistiksel tahminleme, ölçütlerin tespiti, yazılıma ait mimari özelliklerin belirlenmesi, tasarım, geliştirme ve bunlara yönelik çalışma ortamının belirlenmesi ve modellenmesini kapsamaktadır.

1. Model seçme ve düzenlemeye yönelik faaliyetlerin temelinde uygun hedeflerin tespit edilmesi bulunmaktadır.

2. Hata ve aksaklıkların analiz edilmesi için uygun verilerin tanımlanması gerekmektedir. Örneğin, arıza veya hataları önemine göre sınıflandırmak, hatalar arası ortalama süreyi bulmak, hata nedenlerini araştırmak, hataları bulmaya yönelik test verilerine karar vermek.

3. Belirtilen hedeflere yönelik veriler modellenir.

4. Geçmişe yönelik verilerin zaman bilgilerini de içerecek şekilde elde edilerek yazılım geliştirme sürecine dâhil etmek.

5. Yazılım geliştirme sürecinin modellenmesi, hata ile karşılaşılıp, test sürecine başlamak ve model doğrulama işlemlerine gerçekleştirmek.

6. Güvenilirlik tahminleme modelinin seçilmesini sağlamak.

7. Güvenilirlik modeli tarafından kullanılacak olan parametrelerini tespit etmek.

8. Verilen bir noktayı kullanarak gelecekteki olası hatalar hakkında tahmin yapmak.

9. Tahmin edilen hata ve arıza oranları ile gerçekleşen değerleri kıyaslamak.

1. **SİSTEM GERÇEKLEŞTİRİM**

Yazılım Geliştirme Ortamları Yazılım geliştirme ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalıştırılabilmesi için gerekli olan:

 Programlama Dili

 Veri Tabanı Yönetim Sistemi

 Hazır Program Kitapçıkları

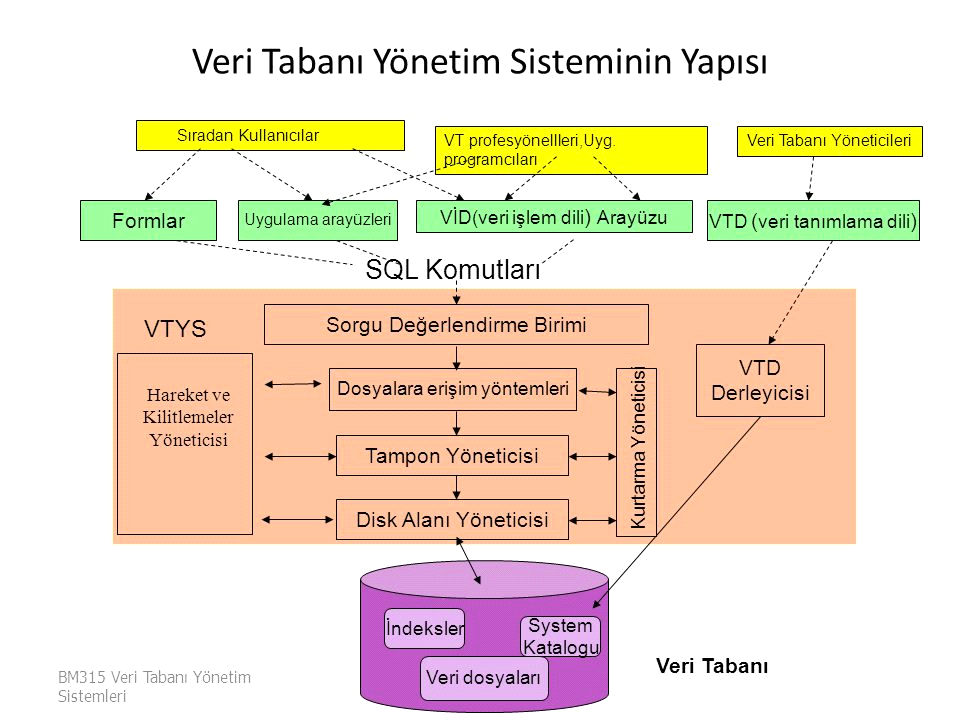
CASE Araçları belirlendi ve yazılım geliştirme ortamı hazırlandı.

**5.1 Programlama Dilleri :**

C# programlama dili ile gerçekleştirilmiştir.

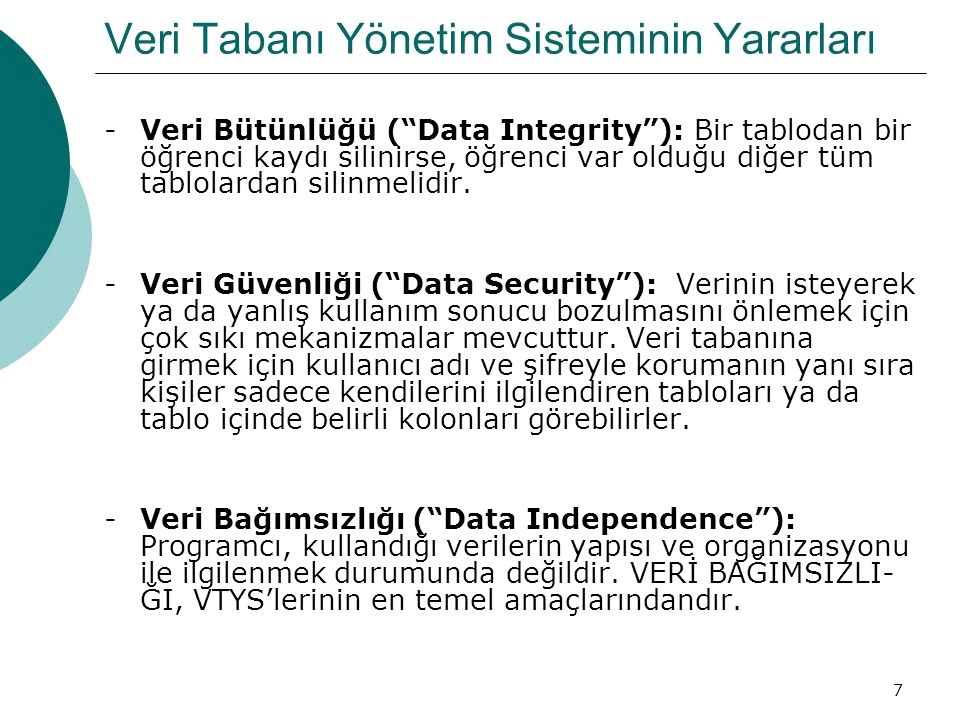
**5.2 Veri Tabanı Yönetim Sistemleri**

Flash ile hazırladığımız sistem ara yüz burada ki kullanıcı/programcının kullandığı uygulama kısmına denk geliyor. Xml sorgu ve programları işleyen yazılıma denk geliyor. Php ise veriye erişen yazılım oluyor. Veri tabanı tanımları ve Depolanmış veri tabanı ise SQL ile çözümleniyor. Veri tabanı yönetiminde kullandığımız hiyerarşiyi bir göstermek gerekirse:



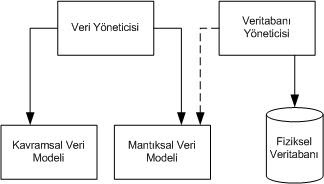
Veri Tabanı Sistemi

**5.3 VTYS Kullanımının Ek Yararları**



Şekil 5.3 VTYS nin Ek Yararları

**5.4 Veri Modelleri**



Şekil 5.4 Veri Modeli

Fiziksel Veri Modelinde verilerin Mysql de tablolar içindeki alanlarda saklanacağı ve birbirleriyle ilişki içinde olduğunu söyleyebiliriz. Kavramsal Veri

Üçlü şema mimarisinde görülen yapıların, kullanıcı gereksinimlerinden yola çıkılarak aşamalı bir şekilde fiziksel olarak gerçekleştirilmesidir.

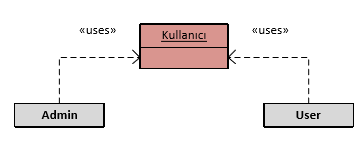
**1. Gereksinimlerin belirlenmesi - Veri tipleri - Veri grupları - Veriler ile ilgili kurallar - Veriler üzerinde yapılması gereken işlemler**

**2. Kavramsal Model Kullanıcıdan elde edilecek gereksinimler ile ilgili bir analiz çalışmasının yapılması ve birbiriyle bağlantılı verilerin gruplanarak bir düzenleme içinde modellenmesi gerekmektedir. Bu modeli grafiksel olarak varlık bağıntı seçenekleri ile gösteririz.**

**3. Mantıksal Model Veri tabanı tasarımlarımızın ilişkisel veritabanı modelinde tablolar ile ifade edilebilmesi için yapılması gereken dönüşümü içerir.**

**4. Fiziksel Model Fiziksel olarak sistemin kurulması sağlanır. Kullanılacak VTYS ile ilgili ilk temas burada kurulur.**

Kullanıcı Şemaları



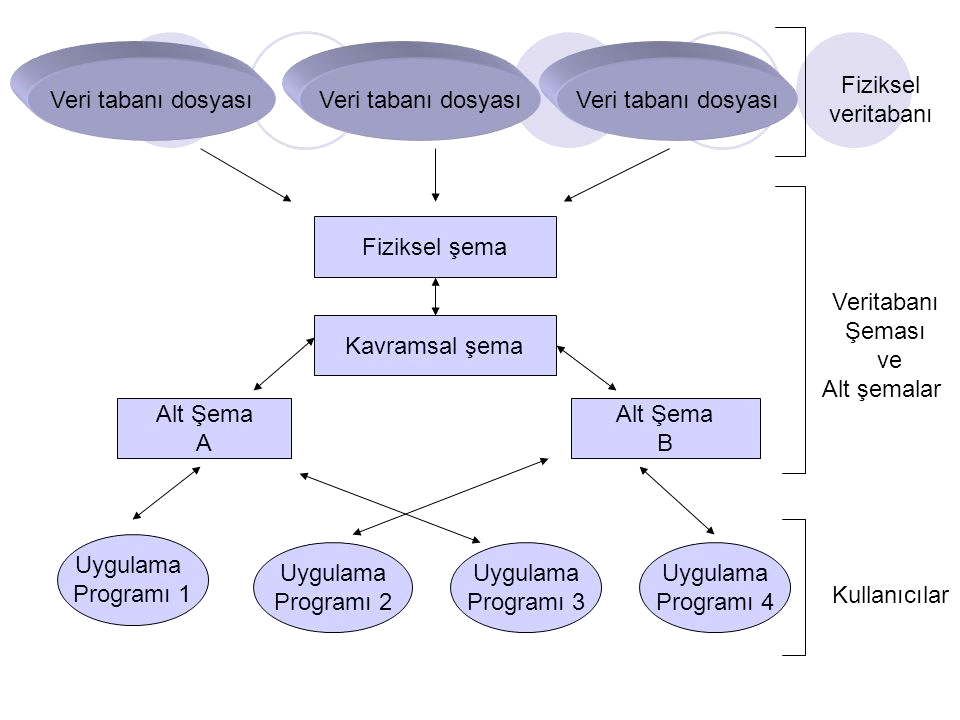
**5.5 Şemalar**

Herhangi bir veri modelinde veri tabanının tanımlanması ile kendisini ayırmak önemlidir. Veri tabanının tanımlamaları veri tabanı şeması veya meta-veri olarak adlandırılır. Veri tabanı

şeması, tasarım sırasında belirtilir ve sıkça değişmesi beklenmez. Pek çok veri modeli şemaları, diyagramlar halinde göstermek için belli gösterim biçimlerine sahiptir. Diyagramlar her kayıt tipinin yapısını gösterir fakat kaydın gerçek örneğini göstermez.

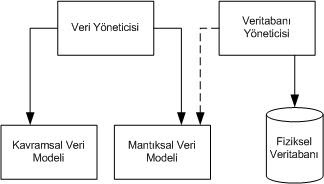
**5.6 VTYS Mimarisi**

VTYS mimarisini üç başlık altında ele aldık bunları özetlemek gerekir ise:

