

Yazılım Gereksinimleri Tanım Belgesi (YGTB/SRS)

Yazılım geliştirme süreci ;

- *Gereksinim Analizi / Requirements (Requirements Analysis)*
- *Spesifikasyon / Specification (Functional Specification)*
- *Mimari / Architecture (Software Architecture)*
- *Dizayn / Design (Software Design)*
- *İmplementasyon (Programlama) / Implementation (Computer Programming)*
- *Test / Testing (Software Testing)*
- *Sahaya Yerleştirme / Deployment (Software Deployment)*
- *Bakım / Maintenance (Software Maintenance)*

aşamalarından oluşmaktadır.

YGTB, belgesinde geliştirilecek olan yazılımın işlevsel ve işlevsel olmayan gereksinimleri tanımlanır. Use-Case'ler, Kullanıcı arayüzleri ve ER diyagramları verilir.

Yazılım mühendisliği kapsamında YGTB için verilen verilen taslak şu şekildeydi:

Yazılım Gereksinimleri Tanımı

1. Giriş

1.1 Amaç

[Belgenin amacını ve hedef kitesini tanımlayın.]

1.2 Kapsam

[Belgenin kapsamını tanımlayın.]

Bu kısımda tanımlanan kapsam, varsa bir üst seviye gereksinimlerde yer alan benzer ifadelerle (örneğin, Vizyon belgesi) tutarlı olmalıdır.]

1.3 Tanımlar ve Kısaltmalar

SRS: Software Requirements Specification

[Belgeyi anlamaya yardımcı olacak ve alana özel terim ve kavramın tanımlarını verin ve belge içinde kullanılan kısaltmaları alfabetik olarak listeleyin.]

1.4 Referanslar

[Bu belgenin referans aldığı tüm kaynakları; referans numarası, referans kodu, başlığı, sürümü ve tarihi ile birlikte listeleyin.]

Bu belge şablonu için, "IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications" referans alınmıştır.]

1.5 Dokümana Genel Bakış

[Belgenin sonraki bölümlerinde neler anlatıldığını özetleyin. Belgenin kullanımı ile ilgili (varsa) güvenlik ve gizlilik koşullarından bahsedebilirsiniz.]

2. Genel Tanım

[Yazılım ürünü ve gereksinimlerini etkileyen genel faktörleri, izleyen başlıklar altında tanımlayın.]

2.1 Ürüne Bakış

[Bu belge kapsamında tanımlanan yazılım ürününün (varsa) diğer ürünlerle ilişkisini anlatın.]

Yazılım ürünü bağımsızsa ve sadece kendini içeriyorsa belirtin. Eğer bu belge, daha büyük bir sisteminin parçası olan yazılım ürününü tanımlıyorsa, üst seviye bir bakışla yazılım ile parçası olduğu sistem arasında ilişkiyi kurun ve sistem ile diğer yazılımlar arasındaki arayüzleri belirtin.

Sistemin ana parçalarını, birbirleriyle olan ilişkilerini ve dış arayüzleri gösteren bir blok diyagram çizebilirsiniz.]

2.1.1 Sistem Arayüzleri

[Yazılım ile diğer sistemler arasındaki arayüzleri ve bu arayüz gereksinimlerinin karşılanması için sağlanması gereken işlevleri belirtin.]

2.1.2 Kullanıcı Arayüzleri

[Yazılım gereksinimlerinin karşılayacak, yazılım ile kullanıcısı arasındaki arayüzleri belirtin. Bu arayüzler için taslak ekran yapılarını, sayfa/ekran yapılarını, menü veya raporların içeriklerini tanımlayın (eklere de referans edebilirsiniz)].

2.1.3 Donanım Arayüzleri

[Yazılım ile sistemin donanım öğeleri arasındaki arayüzlerin mantıksal özelliklerini belirtin. Bu özelliklerin konfigürasyon tanımlarını verin.]

2.1.4 Yazılım Arayüzleri

[Kullanılacak olan diğer yazılımın ürünlerinin (veri yönetimi sistemi, işletim sistemi, modelleme araçları, vs) ve diğer uygulama sistemleri ile (varsa) arayüzlerinin özelliklerini

belirtin. Her bir uygulama yazılımı için, kod, isim, tanım/sürüm no.ve kaynak bilgilerini belirtin.]

2.1.5 İletişim Arayüzleri

[İletişim ile ilgili arayüzler (varsa) belirtilir (protokoller vb.)]

2.1.6 Bellek Kısıtları

[Birincil ve ikincil bellek ile ilgili özellikleri ve kısıtları belirtin.]

2.1.7 Ürünün İşletimi

[Kullanıcı gereksinimi olarak yazılımın normal ve özel çalışma kiplerini, kullanım ile ilgili zamanlama kısıtlarını, yedekleme ve kurtarma kısıtlarını veya gereksinimlerini belirtin.]

