

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

77723

YAZILIMDA KALİTE MODELLERİNİN DEĞERLENDİRİLMESİ

Ender ŞAHİNASLAN
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TEZ DANIŞMANI
Yrd.Doç.Dr E. Murat ESİN

GEBZE - 1998

Bu tez çalışması, G.Y.T.E Mühendislik ve Fen Bilimleri Enstitüsü yönetim Kurulu'nun **10.07.1998** tarih ve **98/18** sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği** Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

JÜRI

Murat Esin

ÜYE : Yrd.Doç.Dr. E. Murat ESİN
(Tez Danışmanı) Gebze Yüksek Teknoloji Enstitüsü

İbrahim Soğukpınar

ÜYE : Yrd.Doç.Dr. İbrahim SOĞUKPINAR
Gebze Yüksek Teknoloji Enstitüsü

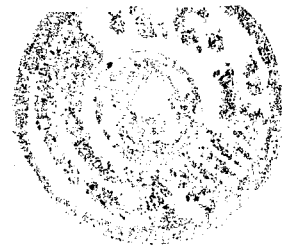
Ertuğrul Karaçuha

ÜYE : Yrd.Doç.Dr. Ertuğrul KARAÇUHA
Gebze Yüksek Teknoloji Enstitüsü

ONAY

G.Y.T.E Mühendislik ve Fen Bilimleri Enstitüsü yönetim Kurulu'nun **23/07/1998** tarih ve **98/20** sayılı kararı.





ÖZET

Yazılım tasarımı ve geliştirme projelerinin büyümesiyle 1970'li yıllarda ortaya çıkan yazılımda kalite sorununun daha sonraları yazılım krizine dönüştü. Yazılımın doğası nedeniyle diğer endüstriyel ürünlerden farklılık göstermesi, kalite ile ilgili olan diğer model ve standartların uygulanmasını güçleştirmiş olup yazılıma özgü yeni yaklaşımlar gerektirmiştir.

Son on yılda yazılım geliştirmede kullanılan kalite sistemleri ve süreçlerini değerlendirmek, yazılımda kalite sertifikasyonunu sağlamak, süreçleri iyileştirmek ve yetenek belirlemek için çeşitli modeller geliştirilmiştir. Modeller arasında kavram düzeyinde temel ilişkiler bulunsa da, başta amaçlar olmak üzere model ve standartlar arasında bire bir izdüşüm ve tam bir uyum olduğu söylenemez. 2000'li yıllara doğru girerken dünyada yazılım üreticileri ve müşterileri arasında kalite sorununa yaklaşımda genel bir görüş birliği yoktur.

Bu çalışmanın amacı, yazılımda temel kalite kavramaları ile yazılımda kalite sertifikasyonu, süreç iyileştirme ve yetenek belirleme amacıyla ortaya çıkan TickIT (ISO 9001), CMM ve SPICE modellerini tanıtmak, aralarındaki benzerlik ve farklılıkları ortaya çıkartmaktır. Yazılımda kaliteli ve güvenilir üretime önem veren, bu amaçla kaynak ayırmak isteyen kuruluşlar için mevcut yeteneklerini ve kalite sistemlerinin eksikliklerini görebilmeleri ve kendilerine uygun model seçebilmeleri konusunda yol gösterici olacaktır.



SUMMARY

In 1970's, when the software design and development projects started becoming more and more complex, quality considerations resulted in software crisis. Due to the nature of software and its differences from other industrial products, other quality related standards were difficult to apply to software and this required new and unique approaches to software quality.

In the past ten years, to assess quality systems and processes in software development, to get software quality certificates, to improve the processes and to determine capability, several models were developed. Although there are some basic relationships at the conceptual level among these models, it is difficult to say that there are one-to-one correspondence and consistency among them. Towards 2000's, there is no single view to software quality problem among software developers and customers.

The purpose of this study is to show the fundamental software quality concepts and to explain TickIT (ISO 9001), CMM and SPICE models which are developed for software quality certification, process improvement and capability determination and to show the similarities and differences among these models. Another purpose is to guide the institutions, which care about quality and reliable software development and want to allocate resources for this purpose, so that they can realize their current capabilities and deficiencies of their quality systems and to help them in choosing among these software quality models.



TEŞEKKÜR

Bu tez konusunu seçmemde beni yönlendiren, çalışmalarım sırasında tecrübelerinden, bilgilerinden istifade ettiğim, gerekli kaynakların sağlanmasında yardımcı olan TÜBİTAK Marmara Araştırma Enstitüsü Bilişim Teknolojileri Enstitüsü Müdürü sayın Doç.Dr.Fuat İNCE ve Yazılım Mühendisliği Bölüm Başkanı sayın Selçuk TARAL Beylere, aynı ilgi ve alakayı gösteren, bizlere güvenen, yön veren, yardımlarını hep yanımda gördüğüm kıymetli hocam sayın Yrd.Doç.Dr. E.Murat ESİN'e, çalışmalarım gerçekleştirmede bana sağladığı imkanlar ve huzurlu bir aile ortamı için Gebze Yüksek Teknoloji Enstitüsü ve Bilgisayar Mühendisliği Bölümüne, tüm çalışma arkadaşlarıma, hayatımın dönüm noktalarından biri olan akademik çalışmalara başlamaya karar vermemde ve sonrasında desteklerini hiç esirgemeyen babam Ahmet Turan ŞAHİNASLAN başta olmak üzere değerli aileme ve çalışmalarım sırasında emeği geçen herkese teşekkürlerimi borç bilirim.

Ender ŞAHİNASLAN



İÇİNDEKİLER

	<u>SAYFA</u>
ÖZET	IV
SUMMARY	V
TEŞEKKÜR	VI
İÇİNDEKİLER	VII
SİMGELER VE KISALTMALAR	X
ŞEKİLLER DİZİNİ	XI
ÇİZELGELER	XII
1. YAZILIMDA KALİTE	1
1.1. Giriş	1
1.2. Kalite Kavramı	1
1.3. Yazılımda Kalitenin Anlamı	2
1.3.1. Kalite Faktörleri	2
1.4. Yazılımda Kalite Güvencesinin Amaçları	4
1.5. Kalite Sistemi	4
1.5.1. Kalite Sisteminin Kullanımı	5
1.6. Kalite Yönetimi	5
1.6.1. Üst Yönetimin Sorumluluğu	6
1.6.1. Günlük Sorunlar	7
1.7. Eğitim	7
1.8. Dokümente Edilmiş Kalite Sistemi	8
1.9. Kalitenin Maliyeti	8
2. YAZILIM VE SÜREÇ/KALİTE MODELLERİ	10
2.1. Giriş	10
2.2. Süreç Kavramı	11
2.3. Süreç/Kalite Modellerinin Gelişimi	11
2.4. ISO 9000 ve TickIT Modeli	12
2.4.1. TickIT Kalite Sistemleri Kavramları	13
2.4.2. TickIT Kalite Sistemi Anahtar Dokümanları	14
2.4.3. TickIT Alt Yapısı ve Organizasyonlarla İlişkisi	14
2.4.4. TickIT Kapsamı	16
2.4.5. TickIT Akreditasyon Süreci	16
2.4.6. TickIT Sertifikasyon Süreci	16
2.4.7. TickIT Değerlendirmesi	17
2.4.8. TickIT Denetim Terminolojisi	17
2.4.9. TickIT Yaklaşımı	18
2.4.10. TickIT Müşteri Rehberi	18
2.4.10.1 ISO 9000-3'de Kalitenin Başarılmasında Müşterinin Rolü	19
2.4.10.2 Müşteri-Satıcı İlişkisi	20
2.4.10.3 Anahtar Müşteri Aktiviteleri	20
2.4.11. TickIT-Satıcı Rehberi	21
2.4.11.1 Satıcı Firma Açısından Yazılım Kalite İsterleri	22
2.4.12. TickIT Kalite Sisteminin Geliştirilmesi	22
2.4.13. ISO 9001 Uyumlu Kalite Sisteminin Geliştirilmesindeki Temel Süreçler	23



2.4.14.	TickIT Süreç Örneği	24
2.4.15.	Sonuç	25
2.5.	Süreç Olgunluk Modeli (CMM-Capability Maturity Model)	25
2.5.1.	Giriş	25
2.5.2.	Süreç Olgunluk Çerçevesi	26
2.5.3.	Süreç Olgunluk Kavramları	26
2.5.4.	Süreç Olgunluk Model Yapısı	27
2.5.5.	Olgun ve Olgun Olmayan Yazılım Organizasyonları	28
2.5.6.	Süreç Olgunluk/Yetenek Seviyeleri	28
2.5.7.	Süreç Olgunluk Seviyelerini Anlama	30
2.5.8.	Anahtar Süreç Alanları	31
2.5.9.	Ortak Özellikler	33
2.5.10.	CMM Değerlendirme Sonuçları	35
3.	YAZILIM SÜREÇ DEĞERLENDİRMESİ VE SPICE MODELİ	36
3.1.	Giriş	36
3.2.	Genel Kavramlar	41
3.2.1.	Süreç Değerlendirme İçeriği	42
3.2.2.	Süreç Değerlendirme Mimarisi	44
3.2.3.	Süreç Değerlendirme Göstergeleri	45
3.2.4.	Değerlendiricilerin Kalitesi	45
3.2.5.	Süreç İyileştirme Kavramları	46
3.2.6.	Süreç Yetenek Belirleme Kavramı	47
3.3.	SPICE Modeli	49
3.3.1.	Referans Modelin Kullanımı	49
3.3.2.	Referans Modelin Yapısı	51
3.3.3.	Süreç Boyutu	54
3.3.3.1.	Müşteri-Satıcı Süreç Kategorisi	55
3.3.3.2.	Mühendislik Süreç Kategorisi	57
3.3.3.3.	Destek Süreç Kategorisi	61
3.3.3.4.	Yönetim Süreç Kategorisi	65
3.3.3.5.	Organizasyon Süreç Kategorisi	67
3.3.4.	Yetenek Boyutu	70
3.3.4.1.	Seviye 0: Tamamlanmamış (Eksik) Süreç	71
3.3.4.2.	Seviye 1: Uygulanan (Varolan) Süreç	71
3.3.4.3.	Seviye 2: Yönetilen Süreç	71
3.3.4.4.	Seviye 3: Kurulmuş (Yerleşmiş) Süreç	72
3.3.4.5.	Seviye 4: Kestirilebilen (Tahmin Edilebilen) Süreç	72
3.3.4.6.	Seviye 5: En İyileşen (Uygun Hale Getirilmiş) Süreç	73
3.3.4.7.	Süreç Niteliklerinin Notlandırılması	73
3.3.4.7.1.	Süreç Nitelik Notlandırma Skalası	73
3.3.4.7.2.	Süreç Nitelik Notlaması	74
3.3.4.7.3.	Başvurulan Notlandırmalar	74
3.3.4.7.4.	Toplam Notlandırma	74
3.3.4.8.	Süreç Yetenek Seviye Modeli	74
3.3.4.8.1.	Süreç Yetenek Seviyeleri Başarısı	74



75

3.3.4.8.2. Yetenek Seviyeleri Toplamı

3.3.5. Bölüm Değerlendirme Sonuçları

76

4. YAZILIM SÜREÇ /KALİTE MODELLERİ KARŞILAŞTIRMASI

77

5. TÜRKİYE'DE YAPILAN ÇALIŞMALAR

83

6. SONUÇLAR

85

KAYNAKLAR

87

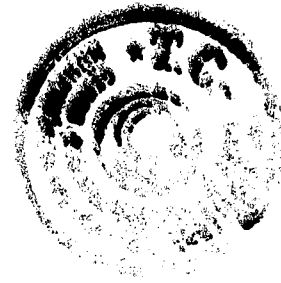
ÖZGEÇMİŞ

88

EKLER

89





SİMGELER VE KISALTMALAR

BCS	İngiliz Bilgisayar Derneği (British Computer Society)
BSI	İngiliz Standartları Enstitüsü (British Standards Institution)
BSI DISC	Bilgi Sistemi için BSI'ya danışmanlık eden birim.
CMM	Yetenek Olgunluk Modeli
EP	En İyi Pratikler (Best Practices)
ESI	Avrupa Yazılım Enstitüsü (European Software Institution)
ICQ	Kalite Güvence Enstitüsü (Institute of Quality Assurance)
IEEE	The Institute Of Electrical and Electronics Engineers
IRCA	Uluslararası Sertifika Kayıt
ISO 90001	Kalite güvencesi tasarımı, geliştirmesi, üretimi, kurulumu ve servisi için modeldir.
ISO 90003	Kalite Sistemleri-Son Muayene ve Deneylerde Kalite Güvencesi Modeli
ISO 9000-3	Yazılımın geliştirilmesi, bakımı ve tedarikinde ISO 9001'in uygulamaları.
ISO/IEC 10011-2	Kalite Kontrol Sistemi için Rehber
ISO/IEC 12207	Yazılım Yaşam Çevrim Süreçleri
JIS	Japon Standartları Enstitüsü
PA	Süreç Niteliği (Process Attribute)
SEI	Yazılım Mühendisliği Enstitüsü (Software Engineering Institution)
SEPG	Yazılım Mühendisliği Süreç Grupları
SPICE	Yazılım Süreç İyileştirme ve Yetenek Belirleme Modeli
Teknik Rapor	ISO/IEC JTC1/SC7 tarafından yayınlanan SPICE'la ilgili Rapor
TickIT	ISO 9001 Uyumlu Sertifikasyon Modeli
TS	Türk Standartları
TSE	Türk Standartları Enstitüsü
UKAS	İngiliz Akredite Servisi (Uni. Kingdom Accreditation Service)



