

# KURULUŞLARIN YAZILIM ŞARTNAMELERİ, NİTELİKLERİ, BAKIM VE EĞİTİM STANDARTLARI ARAŞTIRMASI

1991 yılında DPT;

A Türkiye'de Yazılım Üretimi ve İnsan Gücü Araştırması

\*• Kuruluşların Yazılım Şartnameleri, Nitelikleri, Bakım ve Eğitim Standartları Araştırması,

\*\* Kuruluşların Yazılım Gereksinimlerini Belirleme Araştırması

\*• Uluslararası Yazılım Sektörünü İnceleme ve Türkiye'nin

Bu Alandaki Olası Yerini Belirleme Araştırması

konularında dön: araştırma projesinin gerçekleştirilmesi görevini TÜBİTAK'a vermişti.

Türkiye'nin yarınına yönelik politikalarının saptanmasına önemli katkılar da bulunacağına inandığımız bu proje sonuç raporlarının daha geniş çevrelere duyurulmasında büyük yarar görmekteyiz.

Bu sayımızda, Teletaş uzmanlarınca hazırlanan "Kuruluşların Yazılım Şartnameleri, Nitelikleri, Bakım ve Eğitim Standartları Araştırması" başlıklı raporu yayınlıyoruz.

## 0. KAPSAM VE ORGANİZASYON

### 0.1. Kapsam 0.2. Organizasyon

Bu rapor, yazılım ihtiyaçlarını satın alma ya da sipariş yoluyla, kendi bünyelerinin dışında karşılamak durumunda kalan kuruluşların.

Yazılım Teknik Şartnamelerini hazırlama sürecinde izleyecekleri yöntemler ve Yazılım

Teknik Şartnamesinin nitelikleri ve belgelendirme, eğitim, destek ve bakım hizmetleri standartları konularında yönlendirici olma amacını taşımaktadır.

- Bu raporun kapsamı, mevcut ya da saptanmış bir donanım platformu üzerinde yürütülecek olan yazılımlara yöneliktir. Rapor, yazılımın kendisi dışındaki, donanım ve çevre koşulları gibi tüm diğer unsurların önceden belirlenmiş olduğu kabul edilmektedir.

- Anahtar teslimi bilgi işlem sistemlerinin ediniminde sözkonusu olan, ihtiyaç duyulan donanım ve yazılım birimlerinin etkileşimli olarak birlikte saptandığı süreç bu raporun kapsamının dışındadır.

- Rapor, uygulama yazılımları ele alınmaktadır. Sistem yazılımlarının edinimi ve ilgili standartlar bu raporun kapsamı dışında bırakılmıştır.

- Yazılımın edinimi açısından bu rapor, hazır paket satın alımı ve/veya sipariş ile yazılım geliştirme durumlarına uygulanabilir. Sözkonusu iki farklı yazılım edinimi yolunda izlenecek olan yöntemlerdeki farklılıklar, raporun ilgili bölümlerinde belirtilmektedir.

Bölüm 1'de başlıca yazılım türleri, yazılım edinme biçimleri tanımlanmakta. Yazılım Yaşam Döngüsü kavramı açıklanmakta ve raporun izleyen bölümlerine çatı teşkil edecek olan Yazılım Edinme Aşamaları tanıtılmaktadır..

Bölüm 2'de Kullanıcı İhtiyaçları Tanım Belgesi'nin tanımı, kapsamı ve hazırlanmasındaki ilkeler ele alınmaktadır.

Bölüm 3, raporun temelini oluşturmaktadır. Bu bölümde, Yazılım Teknik Şartnamesi'nin kapsamı ve hazırlanış yöntemleri anlatılmaktadır.

Yazılım Teknik Şartnamesinde ifade edilen hususların dışında, genel olarak belgelendirme, yazılım geliştirme sürecinin izlenebilirliği, eğitim, bakım ve destek hizmetlerine ilişkin temel noktalar raporun izleyen bölümlerinde ele alınmaktadır. Bu bağlamda,

Bölüm 4, sipariş yoluyla yazılım ediniminde, yazılım sürecinin gözlenebilir ve denetlenebilirliği açısından dikkate alınması gereken genel ilkelere-

Bölüm 5, genel olarak yazılım test ve kabul prosedürlerine;

Bölüm 6, genel olarak, yazılımla birlikte müşteriye teslim edilmesi gereken belgelere;

Bölüm 7 ise bakım, eğitim ve destek hizmetlerine ayrılmıştır.

## 1. GİRİŞ

### 1.1. Yazılım

Yazılım, bir bilgisayar sistemi üzerinde işletilerek belirli bir işlevi yerine getiren bilgisayar programı, prosedürler, kurallar, ilişkin veri ve belgeler bütünüdür.

### 1.2. Yazılım Türleri

Yazılım, en genel biçimde,

- 1) Sistem yazılımı,
- 2) Uygulama yazılımı

olarak iki kategoride ele alınabilir [7].

Uygulama yazılımlarının ediniminde kuruluşların izleyecekleri yöntemler bu raporun araştırma konusunu oluşturmaktadır. Sistem yazılımlarının edinimini ise, Bölüm O'da belirtildiği üzere, bu raporun kapsamı dışındadır. Bu iki yazılım türünün tanımını Bölüm 1.2.1 ve 1.2.2'de verilmektedir.

#### 1.2.1. Sistem Yazılımı

Sistem yazılımı, özel bir bilgisayar sistemi ya da bilgisayar sistemi ailesi için tasarlanmış, sistemin ve ilişkili programların işletilmesi ve bakımını gerçekleştiren yazılımdır. Örnek olarak, işletim sistemi, derleyici ve yardımcı programlar v.b. verilebilir.

#### 1.2.2. Uygulama Yazılımı

Uygulama yazılımı, bir bilgisayar sisteminin özel bir işlevsel kullanımı için üretilmiş yazılımdır. Örnek olarak, hava trafik denetim yazılımı, muhasebe yazılımı, kelime işlem yazılımı v.b. verilebilir.

### 1.3. Yazılım Edinme Türleri

Yazılım ihtiyacı olan kuruluşlar bu ihtiyaçlarını aşağıdaki yollardan biriyle karşılayabilirler:

- 1) Yazılımı, kendi bünyeleri içinde geliştirebilirler.
- 2) Yazılımı, hazır olarak, piyasadan satın alabilirler.
- 3) Yazılımı, piyasadaki hazır yazılım araçlarını alarak, kuruluşun ihtiyaçlarına göre konfigürasyonunu ve entegrasyonunu yaptırabilirler.
- 4) Yazılımı, sipariş vererek geliştirebilirler.

Madde 1'de belirtilen, kuruluşların yazılımlarını kendilerinin geliştirmeleri seçeneği bu raporun kapsamı dışındadır.

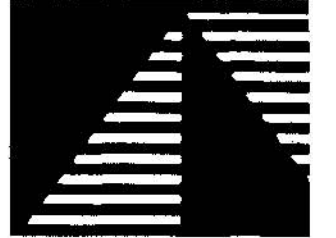
Raporun izleyen bölümlerinde, Madde 2 ve 3'de belirtilen seçeneklerin her İdsi de »atın alma yoluyla yazılım edinimi adı altında geçecektir. Madde 4'de belirtilen seçenek ise sipariş yoluyla yazılım edinimi olarak adlandırılacaktır. Her iki durumda da izlenecek yöntemler ve uygulanacak standartlar bakımından ortak noktalar mevcuttur. Farklılıklar ise, sözkonusu olan yerlerde ayrıca belirtilecektir. Nihayet, sipariş yoluyla yazılım ediniminde, yazılımı geliştiren firma ile yazılım ihtiyacı olan kuruluş arasında, yazılım geliştirme sürecinin takibine yönelik ilişkilerin düzenlenmesi ayrıca ele alınması gereken bir alanı oluşturmaktadır. Bu alan, "Yazılım Geliştirme Gözetim ve Denetimi" başlığı altında Bölüm 4'de ele alınacaktır.

### 1.4. Yazılım Yaşam Döngüsü

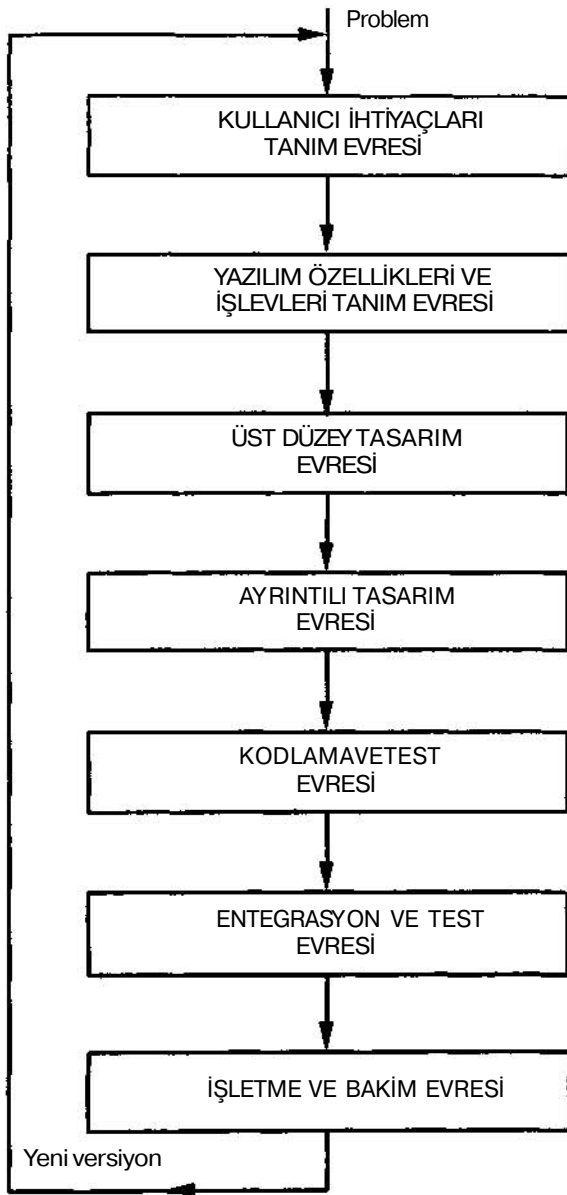
Yazılım Yaşam Döngüsü (Software Life Cycle), bir yazılım ihtiyacının ortaya çıktığı andan başlayarak, yazılımın tüm geliştirme ve bakım süreçlerini kapsayan zaman aralığı olarak tanımlanır [29].

Yazılım mühendisliği literatüründe çeşitli Yazılım Yaşam Döngüsü Modelleri bulunmaktadır. Bu modeller Yazılım Yaşam Döngüsünü, temel yazılım geliştirme ve bakım faaliyetlerini kapsayan ve belirli ürünlerle sonuçlanan evreler türünden tanımlarlar. Yazılım Yaşam Döngüsünün, "Yazılım Geliştirme ve Bakım Evreleri" türünden modellenmesi, ilgili süreçlerin izlenebilir ve denetlenebilir olmasına olanak sağlar. Her bir evre yazılım geliştirme ve bakım süreci için bir kilometre taşı teşkil eden ürünlerle sonuçlanır. Bu ürünler, yazılım belgeleri, planları, raporlar ya da yazılım modülleri olabilir.

Yazılım Yaşam döngüsü modelleri içerisinde en eski ve yaygın olarak kullanılan model, en genel hatlarıyla Şekil 1'de verilmektedir. Bu modelde Yazılım Geliştirme ve Bakım Evrelerinin ardışıl olarak birbirlerini izledikleri gözlenmektedir. Gerçekte bu evreler birbirlerini ardışıl olarak izlemezler, zaman içinde birbirleriyle örtüşürler. Bir diğer deyişle yazılım geliştirme doğrusal değil "iterative" bir süreçtir. Bir evreden önceki evrelere dönüş mümkün ve kimi kez de gereklidir. Örneğin test sırasında tasarım hatalarının gözlenmesi tasarım evrelerine dönüşü gerekli kılabilir. Diğer yandan, özellikle doğrulanma (verificati-on) amaçları açısından ardarda gelen evreler arasında döngüler sözkonusu olabilir, örneğin tasarımın doğrulanması süreci, Üst Düzey Tasarım ile Yazılım Özellikleri ve İşlevleri Tanım Evreleri arasında gidiş dönüşleri gerekli kılar. Nihayet yeniden kullanılabilir yazılım modülleri-



ni destekleyen araçlar, işletilebilen tanım dilleri ve bunlar gibi yazılım tekniklerindeki diğer modern gelişmeler Yazılım Geliştirme ve Bakım Evrelerinin birbirine paralel süreçler halinde düzenlenebilmesine olanak sağlamıştır.