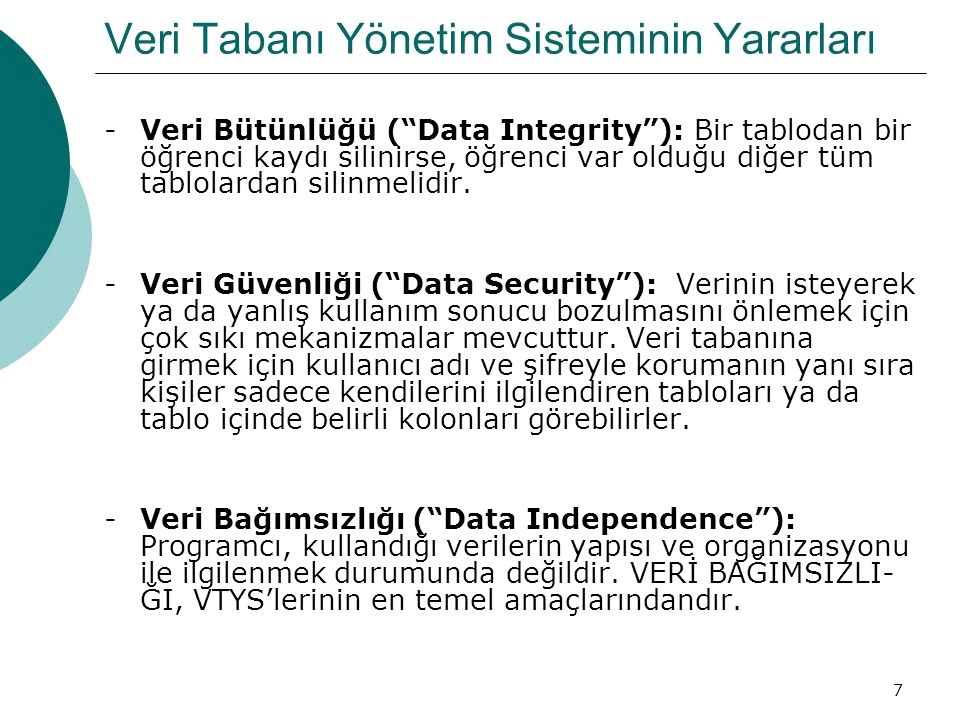


Şekil 5.6 Veri Tabanı Şemaları

**VTYS MİMARİSİ**

****

**5.7 Veritabanı Dilleri ve Arabirimleri**

****

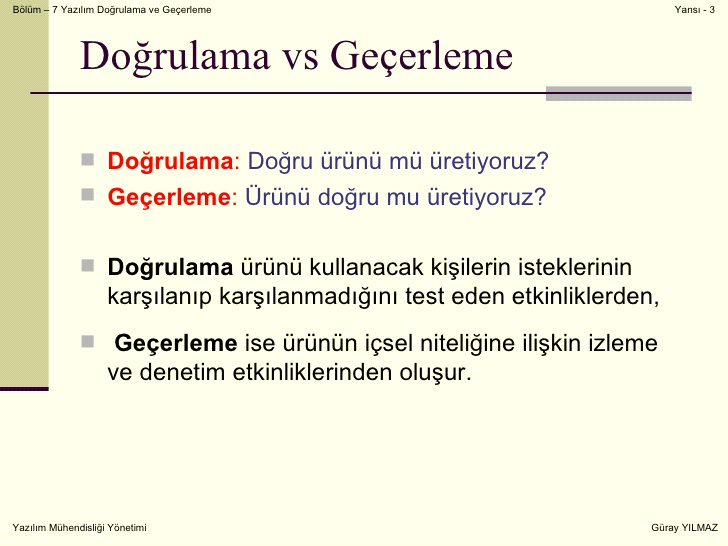
1. **DOĞRULAMA VE GEÇERLEME**

**6.1. Giriş**

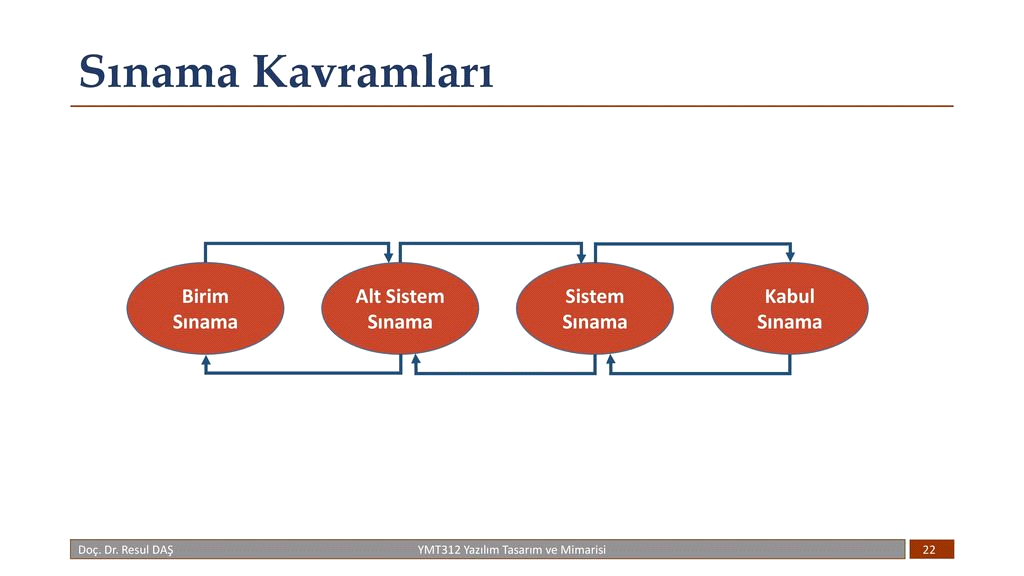
Geliştirilecek bilgi sistemi yazılımının doğrulanması ve geçerlenmesi, üretim süreci boyunca süren etkinliklerden oluşur. Söz konusu etkinlikler:

 Projenin bir aşaması süresince geliştirilen anahtar belirtimlerin önceki belirtimlerle karşılaştırılması.

Yazılım ürünlerinin tüm uygulanabilir gerekleri sağladığının gerçeklenmesi için sınamaların hazırlanıp yürütülmesi biçiminde özetlenebilir.

 Doğrulama ve geçerleme

**6.2. Sınama Kavramları**



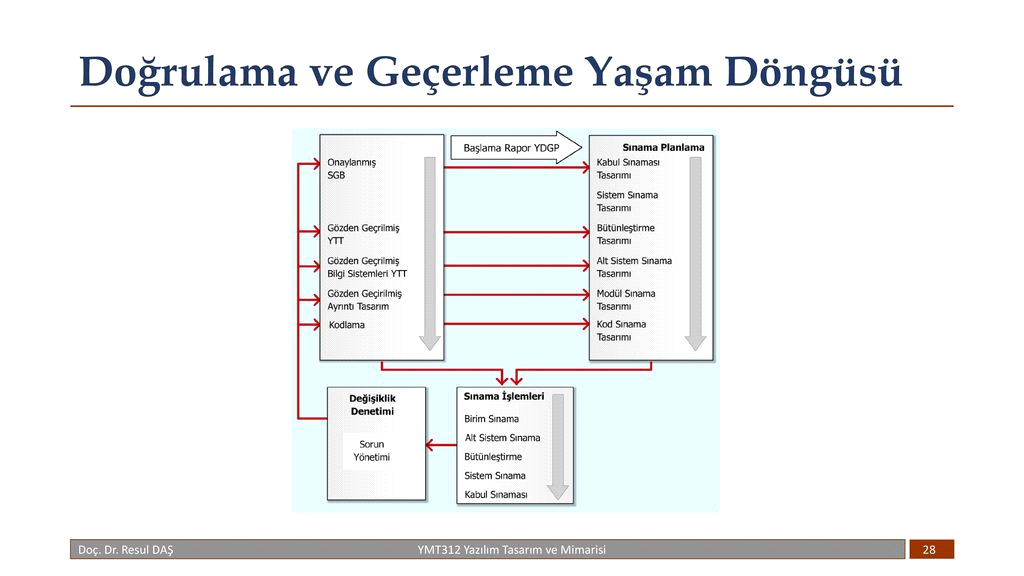
Birim Sınama: Sistemin birimleri olan YSK-YSK il-YSK İlçe-Sandık Kurulu-Seçmen sırasıyla kendi içlerinde birimleri sınandı ve sonuçları çıkartıldı.

Alt Sistem Sınama: Birimlerin birleşmesiyle modüller oluşturulup bunların kendi içinde sınaması yapıldı. Genel olarak arayüzde ki eksiklikler giderildi.

Sistem Sınama: Sistemin bütün olarak sınanması yapıldı ve programın eksiksiz olduğu onaylandı.

Kabul Sınama: Sistem prototipten çıkartılıp gerçek veriler girildi ve sorunsuz olduğu bir kez daha onaylandı.

**6.3. Doğrulama ve Geçerleme Yaşam Döngüsü**

****

**6.4. Sınama Yöntemleri**

Sınama işlemi, geliştirmeyi izleyen bir düzeltme görevi olmak ile sınırlı değildir. Bir "sonra" operasyonu olmaktan çok, geliştirme öncesinde planlanan ve tasarımı yapılması gereken bir çaba türüdür.

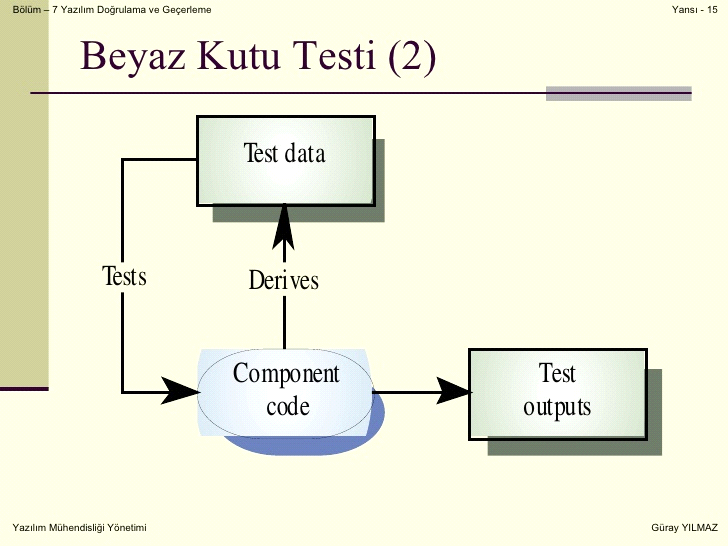
6.4.1 Beyaz Kutu Sınaması Denetimler arasında:

 Bütün bağımsız yolların en azından bir kere sınanması,

 Bütün mantıksal karar noktalarında iki değişik karar için sınamaların yapılması,

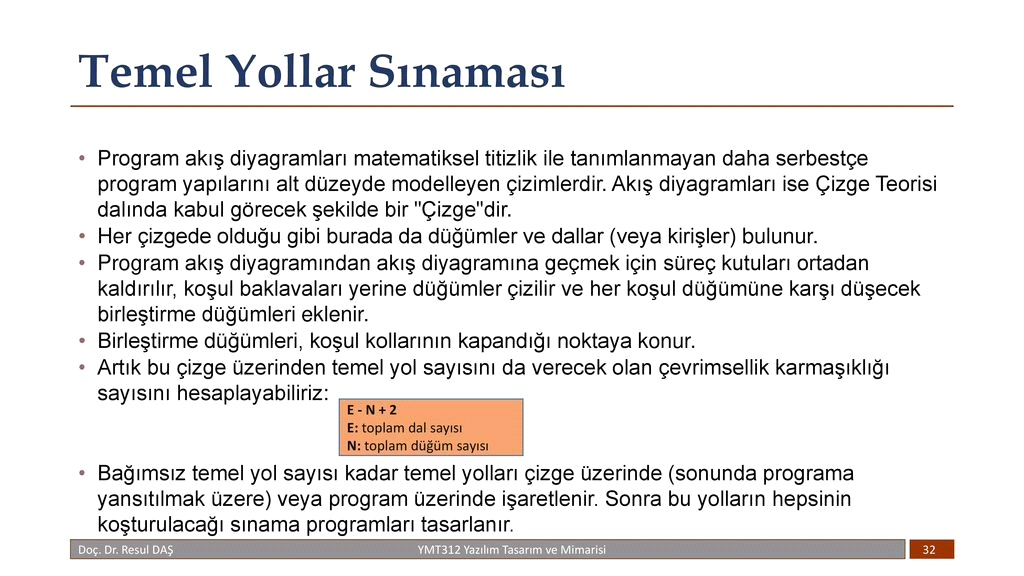
 Bütün döngülerin sınır değerlerinde sınanması,

 İç veri yapılarının denenmesi yapıldı.



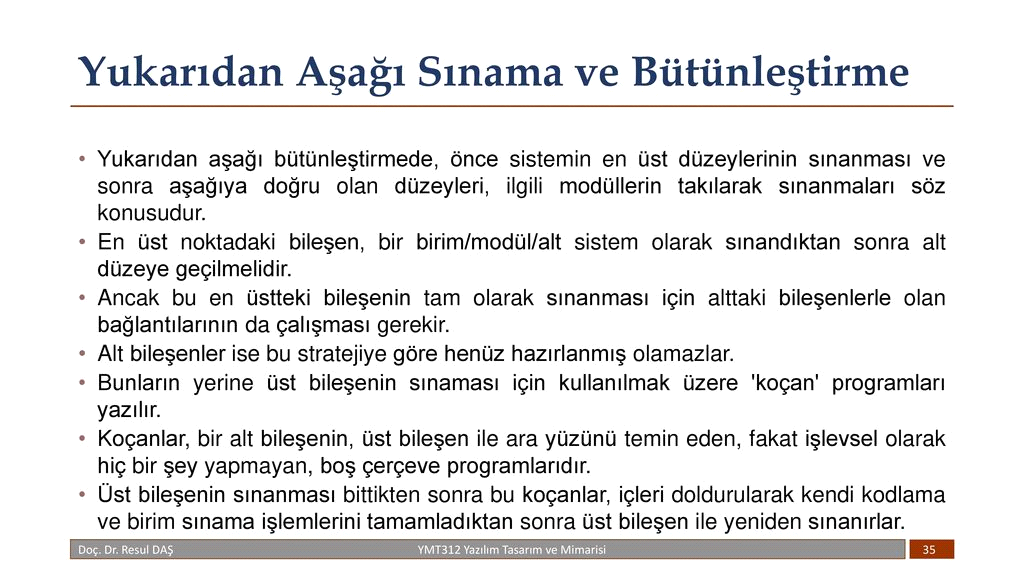
**beyaz kutu sınaması**

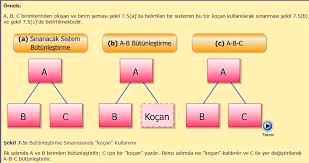
**6.4.2 Temel Yollar Sınaması**

****

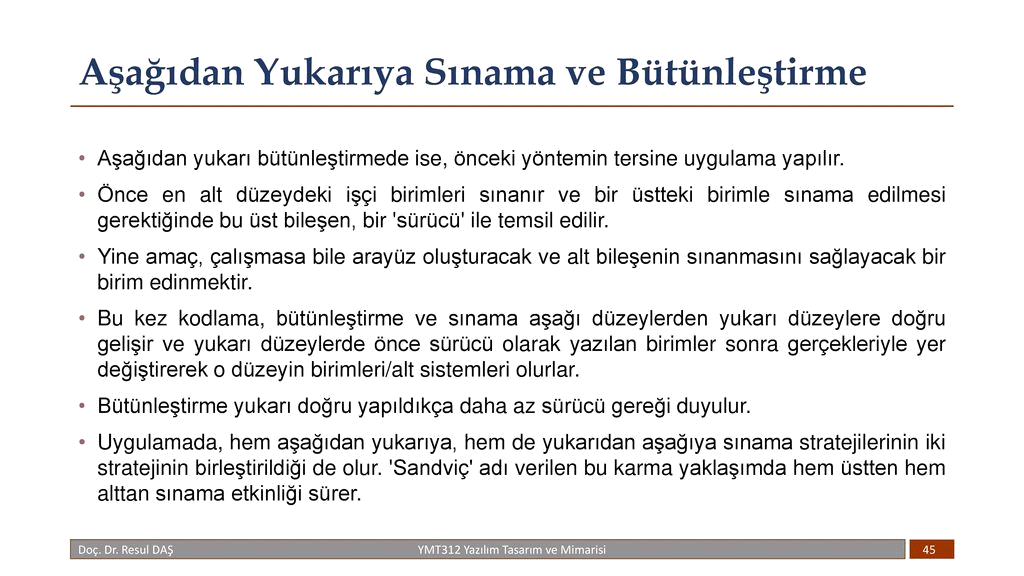
**temel yollar sınaması**

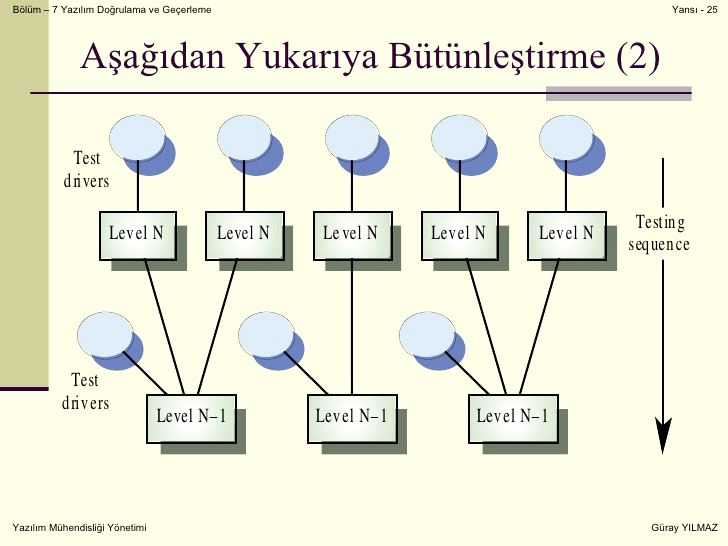
**6.4.3 yukarıdan aşağı sınama ve bütünleştirme**

****

****

**6.4.4 Aşağıdan Yukarıya Sınama ve Bütünleştirme**

****

****

**aşağıdan yukarıya sınama ve bütünleştirme**

**6.6. Sınama Planlaması**

Bir tablo ile özetlemek gerekirse şu şekilde özetleyebiliriz.

Test raporu hazırlanırken şu özellikler mutlaka planda belirtilmelidir;

Test planı kimliği: Test planının adı veya belge numarası

Giriş: Test edilecek yazılımın elemanlarının genel tanıtım özetleri. Ayrıca bu plan kapsamı ve başvurulan belgeler. Kısaltmalar ve terim açıklamaları bu bölümde bildirilmelidir.

Test edilecek sistem: Sistemde bileşenleri sürüm sayıları olarak sıralar ve sistemin özelliklerini bileşenlerini ve nasıl kullanıldıkları açıklanmalıdır. Ayrıca sistemde test edilmeyecek parçalar belirtilmelidir.

Test edilecek ana fonksiyonlar: Sistemin test edilecek ana fonksiyonlarının kısa bir tanıtımı yapılmalıdır.

Test edilmeyecek ana fonksiyonlar: Sistemde test edilmeyecek fonksiyonları ve bunların neden test edilmedikleri açıklanacaktır.

Geçti/Kaldı Kriterleri: Bir test sonucunda sistemin geçmiş veya kalmış sayılacağını açıklanmalıdır.

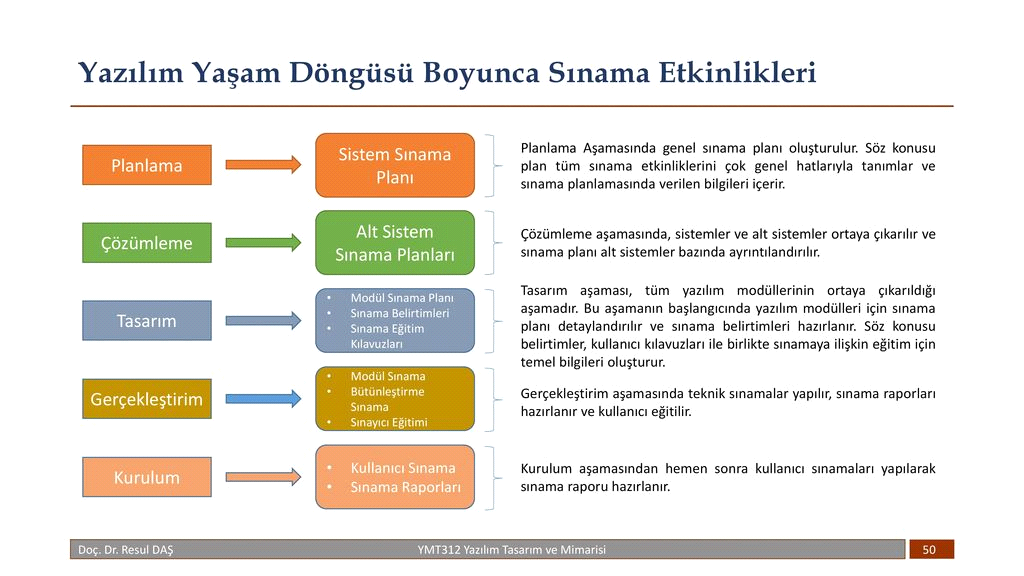
Test dokümanı: Test süresince yapılan işlemleri alınan raporları elde edilen bilgileri rapor içinde sunulmalıdır.

Sorumluluklar: Hangi kişilerin nelerden sorumlu olduğu ve test takım lideri bilgileri mutlaka raporda belirtilmelidir.

Riskler ve Önlemler: Test planında varsayılan ve olası yüksek riskli durumları belirtir ve bu durumların olması durumunda, etkilerinin en aza indirilebilmesi için alınması gereken

**6.7. Sınama Belirtimleri**

Sınama belirtimleri, bir sınama işleminin nasıl yapılacağına ilişkin ayrıntıları içerir.



sınama etkinlikleri

Sınama sırasında bulunan her hata için, değişiklik kontrol sistemine (DKS), "Yazılım Değişiklik İsteği" türünde bir kayıt girilir. Hatalar, DKS kayıtlarında aşağıdaki gibi gruplara ayrılabilir:

• Onulmaz Hatalar: BT projesinin gidişini bir ya da birden fazla aşama gerileten ya da düzeltilmesi mümkün olmayan hatalardır.

• Büyük Hatalar: Projenin kritik yolunu etkileyen ve önemli düzeltme gerektiren hatalardır.

• Küçük Hatalar: Projeyi engellemeyen, ve giderilmesi az çaba gerektiren hatalardır.

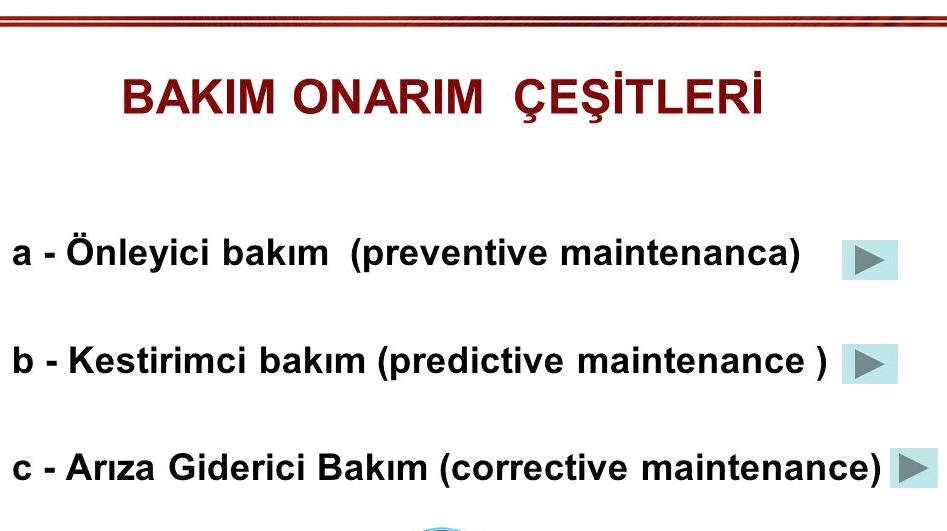
• Şekilsel Hatalar: Heceleme hatası gibi önemsiz hatalardır.

1. **BAKIM**

Klasik BT teknolojilerinde sistem odaları için kapsamlı bir bakım planı vardır. Bu bakım kurum ve kuruluşlar için iş gücü ve maliyet gerektirir. Ancak bulut teknolojilerinde hizmet servis sağlayıcı tarafından alındığı için böyle bir bakıma ihtiyaç yoktur. Kullanıcının sadece internete bağlı olması yeterlidir. Ayrıca servis olarak yazılım hizmeti alındığında kullanılan uygulamaların bakımları yine servis sağlayıcı tarafından yapılmaktadır.

Bakım aşamasına ilişkin yapılan açıklamalarda IEEE 1219-1998 standardı baz olarak alınmıştır.

Bakım çeşitleri



**Bakım Süreç Modeli**

Aslına bakmak gerekirse bakım süreç modeli yukardaki yapılan işlemlerin tümünün baştan yapılması demektir.

1. Adım: Sorunu Tanımlama Süreci İlk önce bakım ne için yapılıyor sorun ne buna bir bakalım.

2. Adım: Çözümleme Süreci Sorun tanımlamadan çıkan karar doğrultusunda problemi kâğıt üzerinde çözelim.

3. Adım: Tasarım Süreci Çözümlenen sistem sonucunda tasarımı güncelleştirmeye geldi sıra.

4. Adım: Gerçekleştirim Süreci Tasarımı yapılan sistemin gerçekleştirmesine sıra geldi.

5. Adım: Sistem Sınama Süreci Artık tekrardan tasarlanan sistemin sınama sürecini tekrar ele almak gerekiyor.

6. Adım: Kabul Sınaması Süreci Kendi içimizde sınadığımız sistemi birde müşteri karşısında sınıyoruz.

7. Adım: Kurulum Süreci Kabul sınamasını geçen sistemimiz artık tekrardan kurulum aşamasına geçiyor.