2.1.8 Saha Uyumlama Gereksinimleri

[Ürünün alandaki işletimi için, yükleme sahasını uyumlama gereksinimlerini (örneğin, veri veya başlatma sırası gereksinimleri) belirtin.]

2.2 Ürün İşlevleri

[Yazılımın gerçekleştireceği ana işlevlerin, detaya girmeksizin bir özetini verin.

Bu bölüm için gereken ana işlevlerin özeti, daha üst seviye gereksinimlerin (örneğin, Vizyon belgesi) söz konusu yazılıma atanmış bölümlerinden alınabilir, ancak bu belgenin “3. Özel gereksinimler” kısmı ile de tutarlı olmalıdır.

İşlevleri, bu belgeyi ilk defa okuyacak herhangi birinin kolaylıkla anlayabilmesi için listelenmiş şekilde düzenleyebilirsiniz.

Farklı işlevleri ve aralarındaki ilişkileri gösterebilmek için metinsel ya da grafiksel gösterim kullanılabilir. Söz konusu diyagramlar ürünün tasarımını göstermez, sadece işlevler arasındaki mantıksal ilişkiyi ifade eder.]

2.3 Kullanıcı Özellikleri

[Kullanıcıların sahip olmaları gereken genel özellikleri (eğitim seviyesi, deneyim vb.) belirtin.]

2.4 Kısıtlar

[Geliştiriciyi kısıtlayabilecek herhangi bir öge ile ilgili genel tanımları verin (örneğin; donanım kısıtları, uyulması beklenen yönergeler, geliştirme ortamı ve teknolojisi ile ilgili gereksinimler, güvenilirlik gereksinimleri, güvenlik ve gizlilik hususları vb.)]

2.5 Varsayımlar ve Bağımlılıklar

[Belgede belirtilen yazılım gereksinimlerini etkileyebilecek olan faktörleri listeleyin. Varsayım ve bağımlılıklar tasarım kısıtları değil, değişimleri halinde gereksinimleri etkileyecek olan faktörlerdir.]

3. Özel Gereksinimler

[Bu kısımda, geliştirme ve test etkinliklerini mümkün kılacak detay seviyesinde, tüm yazılım gereksinimlerini tanımlayın. Tanımlanan gereksinimler, tüm paydaşlar tarafından algılanabilir olmalıdır.]

Gereksinimler, bütün sistem girdilerinin ve çıktılarının tanımı ile girdileri çıktılara dönüştüren işlevlerin tanımını içermelidir. Bunlarla ilişkili ve bunlara ek olarak, işlevsel olmayan (kalite) gereksinimlerini de tanımlayın.

Gereksinimler doğru, kesin, tam, tutarlı, doğrulanabilir, değiştirilebilir, izlenebilir ve önceliklendirilmiş olmalıdır.

Bütün gereksinimleri biricik (“unique”) olarak numaralandırın.]

3.1 Harici Arayüz Gereksinimleri

[Bütün sistem girdilerinin ve çıktılarının detaylı tanımını verin. Bu tanımlar 2. kısımda sunulan bilgileri tamamlamalı / detaylandırmalı, orada verilen bilgilerin tekrarı olmamalıdır.]

Bir sonraki bölümde tanımlanması istenen işlevsel gereksinimlerin işletilmesini destekleyecek kullanıcı arayüzlerine ilişkin detaylar bu bölümde tanımlanabilir. Kullanıcı arayüzü tanımlama için bir şablon Ek-A’da verilmiştir.]

3.2 İşlevsel gereksinimler

[Ürüne ait işlevsel gereksinimlerin detaylı tanımını verin.]

Use-case esaslı gereksinim analizi yaptıysanız bu bölümde use-case modelini ve bu modeldeki öğeleri detaylı olarak tanımlayın. Seçtiğiniz use-case'lere ait senaryoları UML Etkinlik Diyagramları veya metin biçimde detaylandırabilirsiniz. Use-case'lerin metin biçimde detaylandırılması için bir şablon, Ek-B’de verilmiştir.]

3.3 Performans Gereksinimleri

[Yazılım ve/veya kullanıcı arayüzü ile ilgili statik ve dinamik, nicel performans gereksinimlerini tanımlayın. Statik gereksinimlere; desteklenecek eşzamanlı kullanıcı sayısı, desteklenecek istemci sayısı vb. örnek verilebilir. Bu tür gereksinimler ölçülebilir şekilde ifade edilmelidir.]

