



T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

FİNANSAL BİLGİ TEKNOLOJİLERİ

Analiz Dökümanı

Ödevi Hazırlayanlar:

Cem ÇAKIRCA B161210004

Ali BİNGÖL B161210382

Ömer Yaşar KORKMAZ B161210019

Bilal Turhan MAHRABEL U111210018

SAKARYA

Ocak, 2021

Gereksinim Analizi Dokümanı Hazırlama

1. Nedir? Projenin başarıya ulaşabilmesi için yapılması gereken en önemli iştir. Doğru şekilde isterleri ortaya konmuş projeler ancak başarılı olabilirler. İki aşamadan meydana gelir; Gereksinimlerin ortaya çıkarılması. Gereksinimlerin öncelik durumuna göre analiz edilmesi
2. Yöntemleri nelerdir? Yazılım öncesi problemin nasıl çözüldüğünün anlaşılması. Mevcut yazılımların incelenmesi ve varsa eksiklerin not edilmesi. Mevcut yazılımların çıktılarının elde edilmesi ve analiz edilmesi. İlgili sektör çalışanlarıyla yüz yüze görüşme. Olası sistem senaryolarının akıl yürütme yoluyla açığa çıkarılması. Müşteri ile veya potansiyel müşteri ile karşılıklı görüşme. Konuyla ilgili yapılmış tez, makale v.s. inceleme.
3. Gereksinim analiz dokümanı (GAD). Gerçeklenecek sistemin amaç ve kapsamından başlayıp, fonksiyonel ve fonksiyonel olmayan gereksinimlerinin açığa çıkarıldığı, aktörler, olaylar, senaryolar ve arayüz detayına kadar projenin irdelendiği temel dokümandır.
4. GAD bölümleri. Giriş. Teklif edilen system. Fonksiyonel gereksinimler. Fonksiyonel olmayan gereksinimler. Sistem modelleri
5. Giriş bölümü. Temel olarak üç bölümden oluşur; Amaç ve kapsam. Hedefler ve başarı kriterleri
6. Amaç ve kapsam: Projenin başlatılma gerekçesi bu kısımda yazılır. Örnek bir amaç ifadesi şu şekilde olabilir. “Projemizin amacı, okulumuzda öğrenim görmekte olan öğrencilerin kayıt ve işlem verilerini kolayca saklayabilecek, hızlıca öğrenciye erişim imkanı sağlayacak, uzaktan öğrenci kayıt işlemine imkan verecek, not girişleri ve düzeltmelerinin kolayca yapılabilmesi bilgisayar destekli bir sistemin ortaya konmasıdır.” Kapsam ifadesi ise genellikle projede bir budamayı ifade eder. Örneğin; “Proje kapsamında dersliklerin takibi ve haftalık ders programı gibi detaylar yer almayacaktır.”
7. Amaç ve kapsam nasıl olmalı? Projenin amaç ve kapsamı genel hatlarıyla projenin ne yaptığını ifade etmelidir. Bütün detayların verilmesi okuyucuyu sıkacağı için öne çıkan konular bu kısımda verilmelidir. Özgün olduğunu düşündüğünüz (diğer projelerde olmayan) konular mutlaka bu kısımda verilmelidir.
8. Hedefler ve başarı kriterleri: Projenin amacı kısmında verilen ifadelerin bir miktar daha açılarak yazılmasıyla hedefler ortaya çıkar. Hedefler maddeler halinde ve mümkünse ya geliştirilme sırasına göre ya da önem sırasına göre verilmelidir. Örneğin; Projemizin öncelikli hedeflerinden birisi kesintisiz çalışma olanağı sağlamasıdır. Uzaktan ders seçimi yapılabilmelidir. Öğrenciler online olarak anketlere katılabilmelidir. Eğitim kalitesi için bu sistem geri bildirimleri uygun şekilde desteklemelidir.
9. Başarı kriterleri: Projenin başarılı sayılabilmesi için gereken şartlar bu kısımda verilir. Örneğin; Ders seçimi döneminde bütün öğrenciler aynı anda sistemi kullandığında sistemin performans sorunu vermemelidir. Öğrencilere sorunsuz şekilde sistem üzerinden mail atılabilmelidir. Başarı kriterleri ölçülebilir ve açık olmalıdır.

10. Detaylardan arındırılmış şekilde, proje amacından bile daha yüzeysel şekilde projenin anlatılmasıdır. Genel bakış, ürün pazarlama sistemlerinde ürünle ilgili yazılmış kısa bilgilere benzetilebilir. Örneğin; “Eğitim kurumları otomasyonu bir şehirdeki bütün eğitim kurumlarının birbiriyle koordineli şekilde çalışmasına imkan sağlayacak bilgisayar destekli bir uygulamadır.”

11. Sistem detayları: Sistem detayları işin uygulamaya dönük tarafını ifade eder. Giriş kısmında kabaca tarif edilen sistem bu bölümde detaylandırılır. Önce fonksiyonel gereksinimler ile bir nevi yazılımın yerine getireceği işlevler ortaya konulur. Ardından fonksiyonel olmayan gereksinimler sayesinde sistemin görevi yerine getirirken harici unsurları da ihmal etmemesi sağlanır. Örneğin, aynı işi yapan iki yazılımdan daha kullanıcı dostu “user friendly” daha fazla tercih edilir. En sonunda sistemin aktörleri, olayları, senaryoları ve use- case modelleri ortaya konur.

12. Fonksiyonel gereksinimler: Her uygulamanın amacı bir veya birçok fonksiyonu yerine getirmektir. Yerine getirilecek fonksiyonların neler olduğu bu adımda ortaya çıkarılır. □ Bitmiş bir uygulamadaki ana menü ve alt menü detayları veya modüler yazılım geliştirmedeki modül ve alt modüllerin ortaya çıkarılması bu alanda yapılır. Fonksiyonlar bireysel olarak ele alınsa bile fonksiyonları gruplandırarak probleme yaklaşmak daha akılcı olacaktır. Örneğin bir öğrenci işleri sisteminde önce öğrenci kayıt, ders işlemleri, öğretim elemanı kayıt v.s. Belirlenip ardından her biriyle ilgili fonksiyonları yazmak daha akıllıca olacaktır.

13. Fonksiyonel gereksinimler: Fonksiyonel gereksinim yazarken, Sistemin yerine getirmesi gereken her işlev kısa ve anlaşılır şekilde yazılmalıdır. İşlevler kendi arasında önem sırasına göre sıralanabilir olmalıdır. Daha önemli, daha önemsiz gibi etiketler kullanılabilir. Her işlev “-meli”, “-malı” şeklinde gereklilik bildiren ifadeler şeklinde yazılmalıdır. Örneğin, “bu uygulama kimlik numarası üzerinden sorguya imkan sağlamalıdır” gibi.

14. Fonksiyonel olmayan gereksinimler: Yedi madde şeklinde ele alınır. Kullanılabilirlik,,Güvenilirlik, Performans, Desteklenebilirlik, Gerçekleştirim, Arayüz, Gizlilik gereksinimi

15. Kullanılabilirlik: Uygulamanın rahat ve hızlı şekilde kullanımı desteklenmelidir. Örneğin; En kullanışlı menü tasarımları yapılmalıdır (aşağı doğru açılan menüler gibi) Kısa yollar ve hızlı erişim olanakları olmalıdır (klavye kısayolları ve popup menüler) Menülerde ve kısayollardaki standartlara uyulmalıdır. Örneğin, help menüsü en sonda, programdan çıkış ilk ana menünün en sonunda gibi. Butonlar uygun yerlere konulmalıdır. Kapat butonu sağ alt köşe gibi. Yerleşim, fontlar, renk ayarları v.s. Dikkatle yapılmalıdır.