Şekil 1 - YAZILIM Yaşam Döngüsü

### 1.5 Yazılım Edinme Aşamaları

Yazılım gereksinimi olan kuruluş ile yazılımı sağlayacak olan (satıcı ya da üretici firma arasındaki ilişkiler, üç ayrı aşamada ele alınabilir:

- 1) Sözleşme Öncesi Yazılım Tanımlama Aşamaları
- 2) Yazılım Geliştirme Evreleri
- 3) İşletme ve Bakım Dönemi

Bu raporda, yazılımın tanımlanması (sözleşme öncesi evre), geliştirilmesi ve işletim ve bakımı dönemleri boyunca, yazılım ihtiyacı olan kuruluş, yazılım satıcısı/üreticisi firma ve olası diğer firma ve kuruluşlar arasındaki ilişkiler, yürütülecek faaliyetler ve bu faaliyetlerin ürünleri tanımlanmaktadır. Satın alma yoluyla yazılım ediniminde, Yazılım Geliştirme Evresi mevcut değildir.

Şekil 2, Sözleşme Öncesi Yazılım Tanımlama Aşamalarını göstermektedir. Bu aşamada yazılım için kullanıcı ihtiyaçları tanımlanır ve Yazılım Teknik Şartnamesi hazırlanır. Aşama, müşteri kuruluş ve yazılım satıcısı/üreticisi firma arasındaki sözleşme ile sonuçlanır. İlgili belgelerin tanım ve hazırlanış yöntemleri Bölüm 2 ve 3'de verilmektedir.

Yazılım ihtiyacının hazır bir paketle karşılanamadığı ve dolayısıyla yazılımın sipariş edileceği durumlarda sözleşme sonrası Yazılım Geliştirme Evreleri izleyecektir. Bu evreye ilişkin aşamalar, yazılımın türüne, varolan araçların ve teknik birikimin düzeyine bağlı olarak farklılıklar gösterebilir. Yazılım Teknik Şartnamesinde belirtilen özelliklere bağlı olarak müşteri kuruluş ile üretici firma arasında anlaşmaya varılan ve sözleşmeyle onaylanan Yazılım Geliştirme Planı asıl aşamaları belirler. Ancak en genel şekliyle bu evreler, Şekil 3'deki gibi modellenilebilir. Şekil 3'de faaliyetlerin ayrı kutular halinde gruplandırılması olası eşzamanlı faaliyetleri ve değişik aşamalardan geri dönüşleri dışlamamaktadır. Bu evreye ilişkin genel ilkeler, ilgili belgelerin tanım ve hazırlanış yöntemleri Bölüm 4'de verilmektedir.

Yazılımın işleme alınması, işletim sırasındaki aesteek hizmetlerinin verilmesi, olası yazılım hatalarının düzeltilmesi, değişen çevre koşullarına ya da isteklere göre yazılım değişikliklerinin koterılması faaliyetleri İşletme ve Bakım evresini oluşturmaktadır. Bu faaliyetlerin yürütülmesine ilişkin prosedürler ve ilgili belgelerin tanım ve hazırlanış yöntemleri Bölüm 7de verilmektedir.

## 2. KULLANICI İHTİYAÇLARI TANIM BELGESİ

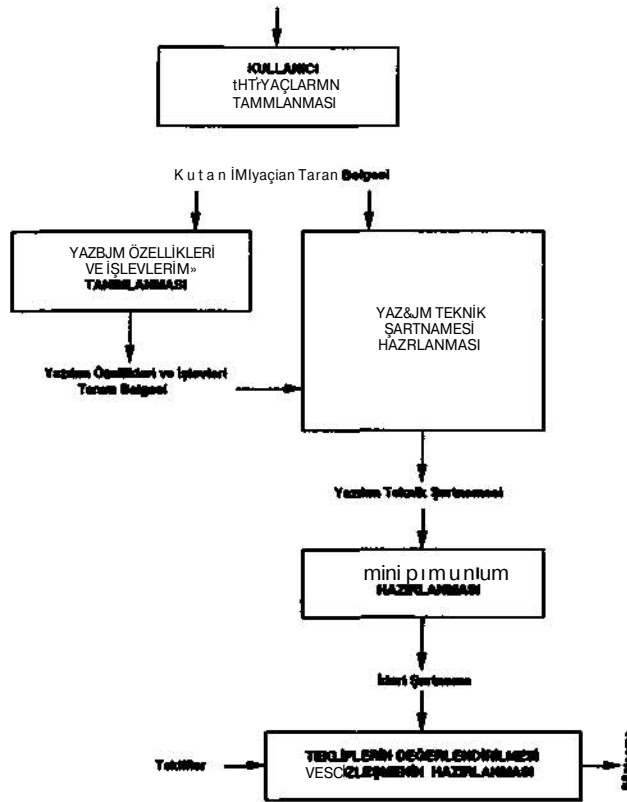
### 2.1. Giriş

Bir yazılımın başarısı, kullanıcı ihtiyaçlarına yanıt verme-deki yeterliliğiyle ölçülür. Başarılı ve kaliteli bir yazılım, kullanıcı ihtiyaçlarını eksiksiz olarak karşılayan yazılım-dır. Kullanıcı ihtiyaçlarının sağlıklı ve eksiksiz analizi bir yazılım ürününün edinilmesi sürecindeki ilk ve en önemli aşamadır.

Yazılım ihtiyacı olan kuruluş, öncelikle yazılımın yerine getirmesi gereken işlevleri ve sahip olması gereken özellikleri saptamalı ve bunları, tam ve açık bir biçimde "Kullanıcı İhtiyaçları Tanım Belgesi"nde belgelemelidir.

### 2.2. Kullanıcı ihtiyaçları Tanım Belgesi: Tanım ve Kapsamı

Kullanıcı ihtiyaçları Tanım Belgesi, kullanıcının zorunlu ve/veya opsiyonel ihtiyaçlarını olabildiğince eksiksiz olarak ve kullanıcının doğal dilinde ifade eder.



Şekil 2 - SÖZLEŞME öncesi Yazılım Tanımlama Aşamaları

Kullanıcı İhtiyaçları Tanım Belgesi, Yazılım Özellikleri ve İşlevleri Tanım Belgesi ile Yazılım Teknik Şartnamesi'nin hazırlanmasında kaynak teşkil eder. (Şekil 2)

Kullanıcı ihtiyaçları

- 1) İşlevsel ihtiyaçlar
- 2) Ortalama uyum ihtiyaçları
- 3) Performans, kapasite ve verimlilik ihtiyaçları
- 4) Bakım, eğitim ve destek ihtiyaçları
- 5) Desteklenmesi istenen standartlar
- 6) Kalite ihtiyaçları
- 7) Yerelleştirmeye (localization) ilişkin ihtiyaçlar

başlıkları altında saptanır.

### 2.2.1. İşlevsel İhtiyaçlar

İşlevsel ihtiyaçlar yazılımın yerine getirmesi istenen işlevlerdir. Yazılımın karşılaması istenen temel işlevler ile karşılanmasında yarar olan yardımcı işlevler arasında ayırım yapılması gerekmektedir.



Şekil 3 - YAZILIM Geliştirme Evreleri

**“Yazılımda kalite,  
tanımı oldukça güç ve  
öznel bir kavramdır.  
Her yazılım için geçerli  
genel kalite ölçütleri  
tanımlamak mümkün  
değildir.”**

2.2.2. Ortama Uyum İhtiyaçları

Yazılımın üzerinde çalışacağı ve/veya etkileşimde bulunacağı kaynaklara bağlı olarak sağlaması gereken özelliklerdir. Başlıcaları aşağıda verilmektedir:

**1) Donanım ve çevre birimlerine bağlı ihtiyaçlar**

(örnek: Yazılımın işletileceği donanım, kullanıcı terminaleri, depolama ortamı, yazıcı ve çiziciler, optik tarayıcılar, diğer çevre birimleri ve bunların gerektirdiği özellikler)

**2) İşletim sistemine bağlı ihtiyaçlar**

(örnek: Yazılımın işletileceği işletim sistemi ya da sistemleri)

**3) Veri tabanına bağlı ihtiyaçlar**

(örnek: Varolan ya da tercih edilen Veri Tabanı Yönetim Sistemi)

**4) Haberleşme ihtiyaçları**

(örnek: Yazılım, tek bir makinada mı yoksa bir bilgisayara ağ üzerinde mi kullanılacak? Ağ üzerinde kullanılacak ise desteklemesi gereken ağ mimarileri nelerdir?)

**5) Varolan/planlanan diğer yazılımlarla beraber çalışma istekleri ve kısıtları**

2.2.3. Performans, Kapasite ve Verimlilik İhtiyaçları

1) Yazılımın beklenen genel performansı ve yerine getireceği işlevlere ilişkin performans ölçütleri verilmelidir.

örnek: (Hız, yanıt süresi, işlem (transaction) süresi, parametre duyarlılığı, v.b.)

2) Kullanıcı sayısı, veri saklama ve işleme kapasitesi v.b. kapasite ölçütleri belirlenmelidir.

3) Yazılımın bellek, merkezi işlem birimi v.b. kaynakları değerlendirme ölçütleri, eğer varsa, belirtilmelidir.

2.2.4. Bakım, Eğitim ve Destek İhtiyaçları

Kullanıcı İhtiyaçları Tanım Belgesinde yazılımın bakımına ilişkin istekler ile istenen eğitim ve destek hizmetleri ve bu doğrultuda yazılımın sağlaması gereken özellikler belirtilmelidir. Bunların başlıcaları aşağıda verilmiştir:

- İstenen belgeler
- Belgelerin güncellenmesi prosedürleri

• Hizmete giriş sırasında istenen bakım istekleri

• İşletme sırasında istenen bakım hizmetleri

• Hata düzeltmeye yönelik istekler

• Müşteri ortamına uyarılma sırasında beklenenler

• Yazılımın yeni versiyonunun sağlanmasına ilişkin beklentiler

• Test gereçleri

• İstenen eğitimin türü ve kapsamı

2.2.5. Desteklenmesi istenen Standartlar

Yazılım ihtiyacı olan kuruluşun, ihtiyaç duyduğu yazılımın özelliklerinden bağımsız olarak, satın aldığı ya da sipariş verdiği yazılımlarda uyulmasını istediği standartlar olabilir. Bunlar, Kullanıcı İhtiyaçları Tanım Belgesinde açık olarak yerelmalıdır.

2.2.6 Kalite İhtiyaçları

Yazılımda kalite, tanımı oldukça güç ve öznel bir kavramdır. Her yazılım için geçerli genel kalite ölçütleri tanımlamak mümkün değildir. Her bir yazılım, kullanıcının o yazılımdan beklediklerine yanıt verme ölçüsünde kalitelidir, örneğin, "güvenilirlik" (reliability) tek başına bir kalite ölçütü değildir, ama bir kalite unsurudur. Bir yazılım, güvenilirlik derecesine göre değil, bu yazılımda güvenilirlik unsuruna duyulan ihtiyacı karşıladığı oranda kalitelidir. Yazılım kalitesi için en uygun tanım, onun öznel içeriğini yansıtan aşağıdaki tanımdır.

Yazılım kalitesi, yazılımın kullanıcı ihtiyaçlarına yanıt vermedeki yeterlilik derecesidir.

Yazılım ihtiyacı olan kuruluş, yazılımda kalite açısından ihtiyaç duyduğu özellikleri saptamalı ve bunları Kullanıcı İhtiyaçları Tanım Belgesi'nde belgelemelidir.