Benzer özellikler: Hız (“speed”), saniyedeki işlem sayısı (“processed transactions/second”), ortalama maksimum cevap süresi (“average, maximum response time”), ekran tazeleme süresi (“screen refresh time”), işlem hacmi (“throughput”), kurtarma/kendine gelme süresi (“recovery time”), kaynak kullanımı (“resource usage”) – hafıza, disk vb., eş zamanlı desteklediği kullanıcı sayısı (“number of simultaneous users to be supported”), kapasite (“capacity”) – sistemin sağlayabileceği müşteri ya da işlem sayısı]

3.4 Mantıksal Veritabanı Gereksinimleri

<Veritabanında tutulacak veriler ile ilgili mantıksal gereksinimleri tanımlayın. Veri varlıklarını ve aralarındaki ilişki ve bütünlük kısıtlarını belirtin. Bu gereksinimleri ifade etmek için E/R Diyagramı kullanabilirsiniz.]

3.5 Tasarım Kısıtları

[Standartlar, kullanıcı istekleri veya donanım kısıtlarına dayanan ve tasarımı gerçekleştirirken uyulması gereken kararları tanımlayın.]

3.6 Kalite Özellikleri

[Yazılım kalite özelliklerini tanımlayın.]

Aşağıda bazı özellikler için açıklamalar ve örnekler verilmiştir.]

3.6.1 Güvenilirlik (“Reliability”)

[Yazılımın güvenilirliği yani yazılımın sunduğu servisin devam etmesi ile ilgili özelliktir.]

Sistemin çökme sıklığı ve şiddeti, iki çökme arasında geçen ortalama süre (“MTTF: mean time to failure”), onarım için geçen ortalama süre (“MTTR: mean time to repair”), hata ya da kusur oranı (“bugs or defect rate”), maksimum hata ya da kusur oranı (maximum bugs or defect rate”) gibi ölçevler tanımlayarak ifade edilebilir.]

3.6.2 Kullanılabilirlik (“Availability”)

[Yazılımın kullanıma hazır olması ile ilgili özelliğidir.]

Yazılımın kullanım için uygun olma yüzdesi (“% of time available=MTTF/(MTTF+MTTR)”) ile ifade edilebilir.]

3.6.3 Güvenlik

[Yazılımın kazayla ya da kötü niyetle erişimine, kullanımına, değiştirilmesine, tahrip edilmesine ya da imhasına engel olmak için taşıması gereken özellikleridir.]

Belirli şifreleme ve kriptografi ile ilgili tekniklerinin kullanımı, amaca özel kayıtların (“log”) ya da geçmiş verilerin tutulması, farklı modüllere/kullanıcılara belirli fonksiyonların atanması, programın bazı bölümleri arasında iletişimin kısıtlanması ve kritik değişkenler için veri bütünlüğünün kontrolü ile ifade edilebilir.]

3.6.4 Bakım-yapılabilirlik (“Maintainability”)

[Yazılımın kullanıma alındıktan sonra kolay bakım-yapılabilmesi için taşıması istenen özellikler belirtilir.]

3.6.5 Taşınabilirlik (“Portability”)

[Yazılımın başka platformlara (işletim sistemi, vb.) taşınabilirliği ile ilgili özellikler belirtilir.]

3.6.6 Kullanılabilirlik (“Usability”)

<Yazılımın kolay kullanılabilirliği ile ilgili özellikler belirtilir.

Uyulması gereken standartlar (kullanıcı arayüzü standartları, kullanıcı arayüzlerinin tutarlılığı, vb), çevrimiçi (“online”) ve içerik duyarlı (“context-sensitive”) yardım, sihirbazlar (“wizards”), kullanıcı dokümantasyonu, eğitim materyalleri ve eğitim süresi gerekleri tanımlanarak ifade edilebilir.]

4. Gereksinimlerin Önceliđi ve Kritikliđi

[Bu belgede tanımlanan gereksinimlerin görelİ önemlerini, önceliklerini ve varsa tanımlanmış ađırlıklarını yazın.]

5. Gereksinimlerin İzlenebilirliđi

[Bu belgenin “3. Özel Gereksinimler” başlıđı altında tanımlanan yazılım gereksinimleri ile bir üst seviye gereksinimler (örneđin, Vizyon belgesi) arasında çift yönlü izlenebilirliđi oluřturun.

İzlenebilirliđi tablo yapısında oluřturabilirsiniz.]

6. Ekler

[Bu belgenin yapısını basitleřtirmek ve anlaşılmasını kolaylařtırmak için,içerikteki bazı bilgileri eklerde verebilirsiniz. Ana döküman içerisinde bilginin normalde sunulduđu yerde verilemeyen tanımlar için bu bölüm kullanılır.

Ekler, alfabetik olarak ve belge içinde referans edildiđi sırada tanımlanmalıdır.]

Bu da IEEE'nin srs tasarım řablonu ve örneđi: