**T.C.**

**FIRAT ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**Proje Dokümantasyonu**

**Tez Kontrol Sistemi**

**Proje Ekibi**

**14545033-Fatih Güleç**

**1855541031-Tarık Kaya**

**Tarih – Versiyon**

|  |
| --- |
| 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 |

* **GİRİŞ**

**Projenin Adı: Tez Kontrol Sistemi**

**1.1** Projenin Amacı

Projemizin temel amacı yazılan tezleri belli kurallara göre kontrol etmek.

**1.2** Projenin Kapsamı

Projemizde .docx türündeki dosyaları giriş olarak alıp belirli kontrollerden geçirerek yazının doğru veya yanlış yazıldığını kontrol etmek.

**1.3** Tanımlamalar ve Kısaltmalar

**2. PROJE PLANI**

**2.1** Giriş

Yazılan tezlerin kontrolünü yapmak çok zaman alıcıdır. Geliştirdiğimiz sistem sayesinde .docx türündeki dosyayı sisteme yükleyerek kurallara uyulmuş mu? Hata var mı? Gibi birçok kontrolleri yaparak gerekli çıktıları üretir.

**2.2** Projenin Plan Kapsamı

Sisteme yüklenen dosyaları, istenilen kontrollere, kurallara göre algoritma kurarak bizi doğru çıktılara götürmesi planlanmaktadır.

**2.3** Proje Zaman-İş Planı

Başlangıç olarak planlanan sistemin tasarımının yapılması ve uygulanması gereklidir.

Ekip proje yapımına başlayacaktır.

Ekip sistemi kontrol edecek ve eksiklikleri, hataları tespit edecek daha sonra gerekli kişileri bilgilendirecektir.

Ekip mevcut sistemi siyah ve beyaz kutu testlerini gerçekleştiricektir.

**2.4** Proje Ekip Yapısı

Fatih Güleç

Tarık Kaya

**2.5** Önerilen Sistemin Teknik Tanımları

Verilen dosyayı masaüstü uygulamamız dosyanın içeriğini aşağıdaki önerilen sisteme göre yapacaktır:

* Windows işletim sistemi
* .Net Framework 4.6.1

**2.6** Kullanılan Özel Geliştirme Araçları ve Ortamları

* Visual Studio
* Microft Office uygulamaları

**2.7** Proje Standartları, Yöntem ve Metodolojiler

Projemizi Sprial modelle yapacağız.

Spiralin başladığı ilk çeyrek içindeki gerekler toplanılır alternatif belirlenir daha sonra karşılaşabilecek sorunlar değerlendirilir belirlenen hedefe ulaştıktan sonra geliştirmek için tartışılır bir sonraki ürün belirlenir. Aynı döngü maksimum performansa gelene kadar devam eder.

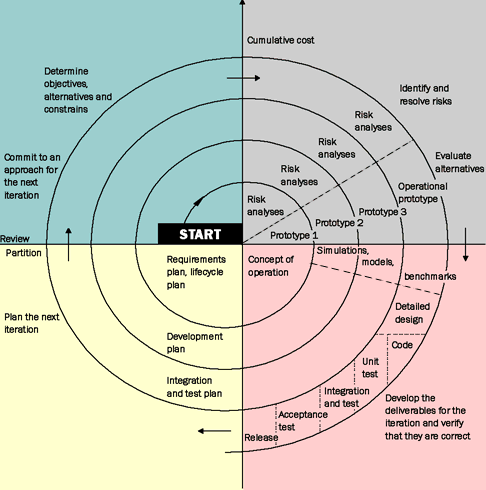
Sprial Model:

* + Amaç belirlenir
  + Alternatifler belirlenir
  + Alternatifler değerlendirilir
  + Risk analizi yapılır
  + Geliştirme yapılır
  + Bir sonraki ürün belirlenir
  + Bir sonraki aşama planlandırılır
  + Kullanıcı değerlendirmeleri alınır

NEDEN SPRİAL MODEL?

Üretim süreci boyunca ara ürün üretme ve üretilen ara ürünün kullanıcı tarafından sınanması temeline dayanır. Yazılımı kullanacak personelin sürece erken katılması ilerde oluşabilecek istenmeyen durumları engeller.

Gerek proje sahibi, gerekse yüklenici tarafındaki yöneticiler, çalışan yazılımlarla proje boyunca var olan sorunlarla karşılaştıkları için daha kolay izleme ve sorunları daha iyi analiz ederler.

****

**2.8** Kalite Sağlama Planı

Kalite planlama yazılım sürecinin erken bir evresinde başlanacaktır. Kalite planı istenen ürün kalitelerini ortaya koyar. Bunlara nasıl değer biçileceğini de tanımlamalıdır. Bu yüzden yüksek kaliteli yazılımın gerçekte ne anlama geldiğini tanımlar. Böyle bir tanım olmazsa, farklı mühendisler zıt yönlerde çalışabilirler böylece farklı ürün özellikleri en iyi duruma getirilmeye çalışılır. Kalite planlama sürecinin sonucu bir proje kalite planıdır.

* Ürün başlangıcı Ürünün, sunulacak pazarın ve ürün için kalite beklentilerinin tanımı
* Ürün planları Önemli sürüm tarihleri ve ürün sorumlulukları yanında dağıtım ve ürün bakım planları
* Süreç tanımlamaları Ürün geliştirme ve yönetimi için kullanılacak geliştirme ve bakım süreçleri
* Kalite hedefleri Önemli ürün kalite özelliklerinin de tanımlandığı ürün kalite hedef ve planları
* Riskler ve risk yönetimi Ürün kalitesini etkileyebilecek anahtar riskler ve bu riskleri karşılayan eylemler.

**2.9** Konfigürasyon Yönetim Planı

Gerekli yazılımsal ve donanımsal güncellemeri kullanıcı feedbackleri ve yazılım mühendislerimizin kaynak kodlu hataları tespiti yapılacak ve sistemimiz sürekli internet ağına bağlı olduğu için gerekli güncelleme hata görüldüğü takdirde düzeltilecektir. Gerekli konfigürasyon planı bu şekilde hazırlanmıştır.

**2.10** Kaynak Yönetim Planı

Projemiz için gerekli insan ve yazılım kaynakları tarafımızca araştırılmakta ve proje süresince gerekli analizler yapılarak oluşturulacaktır.

**2.11** Eğitim Planı

Ekip elamanlarıyla birlikte projeyi tam olarak anladıktan sonra ekibimizle işi ve işin nasıl yapacağını anlatarak, tartışarak en verimli hale getirmek için çalışmaktır.

Müşteriye programın nasıl kullanacağına dair bilgilendirme yapılacaktır.

**2.12** Test Planı

Oluşturulan tasarımın ekip tarafından kontrol edildikten sonra hataların ve eksiklerin giderilmesi, en iyi durumun sağlanması sistemden en iyi verim alana kadar bu döngü devam eder.

**2.13** Bakım Planı

Sistem üzerinden oluşturulacak hataların bakım ekibi tarafından düzeltilmesi ve duruma bağlı olarak sürekli kontrolünün yapılmasıdır.

* **SİSTEM ÇÖZÜMLEME**

**3.1** **Mevcut Sistem İncelemesi**

3.1.1 Örgüt Yapısı

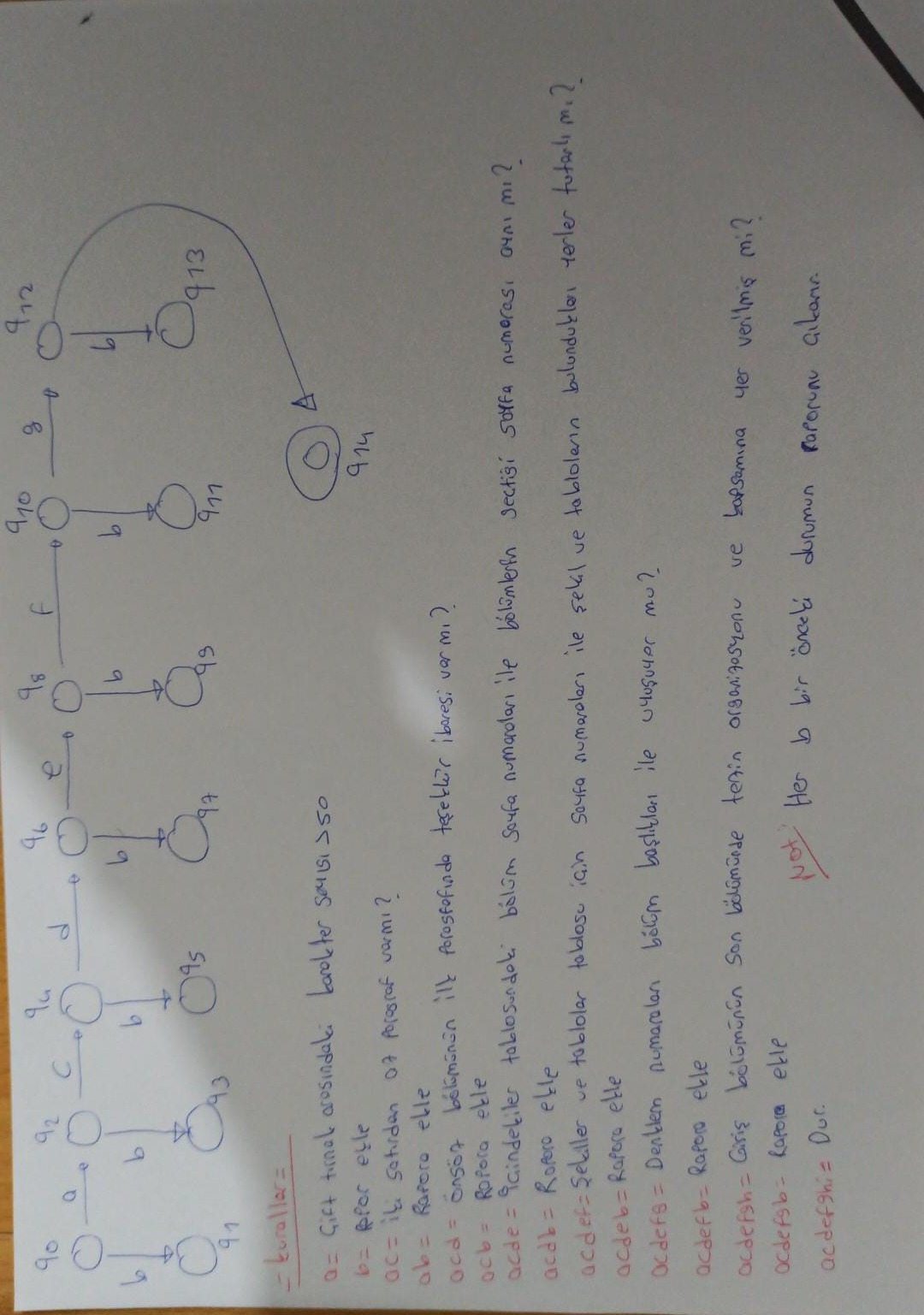
* Yönetici
* Yönetici yardımcısı
* Kullanıcı

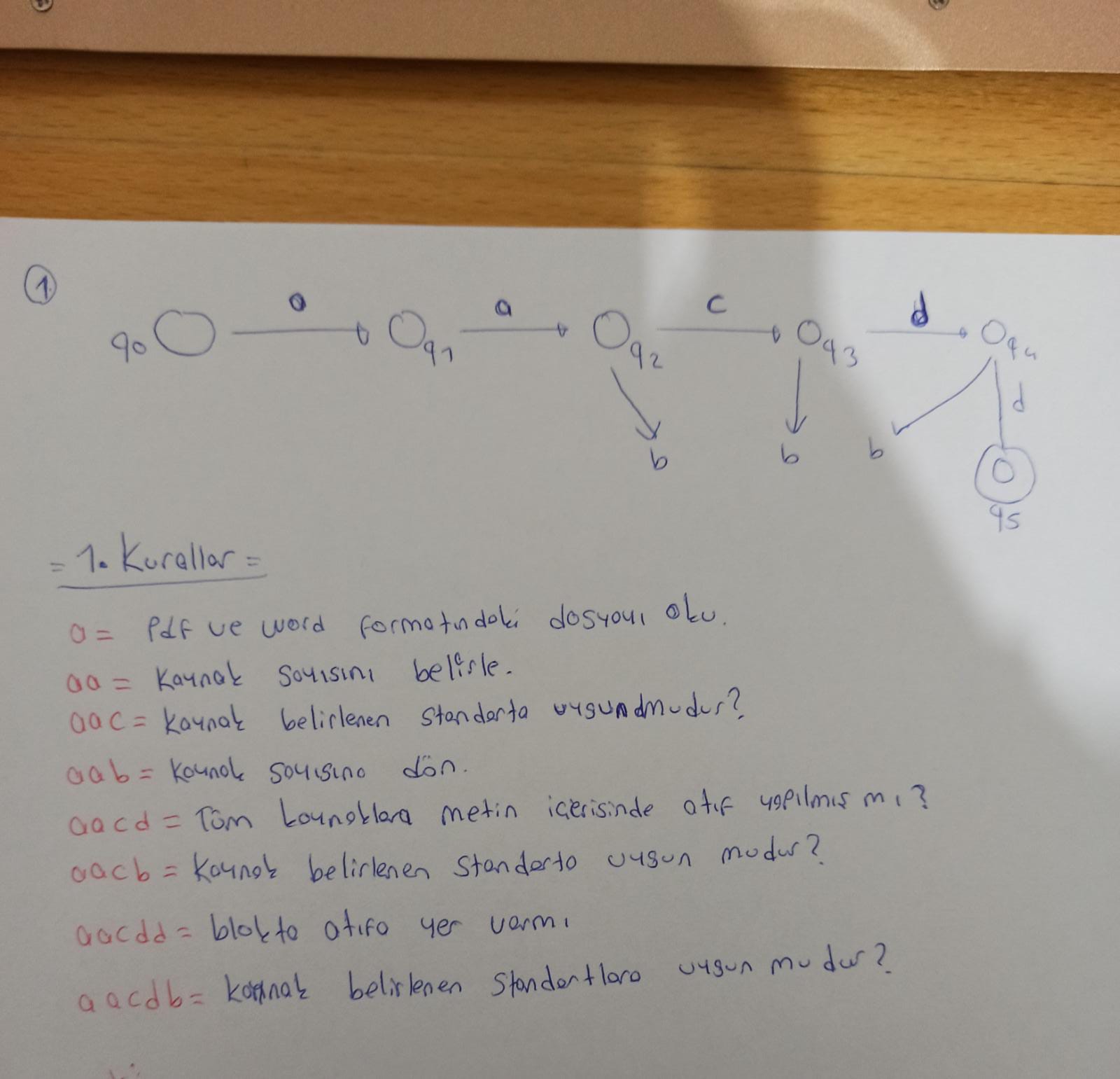
3.1.2 İşlevsel Model

* Kullanıcı sisteme giriş yapar
* Kullanıcı Word dosyasını sisteme yükler
* Kullanıcı rapor oluşturulmasını ister
* Sistem gerekli düzenlemeleri yapar
* Sistem raporu çıkarır

3.1.3 Veri Modeli

Deterministic finite automaton(DFA)





3.1.4 Var olan Yazılım/Donanım Kaynakları

* Windows işletim sistemi
* Visual Studio

3.2.7 İşlevlerin Sıradüzeni

• Word formatındaki dosyayı oku.

• Kaynak sayısını belirle.

• Kaynak belirtilen standarta uygun mudur?

• Tüm kaynaklara metin içinde atama yapılmış mı?

• Kaynak belirlenen standarta uygun mu?

• Blokta atıfa yer var mı?

• Kaynak belirlenen standarta uygun mu?

• Karakter sayısı>50 midir?

• İki satırdan az paragraf var mı?

• Önsöz bölümünün ilk paragrafında teşekkür ibaresi var mı?

• İçindekiler tablosundaki bölüm sayfa numaraları ile bölümlerin geçtiği sayfa numarası aynı mı?

• Şekiller ve tablolar tablosu için sayfa numaraları ile şekil ve tabloların bulundukları yerler tutarlı mıdır?

• Denklem numaraları bölüm başlıkları ile uyuşuyor mu?

• Giriş bölümünün son bölümünde tezin organizasyonu ve kapsamına yer verilmiş mi?

3.2.8 Başarım Gerekleri

* Sistem hızlı çalışmalı
* Ara yüz basit olmalı
* Personel ekip yapısı düzgün seçilmeli
* Personel eğitimi yapılmalı
* Güvenlik sorunları çıkmamalı
* Kullanıcılardan geri dönüş alınmalı.

**3.3** **Arayüz (Modül) Gerekleri**

3.3.1 Yazılım Arayüzü

Projenin yazılım arayüzü kaynak kodları tarafımızca kapalı kaynak kodu olarak belirlenmiştir.Bu yüzden projemiz tamamlandığında kodları yalnızca bizim ekbimiz tarafından düzenlenecektir.

3.3.2 Kullanıcı Ara yüzü

Kullanıcılar tasarladığımız sisteme gerekli dökümanı eklediğinde sistem belirlenen çıktıları verecektir.

3.3.3 İletişim Ara yüzü

Projeminiz bir iletişim arayüzü bulunmamaktadır.

3.4.1 Geliştirme Sürecinin Belgelenmesi

Geliştirme sürecimizin belgelenmesi hem ekibimizin yeni geliştiricilerinin projeyi hızlı tanıması,hemde geriye dönük hata çözümleri ve analizlerimiz için tarafımızca önemli olduğundan belgeleme süreci profosyonel şekilde yapılacaktır.

3.4.2 Eğitim Belgeleri

Projemizin anlaşılması için videolarla anlatım yaparız.Var olan sorunlara e-mail telefonla ulaşarak sorunları çözmek için uğraşacağız.

3.4.3 Kullanıcı El Kitapları

Kullanıcıya sistem açıklaması konferansla verilecektir.

**4.1 Genel Tasarım Bilgileri**

4.1.6 Testler

Yazılım Testi

Yazılım kodlama aşamasında programcı tarafından oluşabilecek hataları gidermek amacıyla düzenli bir şekilde yapılır.

Sistem testi

Oluşturulan tasarımın testerlar tarafından kontrol edildikten sonra hataların ve eksiklerin alanına uzman kişilere belirtirilmesi ve en iyi durumun sağlanması cihazıdan en iyi verim alana kadar bu döngü devam eder.

**4.3 Süreç Tasarımı**

4.3.1 Genel Tasarım

Oluşturulan uml diyagramından faydalınarak sistemdeki görsel ve işitsel bilgileri işlenmesi vasıtasıyla virtual reality ortamının oluşturulması.

4.3.2

Modüller

1. Yönetici modülü:

* Var olan sistemde yapılacak güncellemeler,
* Sistemin Testerlar tarafından test edilmesi

1. Kullanıcı modülü:

* Kullanıcıların görsel ve işitsel sohbet etmesi

4.3.3

Kullanıcı profilleri

Yönetici :kullanıcının hareketliliğini ve davranışlarını izler.

4.3.4

Entegrasyon ve test gereksinimleri

Varolan sistemde hataların,kodlardaki bozuklukların ve kullanıcının istekleri göz önünde bulundurarak yapılacak güncellemelerdir.

* **SİSTEM GERÇEKLEŞTİRİMİ**

**5.1.** Giriş

C# ile oluşturduğumuz formda kullanıcının girdiği Word dosyası üzerinde belirlediğimiz kurallar dahilinde rapor çıkartmaya yarayan programdır.

**5.2.** Yazılım Geliştirme Ortamları

5.2.1 Programlama Dilleri

* C#
* Visual Studio

**5.3.** Kodlama Stili

5.3.1 Açıklama Satırları

Projemizin her kodundan sonra açıklama satırı yazılması tarafımızca önemle belirtilmiştir.

5.3.2 Kod Biçimlemesi

Kod satırlarımızda yorum satırlarına dikkat edilmiştir.

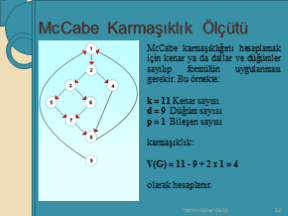
5.3.3 Anlamlı İsimlendirme

Veri Tiplerimiz anlamlı bir şekilde biçimlendirilmiştir.

5.3.4 Yapısal Programlama Yapıları

**5.4.** Program Karmaşıklığı

* + 1. McCabe Karmaşıklık Ölçütü Hesaplama



**5.5.** Olağan Dışı Durum Çözümleme

5.5.1 Olağandışı Durum Tanımları

Olağan dışı durumlarda hata mesajına yönlendiriyor.

5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları

0

**5.6.** Kod Gözden Geçirme

5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi

Kodlarımızı fonksiyon mantığıyla yaptığımız için gözden geçirme imkanımız kolaylaşıyor anında deneyerek hata sayısını minimuma indirgeriz.

5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular

5.6.2.2 Giriş Açıklamaları

Kodlarımızda yorum satırları sayesinde giriş açıklamları yapılmışır.

5.6.2.3 Veri Kullanımı

Verilen girdilere göre veri tipimiz belirlenmiştir. Yüklenen tezin boyutuna göre veri kullanımı belirlenir.

* **DOĞRULAMA VE GEÇERLEME**

6.1. Giriş

Doğrulama: Sistem kullanıcının eksiklerini karşılıyor,tek tek kontrol etmek yerine projemiz tarafından daha az zaman harcayarak gerçekleştirilir.

Geçerleme: Ürünün içsel niteliğine ilişkin izlendi, kontrol edildi ve denendi.

6.2. Sınama Kavramları

Birim Sınama : Var olan bilgiler tek tek sınanmıştır,en sonunda birleştirilerek ana sistemi oluşturmuştur.

Alt Sistem Sınama:Var olan sistem azdan çoğa-temelden bütüne gittiği için her birleştirme aşamasında kontrol edilmiştir.

Sistem Sınama: Birleştirilen sistemler bir bütün olarak son kez test edilmiştir.

6.3. Doğrulama ve Geçerleme Yaşam Döngüsü

Doğrulama ve geçerleme işlemleri yazılım üretim yaşam döngüsünün tüm süreçlerinde ve bu süreçlere koşut olarak sürdülürdü.

Gerçekleştirim aşamasına kadar olan süreçlerde doğrulama ve geçerleme işlemlerinin planlaması yapıldı.

Planlama, genel olarak, birim, alt sistem, bütünleştirme, sistem ve kabul sınamalarının tasarımlarını içerdi.

Gerçekleştirim aşamasının sonunda ise söz konusu planlar uygulandı.

Uygulama sonucu elde edilen bulgular, Yazılım Doğrulama ve Geçerleme raporları biçiminde sürekli olarak raporlandı.

Bu bilgiler, değişiklik denetim sistemi ve sorun yönetim sistemlerinde girdi olarak kullanılır.

Değişiklik Denetim sistemi, sınama süresince elde edilen bulguların izlenmesi amacıyla oluşturulan bir sistemdir.

6.4. Sınama Yöntemleri

6.4.1 Beyaz Kutu Sınaması

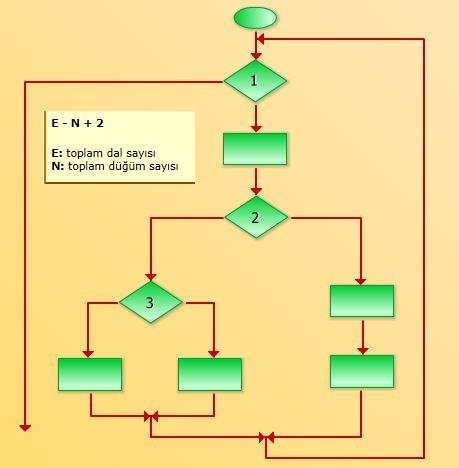
Yapılabilecek denetimler arasında:

Bütün bağımsız yolların en azından bir kere sınandı.

Bütün mantıksal karar noktalarında iki değişik karar için sınamaların yapıldı.

Bütün döngülerin sınır değerlerinde sınandı.

* + 1. Temel Yollar Sınaması



6.5.Sınama ve Bütünleştirme Stratejileri

6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme

Sistemimiz projemiz bittiğinde en üstten başlayarak sınanmıştır. Daha sonra alt sistemlerle var olan problem belirlenmemiştir.

* + 1. Aşağıdan Yukarıya Sınama ve Bütünleştirme

Var olan sistemimiz en alttan tek tek sınanarak bir bütüne doğru giderilerek bütünlük sağlanmıştır.

6.6. Sınama Planlaması

Bir plan güdümünde gerçekleştirildi.

Yazılım yaşam döngüsünün süreçlerine koşut olarak, farklı ayrıntı düzeylerinde birden fazla sınama planı hazırlanır.

Sınama planlarımız :

Birim (Modül) Sınama Planı

Alt Sistem Sınama Planları

Bütünleştirme Sınama Planlar

Kabul Sınama Planları

Sistem Sınama Planları biçimindedir. Her sınama planı, sınama etkinliklerinin sınırlarını, yaklaşımını, kaynaklarını ve zamanlamasını tanımlandı. Plan neyin sınanacağını, neyin sınanmayacağını, sorumlu kişileri ve riskleri göstermektedir. Sınama planları, sınama belirtimlerini içerir.

6.7. Sınama Belirtimleri

Sınanan program modülü ya da modüllerinin adları

Sınama türü

Stratejisi (beyaz kutu, temel yollar vb.)

Sınama verileri

Her sınama planı, sınama etkinliklerinin sınırlarını, yaklaşımını, kaynaklarını ve zamanlamasını tanımlamıştır. Sınama senaryoları olarak belirlenen her kriteri projemiz sağlamıştır.

**7. BAKIM**

7.1 Giriş

Projemizin en önemli kısımlarından biri bakım kısmıdır. Karşılaşılan sorun veya eksik bir durum söz konusu olunca müdahale edilmiştir. Yazılım bakım faaliyetleri yazılım yaşam döngüsünün en son yapılan faaliyeti olması açısından tüm diğer faaliyetler ile ilişkili, çeşitli uzmanlık alanlarını içeren ve müşteriyi 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 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.

7.2 Kurulum

Kurulum yapmayı bilmeyen biri için bile oldukça kolay bir şekilde tasarlanmıştır.Setup dosyamızı indirerek ileri,ileri,ileri,kur kısımlarına basarak kurulum sağlancaktır.

7.3 Yerinde Destek Organizasyonu

Böyle bir hizmetimiz şuan bulunmamaktadır.

* 1. Yazılım Bakımı

. Sürecin gerçekleştirilmesi

. Sorunun ve değişikliğin çözümlenmesi

. Bakımın gerçekleştirilmesi

. Bakımın değerlendirilmesi ve kabulü

. Geçiş

. Tamamlama

Yukarıda verilen adımlar sistemimizde uygulanmıştır.