16. Güvenilirlik: Bir uygulamanın her seferinde aynı girişler için aynı çıkışları vermesine güvenilirlik denir. Güvenilirlik için; Veri kaybı mümkünse sıfıra indirilmelidir. Veritabanı kısıtları ile hatalı veri girişi engellenmelidir. Yazılım mantık hatalarından arındırılmalı ve yazılımın non-deterministik (beklenmeyen) hareketler yapmasının önüne geçilmelidir. Hata yakalama prosedürleri çalıştırılmalı ve yazılımın kesilmesi yerine uygun hata mesajları sunulmalıdır.

17. Performans: Bir uygulamanın kısa sürede istenen işleme cevap vermesi performans başlığı altında ele alınır. Performans problemlerinin çözümü için; Sistemin aynı anda maksimum kaç kullanıcıyla çalışabildiği belirtilmelidir. Performans ihtiyacı varsa yazılımsal (cache mekanizması gibi) veya donanımsal olarak çözümüne gidilmelidir. Ayrıca kimi zaman yazılımın hangi donanım üzerinde en iyi performans vereceği belirtilmelidir.
18. Desteklenebilirlik: Geliştirilecek yazılım hangi ortam koşullarında çalışacağı bu kısımda belirtilir. Örneğin bir yazılımın çalışabilmesi için başka yazılımların yüklenmesi gerekebilir, java eklentileri gibi. Ayrıca yazılımın hangi işletim sistemi tarafından desteklendiği verilmelidir. Mümkünse eklentiler yazılımla birlikte setup şeklinde verilmeli ve uygulama birden çok işletim sistemini desteklemelidir.
19. Gerçekleştirim: Bu kısımda kodlamaya hızlı bir geçiş için uygulamanın sunduğu avantajlar varsa ondan bahsedilir. Örneğin akıllı yazılım araçları kullanımı ve Uml diyagramlarından java kodlarının elde edilmesi bu kategoride yer alabilecek örneklerden bazılarıdır.
20. Arayüz: Uygulamanın girişinde ve çıkışında neler olacağı, uygulamanın komut satırı veya grafik arayüzü desteklemesi durumu belirtilir. Örneğin; Veri girişi klavye ve fare ile yapılacaktır. Çıkış ekran ve yazıcıdan alınacaktır. Çıktı pdf olarak da sunulabilecektir. Seri ve paralel port üzerinden haberleşme imkanı olacaktır.
21. Gizlilik gereksinimi Bir uygulama çok farklı türde verilere erişimi destekleyebilir fakat kullanıcıların bütün verileri görme hakkı olmayabilir. O nedenle kullanıcıların sadece hakları olduğu verileri görmesi istenir. Gizlilik gereksinimi bu konuyla ilgilidir. Örneğin, A kullanıcısı f1, f2 ve f3'e erişebilir, B kullanıcısı f1 ve f4'e erişebilir gibi.
22. Sistem modelleri: Dört farklı bileşenden oluşur. Aktörler, Olaylar, Senaryolar, Use-case modeli
23. Aktörler: Sistemde yer alan, yazılım tarafından bilgileri yönetilecek varlıklardır. Gereksinim analiz dokümanında aktörler tanımlarıyla birlikte verilir. Bu aşamada bir aktör ve onun yapacağı işlevler şeklinde sunulur. Fonksiyonel gereksinimler adımıyla bağlantılıdır. Örnek bir aktör ve tanımı şöyle olabilir. Aktör Tanımı Öğretmen Ders seçimi yapar Ders onayı yapar Not girişi yapar Devam bilgisi girer Öğrenci raporları alır
24. Olaylar: Bu kısımda bütün aktörlerin görevleri bir arada sunulur. Böylece sistemde yapılacak bütün işlemler hakkında bilgi toplanmış olur. Bir ekran birden fazla olayı destekleyebilir. Olay güdümlü programlamada (event driven programming) her bir buton aslında bir olay ile ilişkilidir.
25. Senaryolar • Uygulamadaki akış senaryolar şeklinde sunulur. • Bir senaryo hem aktörün hem de olayların dahil olduğu işlemler bütünüdür.