Satın alma yoluyla yazılım ediniminde kalite ihtiyaçları, yazılımın zorunlu ya da tercih nedeni olabilecek özelliklerine işaret ederler. Örneğin, kalite ihtiyaçları, güvenilirlik, emniyet, hizmette süreklilik gibi kalite unsurlarından birine ya da birkaçına ilişkin olup, yazılımın sahip olması gereken zorunlu özellikleri ifade edebilirler. Ya da kullanışlılık gibi bir kalite unsuru ile bağlantılı olup, işlevsel ve performans ihtiyaçlarının karşılanmasının yanısıra tercih nedeni teşkil edebilecek özellikleri ifade edebilirler.

Sipariş yoluyla yazılım ediniminde kalite ihtiyaçlarının belirlenmesi ayrı bir önem taşımaktadır. Yazılımdan beklenen

nen kalite özellikleri, yazılımın geliştirilme yöntemlerini de belirleyici olacaktır. Yazılımın, kullanıcı ihtiyaçlarına yanıt vermedeki yeterliliğinin saptanmasında Yazılım Değerlendirme (Software Reviews) Yöntemlerine ve testlere başvurulur. Ancak, yazılım kalite düzeyinin yalnızca ürün sonrası test ve incelemelerle sınanması pahalı bir yöntemdir. Yazılım ürününün kullanıcı ihtiyaçlarını karşılamada yetersiz kalması durumunda, geliştirme sürecinin önceki evrelerine, zaman ve maliyet açısından istenmeyen geriye dönüşler gerekli olacaktır. Bunun yerine, Yazılım Geliştirme Evresinin, Yazılım Kalite İhtiyaçlarına göre örgütlenmesinin sağlanması ve bu amaçla Yazılım Kalite Planının oluşturulması ve izlenmesi tercih edilmesi gereken yoldur.

Öte yandan yazılımın kullanıcı ihtiyaçlarını karşılamadaki yeterliliğinin sağlanması kadar, yazılım maliyetinin belirlenmesinde de yazılım kalite ihtiyaçları büyük önem taşır. Belirli kalite unsurları (örneğin güvenilirlik ya da emniyet) geliştirme süresini ve maliyeti artırıcı yazılım geliştirme tekniklerinin ve izleme ve denetim yöntemlerinin uygulanmasını gerekli kılacaktır. Bu nedenle kalite ihtiyaçlarının doğru olarak saptanması, yazılım geliştirme sürecinin zaman ve maliyet açısından gerçekçi bir şekilde planlanmasında temel unsurlardan biridir.

Kalite ihtiyaçları, belirli kalite unsurları başlıkları altında gruplandırılabilir. Aşağıda belli başlı yazılım kalite unsurları ve bunların tanımları verilmiştir.

### **Yazılım Kalite Unsurları:**

Aşağıda, belli başlı Yazılım Kalite Unsurları ve bunların tanımları verilmiştir [24]:

- 1) Güvenilirlik (reliability), yazılımın, belirli koşullar altında, tanımlanmış bir zaman aralığında hatasız çalışma olasılığıdır.
- 2) Emniyet (safety), yazılımın cana ya da mala zarar verecek bir durum oluşturmadan çalışma yeteneğidir.
- 3) Hizmette süreklilik (availability), yazılımın, ihtiyaç duyulduğunda, kimi işlevlerini yerine getirmese bile temel işlevlerin kullanılabilir durumda olma yeteneğidir.
- 4) Kullanışlılık (usability), yazılımın kullanımının ve kullanımı öğrenmenin kolaylık derecesidir.
- 5) Yeniden kullanılabilirlik (reusability), yazılım modüllerinin başka uygulamalarda kullanılabilirliğidir.
- 6) Esneklik (flexibility), yazılımın farklı ortamlara uyarlanma kolaylığıdır.

7) Genişleyebilirlik (expandability), yazılımın yeni ihtiyaçları karşılayabilmek amacıyla, işlevsel yeteneklerini ya da performansını artırabilme kolaylığıdır.

8) Gizlilik/bütünlük (security/integrity), yazılımın, yazılıma ya da veriye istenmeyen ya da kötü niyetli erişimi, kullanımı, değiştirmeyi ve bozmayı önleyici yetenekleridir.

9) Doğrulanabilirlik\* (verifiability), yazılımın doğru çalıştığına gözlenebilmesi ve denetlenebilmesi kolaylığıdır.

10) Birlikte çalışabilirlik (interoperability), yazılımın başka yazılım ve uygulamalarla birlikte çalışabilme yetenekleridir.

11) Taşınabilirlik (portability), yazılımın kullanılageldiği ya da tasarlandığı sistemden farklı bilgisayarlarda ve işletim sistemlerinde kullanılabilme yeteneğidir.

12) Verimlilik (efficiency), yazılımın kendisinden beklenen işlevselliği yerine getirmek için kaynaklarını kullanma oranıdır.

13) Bakım kolaylığı (maintainability), yazılımda olası hataları bulma ve düzeltme kolaylığıdır.

14) Yönetilebilirlik (manageability), yazılımda destek ortamının tamamlığı ve kullanım kolaylığına ilişkindir.

15) Türkçe kalitesi, yazılımın kullanıcı arabasında ve yazılım belgelerinde kullanılan Türkçe'nin gramer ve kelime dağarcığı bakımından doğru ve tutarlı olma derecesidir.



### **2.2.7. Yerelleştirmeye İlişkin İhtiyaçlar**

Satın alınacak yazılımın yabancı bir firmanın ürünü olduğu ya da geliştirilecek yazılımın altyapısında yabancı kaynaklı yazılımların kullanılacağı durumlarda, müşteri kuruluşun bu yazılımın/yazılımların ve belgelerinin ne ölçüde yerelleştirilmesini istediği Kullanıcı İhtiyaçları Tanım Belgesinde belirtilmelidir.

### **2.3. Kullanıcı ihtiyaçları Tanım Belgesinin Hazırlanması**

Kullanıcı İhtiyaçları Tanım Belgesi, asıl olarak problemin

tanımlanması amacını taşır. Bundan sonra, yazılımla bu problemin nasıl çözüleceğinin belirlenmesi aşaması gelir. Çözümün modellenmesi açısından problemin sağlıklı ve eksiksiz biçimde tanımlanmasına gerek vardır.

Kullanıcı ihtiyaçlarının saptanmasında şu yöntemler izlenebilir [24]:

- varolan belgelerin incelenmesi,
- varsa eski sistemin kullanıcılarıyla yapılan görüşmeler,
- planlanan sistemin kullanıcıları ve planlayıcılarıyla yapılan görüşmeler,
- anketler,
- uzmanlara hazırlatılan raporlar

Saptanan kullanıcı ihtiyaçlarının değerlendirilmesi, aralarında olası tutarsızlıkların giderilmesi ve önem derecelerinin belirlenerek Kullanıcı İhtiyaçları paragraf Tanım Belgesinde belgelendirilmesi gerekmektedir.

Kullanıcı İhtiyaçları Tanım Belgesi, Yazılım özellikleri ve İşlevleri Tanım Belgesi ile Yazılım Teknik Şartnamesinin hazırlanmasında kaynak teşkil eder.

Kullanıcı İhtiyaçları Tanım Belgesi yazılım ihtiyacı olan kuruluş tarafından hazırlanır.



### 3. YAZILIM TEKNİK ŞARTNAMESİ

#### 3.1. Giriş

Yazılım ihtiyacı olan kuruluşun, ihtiyaçlarına uygun ürünü saptamasında belirleyici olan Yazılım Teknik Şartnamesidir.

Kuruluş, ihaleye Yazılım Teknik Şartnamesi ile çıkar.

Yazılım Teknik Şartnamesi, Kullanıcı İhtiyaçları Tanım Belgesi kaynak alınarak hazırlanır. (Şekil 2)

Yazılım Teknik Şartnamesi, ürünü teknik olarak tanımlar, ayrıca sipariş yoluyla yazılım ediniminde tasarıma kılavuzluk eder ve teknik bir dille yazılır.

Yazılım Teknik Şartnamesinin hazırlanmasında, istenen yazılımın karmaşıklık derecesine ve çapına göre iki farklı yöntem izlenebilir. Kuruluş, Yazılım Teknik Şartnamesini kendi bünyesi içinde oluşturabileceği gibi, sözkonusu şartnamenin hazırlanması görevini bir diğer firma ya da kuruluşa devredebilir. Yazılım Teknik Şartnamesinin hazırlanmasına ilişkin prosedürler Bölüm 3.3'de ele alınmaktadır.

#### 3.2. Yazılım Teknik Şartnamesi: Tanım ve Kapsamı

Yazılım Teknik Şartnamesi, yazılım ürününden beklenen

- 1) bütün teknik özellikleri,
  - 2) kalite özelliklerini,
- tanımlayan ve
- 3) yazılımla birlikte istenen belgelerin tür ve niteliklerini,
  - 4) istenen eğitim hizmetlerini,
  - 5) bakım ve destek isteklerini,
  - 6) yazılım test ve kabul aşamalarını,
- belirten ve
- 7) yazılımın sipariş yoluyla yaptırılması durumunda, yazılım geliştirme sürecinin aşamalarını ve uygulanacak yazılım geliştirme denetim yöntemlerini de saptayan belgedir.

Yazılımın beklenen teknik özellikleri ile kalite özellikleri, Yazılım özellikleri ve işlevleri Tanım Belgesinde ifade edilir ve adı geçen belge Yazılım Teknik Şartnamesinin önemli bir bölümünü oluşturur.



### 3.2.1. Yazılım özellikleri ve İşlevleri Tanım Belgesi (YÖİTB): Tanım ve Kapsamı

Yazılım Özellikleri ve İşlevleri Tanım Belgesi (Software Requirement Specification), bir yazılımı bütün teknik ve kalite özellikleriyle tanımlayan belgedir [11].

Yazılım özellikleri ve İşlevleri Tanım belgesi, Kullanıcı İhtiyaçları Tanım Belgesinden hareketle yazılır. (Şekil 2)

Yazılım özellikleri ve İşlevleri Tanım belgesi, Yazılım Teknik Şartnamesinin bir bölümünü oluşturur. (Şekil 2)

Yazılım özellikleri ve İşlevleri Tanım Belgesi, ürünü teknik olarak tanımlamaya, ayrıca sipariş yoluyla yazılım ediniminde tasarıma kılavuzluk etmeye elverişli teknik bir dille yazılır.

Yazılım Özellikleri ve İşlevleri Tanım Belgesi'nde ifade edilen her yazılım isteri yazılımın beklenen temel bir özelliği ve işlevini tanımlar. Yazılım isteri, Kullanıcı İhtiyaçları Tanım Belgesinde ifade edilen kullanıcı ihtiyaçlarından kaynaklanır. Kullanıcı İhtiyaçları Tanım belgesi, çözümlenmesi gereken problemi tanımlarken, Yazılım Özellikleri ve İşlevleri Tanım Belgesi tanımlanan problemin formal bir dille ifadesini ve modellenmesini sağlar.

Yazılım Özellikleri ve İşlevleri Tanım Belgesi, yazılım isterlerinin tümünü, doğru ve eksiksiz olarak tanımlamalı, ancak yazılımın geliştirilme ve bakım sürecine ilişkin ayrıntılara ve yazılımla birlikte istenen ve Yazılım Teknik Şartnamesinde yer alacak olan hizmetlerin tanımına girmemelidir. Kaynak [4] ve Kaynak [11], Yazılım Özellikleri ve İşlevleri Tanım Belgesinin hazırlanmasında temel başvuru kaynaklarını oluştururlar.

Yazılım isteri, kullanıcı ihtiyaçlarıyla kısmen örtüşen başlıklar altında toplanır.

- 1) İşlevsel isterler
- 2) Çevre ile arabağlar
- 3) Performans, kapasite ve verimlilik isterleri
- 4) Kalite isterleri
- 5) Desteklenmesi gereken standartlar
- 6) Yerelleştirme isterleri

#### 3.2.1.1. İşlevsel İsterler

İşlevsel isterler, yazılımın yerine getirmesi gereken işlevleri tanımlar. Yazılımın karşılaması istenen temel işlevler ile karşılanmasında yarar olan yardımcı işlevler arasında ayırım yapılması gerekmektedir.

#### Yazılımın İşlevsel İsterlerinin Tanımlanması Yöntemleri:

Yazılımın işlevsel isterlerinin ifadesinde

- 1) Giriş/çıkış çiftlerinin tanımlanması
- 2) Örnek kullanarak davranışın betimlenmesi
- 3) Modelleme

yöntemlerinden biri ya da bir kaçını kullanılabılır [11]. Yazılımdan beklenen işlevsel özelliklerin saptanmasında kullanıcı kılavuzunun taslağının hazırlanması yararlı ve yardımcı bir etkinlik olacaktır.

- 1) Giriş/çıkış çiftlerinin tanımlanması

Yazılımın istenen davranışının betimlenmesinde giriş/çıkış çifti dizilerinin tanımlanması kullanılan etkin yöntemlerden biridir. Tanımlanan yazılımın özelliklerine bağlı olarak farklı yöntemler izlenebilir:

i) Kimi yazılımların davranışı en iyi şekilde istenen çıkışların tanımlanmasıyla betimlenebilir, (örneğin raporlama sistemleri.) Bu tür yazılımlar, genellikle veri dosyaları üzerinde işlem yaparlar. Kullanıcı girişi, denetim bilgisini sağlamaya ve veri dosyası işlemlerini tetkiklemeye yöneliktir.

ii) Kimi yazılımlar ise en iyi şekilde giriş/çıkış çiftleriyle tanımlanabilir. Bu tür yazılımlarda asıl işlem o andaki giriş verisi üzerinden yapılır.

iii) Kimi yazılımlar ise bir giriş karşısında yapılacak olan işleme o andaki girişle birlikte geçmiş giriş/çıkış davranışına göre yanıt verirler. Bu tür yazılımlar sonlu durum makinaları olarak modellenebilirler.

Yazılım davranışının betimlenmesinde giriş/çıkış çiftlerinin kullanılmasını zor hatta olanaksız kılan durumlar sözkonusu olabilir. Pek çok yazılım sonsuz sayıda giriş kabul eder. Giriş/çıkış çiftlerinin hepsinin belirlenmesi, dolayısıyla yazılımın tümüyle tanımlanması bu gibi durumlarda olanaksızdır.

2) Örnek kullanarak davranışın betimlenmesi. Yazılım davranışının betimlenmesinde istenen bir kaç davranış örneğinin verilmesi bir başka seçenek teşkil eder. Bu durumda bütün giriş/çıkış davranışı gösterilemeyecektir. Ancak seçilen örnekler, yazılımın tüm davranışının tanımlanmasında yeterince aydınlatıcı olabilir.



### 3) Modelleme

Yazılım isterlerinin tanımlanmasında bir diğer yöntem ise model kullanımıdır. Özellikle karmaşık davranışların betimlenmesinde modelleme hassas ve etkin bir yaklaşımdır. Matematiksel modeller, işlevsel modeller ve zamanlama modelleri belli başlı modelleri oluştururlar:

#### I) Matematiksel modeller

Matematiksel bir model, yazılım

davranışını matematiksel ilişkiler türünden tanımlar. Matematiksel modeller özellikle havacılık, doğrusal programlama, ekonometri, işaret işleme ve hava durumu analizi alanlarında kullanılırlar.

#### II) İşlevsel modelleme

İşlevsel bir model giriş ve çıkış çiftleri arasında eşlenmeleri gösterir. Sonlu durum makineleri, Petri ağları gibi işlevsel modeller, yazılımın çeşitli özelliklerini tanımlamada ya da amaçlanan yazılım davranışını sergilemekte yardımcı olurlar.

#### III) Zamanlama modelleri

Zamanlama modelleri zaman kısıtları ile genişletilmiş modellerdir. Özellikle gerçek zamanlı sistemlerin davranışını betimlemekte zamanlama modelleri son derece yararlıdır.

#### iv) Diğer modeller

Yukarıdaki genel modellerin dışında, özel uygulamalar kendilerine uygun modellere sahip olabilirler. Özellikle belirli bir Formel İster Tanım Dilinin (Formal Requirements Specification Language) kullanımı beraberinde belirli bir modelin kullanımını gerektirebilir.

Model kullanımında dikkat edilecek noktalar:

Hangi model kullanılırsa kullanılsın ilgili Yazılım Özellikleri ve İşlevleri Tanım belgesinde kullanılan model tam olarak tanımlanmalıdır. Modelle ilgili olarak

- 1) Model parametrelerinin sınırları
- 2) Sonuçların istenen hassaslığı
- 3) İş yükleme kapasitesi
- 4) İstenen işlem süresi
- 5) Normal yanıtı ile hata yanıtı

belirtilmelidir.



#### 3.2.1.2. Çevre ile arabağlar

Yazılımın, donanımla, çevre birimleri ile, diğer yazılımlarla ve insanlarla arabağlarını tanımlayan isterlerdir.

#### 3.2.1.3. Performans, Kapasite ve Verimlilik İsterleri

Yazılımın,

- 1) genel olarak ve işlevlerine ilişkin hız, yanıt süresi, işlem süresi, parametre duyarlılığı v.b. performans ölçütlerini;

çütlerini;

- 2) kullanıcı sayısı, veri saklama ve işleme kapasitesi v.b. kapasite ölçütlerini;

- 3) bellek, merkezi işlem birimi v.b. kaynakları değerlendirme ölçütlerini tanımlayan isterlerdir.

#### 3.2.1.4. Kalite İsterleri

Kullanıcının yazılımdan beklediği kalite ihtiyaçlarının karşılanması için yazılıma ve sipariş yoluyla yazılım ediniminde yazılım geliştirme süreçlerine ilişkin isterlerdir.

Yazılım Kalite İsterleri, yazılım ihtiyacı olan kuruluşun yazılımdan beklediği kalite özelliklerinin sağlanması için gerçekleştirilmesi gereken

- 1) Yazılımın kendisine,
- 2) Tasarım sürecine
- 3) Kodlamaya (programların yazılması)
- 4) Test süreçlerine
- 5) Yazılım gözetim ve denetim süreçlerine
- 6) Belgelendirmeye

ilişkin isterler olarak gruplandırılabilir.

Bu isterlere ilişkin bir kaç örnek aşağıda verilmektedir:

#### 1) Yazılımın kendisine ilişkin kalite isterlerine örnekler:

örnek (1) : "Kullanıcıya veri girişlerinde öntanımlı (default) değer sunulmalıdır." (İlgili kalite unsuru: Kullanışlılık)

örnek (2) : "Bellek kullanımı dinamik olmalıdır." (İlgili kalite unsuru: Verimlilik)

örnek (3) : "Ana haberleşme hattında arıza durumun-

da alternatif haberleşme hatları seçilmelidir.\* (kg\* kalite unsurları: Güvenilirlik, emniyet, hizmette süreklilik)

**2) Tasarım sümek\* ilişkin kalite isterleri örnekleri:**

örnek (5) : "Performans isterleri uyarlanabilir parametreler olarak gerçekleştirilmelidir.\* (ilgili kalite unsuru: Genişleyebilirlik\*)

örnek (6) : "Yapısal tasarım teknikleri kullanılmalıdır." (ilgili kalite unsuru: Genişleyebilirlik\*, esneklik, birlikte çalışabilirlik, bakım kolaylığı, taşınabilirlik, yeniden kullanılabilirlik, doğrulanabilirlik)

**3) Kodlamaya ilişkin kalite isterleri örnekleri:**

örnek (7) : "Makina dili ya da birleştirici dili kullanmaktan kaçınılmalıdır." (ilgili kalite unsuru: Birlikte çalışabilirlik, taşınabilirlik, yeniden kullanılabilirlik)

örnek (8) : "Modül kod satır sayısı 100 ile sınırlı olmalıdır." (ilgili kalite unsuru: Genişleyebilirlik, esneklik, birlikte çalışabilirlik, bakım kolaylığı, taşınabilirlik, yeniden kullanılabilirlik, doğrulanabilirlik)

**4) Test süreçlerine ilişkin kalite isterleri örnekleri:**

örnek (9) : "Bütün modül arabaları test edilmelidir." (ilgili kalite unsuru: Bakım kolaylığı, doğrulanabilirlik)

**5) Yazılım gözetim ve denetim süreçlerine ilişkin kalite isterleri örnekleri:**

örnek (10) : "Bütün kritik modüllerde kod incelemesi yürütülmelidir." (ilgili kalite unsuru: Güvenilirlik, emniyet, bakım kolaylığı)

örnek (11) : "Yazılım doğrulama ve test faaliyetleri bağımsız bir kuruluş tarafından yürütülmelidir." (ilgili kalite unsuru: Güvenilirlik, emniyet, bakım kolaylığı)

Örnek (12) : "Taslak kullanıcı kılavuzu tasarım evresinde yayınlanmalıdır." (ilgili kalite unsuru: Birlikte çalışabilirlik, kullanılabilirlik)

**6) Belgelendirmeye ilişkin kalite isterleri örnekleri:**

Örnek (13) : "Kod listeleri belgelendirmeye dahil edilmelidir." (ilgili kalite unsuru: Yönetilebilirlik, yeniden kullanılabilirlik)

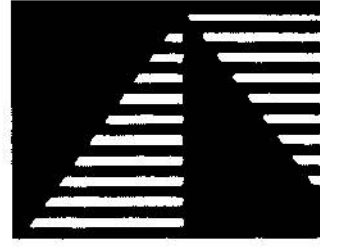
örnek (14) : "Yazılım belgelerinde indeks bölümü bulunmalıdır." (ilgili kalite unsuru: Yönetilebilirlik, yeniden kullanılabilirlik)

**3.2.1.5. Uyulması Gereken Standartlar**

Yazılım özellikleri ve İşlevleri Tanım Belgesinde, yazılı-

mın ve ayrıca sipariş yoluyla yazılım ediniminde Yazılım Geliştirme eylemlerinin uyumlu olması istenen standartlar var ise belirtilmelidir. Bu standartlar, uluslararası ya da ulusal standart organizasyonları tarafından kabul edilen standart ve öneriler, de facto endüstri standartları ya da firmaların kendi standart ya da kılavuzları olabilir. Örnek olarak

- Veri Gösterimi Standardı (Örneğin ASCII, EBCDIC)
- Tasarım Gösterim Standardı (Örneğin SDL, ANSI/IEEE Std 1016) [16]
- Program Kodlama standartları (ANSI-C- FORTRAN 77, SOL)
- Kullanıcı arabacı standartları (örneğin X-Window's)
- Haberleşme standartları (örneğin IEEE 802.3)
- Belgeleme standartları (DOD STD 2167, ANSI/IEEE 829) [3,10]
- Kod yazım kuralları (Çeşitli kuruluş ve firmalara ilişkin)
- İşletim sistemi arabacı \_ \_ \_ \_ \_ " \_ \_ \_ \_ \_ standardı (System Call Interface) (UNIX SVID Rel. 2) verilebilir.



**3.2.1.6. Yerelleştirme İsterleri**

Satın alınan yazılımların dış kaynaklı olması ya da geliştirilecek yazılımın dış kaynaklı yazılımları kullanması durumunda yazılımın yerelleştirilmesi gündeme gelmektedir. Yerelleştirme,

- 1) ekran bilgisinin Türkçe çıkması,
- 2) girişlerin Türkçe yapılması,
- 3) çıkışların Türkçe sunulması,
- 4) Türkçe karakter içeren bilginin yazılım tarafından işlenebilmesi (sıralama, büyük harf/küçük harf çevrimi, v.b.),
- 5) hata mesajlarının Türkçe olması,
- 6) yardım mөнüsünün Türkçe sunulması,
- 7) yazılım belgelerinin (Bölüm 6.1'de tanımlanan) Türkçe olması,
- 8) tarih, saat, para birimi, sayı gösterimi ve diğer ölçü birimlerinin Türkiye'de kabul edilen ve kullanılan notas-

yonda girilmesi ve sunulması olanaklarının gerçekleştirilmesini içerir.

Bir yazılımın yukarıda ifade edilen bütün özellikleri sağlayacak biçimde yerleştirilmesi yoğun emek gerektiren ve maliyet yükseltici bir süreçtir. Müşteri kuruluş, hangi yerleştirme olanaklarını talep ettiğini Yazılım özellikleri ve İşlevleri Tanım belgesinde belirtmelidir.

Kod Tablosu:

Yerleştirmede bugüne kadar karşılaşılan önemli sorunlardan biri, Türkçe karakterler için kabul edilmiş bir standart Kod Tablosunun bulunmamasıydı. Bu durum, şimdiye kadar yapılan yerleştirmelerde firmaların kendilerine özgü Türkçe karakter kod tabloları kullanmalarına ve bundan doğan uyumsuzluklara neden olmuştur. Bugün, uluslararası (ISO) ve ulusal (TSE) standart organizasyonları tarafından onaylanmış Türkçe karakterler için 8 ikilik (bit) kod tablosu mevcuttur (TSE 5881 Bilgisayar ve Veri iletişimi Kullanılan 8 İkil Uzunluğunda Türkçe karakter Kodlama Kuralları/ISO 8859 Text Communication - Registration of graphics character subreperitoires - 8-bit single byte coded graphic character sets - Table 9 (Latin 5) ) [33, 34]. Yerleştirmelerde, sözleşmede aksi belirtilmediği sürece, TSE 5881'e uyulmalıdır.

Terminoloji

Yazılımın yerleştirilmesi sürecinde Türkçe terminoloji büyük önem taşımaktadır. Müşteri kuruluş, satıcı/üretici firmadan çeviri sırasında belirli bir Türkçe terim sözlüğünün kullanılmasını talep edebilir, yazılımla birlikte kullanılan terim sözlüğünün teslimini isteyebilir ve/veya yazılımla birlikte teslim edilecek Türkçe Belgelerin her birinin terim sözlüğünü içermesini talep edebilir.

3.2.2. Yazılım özellikleri ve İşlevleri Tanım Belgesinin Özellikleri

Yazılım isterlerinin tümü, doğal olarak, eşit öneme sahip değildir. Yazılım Özellikleri ve İşlevleri Tanım Belgesi'nde yer alan yazılım isterlerinin görece önemlerinin belirtilmesi gerekmektedir. Her bir yazılım isterinin görece önemi açık ve kesin olmalıdır [11].

1) Yazılım isterlerinin bir kısmı yazılımın işletim süresi boyunca sabit kalacaktır. Kimi yazılım isteri ise değişikliğe açık ya da geçici olabilir. Her bir isterin sürekli, ya da geçici olup olmadığı, değişiklik beklentisinin bulunup bulunmadığının belirtilmesi gerekmektedir.

2) Her bir yazılım isteri "zorunlu" ya da "tercih edilir" niteliklerinden biri kullanılarak tanımlanmalıdır. "Zorunlu" olarak tanımlanan bir ister yazılımın kabulünde belirleyici olacaktır. "Tercih edilir" olarak nitelenen bir ister

ise kabul kriteri teşkil etmeyecek, ancak gerçekleşmesi halinde yazılımın daha üstün nitelikli olmasını sağlayacağından tercih nedeni olabilecektir.

Yazılım özellikleri ve İşlevleri Tanım Belgesi, tüm yazılım isterlerini doğru ve eksiksiz olarak tanımlamalıdır. Ancak YÖİTB tasarım ya da doğrulanma gibi yazılım geliştirme süreçlerine ya da proje yönetimine ilişkin ayrıntıları kapsamamalıdır.

Yazılım özellikleri ve İşlevleri Tanım Belgesi, sözkonusu yazılımın nasıl olacağını tanımlar. Olası tasarım kısıtlamaları dışında, geliştirilmesine ilişkin konuları ele almaz.

Yazılım Özellikleri ve İşlevleri Tanım Belgesinin özellikleri:

1) YÖİTB açık ve kesin olmalıdır.

YÖİTB'de ifade edilen her isterin tek bir yorumu olmalıdır. Bu ise yazılımın özelliklerini belirten her terimin belirli tek bir anlama sahip olmasıyla mümkündür. Özel bir bağlam içerisinde kullanılan herhangi bir terim, bir kaç anlama sahipse, ilave edilen bir terim sözlüğü ile kullanılan anlamı netleştirilmelidir.

Gündelik konuşma diliyle (örneğin Türkçe) yazılım isterlerinin ifadesinde doğabilecek olası belirsizliklerin önlenmesinde güvenli bir yöntem olarak Formel İster Tanım Dilleri (Formal Requirements Specifications Languages) kullanılabilir.

2) YÖİTB eksiksiz (complete) olmalıdır.

Eksiksiz bir YÖİTB,

i) Bütün yazılım isterlerini içermelidir;

ii) Yazılımın, bütün gerçekleştirilebilir durumlarda, bütün gerçekleştirilebilir giriş veri sınıflarına yanıtlarını tanımlamalıdır;

iii) İlgili standartlarına uymalıdır. Özel bir nedenle ilgili standartın bir bölümü uygulanamaz ise, uygulanamayan bölüm ve uygulanamama nedeni belirtilmelidir.

iv) İçindeki bütün şekil, şema ve tablolar eksiksiz olarak adlandırılmalı ve numaralandırılmalı ve kullanılan bütün terimler ile ölçü birimleri tanımlanmalıdır.

"Daha sonra belirlenecek" ifadeleri üzerine:

İçinde "daha sonra belirlenecektir" ifadesini kullanan bir YÖİTB eksiksiz olamaz. Ancak, bir YÖİTB'de "daha sonra belirlenecektir" ifadesinin kullanılması çeşitli nedenlerle gerekli olabilir. Bu durumlarda

- i) Belirsizliğin çözümlenebileceği zaman belirtilmelidir.
- ii) Belirsizliğin ortadan kalkması için ne yapılması gerektiği belirtilmelidir.

Hazır yazılım paketi alımına gidilen durumlarda YÖİTB "daha sonra belirlenecektir" ifadesi içeremez.

Sipariş yoluyla yazılım ediniminde ise sözleşmeyle kesinleşen YÖİTB "daha sonra belirlenecektir" ifadesi içeremez.

Sipariş yoluyla yazılım ediniminde sözleşmeye kadar olan evrede kesinleşmemiş YÖİTB'de "daha sonra belirlenecektir" ifadesi yer alıyor ise, buna dayanılarak üretilen her belge sözkonusu YÖİTB'nin versiyon ya da baskı numarasını belirtmelidir.

#### 3) YÖİTB doğrulanabilir olmalıdır.

İçindeki her yazılım isteri doğrulanabilir olan bir YÖİTB doğrulanabilir. Bir ister, sonuçtaki yazılımın kendisini sağlayıp sağlamadığı maliyet etkin bir şekilde bir makina ya da insan tarafından kontrol edilebiliyor ise doğrulanabilir.

#### 4) YÖİTB tutarlı (consistent) olmalıdır

Birbiriyle çelişen yazılım isteri barındırmayan YÖİTB tutarlıdır. Başlıca tutarsızlık türleri şöyle olabilir:

- İki ya da daha fazla yazılım isteri aynı nesneye farklı adlarla başvurursa;
- İki ya da daha fazla yazılım isteri aynı nesneyi farklı ve çelişen özelliklerle tanımlarsa;
- İki ya da daha fazla yazılım isteri tarafından tanımlanan işlevler arasında bir çatışma varsa

sözkonusu YÖİTB tutarsız olur.

#### 5) YÖİTB kolaylıkla değiştirilebilir (modifiable) olmalıdır.

Yapısı ve yazım biçimi, isterlerde gereken değişikliklerin kolayca, eksiksiz ve tutarlı biçimde yapılmasına elveren bir YÖİTB değiştirilebilirdir. Değiştirilebilir bir YÖİTB şu özelliklere sahiptir:

- i) Kolay okunabilen ve izlenebilen, bağlantılı bir yapısı olmalıdır. İçindekiler ve indeks bölümleri bulunmalıdır. Bütün bölümler arasındaki çapraz referansları açık olmalıdır.
- ii) Aynı yazılım isteri YÖİTB içinde birden fazla yerde bulunmamalıdır. Bir isterin bir kaç yerde dile getirilmesi belgeyi daha okunabilir kılmakla birlikte ilgili isterdeki değişiklik durumunda değişikliğin her yere yansıtılmaması

ve dolayısıyla YÖİTB'nin tutarsızlaşması riskini barındırır.

#### 6) YÖİTB izlenebilir olmalıdır.

İçindeki her yazılım isterin kaynağı açık bir şekilde belirtilen ve ilerideki yazılım değişiklikleri ya da yeni versiyon geliştirme evresinde isterlere referansı kolaylaştıran bir YÖİTB izlenebilirdir.

#### 7) YÖİTB işletme ve bakım evresinde de kullanılabilir olmalıdır.

YÖİTB, işletme ve bakım evresindeki ihtiyaçlara yanıt verebilmelidir. Geniş çaplı değişiklik gerektiğinde ya da yeni versiyon geliştirme durumunda YÖİTB ve tasarım belgeleri kaynak teşkil edecektir. Bu bakımdan YÖİTB (6)'da belirttiği biçimde izlenebilir olmalıdır. Ayrıca YÖİTB'deki her yazılım isterin

- i) önem derecesi,
  - ii) geçici ihtiyaçlara karşılık gelip gelmediği ve
  - iii) kaynağı
- belirtilmelidir.

#### 3.2.3. Belgeleme İsterleri

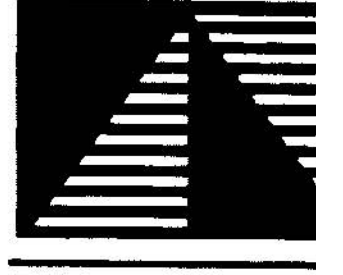
Yazılım Teknik Şartnamesinde, yazılımla birlikte istenen belgelerin

- i) türleri
  - ii) kapsam ve nitelikleri
  - iii) varsa belgelendirmede uyulması istenen standartlar
  - iv) belgelerin güncellenmesi prosedürlerine ilişkin isterler
- olarak belirtilmelidir.

Yazılım Teknik Şartnamesinde belirtilen belgelendirme isterlerinin haricinde, her tür yazılım tesliminde, yazılımın türü, karmaşıklığı ve büyüklüğünden bağımsız olarak üretici/satıcı firmanın sağlaması gereken belgeler ve içerikleri Bölüm 6'da verilmektedir.

#### 3.2.4. İstenen Eğitim Hizmetleri

Yazılım kullanıcılarının, yazılımın işletme ve bakımından sorumlu olacak personelin eğitim gereksinimleri ve bu doğrultuda yazılım üreticisi/satıcısı firmadan istenen eğitimin kapsam ve süresi Yazılım Teknik Şartnamesinde belirtilmelidir.



### 3.2.5. Bakım v\* Destek İsterleri

- Yazılımın hizmete alınması sırasında beklenen destek hizmetleri
- Müşteri ortamına uyarlamaya ilişkin destek hizmetleri
- Yazılımla birlikte istenen test gereçleri, performans gözleme gereçleri
- Hatalı çalışma durumunda verilecek düzeltmeye yönelik bakım hizmetlerinin çerçevesi

- Yeni versiyon geliştirme ya da güncelleme prosedürleri

v.b. bakım ve destek hizmetlerine ilişkin isterler Yazılım Teknik Şartnamesinde saptanmalıdır.

### 3.2.6. Yazılım Kabul ve Kabul Testi

Yazılım Teknik Şartnamesinde, Yazılım Geçici ve Kesin Kabul Prosedürleri belirtilmeli, Yazılım Kabul testleri ve ölçütleri tanımlanmalı ve kabul için yazılımla birlikte verilmesi gereken diğer materyelin (belgeler, kod listeleri, yazılım destek gereçleri, v.b.) listesi verilmelidir.

Yazılım Kabul Testi, yazılımın kabul ölçütlerini sağlayıp sağlamadığının ve müşterinin yazılımı kabul edip etmeyeceğini saptar, yazılımın kullanım ortamında yürütülür ve bağlayıcıdır [7].

Yazılım Teknik Şartnamesinde, yazılım kabul ölçütleri belirtilmeli ve Yazılım Kabul Test isterleri tanımlanmalıdır. Yazılım Kabul Test isterleri,

- 1) Test nesnelere (yazılım artsistemleri)
  - 2) Test edilecek özellikler
  - 3) Test nesnelere için kabul/ret ölçütleri
  - 4) Test belgeleri
  - 5) Test işlemleri
  - 6) Ortam koşulları
- olarak belirtilmelidir.

### 3.2.7. Yazılım Geliştirme Planı ve Yazılım Geliştirme Gözetim ve Denetimine Yönelik İsterler

Yazılım Geliştirme Planı, yazılımın yaşam döngüsünün her evresinde yürütülecek başlıca işleri, ulaşılabilecek aşamaları, bu aşamalarda ortaya çıkarılacak ürünleri, bu iş

**“Sipariş yoluyla yazılım ediniminde, müşteri kuruluş ile üretici firma arasında imzalanan sözleşme Yazılım Geliştirme Planını kapsamalıdır. \*\***

ve aşamalara ait takvim ve ayrılacak kaynaklarla birlikte belirten, söz konusu aşamalarda hedeflere ulaşıp ulaşılmadığının doğrulanma prosedürlerini tanımlayan, müşteri ile üretici arasındaki her aşamadaki ilişkileri belirleyen ve belgelendirme sistemini tanımlayan plandır [7].

Sipariş yoluyla yazılım ediniminde, müşteri kuruluş ile üretici firma arasında imzalanan sözleşme Yazılım Geliştirme Planını kapsamalıdır. Ya-

zılım Geliştirme Planı, yazılım üreticisi firma tarafından hazırlanır, teklifle birlikte sunulur ve müşteri kuruluşun kabulü sonucu sözleşmeyle kesinlik kazanır. Amacı, yazılım ihtiyacı olan kuruluş için yazılım geliştirme sürecini gözlenebilir ve denetlenebilir kılmaktır. Yazılım Geliştirme Planının kapsam ve ayrıntıları projenin türüne, büyüklüğüne ve karmaşıklığına göre farklılıklar gösterecektir. Müşteri kuruluş Yazılım Teknik Şartnamesinde Yazılım Geliştirme Planına yönelik isterlerini belirtmelidir.

Yazılım Teknik Şartnamesindeki yazılım geliştirme planıyla ilgili isterler, müşteri kuruluşun yazılımın geliştirilmesi sırasında talep ettiği denetim ve gözetimin ölçü ve tekniklerini saptar. Bu doğrultuda, sözleşmeyle bağlanacak olan plana kaynak teşkil etme amacını taşır.

Bu bağlamda müşteri kuruluş Yazılım Teknik Şartnamesinde:

- 1) Yazılım Değerlendirme faaliyetlerinin (Bkz. Bölüm 4.1) yürütülmesi, izlenmesi, belgelendirilmesi ve değerlendirilmesine ilişkin isterlerini ifade eder.
- 2) Yazılım geliştirme süreci aşamalarının ürünleri olan belgeler içerisinde, kendisine teslim edilmesi gereken belgeleri belirtir. (Tasarım belgeleri, Test Belgeleri, Test Raporları, Yazılım Değerlendirme Raporları, Test Planları, Kalite Denetim Plan ve Belgeleri, v.b.)
- 3) Yazılım Geliştirme süreci sonunda yazılımın toptan teslimi yerine, varsa, bağımsız yazılım altsistemlerini teker teker teslimini talep edebilir ve bu doğrultudaki isterlerini belirtir.
- 4) Geliştirme sürecinin çeşitli aşamalarında yazılımda gerçekleştirilen işlevselliği gözlemek üzere ara gösteriler (demonstration) talep edebilir ve bunları tanımlar.
- 5) Entegrasyon Testlerinin (Bkz. Bölüm 4.2) yürütülmesi, izlenmesi, belgelendirilmesi ve değerlendirilmesine ilişkin isterlerini ifade eder.

## KURULUŞLARIN YAZILIM ŞARTNAMESİ, NİTELİKLERİ, BAKIM VE EĞİTİM STANDARTLARI ARAŞTIRMASI

Geçici Kabul öncesi yazılım faaliyetlerinin kısmi ücretlendirilmesi, Yazılım Geliştirme Planında belirtilen aşamalar temel alınarak belirlenir. Sözleşme, Yazılım Geliştirme Planındaki belirli aşamaların geçilmesi durumunda o aşamalara ilişkin ödenecek proje bedeli oranını gösterir. Bu bedel, ilgili aşamaya ilişkin ürünlerin (alt sistem ve/veya belgeler) teslimi karşılığında ödenir. Sözleşme, ara ödemenin yapılacağı aşamayı (örneğin, yazılımın bir alt-sistemini teslimi, ya da tasarım değerlendirmesinin olumlu sonuçlanması, entegrasyon testlerine başlanması, v.b.), bu aşamaya erişildiğinde teslim edilmesi gereken ürünleri ve ödenecek bedeli tam ve açık olarak tanımlamalıdır. Ara ödemelerle ilgili öneriler Bölüm 4.3'de verilmektedir.

Yazılım Geliştirme Planındaki aşamalara ulaşmadaki olası gecikmeler, sözleşme ile tanımlanan cezai yaptırıma tabi tutulabilir.

Müşteri kuruluş, Yazılım Geliştirme Planının izlenmesini ya da yazılım geliştirme süreci içerisinde belirli denetim faaliyetlerinin (doğrulama ya da testler) yürütülmesini üretici firmanın dışında ayrı bir firma ya da kuruluştan talep edebilir. Yazılım Teknik Şartnamesinde, varsa bu talep ifade edilmelidir. Bu durumda, yazılım geliştirme sürecini denetleyecek ve/veya bağımsız bir kuruluş olarak testleri ya da doğrulama faaliyetlerini yürütecek olan firma ya da kuruluşla ilişkiler ayrı bir sözleşme konusudur.

### 3.3. Yazılım Teknik Şartnamesinin ve Yazılım Özellikleri ve İşlevleri Tanım Belgesinin Hazırlanması

Yazılım Teknik Şartnamesi ve Yazılım Özellikleri ve İşlevleri Tanım Belgesi, ihtiyaç duyulan yazılımın karmaşıklık derecesine ve büyüklüğüne bağlı olarak iki şekilde hazırlanabilir.

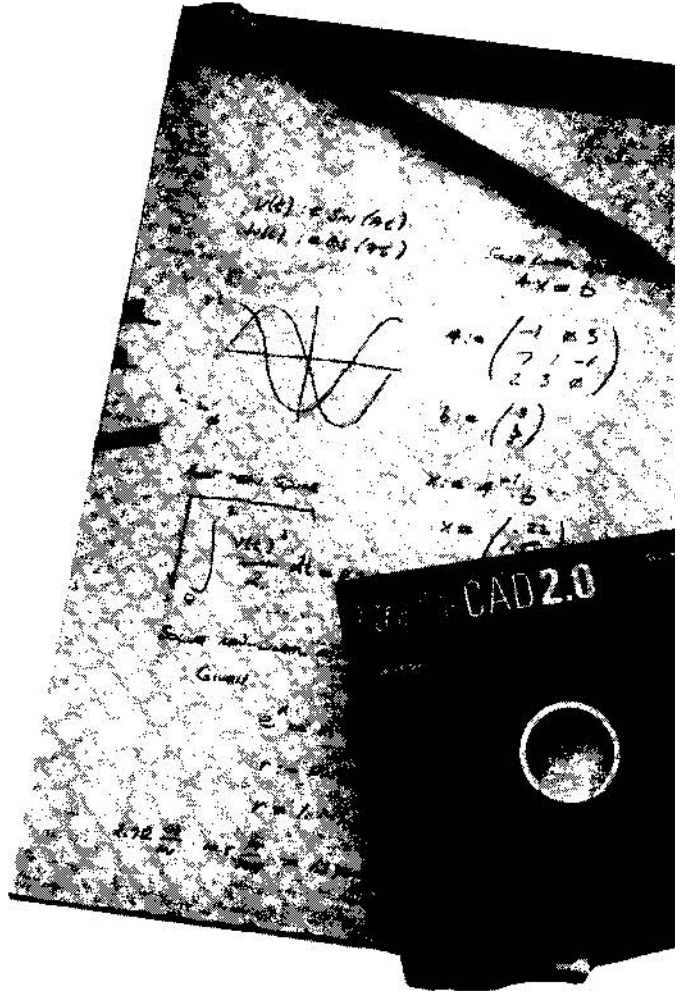
1) İhtiyaç duyulan yazılımın görece basit ve küçük ölçekte olduğu ve yazılım ihtiyacı olan kuruluşun bünyesinde sözkonusu yazılımı teknik olarak tanımlamaya yetecek birikimin mevcut bulunduğu durumlarda adı geçen belgeler yazılım ihtiyacı olan kuruluş tarafından hazırlanır.

2) Yazılımın karmaşıklık derecesi ve büyüklüğünün, yazılım isterlerinin tanımlanmasında ve teknik olarak ifade edilmesinde özel uzmanlığı gerekli kıldığı durumlar olabilir. Sağlıklı bir Yazılım Teknik Şartnamesinin ve Yazılım Özellikleri ve İşlevleri Tanım Belgesinin yazılması, bu gibi durumlarda yazılım ihtiyacı olan kuruluşun ya da satıcı/üretici firmanın sahip olduğu uzmanlığın dışında olabilir. Böyle durumlarda yazılım ihtiyacı olan kuruluş, Yazılım Teknik Şartnamesinin ve Yazılım Özellikleri ve İşlevleri Belgesinin hazırlanması işini bu konuda uzman firma ya da kuruluşlara devreder.

Yazılım Teknik Şartnamesi ve Yazılım Özellikleri ve İşlevleri Tanım Belgesinin Uzman Firma ya da Kuruluşlar tarafından Hazırlanması

Bu durumda yazılım ihtiyacı olan kuruluş, Kullanıcı Özellikleri Tanım Belgesini hazırlayarak Yazılım Teknik Şartnamesinin ve Yazılım Özellikleri ve İşlevleri Tanım Belgesinin yazılmasını ayrı bir proje olarak bir ya da bir kaç firma ya da kuruluştan talep eder.

Yazılım ihtiyacı olan kuruluş, bu firma ya da kuruluşlar tarafından hazırlanan Yazılım Teknik Şartnamesinden birini kabul edebilir, ya da bunları kullanarak kendisi yeni bir şartname hazırlayabilir. Her durumda, Yazılım Teknik Şartnamesini hazırlayan firma ya da kuruluşların her biri ücretlendirilir.



#### 4. YAZILIM GELİŞTİRME GÖZETİM VE DENETİMİ

##### 4.1. Yazılım Değerlendirme Faaliyetleri

Sipariş yoluyla yazılım ediniminde, yazılım üreticisi firma, geliştirme sürecinin çeşitli aşamalarında ortaya çıkan ürün ve belgelerin istenen özellikleri sağlayıp sağlamadığının tespiti amacıyla Yazılım Geliştirme Planı ve Yazılım Kalite Planı uyarınca Yazılım Değerlendirme faaliyetleri yürütür. Bu faaliyetleri müşteri kuruluşun izlemesi ve/veya müşteri kuruluş temsilcilerinin katılımı, Yazılım Teknik Şartnamesindeki istekler doğrultusunda saptanır ve Yazılım Geliştirme Planı'nda ve Yazılım Kalite Planı'nda yer alır.

Yazılım Değerlendirme, yazılım ürünlerinin ya da yazılım geliştirme projesinin eriştiği durumun, hedeflere ulaşmadığının incelenmesi, değerlendirilmesi ve düzeltici ve/veya geliştirici önerilerin oluşturulması faaliyetidir. Yazılım Değerlendirme faaliyetinin,

- 1) Yönetim Değerlendirmesi (Management review),
- 2) Teknik Değerlendirme (Technical review)
- 3) Yazılım inceleme (Software inspectbn),
- 4) Yazılım Tarama (Walkthrough)

olarak dört genel türü tanımlanabilir [17].

Yönetim Değerlendirmesi, yazılım geliştirme faaliyetlerinin Yazılım Geliştirme Planına göre ilerlemesinin sağlanması ve kaynakların bu hedefler doğrultusunda uygun olarak kullanılması amacıyla yürütülür. Proje yönetim belgeleri ve raporları, tamamlanmış diğer değerlendirme faaliyetlerinin raporları ve tamamlanmış test raporları Yönetim Değerlendirmesi faaliyetinin kaynaklarını oluşturur.

Teknik Değerlendirme, Yazılım İnceleme ve Yazılım Tarama faaliyetleri ise, yazılım ürünlerinin istekleri sağlayıp sağlamadığının ve standartları ve kılavuzlara uyumluluğunun incelenmesi, denetlenmesi, hataların araştırılması, gözlenen problemlerin belgelendirilmesi amacını taşır. Tasarıma ya da kodlamaya alternatif üretilmesi bu faaliyetlerin kapsamının tamamen dışındadır. Tasarım belgeleri, test belgeleri, test planları, kaynak kodlar ve test raporları başlıca kaynak materyeli oluşturur.

Teknik Değerlendirme, sonucun değerlendirilmesine yöneliktir, dolayısıyla bir aşama sonrası faaliyetidir (Örneğin, tasarım sonrası Tasarım Değerlendirme). Teknik Değerlendirme ekibinde müşteri temsilcisi ya da bağımsız bir denetleyici bulunabilir.

Yazılım Tarama, ilgili ürünün tasarım, kodlama ve test

süreçlerinden sorumlu personelin kendi arasında yürütülen ve süreç içi bir faaliyettir. Hata ve uyumsuzlukların bulunmasına, çözüm ve alternatifin üretilmesine ve geliştirme ekibinin iç eğitimine yöneliktir.

Yazılım İnceleme'de ise, inceleme ekibini yazılım ürünün geliştirilmesinde yer almayan farklı bir teknik grup oluşturur. Bu çalışmanın amaçları da hata ve uyumsuzlukların bulunmasıdır, ancak tasarım ya da kodlamaya alternatif geliştirilmesi Yazılım inceleme grubunun amaçlarının ve sorumluluklarının tamamen dışındadır.

Yazılım Değerlendirme faaliyetleri, tasarımın, kodlamanın, test sonuçlarının değerlendirilmesi için ve ürünün sonlandırılması ve değerlendirilmesi amacıyla, geliştirme süreci içinde farklı aşamalarda düzenlenebilir.

Üretici firma, Yazılım Geliştirme Planı'nda, geliştirme süreci içerisinde yürütülecek Yazılım Değerlendirme faaliyetlerini tanımlar ve planlar. Planlanan her değerlendirme faaliyeti için, aşağıdaki bilgilerin saptanması ve geliştirme planında belgelendirilmesi gerekir:

- 1) Amaç,
- 2) Değerlendirme çalışmalarının başlama zamanı, planlanan süresi, düzenlenecek toplantıların zamanı ya da görelisi planı,
- 3) Değerlendirme ekibinin tanımı ve sorumlulukları,
- 4) İncelenecek ya da teftiş edilecek materyalin listesi,
- 5) Yöntem,
- 6) Değerlendirme raporunun kapsamı, dağıtımı ve kullanımı,
- 7) Faaliyetin sonlandırılma ölçütü.

##### 4.2 Entegrasyon Testleri

Entegrasyon testi, yazılım modüllerinin, modül testleri yapıldıktan sonra, diğer modüller ve donanım ile belirli bir plan dahilinde, teker teker entegre edilerek test edilmesi sürecine denir [7]. Yazılımın kodlanması, test planındaki entegrasyon hedeflerine uygun sırada gerçekleştirilir.

Sistem testi, yazılım ve donanımın, entegrasyon sonrası, bir bütün olarak test edilmesi sürecidir [7]. Donanımın hazır olduğu ve yazılımla birlikte geliştirilmediği durumlarda bu test, tüm yazılım özelliklerinin ve donanımla birlikte işleyişinin testi anlamını taşır ve bu bağlamda Yazılım Teknik Şartnamesi ve Yazılım Test Planının konularından birini oluşturur. Donanım ve yazılımın birlikte geliştirilmesi ve bu durumdaki sistem testleri ise bu raporun kapsamının dışındadır.



Test Planı, taslak halinde tasarım öncesinde oluşturulmalıdır [24]. Planın ayrıntılandırılması ve attsistemler ile ilgili altplanlar süreç içerisinde oluşturulur ve plan güncellenir. Bu plan, müşteri kuruluş Yazılım Teknik Şartnamesinde talep ettiği takdirde, Yazılım Geliştirme Planı ile birlikte müşteri kuruluşun ya da müşteri kuruluşun seçtiği denetleyici firma ya da kuruluşun incelemesine sunulur.

#### 4.3. Yazılım Geliştirme Sürecinde Öcretlendirme Dağılımı

Yazılım siparişinde kısmi ücretlendirme, Bölüm 3.2.7'de belirtildiği üzere sözleşme tarafından belirlenen aşamalar esas alınarak yapılır. Ancak, sözleşmede belirlenecek bu kısmi ödemeler için, aşağıda belirtilen ödeme planı, genel bir örnek olarak, sunulabilir:

- 1) Sözleşmenin bağlanmasıyla birlikte toplam proje bedelinin %10-15'i oranında ilk ödeme (avans) yapılır.
- 2) Kodlamaya geçilmeden önce, toplam proje bedelinin %25-40'ı oranında ödeme gerçekleştirilir.
- 3) Entegrasyon testlerine geçilmeden önce toplam proje bedelinin %50-60'ı oranında ödeme gerçekleştirilir.
- 4) Yazılım alt sistemlerinin peyder pey tesliminde yapılan ödemelerde, yukarıdaki (Madde 1-3) ilkeler gözönüne alınır.
- 5) Geçici Kabul ile birlikte toplam proje bedelinin %90-95'i ödenir.
- 6) Kesin Kabul sonucunda bütün proje bedeli ödenir.

#### 4.4. Yazılım Geliştirme Sürecinde Değişiklik İstekleri

Sipariş yoluyla yazılım ediniminde, sözleşmeyle kesinlik kazanan Yazılım Teknik Şartnamesi bağlayıcıdır. Yazılım Teknik Şartnamesinde ifade edilen yazılım isteklerinden herhangi birinde, ya da birkaçında değişiklik, bir özellikten tamamen vazgeçme ya da yeni bir özellik isteme şeklindeki her tür talep, hangi taraftan gelirse gelsin yeni bir pazarlığa yolaçabilir. Müşteri kuruluş ve üretici firma arasında bu tür isteklerin değerlendirilmesi sonucunda, talebin Yazılım Geliştirme Planı ya da yazılım maliyeti üzerinde herhangi önemli bir değişiklik getirmediği üzerinde mutabakat sağlanırsa Yazılım Teknik Şartnamesine, sözleşmenin diğer koşulları sabit kalmak kaydıyla gereken değişiklikler yapılabilir. İstenen değişiklik, program ve maliyete etki ediyor ise, sözleşmenin feshine ve yeni bir Yazılım Teknik Şartnamesi ile sözleşmeye gidilebilir.

Yazılım değişiklik önerisinin taraflarca onaylanması durumunda, bütün ilgili ürün ve belgelerde gereken güncelleştirmenin yapılması gerekmektedir.

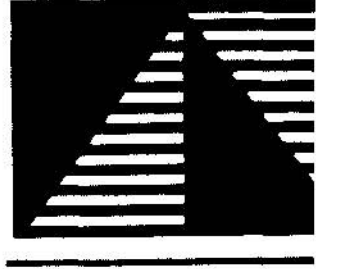
#### 5. YAZILIM TESTLERİ VE YAZILIM KABUL PROSEDÜRLERİ

Yazılım, sözleşme ile kesinlik kazanmış Yazılım Teknik Şartnamesinde tanımlanan Kabul Testleriyle Geçici Kabul alır. Geçici Kabul alan yazılım için, sözleşmede belirlenen oranda Ücretlendirme yapılır. Bu oran için toplam proje bedelinin %95-100'ü önerilmektedir.

Geçici Kabul alan yazılım kullanıma alınarak, sözleşme ile belirlenen süre içerisinde olağan çalışma koşullarında kullanılır. Olağan çalışma koşullarındaki bu kullanımın sonucu tatminkar ise yazılım kesin kabul kazanır. Yazılımın Kesin kabul kazanmasıyla bedelinin geri kalanı da ödenir. Bu dönem içerisinde yazılım hataları ve/veya Teknik Şartnamede belirtilen yazılım isteklerinin birinin ya da birkaçının karşılanamamış olduğu gözlemlenebilir. Bu durumun kotarılması için,

- 1) Yazılım ihtiyacı olan kuruluşun durumu üretici firmaya ne kadar süre içerisinde ve ne şekilde bildireceği,
- 2) Üretici firmanın sorunu ne kadar süre içerisinde çözümlenmesinin beklendiği; sözleşmeyle saptanır.

Kesin kabul öncesi bu tür bakım hizmetleri ücretsiz olmalıdır.



## 6. YAZILIM BELGELENDİRME

### 6.1. Yazılımla Birlikte Teslimi Zorunlu Belgeler

Yazılımın tür, büyüklük ve karmaşıklığından ve yazılım edinme yolundan bağımsız olarak, müşteri kuruluşa yazılımla birlikte, Kullanıcı kılavuzu (ve/veya Kullanıcı Başvuru kılavuzu) ve İşletme kılavuzunun teslimi gerekmektedir. Kullanıcının aynı zamanda yazılımın işleticisi olduğu, ayrı bir işleticiye ihtiyaç duyulmayan kimi yazılımlarda, bu iki belge tek bir belgede toplanabilir.

Yazılım ihtiyacı olan kuruluş, bu belgenin dışında belgeler talep edebilir ve/veya belgelerin içerikleri ve biçimleri üzerinde isteklerde bulunabilir. Yazılım beklendirmesine ilişkin bütün bu istekler Yazılım Teknik Şartnamesinde yer almalıdır.

Kullanıcı Belgelerin hazırlanmasında, Kaynak [20] referans alınabilir.

#### 6.1.1. Kullanıcı Kılavuzu ve Kullanıcı Başvuru Kılavuzu

Kullanıcı kılavuzu şu bilgileri kapsamalıdır:

- i) Yazılımın temel işlevleri ve özellikleri tanımlanmalıdır.
- ii) Yazılımın giriş komutları ve formatları, giriş ortamı, giriş prosedürleri, yazılımın işlemesi için zorunlu giriş değerleri ile seçeneğe bağlı özellikler için gerekli giriş değerleri verilmelidir. Kullanım için gerekli bu bilgiler, birkaç yaygın kullanım örneği alınarak, yazılım temel işlevlerinin yerine getirilebilmesi için ardarda yapılması gereken işlemler dizisi olarak sunulmalıdır.
- iii) Ekran ya da baskı formatları verilmelidir.
- iv) Kullanıcıyı ilgilendiren hata mesajları, format ve içerikleri ve bu mesajlar alındığında yerine getirilmesi gereken işlemler belirtilmelidir.

Kullanıcı Başvuru kılavuzu, yazılımın bütün giriş komutlarını, format ve açıklamaları ile birlikte sıralı olarak sunan başvuru belgesidir. Her tür yazılım için Kullanıcı Kılavuzu ile Kullanıcı Başvuru kılavuzunun ayrı iki belge olarak teslimi gerekmez. Bu gibi durumlarda, içeriği her ikisini de kapsayan tek bir Kullanıcı Kılavuzu yeterli olabilir.

#### 6.1.2. İşletme Kılavuzu

İşletme Kılavuzu aşağıdaki bilgileri içerir:

- i) Yazılımın hizmete alınması, ortama uyumun gerçekleştirilmesi ve konfigürasyon için gereken bilgi ve prosedürler.

ii) Yazılımın çalıştırılması için gerekli komut ve prosedürler; durdurulması için gerekli komut ve prosedürler; herhangi bir nedenle yazılım durduğunda yeniden başlatılması için gereken komut ve prosedürler.

iii) Hata mesajları, bunların format ve anlamları, hata durumunda yapılması gereken işlemler, hata tanıma işlemleri (diagnostics).

iv) Yazılımın performansını gözlemlemek için gerekli komut ve prosedürler.

### 6.2. Bakım Kılavuzu

Bakım kılavuzu genel olarak aşağıdaki bilgileri içerir:

- i) Yazılımın tanımı, teknik özellikleri ve işlevler;
- ii) Ayrıntılı Tasarım; (Bütün tasarım adımları verilmelidir. "Optimality" gibi tasarımda hemen göze çarpmayacak özellikler açıkça belirtilmelidir.)
- iii) Okunabilir ve anlamlı açıklamalar bulunduran kaynak kod;
- iv) Test planları, modül testlerinin ve genel testlerin sonuçları.

Bakım kılavuzunun müşteri kuruluşa hangi durumlarda verileceği ve kapsamının saptanması konusu Bölüm 7'de ele alınmaktadır.

## 7. BAKIM, EĞİTİM VE DESTEK HİZMETLERİ

### 7.1. Bakım Hizmetleri

Yazılımın yaşam süresi boyunca;

- 1) Yazılım hatalarını ve performans yetersizliklerini saptamak ve düzeltmek,
- 2) Yazılımı, veri yapılarında ve/veya çalışma ortamında ortaya çıkan değişikliklere uyarlamak,
- 3) Yazılımın performansını, maliyet-etkinliğini, işlem verimliliğini vb. özelliklerini iyileştirmek,

amacıyla yazılımda düzeltici (corrective), uyarlayıcı (adaptive) ya da iyileştirici (perfective) değişiklikler yapma gerekliliği ortaya çıkacaktır. Yazılımın müşteriye tesliminden sonra gerekli olan bütün bu değişiklikler yazılımın bakım sürecini oluşturur [31].

Bakım hizmetleri içerisinde, yukarıda Madde 1'de tanımlanan hata düzeltmeye yönelik bakım Düzeltici Bakım olarak nitelendirilir. Hatasız yazılım, bir hedef olmakla birlikte gerçekçi olmaktan uzaktır. Yazılım hataları, yazılımın müşteriye teslimini izleyen ilk dönemde ağırlıkla or-

taya çıkarlar. Hatalı tasarım, hatalı kodlama ya da yazılım isterlerinin yanlış yorumlanmasından kaynaklanan bu tür yazılım hatalarını onarmaya yönelik düzeltici bakım, bakım hizmetleri içindeki ağırlığın bir süre sonra yitirir. Ancak, yazılımın yaşam süresi içinde yazılımı değişen koşullara uyarlamak ya da yeni özellikler eklemek ve/veya özelliklerini geliştirmek amacıyla değişiklikler gerekli olacaktır. Bu tür yazılım değişikliklerini karşılamaya yönelik, sırasıyla madde 2 ve 3'de tanımlanan Uyarlayıcı Bakım ve İyileştirici Bakım hizmetleri sonunda yazılımda yeni hatalar üretilmesi kaçınılmazdır. Böylece, yazılımın artık ihtiyaçları karşılamamaya başladığı dönemde düzeltici bakımın ağırlığı yeniden artmaya başlayacaktır. Yazılımın yaşam süresi boyunca ortaya çıkacak bakım hizmetlerinin yoğunluğundaki değişim için Şekil 4'deki eğri genel ve yaygın bir gösterimdir.

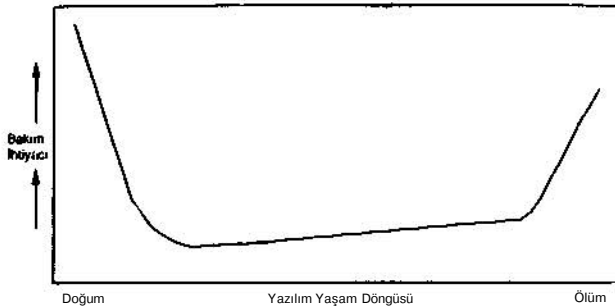
#### Bakım Garantisi:

Hata onarımına ilişkin düzeltici bakım yükümlülükleri asıl olarak sözleşmeyle saptanır. Ancak satıcı/üretici firma, yazılımın tesliminden sonra belirli bir süre için, örneğin, sipariş yoluyla yazılım ediniminde. Kesin Kabulü takiben ilk bir yıl içerisinde, ücretsiz bakım garantisi vermelidir.

#### Bakım Anlaşmaları:

Bakım garanti süresi sonrasındaki düzeltici bakım yükümlülükleri Bakım Anlaşmaları çerçevesinde yerine getirilecektir. Sözleşme, sözleşme sonrası düzeltici bakım ihtiyaçlarının karşılanma yöntemlerini ve tarafların sorumluluklarını belirleyerek ilerdeki bakım anlaşmalarının çerçevesini çizmelidir.

Uyarlama ve/veya iyileştirmeye yönelik bakım hizmetleri asıl olarak ayrı sözleşme konularını oluşturur. Ancak, yeni versiyon geliştirme ya da güncellemeye yönelik yükümlülükler ve Yazılım Teknik Şartnamesinde belirtilen (Bölüm 3.2.5) diğer bakım isterlerinin karşılanma yöntemleri sözleşmeyle bağlanır.



Şekil 4 - YAZILIM yaşam Döngüsü Boyunca Bakım İhtiyaçlarının Değişimleri

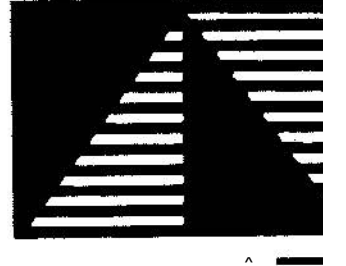
#### Bakım Kılavuzu:

Bakım Kılavuzu, Bölüm 6.2'de belirtildiği gibi, yazılım değişikliklerinin koterilmesine yönelik kapsamlı bir yazılım belgeleri grubudur. Bakım Kılavuzunun müşteri kuruluşu teslimi, bakıma yönelik değişiklikleri yürütebilecek bir yazılım ekibinin müşteri kuruluş bünyesinde olması durumunda ya da bakım hizmetleri yazılım üreticisi firmanın dışında bir başka firma tarafından koterilcek ise anlam kazanır. Her iki durumda da yazılımın mülkiyeti konusu gündeme gelir. Dolayısıyla Bakım Kılavuzunun müşteri kuruluşu teslimi, yukarıdaki ihtiyaçların varlığı durumunda taraflar arasında çözülmesi gereken bir mülkiyet anlaşmasıyla bağlanabilir.

#### 7.2. Eğitim Hizmetleri

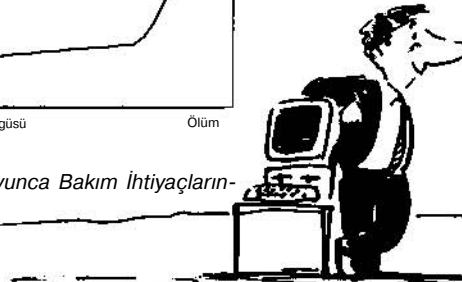
Müşteri kuruluş, sundaki ihtiyaçlarını ve isterlerini Yazılım Teknik Şartnamesinde ifade eder. Taraflar arasında bağlanan sözleşme, satıcı ya da üretici firmanın vereceği eğitim hizmetlerini belirler. Bu hizmetler, kullanıcıya ya da işletme personeline yönelik olabilir, basit ve küçük yazılımlar için kısa süreli eğitim programı olabileceği gibi, daha karmaşık yazılımlar için yaygın ve yoğun eğitim programlarına ihtiyaç duyulabilir. Yazılım Teknik Şartnamesi, varsa Yardım münüsü (Help Menu) talebini ve bu mününün özelliklerini belirtir. Ayrı bir eğitici yazılım ve/veya Eğitim Belgesi ihtiyacı yine Yazılım Teknik Şartnamesinde ifade edilir.

eğitim konu-



#### 7.3. Destek Hizmetleri

Yazılımın kullanım süreci boyunca yazılımda değişiklik gerektirmeyen, ancak kullanıcı ya da işletici personelin yazılımın olanaklarını daha iyi değerlendirebilmeleri, hata belirleme ya da performans ölçme türünden işleyiş anlamaya yönelik faaliyetleri yürütebilmeleri, v.b. işler için üretici ya da satıcı firmadan bekledikleri destek hizmetleri olacaktır. Bu tür yazılım destek isterleri Yazılım Teknik Şartnamesinde ifade edilir ve sözleşme ile yükümlülükler saptanır.



**K A Y N A K Ç A**

1. DOD-STD-480, Configuration Control-Engineering Changes, Deviations and Waivers.
2. DOD-STD-1467, Software Support Environment,
3. DOD-STD-2167. Defense System Software Development.
4. DOD-STD-2167A, Software Requirements Specifications.
5. DOD-STD-2168, Software Quality Evaluation.
6. DOD-STD-7935, Automated Data Systems Documentation.
7. ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.
8. ANSI/IEEE Std 730.1-1989, IEEE Software Quality Assurance Plans.
9. ANSI/IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans.
10. ANSI/IEEE Std 829-1983. IEEE Standard for Software Test Documentation.
11. ANSI/IEEE Std 830-1984, IEEE Guide to Software Requirements Specifications.
12. ANSI/IEEE Std 983-1986, IEEE Guide for Software Quality Assurance Planning.
13. ANSI/IEEE Std 1002-1987, IEEE Standard Taxonomy of Software Engineering Standards.
14. ANSI/IEEE Std 1008-1987. IEEE Standard for Software Unit Testing.
15. ANSI/IEEE Std 1012-1986, IEEE Standard for Software Verification and Validation Plans.
16. ANSI/IEEE Std 1016-1987. IEEE Recommended Practice for Software Design Descriptions.
17. ANSI/IEEE Std 1028-1988, IEEE Standard for Software Reviews and Audits.
18. ANSI/IEEE Std 1042-1987, IEEE Guide to Software Configuration Management.
19. ANSI/IEEE Std 1058-1987. IEEE Standard for Software Project Management Plans.
20. ANSI/IEEE Std 1063-1988, IEEE Standard for Software User Documentation.
21. IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.
22. IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.
23. TR-TSY-000179. Bellcore Technical Reference, Software Quality Program Generic Requirements.
24. Deutsch, M.S., White, R.R. Software Quality Engineering: A Total Technical and Management Approach, Prentice Hall, 1988.
25. Ledgard, H., Tauner, J., Software Engineering Concepts, Addison-Wesley, 1987.
26. Fox, J.M.. Software and its Developments, Prentice-Hall, 1982.
27. Matsumoto, Y., Ohno, Y.. Japanese Perspectives in Software Engineering, Addison-Wesley, 1987.
28. DeMarco, T., Structured Analysis and System Specification, Yourdon Press, 1979.
29. Macro, A., Buxton, J., The Craft of Software Engineering, Addison-Wesley, 1967.
30. Booch. G., Software Engineering with Ada, The Benjamin/Cummings Pub. Co., 1983.
31. Martin, J.. Software Maintenance, Prentice-Hall, 1983.
32. C. Summers, Software Quality Assurance, Reliability and Testing, UNICOM Technical Press, 1987.
33. TSE 5881. Bilgisayar ve Veri İletişiminde Kullanılan 8 Bit Uzunluğunda Türkçe Karakter Kodlama Kurulması.
34. ISO 8859 Text Communication - Registration of graphics character subrepertoires-8 bit single byte coded graphic character sets.



**EDA BATKAL**

Odamız 13745 Sicil Nolu  
Üyesi  
Eda BAYKAL'ı kaybettik.

AİLESİNE, YAKINLARINA  
VE ODAMIZ  
TOPLULUĞUNA  
BAŞSAĞLIĞI DİLERİZ.

**M.FATİH DİNÇER**

Elektrik Mühendisi  
M.Fatih DİNÇER'i  
kaybettik!

AİLESİNE, YAKINLARINA  
VE ODAMIZ  
TOPLULUĞUNA  
BAŞSAĞLIĞI DİLERİZ.