ŞEKİLLER

	Sayfa
<i>Şekil 1.1 - TickIT Sertifikasyonun Organizasyonlarla Olan İlişkileri</i>	15
<i>Şekil 2.1 - CMM Model Yapısı</i>	27
<i>Şekil 2.2 - Yazılım Süreç Olgunluk Seviyeleri</i>	29
<i>Şekil 2.3 - Olgunluk Seviyesine Göre Anahtar Süreç Alanları</i>	31
<i>Şekil 2.4 - CMM 2. Olgunluk Seviyesi Anahtar Süreç Alanı</i>	32
<i>Şekil 3.1 - Yazılım Süreç Değerlendirme</i>	39
<i>Şekil 3.2 - Standart Parçalar Arasındaki İlişkilerin Genel Görünümü</i>	42
<i>Şekil 3.3 - Süreç Geliştirme Genel Durumu</i>	43
<i>Şekil 3.4 - Süreç İyileştirme</i>	46
<i>Şekil 3.5 - Süreç Yetenek Belirleme</i>	48

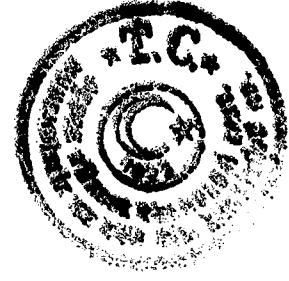




ÇİZELGELER DİZİNİ

Çizelge	Sayfa
<i>Tablo 2.1 - ISO 9000-3 Yazılım Geliştirme Temel Süreçleri</i>	24
<i>Tablo 3.1 - Süreç Kategori Grupları</i>	45
<i>Tablo 3.2 - Referans Modelin Kullanımı</i>	50
<i>Tablo 3.3 - Yetenek seviye notlandırması</i>	75
<i>Tablo 4.1 - Değerlendirme Model Karşılaştırmaları</i>	80
<i>Tablo Ek-1 - SPICE Referans Modeli Süreçler ve Süreç Kategorileri</i>	89
<i>Tablo Ek-2 - SPICE Referans Modeli Yetenek Seviyeleri ve Süreç Nitelikleri</i>	90





1. YAZILIMDA KALİTE

1.1. Giriş

Yazılımda kalite güvencesine büyük önem verilmesinin bir çok nedenleri vardır. Örneğin, uçak yolculuğu yapan herkes hava trafik kontrol yazılımının kalitesine tam bir güven duymak ister. Aynı şey bir bankacılık yazılımı, hava savunmasında kullanılan yazılımlar veya bir hava tahmin yazılımı için de şüphesiz geçerlidir.

Bu bölümde kalite kavramı, yazılımda kalitenin ne anlama geldiği, kalite sistemi, bir kalite sistemi kurmak isteyen kuruluşların yönetimde neleri dikkate almaları gerektiği, kalitenin maliyeti ve yazılımda kalite ilkelerinin belirlenmesinde rol oynayan en iyi pratikler anlatılmaya çalışılmıştır.

1.2. Kalite kavramı

Kalite kavramı, Japonya'nın 1950'li yılların sonundan itibaren sanayi kollarında elde ettikleri önemli başarılar ve hızlı gelişmeler, 1970'li yılların ortalarına doğru batılı ülkeleri önemli ölçüde geride bırakmaları sonucunda sürekli olarak gündemde kalmış ve tüm dünyaya yayılmıştır.

Kalite tanımı, geçmişten bugüne kadar bu konuda çalışmalarda bulunmuş kalite uzmanları tarafından çeşitli şekillerde yorumlanmış ve tanımlanmıştır. Bunlar;

- i) Kalite, tanımı zor, tanımı kolay, ölçülmesi imkansız (Kitchenham).
- ii) Kalite, amaca uygunluk (Dr. Joseph M. JURAN).
- iii) Kalite, müşterilerin gelecekteki beklentilerinin doğru tahminine göre yapılan yenilikler (Dr. W. Edwards DEMING).
- iv) Kalite, sıfır hata ile müşteri isteklerinin karşılanmasıdır (Philip B. CROSBY).
- v) Kalite, ürün ya da hizmeti ekonomik bir yoldan üreten ve tüketici isteklerine cevap veren bir üretim sistemidir (JIS-Japon Standartları Enstitüsü)



- vi) Kalite, bir ürün veya hizmetin, belirlenen ihtiyaçları karşılayabilme yeteneğine yönelik özelliklerin bütünüdür (IEEE).
- vii) Kalite, bir ürün yada hizmetin belirlenen veya olabilecek ihtiyaçlarını karşılama kabiliyetine dayanan özelliklerin toplamıdır (TS 9005 veya ISO 8402:1994).

Tüm bu kalite kavramlarından da anlaşılacağı gibi bir ürünün kalitesi denildiğinde akla gelecek önemli kalite kriterleri olarak; kullanıcı isterlerine cevap verebilme, hataların en aza indirilmesi, arızalar arası zaman miktarının kısa olması, satış sonrası yapılan destek ve ürün geliştirmelerinin sürekliliğinin sağlanması gibi faktörlerin ön plana çıktığını görmekteyiz.

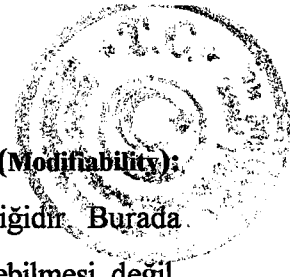
1.3. Yazılımda Kalitenin Anlamı

Yazılım mühendisliğinde kalite güvencesi konusunda yapılan bir çok yayında kalitenin tanımı yapılırken, ‘**ürünün amaca uygunluğu**’ olgusu ön plana çıkmaktadır. ‘Amaca uygunluk’ kalite kavramının önemli bir özelliğini vermekle birlikte, tek başına yeterli bir tanım değildir. Çağdaş görüşe göre, kaliteli bir ürünün bir dizi kalite faktörü ile ilişkisi vardır. Bu faktörler üç kategoride toplanabilir; Birincisi isterler belgesinde yer alan faktörler olup isterler belgesi, bir yazılım sisteminden istenilen tüm işlevlerin anlatıldığı dokümandır. Örneğin, böyle bir kalite faktörü taşınabilirlik olabilir, yani kullanıcı yazılımın farklı ortamlarda çalışabilir olmasını isteyebilir. İkinci kategoride, kolay kullanılabilirlik gibi kalitenin daha çok kültürel tarafını ilgilendiren faktörler vardır. Üçüncüsü ise kullanıcıdan çok geliştiriciyi ilgilendiren faktörlerdir. Örneğin tekrar kullanılabilirlik ve sınanabilirlik böyle bir özelliktir.

1.3.1 Kalite Faktörleri

En önemli yazılım kalite faktörleri şunlardır;

- **Doğruluk:** Bir yazılım sisteminin isterler belgesinde yer alan özelliklere tam olarak sahip olması; bu her zaman sağlanması şart olan bir kalite faktörüdür.



- **Bakımlanabilirlik (Maintainability) veya Uyarlanabilirlik (Modifiability):**

Yazılım sisteminin üzerinde kolaylıkla değişiklik yapılabilir olma özelliğidir. Burada söz konusu olan, sadece yapılan hataların kolay tespit edilip düzeltilebilmesi değil, isterlerde meydana gelebilecek sıkça rastlanan bir durum olarak değişikliklere veya ilavelere göre de sistemin kolay uyarlanabilir olması faktörüdür. Diğer bir değişiklik ise sistemi, işlevini değiştirmeden, iyileştirmek, başarısını artırmak vb. şeklinde olabilir.

- **Sınanabilirlik (Testability):** Geliştirici için doğrudan önem taşıyan bu faktör, sistemin veya bir bölümün test edilebilir olma özelliğini açıklar. Örneğin, isterler belgesi yeteri derecede açık olmayan bir sistemin test edilmesinde problemlerle karşılaşılması kaçınılmaz olacaktır.

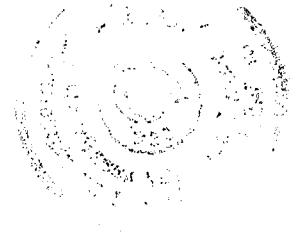
- **Güvenilirlik (Reliability):** Sistemin işlevlerinde bir kesintiye uğramadan çalışmaya devam edebilme özelliğidir. Bu kalite faktörü emniyet açısından kritik olan sistemlerde büyük önem taşır.

- **Bütünlük (Integrity):** Faktör, sistemin ve verilerin yetkisiz kullanıcılarca kullanımına karşı ne ölçüde korunduğunu ve bozulmaya karşı dayanıklılığını belirtir. Bu faktör bazı sistemler için isterler belgesinde ayrıntılı olarak verilir. Örneğin, finans uygulamalarında sisteme hangi seviyede müdahale edilebileceği isterler belgesinde belirtilir.

- **Tekrar Kullanılabilirlik (Reusability):** Geliştirici açısından giderek büyük önem taşıyan bu faktör bazı yazılım eleman parçalarının başka sistemlerde de ne ölçüde kolay kullanılabilir olduklarını anlatır.

- **Taşınabilirlik (Portability):** Daha önce de bahsedilen bu faktör, sistemin bir donanım ortamından bir diğerine kolay taşınabilir olması özelliğidir. Bu faktör kullanıcı için önemliyse, isterler belgesinde ayrıntılı olarak yer alınmalıdır.

Proje yürütücüsü tüm yazılım kalite faktörlerini inceledikten sonra hangilerinin projesinde kontrol edilmesi gerektiğine karar verir.



1.4. Yazılımda Kalite Güvencesinin Amaçları

Yazılımda kalite güvencesinin amaçları şunlardır:

- i) Yazılım geliştirme sürecini kontrol ederek yazılım kalitesinde ilerleme sağlamak,
- ii) Yazılım süreci için geliştirilmiş tüm standartlar ve yöntemlere uyum sağladığının güvencesini vermek,
- iii) Üründe, süreçte veya standartlardaki herhangi bir yetersizliğe yönetimin dikkatini çekmek, böylece bunların düzeltilmesini sağlamak.

Yazılım kalite güvencesi organizasyonu kaliteli ürünler çıkarmaktan sorumlu değildir; bunlar geliştirmeyi yapanların işidir. Yazılım kalite güvencesinin sorumluluğu kalite faaliyetlerini denetlemek ve sapmalardan yönetimi haberdar etmektir.

Etkili olabilmek için kalite güvencesi geliştirmeye çok yakın çalışır. Geliştiriciler yazılım kalite güvencesini kendilerine düşman olarak görürlerse, kalite sisteminin etkili olabilmesi çok zor olur. Bu nedenle destekleyici bir davranış ve ortak çalışma anlayışı önemlidir.

1.5 Kalite Sistemi

Kalite sisteminin merkezinde bir **kalite kılavuzu** yatmalıdır. Kalite kılavuzunda *standartlar*, *yöntemler* (procedures) ve *rehberler* (guidelines) bulunur. Bunlar yazılım geliştirme sürecinin tüm faaliyetlerini bir disiplin içine sokan özellikleri oluştururlar. Kalite kılavuzunun önemli bir özelliği, bir projeye uygulanabilecek bir dizi kalite faktörünü içermesidir.

Tipik kalite faktörleri ve bunların mevcudiyetini kanıtlayan dokümantasyonlara örnekler şöyle verilebilir:



- i) Geliştirilen kodun, kuruluşun standartlarına uygunluğunu denetleyen bir yazılım aracıyla kontrol edilmesi; bu uygulamanın çıktısı, herhangi bir ihlalin olmadığını kanıtı olacaktır.
- ii) Sistemin bir fonksiyonun test edilmesi; ortaya çıkan doküman testin yapıldığını ve doğru sonuçların alındığını gösteren bir test raporudur.

Kalite sistemi proje yöneticisine, kullanabileceği kalite kontrollerinin neler olduğunu eksiksiz anlatmalıdır. Bu kontroller standartlar, yöntemler ve sistem dokümantasyonu ile ilgilidir. Standart ve yöntemler sadece gelişme faaliyetlerine değil, kalite güvencesine yönelik faaliyetlere de uyarlanabilir olmalıdır.

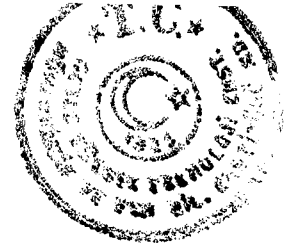
1.5.1. Kalite Sisteminin Kullanımı

Kalite kılavuzunun önemli bir bölümü kalite planının nasıl geliştirileceğinin açıklanmasına ayrılmalıdır. Proje yöneticisine, ileride yapılacak bir projede hangi faktörlerin yer alacağını ve ne zaman uygulanacağını gösteren bilgiler kılavuzda anlatılmalıdır. Risk analizi, bir dizi sorunun yanıtlanmasıyla ortaya çıkan ve uygulama alanı, proje takımının deneyimi, proje organizasyonu, kullanılacak yazılım ve donanım araçlarının yeniliği gibi bir takım faktörleri inceler.

1.6. Kalite Yönetimi

Etkili bir yazılım kalite güvencesi yönetimi, tüm geliştirme sürecinin denetimi sağlayan bir organizasyonun kurulmasını gerektirir. Böyle bir organizasyon kurulmadan önce kalitenin kuruluş için ne kadar önemli olduğu tespit edilmelidir. Yönetim kaliteye verdiği önemi günlük faaliyetlerle gösteremezse istikrarlı bir gelişme beklemek güç olur.

Yazılım kalite güvencesinin esas yararı, yönetime, resmen kurulu olan yazılım geliştirme sürecinin gerçekten uygulanan olduğuna dair güven vermesidir. Bu güvence şu unsurların var olmasıyla sağlanır;



- i) Uygun bir geliştirme metodolojisinin yerleşmiş olması,
- ii) Proje faaliyetlerinde standart ve yöntemlerin kullanılıyor olması,
- iii) Bağımsız denetlemelerin yapılması,
- iv) Bakım ve geliştirme faaliyetlerini destekleyen dokümantasyon üretilmesi,
- v) Dokümantasyonun proje nihayetinde değil, proje süresince geliştiriliyor olması,
- vi) Değişiklikleri kontrol edebilecek mekanizmaların kurulmuş ve kullanılıyor olması,
- vii) Standart ve yöntemlerden sapmaların derhal ortaya çıkması.

Kalite sisteminin yönetim desteğini alması şarttır. Kalite sisteminden üst düzey bir yönetici sorumlu olmalıdır. Kalite güvencesine yönelik yönetim kurulu seviyesinde sorumluluklar şöyle sıralanabilir.

- i) Yönetim kuruluna düzenli olarak kalite konularını getirmek,
- ii) Kalite sistemiyle ilgili önemli konuları çeşitli zamanda tartışmaya açmak, kalite sistemini düzenli olarak gözden geçirmek,
- iii) Şirketin geliştirme ve kalite güvencesi işlevleri arasında oluşacak problemlere çözüm getirmek.

1.6.1. Üst Yönetim'in Sorumluluğu

Yönetim kurulu kalite güvencesine yeterli kaynak ayırmasını ve tüm denetleyici toplantılarına, kalite fonksiyonuyla ilgili raporlar gelmesini sağlar.

Bir yazılım sistemi gibi kurulmuş olan kalite sistemi de zamanla değişmek zorundadır. Değişikliğe bir çok faktör neden olabilir. Örneğin yazılım teknolojisinde görülen değişiklikler ve yeni programlama dilinin kullanımı bize yeni standartları kullanmamızı gerektirir. Yazılımda kalitenin geliştirilmesi çalışmalarında yönetimin tam desteği alınmış olmalıdır. Yönetim başta olmak üzere organizasyonun tüm birimlerinde kalitenin geliştirilmesini bir kültür olarak benimsenmeli ve sürekli iyileştirme çalışmaları yönetimin tüm bireylerinin desteği alınarak devam ettirilmelidir.



1.6.2. Günlük Sorumluluklar

Kalite sisteminin günlük yönetimi için bir kişi görevlendirilmelidir. Büyük firmalarda bu kişi kalite bölümünün başı durumundadır. Bu kişinin sorumlulukları şunlardır:

- i) Kalite elemanlarının günlük faaliyetlerini kontrol etmek,
- ii) Kalite güvencesi fonksiyonu için verilen kaynakları takip etmek,
- iii) Projelerin günlük işleyişlerinde karşılaşılan problemlere çözüm bulmak; örneğin kalite grubundan bir projenin denetlenmesi sırasında kalite kontrollerinden birine uyulmadığı tespit edilirse, proje yöneticisini uyarmak,
- iv) Kalite verilerinin alınması ve yönetim kurulları için özet raporların hazırlanması,
- v) Kalite sisteminin etkinliğinin devamlı takibi,
- vi) Kalite güvencesi fonksiyonuyla ilgili günlük sorumluluk taşıyan kişi kaliteyle sorumlu yönetim kurulu üyesine rapor eder.

1.7. Eğitim

Her yeni işe alınan eleman kuruluşun geliştirme ve kalite güvence uygulamaları hakkında bir eğitim programına sokulmalıdır.

Programın konuları şunları içermelidir:

- i) Program, kalite kılavuzunun rolü, sistemin işleyiş şekli, standart ve yöntemlerin koyulması ve hedefleri; çalışılan konu kalite sistemini geliştirmeye yönelik katkı sağlayabilecekleri mekanizmalar ve geçmişten bazı kötü örnekler,
- ii) Temelde kötü sonuçlanan bazı projeler örnek olarak verilmelidir. Eğitimin birinci bölümünde karşılaşılan problemler, İkinci bölümde ise bu problemlerin tekrarlanmamasını sağlayacak kalite kılavuzunun ilgili bölümleri gösterilmelidir.

Standartların ve yöntemlerin geliştirilmesi sıkıcı bir iş olarak görülebilir. Ancak söz konusu standartların bir yazılım projesinde ortaya çıkan hatalardan ve düzensizliklerden nasıl ortaya çıkarıldığının gösterilmesi, sistem geliştirici personele çok yararlı olacaktır.



Kalite kılavuzunun, yıllarca çalışılan projelerden çıkan deneyimlerinin iyi veya kötü sonuçları süzgeçten geçirildikten sonra elde edilen deneyim ve izlenimlerden oluşan yöntemler olduğu gösterilmelidir.

1.8. Dokümanite Edilmiş Kalite Sistemi

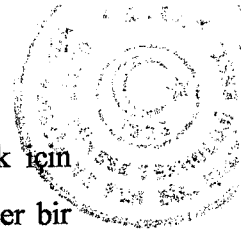
Standart, yöntem ve yol gösterimlerin dokümanite edildiği bir kalite kılavuzu bulunmalıdır. Kalite kılavuzu sadece geliştirmeye ilgili konuları değil, kalite güvencesi faaliyetlerini de içermelidir. Birinci sayfasında, yönetim kurulunun kalite ile ilgili politikasının tanımlanması bulunmalıdır. Kurumun kalite politikası fazla ayrıntıya girilmeden verilmelidir. [İnce, D., 1994]

Kuruluşun geliri, ürettiği ürünlerin kalitesine bağımlıdır. Kalitenin en üst düzeyde olmasını sağlamak için bir kalite yönetim sistemine ihtiyaç vardır. Bu sistem standartlar, prosedürler ve rehberlerden oluşur ve hataları en aza indiren yazılım sistemini müşteriye zamanında verilmesini sağlayan en iyi geliştirme pratiğini ortaya koyar. Her projenin başlangıcında proje yürütücüsü uygulayacağı kalite kontrollerini ve seviyelerini kalite kontrol bölümüne danışır. Bu kontrollere tüm seviyelerde uyulması herkesten beklenir.

Her tür projeye uygulanan bir kalite kılavuzuna sahip olmak yalnız başına çözüm değildir. Projeler birbirinden çok farklı özellikler gösterebilirler, örneğin gerçek zamanlı ve yüksek ölçüde güvenlik gerektiren bir yazılım projesiyle basit bir muhasebe yazılımına uygulanacak kalite kontrol faktörleri arasında önemli farklar vardır.

1.9. Kalitenin Maliyeti

Bir ürün ya da hizmetin, müşteri istekleri doğrultusunda ekonomik bir şekilde üretilmesini sağlayan, pazardaki rekabet ortamına uydurabilmesi, tüketici ihtiyaçlarını karşılayabilme başarısı ve verimin artırılması için gerekli olan kalitenin bir ürün üzerinde ilk başta bir maliyet getireceği bir gerçektir.



Yazılım ürünü üzerinde ki kalite maliyetini düşük seviyede tutabilmek için aşama aşama bazı kontrollerin yapılması gereklidir. Bu kontrol aşamaları ve her bir aşamada yapılması gerekli olan faaliyetler şunlardır:

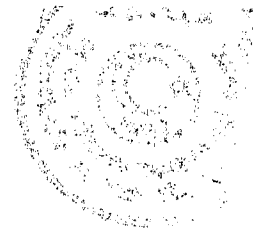
- *Önlem aşaması:* Planlama, eğitim ve süreç iyileştirmeleri faaliyetleri,
- *Denetleme/Değerlendirme:* Tasarım ve kod kontrolü, testler vb,
- *Hata düzeltme (İç):* Kod düzeltme, tekrar test, iç denetleme,
- *Hata düzeltme (Dış):* Müşteri garantisi, alanda destek vb.

Yapılan çalışmalar hata maliyetinin iş ilerledikçe arttığını göstermekle birlikte genelde bir hatanın oluşmadan önlenmesinin en ekonomik yol olduğunu göstermektedir.

Yazılımda kalite ilkeleri, geçmişteki yaşanan tecrübelerden oluşan en iyi pratikler (EP) ile ortaya çıkmıştır (Best Practices, BPs, Best Current Practices, BCPs). En iyi pratikler birer rehber konumunda olup mutlak koşul değildir. Ancak kabul edilen kalite modellerinin temellerini bunlar oluşturmaktadır.

En iyi pratik (EP)'ler şu şekilde verilebilir:

- i) Erken tanı, erken çözüm maliyeti düşürür,
- ii) Ürün değil, süreç önemlidir. Süreç iyileştirilmeli,
- iii) Sürekli iyileştirme, geliştirme yolu açık olmalı,
- iv) Üretimde standartlar ve referans modellerdeki temel ölçütler kullanılmalı,
- v) Müşteri tatmin araştırması yapılmalı,
- vi) Proje kestirim araçları kullanılmalı,
- vii) Süreçleri iyi belgelenmeli,
- viii) Metodoloji kullanılmalı ve iyi belgelendirilmeli,
- ix) Kritik tasarım değerlendirmesi yapılmalı,
- x) Kod denetlemesi yapılmalı,
- xi) Konfigürasyon yönetimi araçları kullanılmalı,
- xii) Hataları ve güvenilirliği ölçümlenmeli, kaydedilmeli, analizi yapılmalı,
- xiii) Proje sonrası değerlendirme yapılmalı.



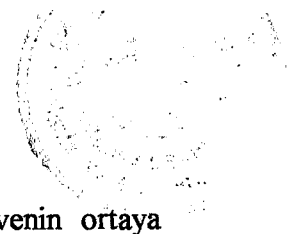
2. YAZILIM SÜREÇ / KALİTE MODELLERİ

2.1. Giriş

Yazılım projelerinin büyümesiyle 1970'lerde ortaya çıkan yazılımda kalite sorunu 1980'lerde *yazılım krizi* olarak anılacak düzeye ulaştı. 1980'lerin ortalarından bu yana yapılan çalışmalarla, yazılımda kalite sertifikasyonu, süreç iyileştirme ve yetenek belirleme amaçlarıyla çeşitli modelleri ortaya konmuş bulunmaktadır. Doğal olarak aralarında kavram düzeyinde temel ilişkiler bulunsa da, başta amaçlar olmak üzere henüz bu model ve standartlar arasında bir uyum ve bire bir izdüşüm olduğu söylenemez. XXI. yüzyıla girerken dünya üzerindeki yazılım üreticileri ve müşterileri arasında, kalite sorununa yaklaşımda, genel görüş birliğinden çok çeşitlilikten söz etmek daha doğru olur.

Yazılıma özgü ilk süreç modeli ABD Savunma Bakanlığı desteği ile Carnegie Mellon Üniversitesinde kurulan Yazılım Mühendisliği Enstitüsünde (SEI) geliştirilen Yetenek Olgunluk Modeli, CMM (Capability Maturity Model) olmuştur. 1987'de ortaya konduğundan bu yana, belki de ilk olmanın verdiği ivme ile başta ABD olmak üzere yazılım dünyasında en çok tartışılan, kullanılan ve örnek alınan süreç/kalite modeli, CMM olmuştur. 1990'lı yılların başlamasıyla ağırlıkla Avrupa'da olmak üzere ISO 9000 temelli yazılım kalite sistem sertifikasyonlarında önemli artış gözlenmektedir. ISO 9001 ve ISO 9000-3 temel alınarak ortaya konan *TickIT* standardı yalnız çıkarıldığı ülke olan İngiltere'de değil tüm Avrupa ve Asya'daki yazılım çevrelerinde tanınır ve kullanılır olmaktadır.

Bunların dışında Avrupa'da ESPRIT programı çerçevesinde geliştirilen *Bootstrap*, Kanada'da telekom endüstrisinde yazılım süreç değerlendirmesi için geliştirilen *Trillium*, özel firmalar tarafından geliştirilen *Healthcheck*, *SQPA*, *STD* gibi modeller de vardır ve bunlar belirli ortam ve çevrelerde kullanılmaktadır.



Modellerin çoğalması 1991 yılında bir üst model veya çerçevenin ortaya konması girişimine yol açmış ve 1993'te ISO/IEC JTC1/SC7 tarafından SPICE (Software Process Improvement and Capability dEtermination) projesi 20 ülkeden kuruluşların katılımı ile başlatılmıştır. Haziran 1995'te birinci, Kasım 1996'da ikinci versiyonu çıkan ve resmen ISO 15504 taslak standardı olarak anılan SPICE modelinin, halen sürmekte olan denemelerden sonra 1998 yılı sonu veya 1999 yılı başlarında resmi ISO standardı olması beklenmektedir.

2.2. Süreç Kavramı

Yazılımda kullanıcı ihtiyaçlarını tam karşılayabilen bir standart sağlamak ve bunu kabul edilebilir bir zaman aralığında kullanışlı çıktılar sağlayabilmek için deneme ve geliştirmeleri belli bir yöntem, düzen, adım ve işlemlerden oluşan bir düzene oturtma işidir.

Süreç amacı; bir standart oturtmak, değişkenlikleri azaltmak ve devamlı iyileştirmeler sağlamaktır. Bir süreç yazılı ve grafiksel olarak belgelenen, tekrarlanabilen, girdi ve çıktıları olan basit ve karmaşık olarak tanımlanabilir. Örneğin, isterler analizi süreci, pazar araştırması, müşteri isteklerinin belirlenmesi, fizibilite çalışması, spesifikasyonların belirlenmesi ve isterlerin modellenmesi gibi alt süreçlere ayrılabilir.

2.3. Süreç/Kalite Modellerinin Gelişimi

Yazılımda kalite sorunları 1968 yılından sonra önemli ölçüde artarak ilk “Yazılım Krizi” olarak karşımıza çıkmaktadır. Yazılım projelerinin büyümesi, donanım ve yazılım alanındaki gelişmelerin hızlanması ve kullanım alanlarının yaygınlaşarak büyümesi yazılımda kalite boyutunun gözden geçirilmesinin gerekliliği konusunun ön plana çıkarmaktadır.



ABD’de 1970’lerde kamu tarafından sipariş yoluyla satın alınan yazılım ürünlerinin ancak %5’i kullanılabilmiştir. Bu yazılımların %40’ı hiç kullanılmamış, geriye kalanı ise maliyet artışı, zaman aşımı, performans yetersizliği yüzünden kullanılmamış veya ancak özelliklerinden ödünler verilerek kullanılabilmiştir. Artarak devam eden kalite sorunları, 1980’lerde “yazılım krizi”nin yoğun bir şekilde kullanılmasına neden olmuş ve 1990’lı yıllarda ise büyük projelerdeki başarı oranları çok düşük gerçekleşmiştir. Tüm bu nedenler son on yıl içerisinde yazılım geliştirmede kullanılan kalite sistemlerini ve süreçleri değerlendirmek iyileştirmek ve sertifikasyonunu sağlamak amacıyla çalışmalar başlatılmış ve çeşitli modeller ortaya konularak geliştirmeleri sağlanmıştır.

Yazılımda ilk yaklaşımlar yapılırken diğer mühendislik dallarından örnekler alınarak aşağıdaki çalışmalar yapılmıştır:

- Yapisallaştırma (Parçala ayırma),
- Metodolojiler (SSADM, OOM vs),
- CASE ve ICASE yaklaşımları,
- 1984 ABD DOD girişimi ve SEI kuruluşu (Carnegie Mellon Üniversitesi Software Engineering Institute) Watts Humphrey ve Yetenek Olgunluk Modeli (Capability Maturity Model, CMM),
- ISO 9001, ISO 9000-3, TickIT modelleri,
- Diğer modeller: Bootstrap (AK), Trillium (Kanada) vb,
- SPICE modeli (ISO 15504 taslak standardı), ESI (Avrupa Yazılım Enstitüsü)
- Sertifikasyon.

Yazılımın mühendislik olması için standartlara dayanması ve ölçülebilmesi gerekir. Kalite standartları kalitenin ölçülebilmesini, verim standartları ise verimin ölçülebilmesini sağlamaya yöneliktir.

2.4. ISO 9000 ve TickIT Modeli

ISO 9000 ve ISO 9001’in yazılımda kullanımı için rehber olan ISO 9000-3’e dayanarak İngiltere’de geliştirilmiş olan bir sertifikasyon modelidir.



ISO 9000 standardı, yazılım endüstrisi için yazılmış bir standart olmayıp bütün endüstriler için uyulması gereken genel kalite prensiplerini açıklayan, kalite sistemi ve kalite güvencesi standartları için kalite sistemi modeli olarak uygulanan bir standarttır. Bu standartların müşteriler tarafından kabul görmesiyle firmalar bu modele uyduklarını tarafsız organlarca belgelendirmek istemeye başladılar. Müşteriler kalite güvencesinin bir göstergesi olarak satıcı firmada ISO 9001 sertifikasyonu aramaktadır.

ISO 9000 serisinin ilk sürümlerinin 1987'de yayınlanmasından sonra İngiliz Ticaret ve Sanayi Bakanlığı, ISO 9000 serisi ve bazı diğer standartların yazılım üretimi ile olan ilgisini gözden geçirerek şu sonuçları elde etmiştir.

- i) ISO 9001'i yazılıma uyumlu bir yol olarak kullanmak,
- ii) Yazılım sektöründe üçüncü parti sertifikasyona olan güveni akreditasyon yoluyla sağlamak,
- iii) Yazılım sektöründeki kalite denetiminin profesyonellik düzeyini iyileştirmek,
- iv) Yazılım ürününün dokümantasyonunu sağlamak olarak özetlenebilir.

2.4.1. TickIT Kalite Sistemi Kavramları

Bir kalite sistemi, bir ürün veya servisten istenen kaliteyi başarmak için gerek duyulan aktivitelerin koordinasyon ve kontrolünü sağlayan mekanizmadır. Kalite sistemi teslim edilen ürünün kalitesini doğrudan ve dolaylı etkileyen tüm organizasyon elemanlarını içerir. Tipik olarak bir kalite sistemi; kalite elkitabı (quality manual) ve bağımlı politikalar, prosedürler ve standartlar ile dokümante edilir. ISO 9000 serisi standartlar, satıcının kalite sisteminin yeterliliğinin nesnel 'benchmarking'i için temel görev yaparlar (ISO 9001:1994, bölüm 4.2.).

Kalite yönetimi, kalite politikasını uygulama yollarıdır. Projede kalite planlaması, güvencesi, kontrolü işleriyle başılır. Kalite yönetimi tedbirlerinin uygulandığını güvenceye almak için proje yürütücüsü ve elemanlarının görevidir (ISO 9001:1994, madde 4.1.2.3).

Kalite güvencesi, kalite sistemi içinde yürürlüğe koyulan, gerektiğinde gösterilebilen planlı ve sistematik aktivitelerdir. Bunların amacı, bir ögenin kalite gereksinimlerini karşıladığına dair yeterli güvence sağlamaktır (ISO 8402).

Kalite güvencesinin getirdikleri;

- i) Kalite kontrol aktivitelerinin uygulandığının sağlanması,
- ii) Destekleyici faaliyetlerin kalite seviyelerinin kabul edilebilir olduğunun doğruluğunun sağlanması (ISO 9001:1994, madde 4.1.2.1).

Kalite kontrolü, kalite isterlerinin tatmini için kullanılan operasyonel teknikler ve aktivitelerden oluşur (ISO 8402). ISO 9001’de tanımlanan bir proje yöneticisinin sorumluluğu; kalite kontrol organizasyonu, standartları, yöntem ve mekanizmaları proje için tanımlamak ve bunları kalite sistemi çerçevesinde bir kalite planı şeklinde doküman etmek (ISO 9001:1994, madde 4.2.3).

2.4.2. TickIT Kalite Sistemi Anahtar Dokümanları

TickIT kalite sistemi içerisinde kullanılan anahtar dokümanlar şunlardır:

- i) ISO 9001:1994, *Kalite Sistemi Standardı*, tasarım, geliştirme, kurma (installation) ve servis için bir model tanımlar,
- ii) ISO 9002:1994, Kalite Sistemi Standardı ikinci bölümüdür. ISO 9001’den farkı tasarım aktivitelerini içermez,
- iii) ISO 9000-3:1991, yazılım için kalite sistemlerinin uygulayıcıları ve denetimcileri için özel rehberlik bilgileri verir.

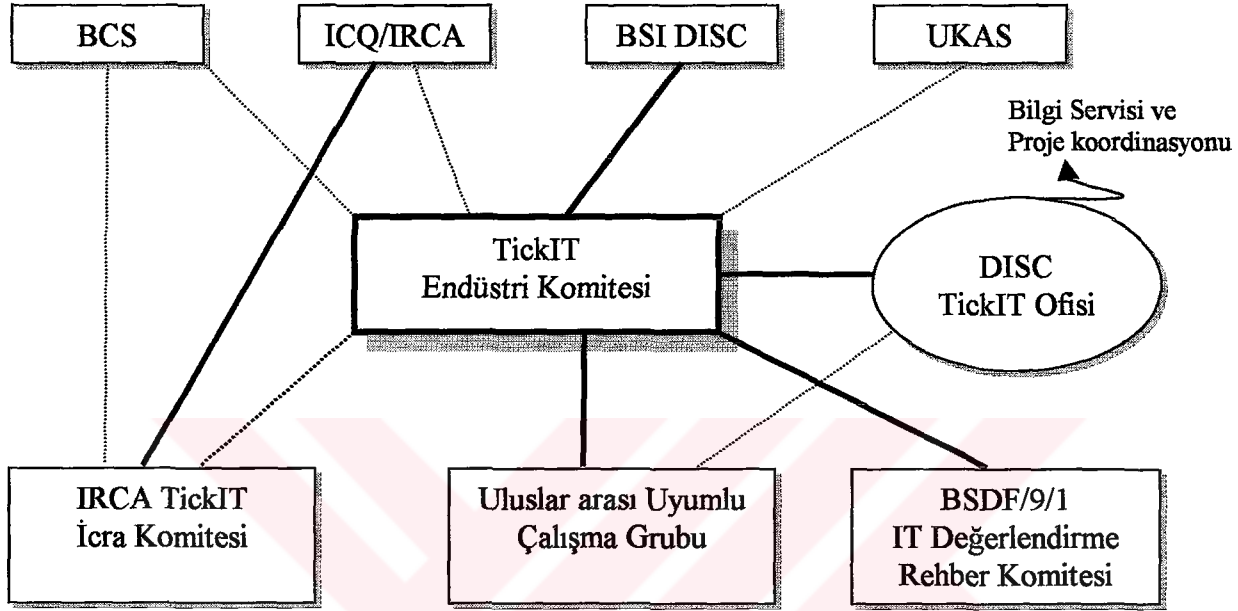
2.4.3. TickIT Altyapısı ve Organizasyonlarla İlişkisi

TickIT yazılım kalite sistemi standardı içerisindeki anahtar organizasyonlar ile olan ilişkileri şekil 2.1 ve ilişkilerde geçen terimlerin anlamları aşağıda belirtilmiştir.

BCS : İngiliz Bilgisayar Derneği (British Computer Society)

BSI : İngiliz Standartları Enstitüsü (British Standards Institution)

- BSI DISC** : Bilgi Sistemi için BSI'ya danışmanlık eden birim.
- IQA** : Kalite Güvence Enstitüsü (Institute of Quality Assurance)
- IRCA** : Uluslararası Sertifika Kayıt
- UKAS** : İngiliz Akredite Servisi (Uni. Kingdom Accreditation Service)
- ABCB** : İngiliz Sertifikasyon Kurulu Gövdesi



Şekil 1.1 - TickIT Sertifikasyonun Organizasyonlarla Olan İlişkileri

- : Rapor Hattı (Reporting Line)
- : Temsil Etme (Representation)

TickIT alt yapısı DISC tarafından korunur. TickIT Endüstri Komitesi bu amaçla kurulmuştur. Burada IT sistemleri alıcıları, üreticileri, sertifikasyon endüstrisi ve profesyonel endüstriler temsil edilir. DISC, sahibi olduğu TickIT için anahtar konumunda olan dokümanları ve operasyonel prosedürleri saklamaktan sorumludur. Ayrıca UKAS'a yazılım sektörünün gereksinimleri ile ilgili önerilerde de bulunmaktadır. TickIT denetimci kayıt prosedürleri IQA/IRCA tarafından 'auditor attribute standards' ile uyumlu yönetilir. TickIT Endüstri Komitesi yakın ilişkide olduğu DISC BSFD/9/1 komitesi TickIT için rehber materyali sağlar ve endüstriye açık çalışır. TickIT Endüstri Komitesi-ITQS (Avrupa Sertifikasyon Organlarının Anlaşma Grubu) ilişkisi, IT kalite sistemi sertifikasyonu için uluslar arası düzenlemeler yapmaktadır.



2.4.4. TickIT Kapsamı

TickIT değerlendirmesi yapılacak alanların belirlenmesi amacıyla DISC BSFD/9/1 tarafından ABCB ile yapılan ortak çalışmayla sonucunda belirlenen TickIT uygulama alanları şu bölümlerden oluşmaktadır.

- i) Yazılım ürünü ve servisi geliştirme,
- ii) İç (internal) yazılım geliştirme,
- iii) Yazılımla ilgili servis sağlama.

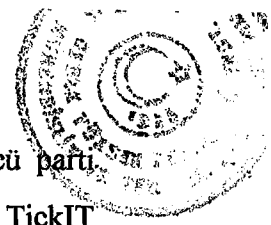
2.4.5. TickIT Akreditasyon Süreci

Sertifikasyon veren organların (Certification bodies) merkesi bir otorite tarafından kaydedilme sürecine **akreditasyon**, firmaların kaydedilme sürecine ise **sertifikasyon** denir. Akreditasyon süreci, sertifikasyon sürecine paralel ancak daha üst düzeyde , bir sertifikasyon organın resmen tanınması anlamına gelir. İngiltere’de UKAS akreditasyon sürecinden sorumludur.

TickIT akreditasyonunun amacı, piyasada güven tahsis etmek, ikinci ve üçüncü sertifikasyon süreçlerini kontrol etmek olarak tanımlanabilir. Avrupa akreditasyon otoriteleri, kalite sistemi sertifikasyonunda tutarlılığı ve karşılıklı kabulü sağlamak için EN 45012 normunu (General Criteria for Registration Bodies Operating Quality System Registration) kullanmaktadırlar.

2.4.6. TickIT Sertifikasyon Süreci

Sertifikasyonun endüstride bir değerinin olması için sertifikasyon organlarının tanınmış ulusal bir akreditasyon organınca akredite edilmesi gerekir. Ayrıca ortak bir sertifikasyona gidilmesi gerekir. Buradaki asıl amaç, TickIT servisi veren tüm sertifikasyon organlarının kendilerinden beklenen servisi sağlayacaklarına olan güveni sağlamaktır.



Sertifikasyon veya bazı ülkelerde “registiration” olarak bilinen üçüncü parti değerlendirmesi bağımsız bir kuruluşça belirli standartlara karşı yapılır. TickIT içerisindeki değerlendirme ISO 9001:1994 ve bazen ISO 9002:1994 standartlarına göre yapılır. Organizasyon tarafından belirtilen sertifikasyon kapsamına giren tüm alanlarda standarda uyumluluk esas alınır. ISO 9000-3 ve TickIT rehberleri, sertifika için aday firma tarafından dikkate alınır; ancak uyumsuzluklar ISO 9001’in maddelerine göre çıkartılır, ISO 9000-3’e göre yapılmaz.

2.4.7. TickIT Değerlendirmesi

Değerlendirme sırasında denetimciler, mevcut pratikleri kalite sistemine karşı incelerler. Bu işi, çalışanlarla mülakat, çalışma dokümanlarını, prosedürleri ve kayıtları inceleyerek yaparlar.

Bir TickIT değerlendirmesi en az iki aşamalı bir süreçten oluşur:

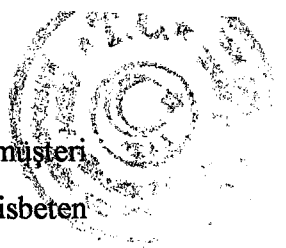
- i) Organizasyonun kalite sistemi, Standarda (TickIT için ISO 9001) göre değerlendirmesi yapılır,
- ii) Organizasyon, pratikte gerçekten kendi kalite sistemine ve standarda uyumlu çalışıp çalışmadığı denetlenir ve kalite sisteminin etkinlik derecesine bakılır.

Denetimcilere rehberler verilerek ilgili kişilere bu rehberler aracılığı ile kalite değerlendirmesi yapmalarında yardımcı olunması sağlanır. Değerlendirme sonuçlarına göre çoğu zaman organizasyonlardan yetersizliklere karşı düzeltici hareketler yapması beklenir ve sertifika vermeden bunların doğrulanması istenir.

2.4.8. TickIT Denetim Terminolojisi

TickIT değerlendirmelerinde kullanılan denetim terimleri ve anlamları aşağıda verilen anlamları içermektedir:

- **Birinci parti denetim:** Bir organizasyonun iç denetimidir. Genellikle sertifikasyon için üçüncü parti çağrılmadan önce, hazırlık amacıyla yapılır.

- 
- **İkinci parti denetim:** Organizasyonun ilgili ürün ve süreçlerden müşteri tarafından denetimidir. Bu denetimlerin kapsamı diğerlerine göre nisbeten dardır.
 - **Üçüncü parti denetim:** Organizasyonun kendi isteği ile tarafsız bir sertifikasyon organınca değerlendirmeye tabi tutulmasıdır.

2.4.9. TickIT Yaklaşımı

TickIT, ISO 9001'in temel isteklerine yeni bir şeyler eklememektedir. Bir kalite sistemi kurarken uygulayıcının yükümlülüğü, ISO 9001'in isteklerini, ISO 9000-3 ve TickIT rehberliğinde organizasyonun ihtiyaçlarına göre yorumlayarak uygulamaktır. TickIT rehberi, emir (prescriptive) olarak alınmamalı. Her organizasyon kendi süreçlerini tanımlamalı ve iş süreçlerine değer katmalıdır.

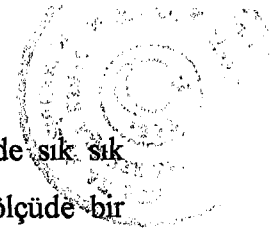
TickIT'in etkinliği üç faktöre dayanır:

- i) Sertifikasyon organlarının birbirini karşılıklı tanınması,
- ii) TickIT rehberi, ve
- iii) Uluslar arası piyasalara uygun bir yöntem yapısı.

2.4.10. TickIT Müşteri Rehberi

TickIT rehberinin bu bölümünün amacı, yazılımda kalite sertifikasyonu konularına müşteri açısından bakmak ve müşterilerin teslim edilecek ürünün kalitesine nasıl katkı sağlayacaklarını incelemektir.

Kalitenin başarılmasında müşterinin rolü incelendiğinde müşteriler; gereksinimlerini işlevsellik, maliyet, zaman açısından belirleyebilmeli ve tanımlayabilmelidir. Bu gereksinimleri karşılayabilecek ürünleri teslim edebilecek yetenek, kapasite ve sertifika sahibi tedarik ediciler seçilebilmelidir.



Müşteriler edindikleri yazılım ürünleri ve yazılım içeren projelerinde sık sık önemli sorunlarla karşılaşmaktadırlar. Yazılım, üretimden ziyade büyük ölçüde bir tasarım ve geliştirme faaliyetidir.

Sorunlar en çok şu alanlarda görülmektedir:

- i) Müşteri için isterlerin tam olarak belirtimi zor olmakta,
- ii) Satıcının müşteri isteklerini tam olarak anlamasının zor olması,
- iii) İsterlerin tanımı ve anlaşılmasında büyük miktarda bilginin iletişimi ve sürekli olarak özümlemesinin gerekliliği,
- iv) Değişen isterlerin sistem ile uzlaştırılmasındaki güçlükler,
- v) Yazılımın ve yazılım geliştirme sürecinin büyük ölçüde soyut olması deneyim, tecrübe ve hayal gücü gerektirmesi,
- vi) Zaman ve maliyeti düşürmek zorunda kalan satıcıların isterleri tümüyle işlevsel ve ayrıntılı tasarıma dönüşmeden kodlamaya başlamaları,
- vii) Yazılımı tümüyle bir bütün olarak test etmenin zorlukları.

Yazılım geliştirme, bakım ve destek faaliyetlerini, yaşam döngüsünün tüm aşamalarında daha görünür ve yönetilebilir bir sistemin oluşturulması gereklidir. Böyle bir kontrol sistemi, bir kalite sistemini oluşturur. Amacı ise temelde müşteri ve satıcı firmayı bir araya getirmektir. TickIT, sundukları ürünlerin bir elemanı olarak yazılım tasarımı yapan ve geliştiren firmaların ISO 9001 sertifikası almalarını kolaylaştırmak için tasarlanmıştır.

2.4.10.1. ISO 9000-3’de Kalitenin Başarılmasında Müşterinin Rolü

ISO 9000-3’de müşterinin anahtar rolünü gösteren, kaliteli ürün geliştirme sürecine nasıl katılabileceği konusunda bölümler bulunmaktadır. Müşterinin satıcı ile etkileşimi, yönetimin işbirliği, gözden geçirmeler, isterlerin tanımı ve kabul süreci hedeflenir. ISO 9000-3’ün ilgili bölüm ve maddeleri şunlardır.

Bölüm 4.1.Yönetimin sorumluluğu,

- Madde 4.1.2. Müşteri yönetiminin sorumluluğu
- Madde 4.1.3. Ortak gözden geçirmeler

Bölüm 5.3. Müşteri isterler belirtimi



- Madde 5.3.1. Genel
- Madde 5.3.2. Karşılıklı işbirliği

Bölüm 5.8. Kabul

- Madde 5.8.1. Genel
- Madde 5.8.2. Kabul testi planı

2.4.10.2. Müşteri – Satıcı İlişkisi

Müşteri yazılımı tedarik eden firma ile yakın ilişki kurmak zorundadır. Bu ilişki karşılıklı güven ve birlikte çalışmaya dayanmalıdır. Müşteri yazılım geliştirilirken uygulanan kalite kontrollerinin sağlıklı olduğunu belirleyebilmelidir. Gerçekte, müşteri koyduğu kalite hedeflerine ulaşabilmesi için yazılım geliştiricilere yardımcı olabilmeli ve kalitenin geliştirilmesinin yazılım ürünün imalatı sırasında gerçekleştiğini özümleyebilmelidir (ISO 9001:1994, Bölüm 4.6.2).

Müşteri bir sipariş vermeden önce satıcı firmayı denetleyebilir ve bir denetimin sonuçlarını isteyebilir (ISO 9001:1994, Bölüm 4.6.2). İngiltere’de müşteriler satıcı firmanın ISO 9001 sertifikasyon statüsünü İngiliz Ticaret ve Sanayi Bakanlığı’ndan öğrenebilirler.

Sözleşme, gerekirse müşterinin satıcının (üzerinde anlaşılan süreç, standart ve prosedürlerin uygulandığının bir güvencesi olarak) operasyonlarını denetleyeceğini belirleyebilmelidir (ISO 9001:1994, Bölüm 4.3, 4.16, 4.17).

2.4.10.3. Anahtar Müşteri Aktiviteleri

Müşteri, ISO 9001 sertifikalı bir maksimum fayda sağlayabilmesi için şu pratikleri benimsemeli ve kendine bir kültür edinerek; isterleri nesnel, tam, tutarlı ve gerçekçi olarak belirtmeli ve tedarikçi firma ile kalite planın değerlendirilmesi ile yeterliliğinde anlaşmaya varılması, geliştirmelerin izlenmesi ve gerekirse denetlenmesi işlemlerini firma diyalog halinde sürdürmeyi sağlamalıdır.



Müşteri projeyi izlemeli, üzerinde karşılıklı olarak anlaşılan aktivitelerin gerçekten yapıldığına ve kalite, zaman ve maliyet hedeflerinin başarıldığından emin olmalıdır (ISO 9001:1994, Bölümler 4.3, 4.4, 4.9). Müşteri satıcının üzerinde anlaşılan kalite hedeflerini karşıladığını güvence altına alabilmek için doğrulanması gereken nitelikleri tanımlar ve satıcı ile her nitelik için yaşam döngüsünün hangi noktasında doğrulacağı ve doğrulama noktalarındaki doğrulama kriterlerinin neler olacağı konusunda anlaşması gerekir.

Ürün tasarımı ve uygulamasını izlemek için gerekli olan müşteri aktiviteleri;

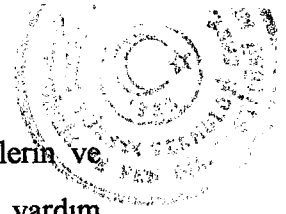
- i) İsterlerin belirtimi aşaması (ISO 9001:1994, Bölümler 4.3, 4.4),
- ii) Proje planlaması (ISO 9001:1994, Madde 4.4.4),
- iii) Tasarım ve Geliştirme (ISO 9001:1994, Madde 4.4.4-5-6-7),
- iv) Doğrulama, sağlama, test (ISO 9001:1994, Bölümler 4.10, 4.12 ve maddeler 4.4.7, 4.4.8),
- v) Ürün kabulü (ISO 9001:1994, Bölümler 4.6, 4.10, 4.15 ve madde 4.4.2, 4.4.7, 4.4.8),
- vi) Dokümantasyon ve veri (ISO 9001:1994, Bölümler 4.5 ve 4.16),
- vii) Konfigürasyon yönetimi: müşteri, ürün kimlik testi, hata raporlarından kaynaklanan değişiklik kontrolü, yazılım modifikasyonu ve yapım ve izleme ile ilgilenmek durumundadır (ISO 9001:1994, Bölümler 4.7, 4.8 ve 4.15).

2.4.11. TickIT-Satıcı Rehberi

Yazılımı tedarik eden firma açısından bakıldığında kalitenin başarılması için üç ana hedefin gerçekleşmesinin gerekliliği anlaşılmaktadır. Bunlar;

- Neyin, ne zaman başarılabileceğinin açık tanımının yapılması,
- İcra edilmesi gerekli faaliyetlerin ve işlevlerin tasviri (standartlar ve prosedürler),
- Faaliyetlerin, işlevlerin ve bunların sonuçlarının kontrol edilmesi ve izlenilmesini (kalite güvencesi ve kalite kontrolü) sağlanmalı.

Başarılı firmaların ve organizasyonların önemli yeteneklerinin temelinde, istikrarlı siparişler kazanmaları ve müşteri isteklerine düzenli olarak zamanında ve en



ekonomik bir şekilde sağlamalarını görmekteyiz. Rehberde açıklanan süreçlerin ve kontrol mekanizmalarının etkin olarak uygulanması bu hedeflere ulaşmada yardım edecektir. Rehberdeki prensiplerin uygulanmasında kullanılacak yöntemler işin büyüklüğüne, karmaşıklığına ve organizasyon veya bölümün yapısına göre değişiklikler gösterir

2.4.11.1. Satıcı Firma Açısından Yazılım Kalite Sistemi İsterleri

Bir kalite sistemi gerçek operasyonları yansıtmalıdır. ISO 9001 sertifikası alan bir firma müşterileriyle birlikte daha önce tanımladıkları anlaşma çerçevesinde çalışmak zorundadır. ISO 9001'in 4.2.3 maddesi, belli ortamlarda kalite isterlerinin nasıl karşılanacağını tanımlamasını ve dokümente edilmesini gerektirir. Bir çok yazılım projesi için bu gereksinim bir *kalite planı* ile karşılanır. Kalite planının kapsamı gereken bazı elemanlar şunlardır:

- Proje sorumlulukları ve gerekli eğitim seviyeleri,
- Uygulanacak yazılım yaşam döngüsü modeli,
- Geliştirme yönetimi ve kullanılacak kontroller,
- Test, kabul ve teslimatın nasıl yapılacağı,
- Müşteri ile gözden geçirme isterleri ve
- Risk yönetim prosedürleri.

2.4.12. TickIT Kalite Sisteminin Geliştirilmesi

ISO 9001 ve TickIT ile uyumlu bir kalite sisteminin geliştirilmesi için yazılım, işletmenin temel hedeflerini destekleyen nitelikte olacak şekilde tasarlanmalı ve geliştirilmesinde organizasyondaki tüm bölümlerin rol alması sağlanmalıdır.

Kalite sisteminin geliştirilmesinde ilk görev; müşteri tatminini ana hedef alan, kalitenin üretici firma ürünleri ve servislerini ne yönde etkilediğini açıklığa kavuşturan, sürekli iyileştirmeleri sağlayan ve kalite için sorumluluk taşıyan bir *kalite politikası* tanımlanmalıdır. Bir kalite sistemi ancak her seviyede çalışanlarla uyumlu ve



kullanılabilir olduđu zaman etkin olabilir. Kalite sistemini kurma giriřimi uzun vadeli çalıřmaları gerektirir fakat genelde yöneticilerin çok aceleci davranmalı yüzünden giriřim başarısız olarak sonuçlanır. Kalite kültürünün bir parçası bir takım denetim süreçlerinden geçer. Denetlemelerin asıl amacı ise sadece uyumsuzlukları belirtmek deđil, süreçleri iyileřtirmek ve çalışanlarında bu süreçlere katkılarını sağlamaktır.

Yazılımda kalite sisteminin geliştirilmesi bir proje olarak ele alınmalı ve kaynakları tedarik edilmelidir. Kalite sisteminin geliştirilmesi projesinin amaçları; müşteri ve çalışanların tatmini, mevcut süreçleri belirleme ve dokümantasyonu, kalite sorunlarını azaltmak, daha etkin süreçler kurmak ve ISO 9001 sertifikasyonunu sağlamak olarak özetlenebilir. Yazılım kalite sisteminin geliştirilmesinde dokümanite edilmiř bir kalite sisteminin hiyerarřı yapısı incelendiđinde;

- *Kalite el kitabı (Quality manual):* ISO 9001 isterlerini kapsar.
- *Prosedürler:* Süreçlerin detaylarını, neyin ne zaman kim tarafından yapılacağını içerir.
- *İř talimatları:* Süreçlerle ilgili doküman yapısı gibi özel bilgileri içerir.

Kalite sistemi deđiřikliđi ve iyileřtirmelere açık olmalıdır. Sadece ISO 9001 isterlerini yerine getiren, zamanla iyileřmeyen bir kalite sistemi gerçekte gereksinimleri karşılamıyor demektir. Yazılım süreçlerinin iyileřtirilmesi için temel araç ölçmektir. Kalite sistemi, sürekli deđiřimi ve iyileřtirmeleri kolaylařtırıyorsa yařam tarzı olur, dramatik yükselmeler yerini sürekli iyileřtirmelere bırakır. Ancak kalite sistem organizasyonu firma tarafından günlük yařamın bir parçası olarak ele alındıđında maliyet, verimlilik ve tatmin gibi gerçek yararları gerçekteřmeye başlayabilir.

2.4.13. ISO 9001 Uyumlu Kalite Sistemi Geliřtirilmesindeki Temel Süreçler

ISO 9001 uyumlu bir yazılım kalite sistemi geliştirme konusunda, yazılım geliştirme, bakım ve destek faaliyetleri için ISO 9000-3:1991’de verilen temel süreçler esas alınmıřtır. Temel ISO 9000-3’de verilen temel süreçler Tablo 2.1’de verilmiřtir.



ISO 9000-3'de verilen Temel Süreçler		
Çerçeve Süreçler	Yaşam Döngüsü Süreçleri	Destek Süreçleri
- Organizasyon Süreçleri, - Kalite sisteminin, tanımlanması, bakımı ve iyileştirmesi - İç denetimlerin yapılması	- Sözleşmelerin gözden geçirilmesi, - İsterlerin tanımlanması, - Geliştirme planlaması, - Kalite planlaması, - Tasarım ve uygulama, - Test, doğrulama ve sağlama, - Kabul, - Teslimat ve kurma.	- Konfigürasyon yönetimi, - Doküman ve veri kontrolü, - Kalite kayıtları kontrolü, - Araç ve tekniklerin kontrolü - Satın alma - Eğitim

Tablo 2.1. - ISO 9000-3 Yazılım Geliştirme Temel Süreçleri

Süreçler için amaç, aktiviteler, kalite sistemi elemanları, kalite kayıt örnekleri, ISO 9001 ve isterlerle uyumluluk göstergeleri, ek rehberlik gerektiren notlar ve ISO 9001:1994 ve ISO 9000-3:1991'e göndermeler konularında ortak bir yapı oluşturulmalıdır.

2.4.14. TickIT Süreç Örneği

ISO 9001 uyumlu bir yazılım kalite sistemi geliştirmede konfigürasyon yönetim (destek süreç) süreç örneği şu şekilde verilebilir:

- **Amaç:** Yazılım ve ilgili dokümanın konfigürasyon statülerinin tasarım, geliştirme, test ve destek süresince tam kontrolünü güvenceye almak.
- **Tipik aktiviteler:** Konfigürasyon öğelerinin belirlenmesi ve kontrolünün sağlanması.
- **Kalite Sistemi Elemanları:** Prosedürler ve konfigürasyon
- **Kalite Kayıt Örnekleri:** Konfigürasyon yönetim planının dokümante edilerek gözden geçirilmesi ve onayı ile her konfigürasyon öğesi için kimlik kayıtlarının tutulması.
- **Göstergeler:** Konfigürasyon yönetim planı, sürüm (versiyon) kontrolü kanıtı ve daha önce çıkarılmış bir yazılımın yeniden inşa edilebilirliğinin kanıtı.
- **Referanslar:**



a) ISO 9001:1994

i) 4.4.9 tasarım değişiklikleri,

ii) 4.5.2 doküman ve veri onayı,

iii) 4.8 ürün kimlik belirleme ve izlenebilirlik,

iv) 4.12 inceleme ve test statüsü.

v) 4.13 uyumsuz ürün kontrolü

vi) 4.13.1 genel

b) ISO 9000-3:1991

i) konfigürasyon yönetimi

2.4.15. Sonuç

TickIT sertifikasyon değerlendirme çalışmalarının ilk üç yılında sertifika alan şirketlerin %19 kadarı İngiltere dışındaki şirketlere verilmiştir. TickIT, İngiltere Savunma Bakanlığının 2.parti sertifikasyon isterlerinin (AQAP 1,13) yerini almıştır. Bazı Avrupa ve Uzak Doğu ülkeleri TickIT'i sertifikasyon programları için temel almaktadırlar.

ISO 9001 ve TickIT uyumlu bir kalite sisteminin elde edinilmesi diğer kalite geliştirme standartlarının (Örneğin; IEEE 730-1989, MIL-STD-498, DOD-STD-2167A-1988, DOD-STD-2168-1998 vs) kullanılmasını dışlamaz.

2.5. Süreç Olgunluk Modeli (CMM-Capability Maturity Model)

2.5.1. Giriş

CMM etkin bir yazılım sürecinin anahtar elemanlarını tanımlayan bir çerçeve modeldir. Olgun olmayan bir süreçten olgun ve disiplinli bir sürece giden evrimsel bir yol çizer. CMM yazılım geliştirme ve bakımının, mühendislik ve yönetsel pratiklerini



kapsar. CMM yazılım sürecinin olgunluğu üzerine hüküm vermek ve endüstrideki pratiklerle karşılaştırmak için bir ölçüm aracıdır (yardstick).

İlk yazılım olgunluk süreç modeli çalışması, ABD hükümetinin kontrat yaptığı yazılım firmalarının yeteneklerini ölçmek için bir yöntem gere duymasıyla başlatıldı. ABD Savunma Bakanlığı'nın girişimi ile Carnegie Mellon Üniversitesi'nde kurulan Yazılım Mühendisliği Enstitüsü, MITRE Co. ile birlikte, yazılım süreç olgunluk çerçevesi ve olgunluk anketini çıkardı (1986). ve 1987 yılında ilk yazılım süreç modeli olarak ortaya koydu. Yazılım Mühendisliği Enstitüsü, 1991 yılında yazılım süreç olgunluk çerçevesi, süreç değerlendirme sonuçları ve diğer bilgiler ışığında CMM olarak yeniden düzenlendi.

CMM, yazılımda süreç olgunluğu ilkelerini ve uygulamalarını belirlemeyi ve gelişigüzel yazılım geliştirmekten, adım adım, olgun ve disiplinli uygulamalara geçişlerde yardımcı olmayı amaçlar. CMM'in ikinci amacı yetenek belirlemektir. Yazılım geliştiren kuruluşların yeteneklerinin belirlenmesi, yazılım edinecek müşteriler açısından çok önemlidir. Belirlenen yetenek düzeyi, edinilecek yazılımın kalitesi ve o yazılım grubu ile çalışmada yatan risklerin değerlendirilmesinde bir göstergedir.

2.5.2. Süreç Olgunluk Çerçevesi

Organizasyon çapında yazılım sürecinin tanımlı olmadığı durumlarda, projelerde başarının tekrarı kişilere bağımlıdır. Bu durum, uzun süreli verimliliği ve kaliteyi iyileştirmeye olanak vermez. Sürekli iyileşme ancak, etkin yazılım mühendisliği ve yönetim pratiklerinden oluşan bir süreç altyapısı inşa ederek mümkündür. CMM olgun bir yazılım organizasyona giden *aşamaları*, öncelik sırasıyla veren bir modeldir.

2.5.3. Süreç Olgunluk Kavramları

Bir *yazılım süreci*, yazılımı ve onun parçası ürünleri (proje planları, tasarım dokümanları, kullanıcı kılavuzları vb.) geliştirmek, ve bakımını yapmak için aktiviteler,

yöntemler ve pratikler kümesi olarak tanımlanabilir. Etkin bir süreç, denenmiş, dokümanite edilmiş, uyulan, ölçülen ve iyileştirilebilen bir süreçtir.

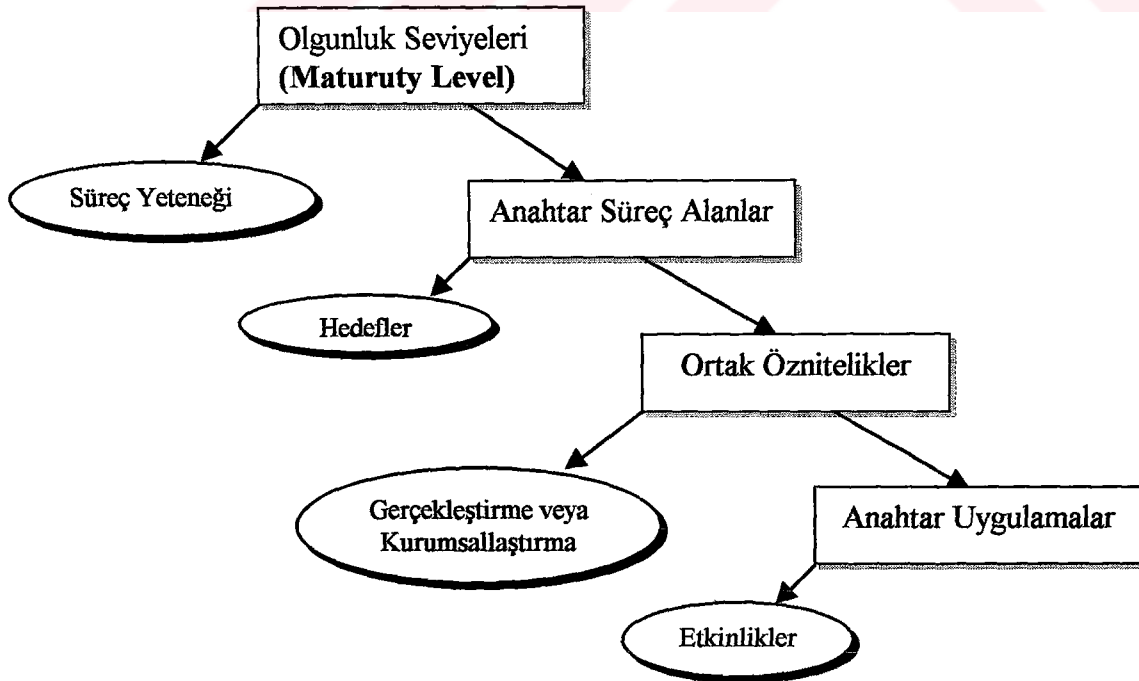
Yazılım süreç yeteneği, bir yazılım sürecinin izlenmesinden beklenen sonuçların menzildir. *Yazılım süreç performansı*, sürecin izlenmesiyle elde edilen gerçek sonuçları temsil eder. *Yazılım süreç olgunluğu*, bir sürecin ne ölçüde açık tanımlandığını, yönetildiğini, ölçüldüğünü, kontrol edildiğini ve etkinliğini belirler.

Olgunluk, organizasyonun yetenek geliştirme potansiyelini, yazılım sürecinin zenginliğini ve projelere yansıtılan tutarlılığı ima eder. Yazılım organizasyonu, süreç olgunluğunu kazandıkça, altyapısı ve organizasyon kültürünü kurarak kurumsallaştırır.

Kurumsallaşma (institutionalization) organizasyonun iş yapma yöntemini, pratik ve prosedürlerini destekleyen bir altyapı ve organizasyon kültürünün oluşturulmasını getirir. Bu kültürü oluşturanlar organizasyonu terk etmiş olsalar bile yöntemler, pratikler ve prosedürler kalır.

2.5.4. Süreç Olgunluk Model Yapısı

Süreç Olgunluk Model (CMM)'in yapısı Şekil 2.1'de gösterilen biçimdedir.



Şekil 2.1 - CMM Model Yapısı



2.5.5. Olgun ve Olgun Olmayan Yazılım Organizasyonları

Olgun bir yazılım organizasyonun karakteristikleri şu şekillerde olmalıdır:

- i) Yazılım geliştirme ve bakımını yönetecek organizasyon çapında yetenekleri mevcut olmalı,
- ii) Yazılım süreci, çalışanlara ve yeni elemanlara doğru bir şekilde iletiliyor ve aktiviteler plana uygun bir şekilde icra ediliyor olmalı,
- iii) Tanımlı süreçlerde roller ve sorumluluklar tüm proje ve organizasyonda belirli, disiplinli bir süreç takip etmelidir. Çünkü katılan herkes bunun değerini bilinci içerisinde.

Olgun olmayan bir yazılım organizasyonun karakteristikleri ise şu şekillerde gözlemlenir:

- i) Yazılım süreçleri proje sırasında yönetici ve uygulayıcılar tarafından doğaçlanır,
- ii) Bir yazılım süreci, belirtimi yapılmış olsa bile, kuvvetle takip edilmez ve mecbur tutulmaz,
- iii) Yöneticiler kriz çözmekle meşguldür,
- iv) Ürün kalitesini ölçmek için nesnel bir temel yoktur,
- v) Kaliteyi yükseltici etkinlikler (gözden geçirme, test gibi), projeler gecikince, çoğu kez atlanır.

2.5.6. Süreç Olgunluk/Yetenek Seviyeleri

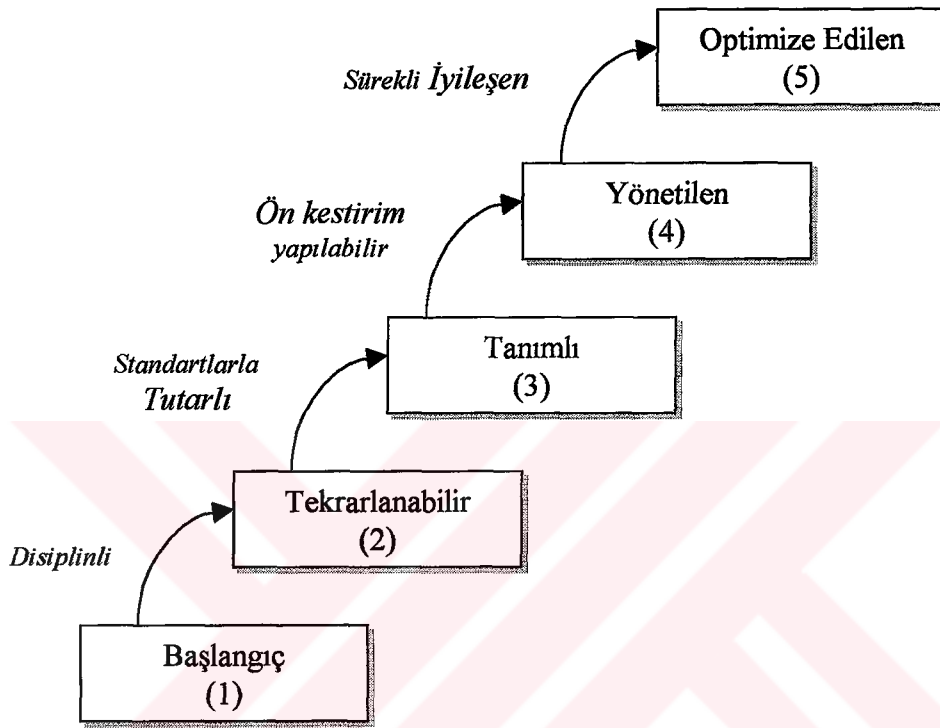
Olgunluk seviyesi, olgun bir yazılım sürecini başarmaya giden evrimsel yolda iyi tanımlanmış bir takım adımlardan oluşur. Her olgunluk seviyesi, süreç yeteneğinin bir seviyesini gösterir. Örneğin, 2. seviyedeki bir organizasyonun süreç yeteneği, gelişmiş olmaktan çıkmış, proje yönetimi kontrollerinin yerleştirilmesiyle disiplin altına alınmıştır. CMM'de beş olgunluk seviyesi vardır.

Süreç olgunluk seviyeleri ve davranış karakterizasyonları:



i) *Başlangıç (initial)*

Yazılım süreci geliş güzel (ad-hoc) ve hatta bazen kaotiktir. Başarı kişisel gayretlere bağlıdır. Kriz durumunda, tipik olarak plan terk edilir, kodlama ve teste dönülür. Yazılım süreç yetenekleri, yazılım süreci, proje yürürken sürekli değiştirildiği veya yeniden düzenlendiği için önceden kestirilemez.



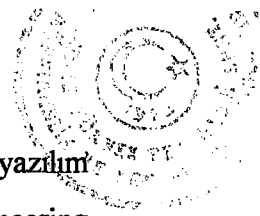
Şekil 2.2 - Yazılım Süreç Olgunluk Seviyeleri

ii) *Tekrarlanabilir (repeatable)*

Proje yönetimi politikaları ve bunları uygulamak için prosedürler oturmuştur. Temel proje yönetimi kontrolleri kurulmuştur. Benzer projelerde başarının tekrarı için gerekli süreç disiplini kurulmuştur. Yazılım istekleri ve bunları sağlamak için geliştirilen ürünlerin için dayanak (baseline) oluşturulur ve bütünlükleri kontrol edilir. Süreç yetenekleri *disiplinli* diye özetlenebilir. Proje yönetim sistemi, proje sürecini etkin kontrol eder.

iii) *Tanımlı (defined)*

Standart yazılım geliştirme ve bakımı *süreci* (mühendislik ve yönetim süreçleri) organizasyon çapında dokümanite edilmiş ve tutarlı bir bütün oluşturur. Bu sürece



CMM’de *organizasyonun standart yazılım süreci* adı verilir. Organizasyonun yazılım süreci aktivitelerinden sorumlu bir grup bulunur. Bu gruba, ‘software engineering process group’ (SEPG) adı verilir. Tüm projelerde standart yazılım geliştirme ve bakım sürecinin projeye göre uyarlanmış (tailored) şekli kullanılır. CMM üçüncü seviyede, yazılım süreç yeteneği standart ve tutarlı olarak tanımlanabilir, çünkü yazılım mühendisliği ve yönetimi aktiviteleri dengeli ve tekrarlanabilir. Tanımlanmış süreçte, süreç yeteneği, aktivitelerin, rollerin ve sorumlulukların ortaklaşa anlaşılmasına dayanır.

iv) *Yönetilen (managed)*

Yazılım sürecinin ve ürün kalitesinin ayrıntılı ölçümleri toplanır. Organizasyon çapında bir yazılım süreç veri tabanı, projelerin süreçlerinin toplanması ve analizi amacıyla kullanılır. Bu seviyede yazılım süreç yeteneği *tahmin edilebilir* olarak özetlenebilir. Süreç ölçülür ve ölçülebilen sınırları içinde işler.

v) *En iyileştiren (optimizing)*

Süreçten gelen nicel geri besleme ve yeni teknoloji ve buluşlardan pilot uygulamalarla yararlanılarak, sürekli süreç iyileştirilmesi yapılır. Tüm organizasyon süreç iyileştirmeye odaklanmıştır. Beşinci seviyede yazılım süreç yeteneği *sürekli iyileşen* olarak karakterize edilebilir.

2.5.7. Süreç Olgunluk Seviyelerini Anlama

Yazılım süreç iyileştirmesi, organizasyonun stratejik planları ve iş hedefleri, yapısı, kullandığı teknoloji ve sosyal kültürü ile bağlantılıdır. CMM toplam kalite yönetimin süreçlerinin yazılım dışında kalan hususlarını da göz önüne alır.

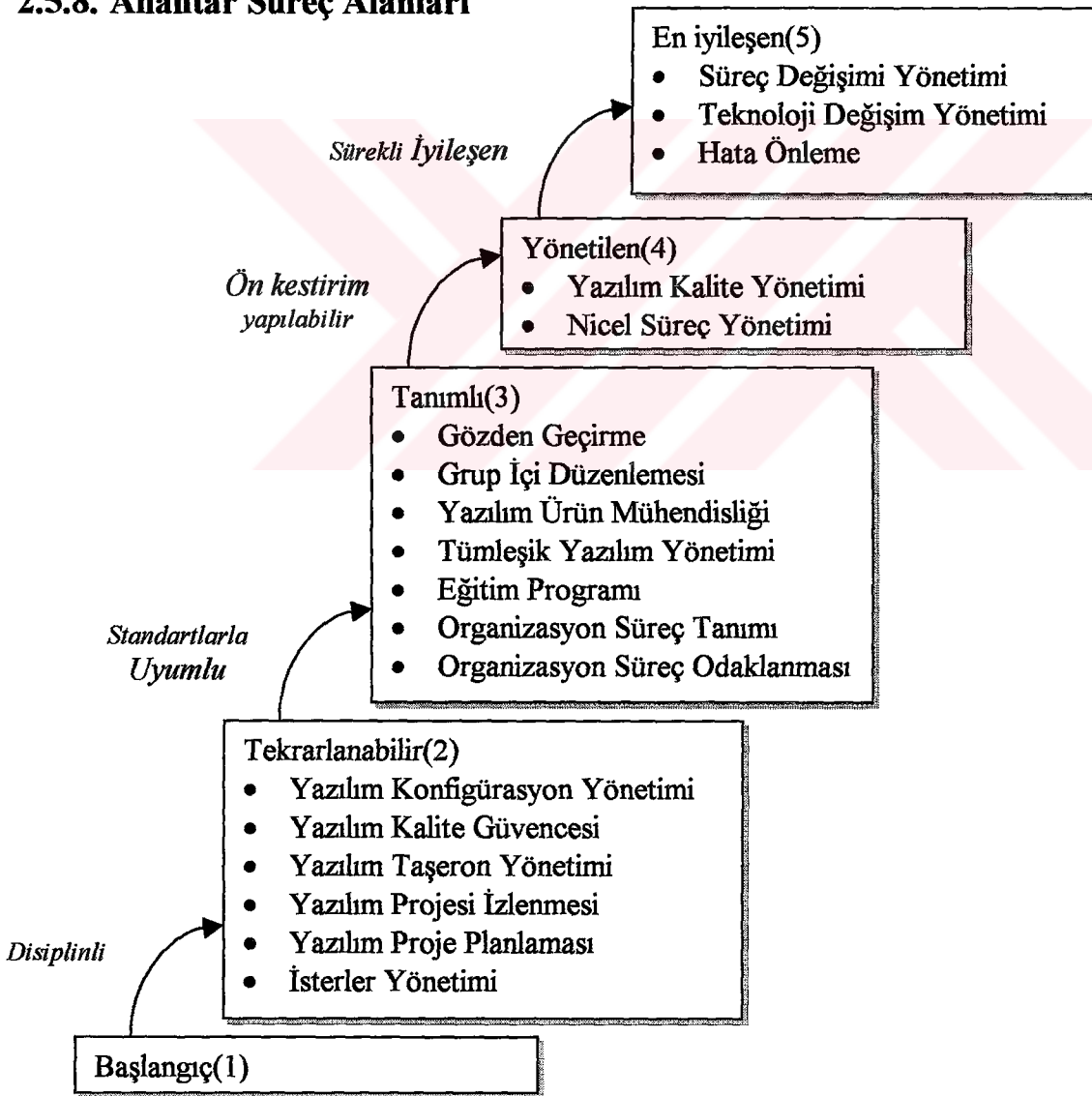
Birinci seviyede organizasyon, maliyet, süre, işlevsellik ve kalite hedeflerinden büyük sapmalar gösterir. İkinci olgunluk seviyesine ulaşmak için yönetim kendi süreçleri üzerine odaklanır. Yönetim 2. seviyeye çıkmak için proje yönetim süreçlerini dokümante eder ve bunlara uyarak liderlik yapar. Üçüncü seviye, proje yönetimi



süreci üzerine, bütün yazılım süreci tanımları entegre edilerek ve dokümanite edilerek inşa edilir. Dördüncü seviyenin en önemli sorumluluğu ve ortak noktası süreç kontrolüdür. Olgunluk seviyesi yükseldikçe projelerde hedeflenen sonuçlarla gerçek sonuçlar arasındaki fark azalır.

CMM'de mühendislik süreçleri (isterler analizi, tasarım, kodlama, test) üçüncü seviyeye kadar tartışılmaz, ancak 1. seviyedeki bir organizasyon bu aktiviteleri yapmak zorundadır. Örneğin, 'peer' gözden geçirme başarıyla yapılabilir. Yukarıda verilen örnekte, ileri seviyelere ait süreçler tam potansiyellerine ulaşamazlar. Örneğin, 'peer' gözden geçirme tutarlı olarak uygulanmadan tam etkili olmaz.

2.5.8. Anahtar Süreç Alanları



Şekil 2.3 - Olgunluk Seviyesine Göre Anahtar Süreç Alanları

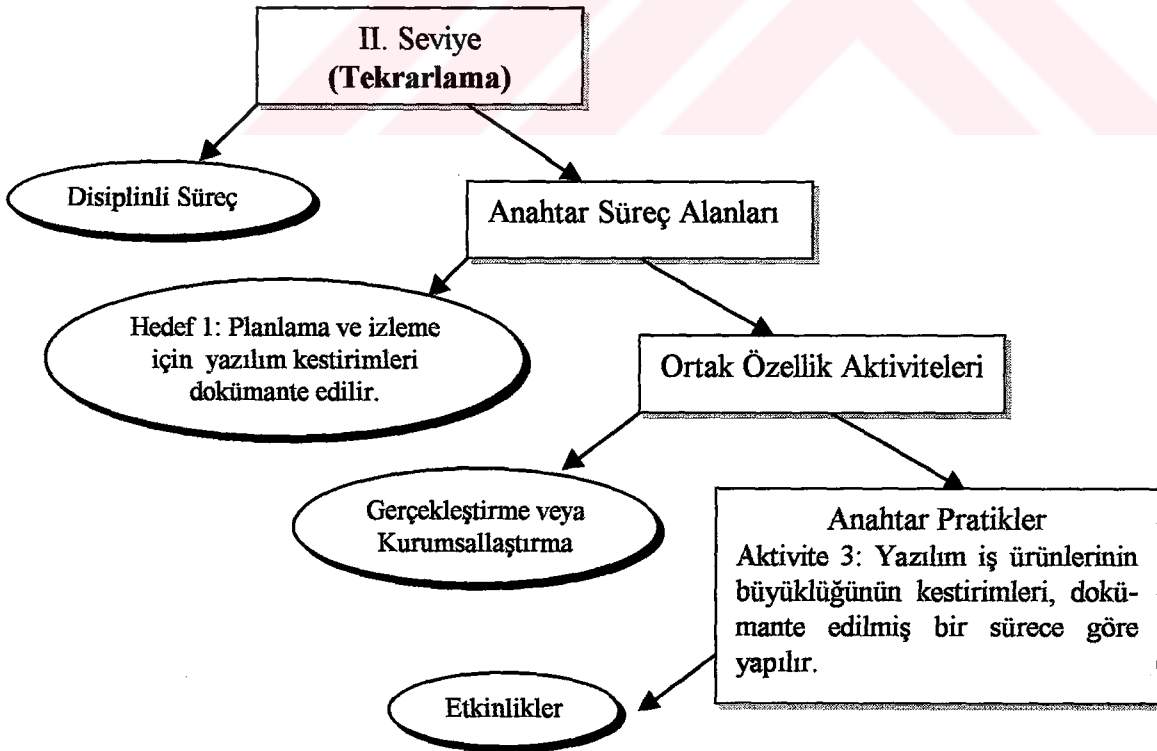


Birincisi dışında, her olgunluk seviyesi çeşitli *anahtar süreç alanlarına* bölünür. Her anahtar süreç alanı birbiriyle ilişkili aktivitelerden oluşur ve bunların birlikte icrası ile anahtar süreç alanı hedeflerine varılır. Bir anahtar süreç alanı tüm hedeflerine ulaşılmasıyla başarılıdır. Bir anahtar süreç alanının hedefleri tüm projelerde tutarlı olarak başarılyorsa, o alanın karakterize ettiği süreç yeteneğinin kurumsallaştığı anlamına gelir.

Her anahtar süreç alanındaki özel pratikler kapsamı, organizasyon ileri seviyelere ulaştıkça genişler. Örneğin, 2. seviyede yazılım proje planlaması anahtar süreç alanında proje önkestirim yetenekleri, ileri seviyelerde gelen yeni verileri de kapsayacak şekilde gelişir.

İkinci seviye için anahtar süreç alanları:

İsterler Yönetimi anahtar süreç alanının amacı, müşteriyle projenin isterleri üzerinde ortak bir anlayış oluşturmaktır. Müşteriyle ilişkinin kontrolü, etkin bir değişiklik kontrol sürecinin takip edilmesine bağlıdır.



Şekil 2.4 - CMM 2. Olgunluk Seviyesi Anahtar Süreç Alanı



Yazılım Projesi Planlaması'nın amacı, yazılım projesini yönetmek ve mühendislik aktiviteleri için mantıklı planlar hazırlamaktır. Gerçekçi planlar olmadan yazılım projesi yönetilemez.

Yazılım Kalite Güvencesi'nin amacı, yazılım projesinde kullanılan sürece ve ürünlere uygun görünürlüğü yönetime sağlamaktır.

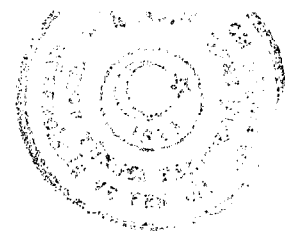
Yazılım Konfigürasyon Yönetimi'nin amacı, projenin yaşam döngüsü süresince proje ürünlerinin bütünlüğünü sağlamak ve korumaktır.

Her anahtar süreç alanı, amaçlarını sağlamaya yardımcı anahtar pratikler aracılığıyla tanımlanır. Anahtar pratikler, anahtar süreç alanının etkin uygulanmasına en çok yardım eden altyapı ve pratikleri tanımlar. Anahtar pratikler '**ne**' yapılması gerektiğini tanımlar, '**nasıl**' yapılacağını tanımlamaz.

2.5.9. Ortak Özellikler

Her anahtar süreç alanında anahtar pratikler ortak özelliklere göre organize edilir. Ortak özellikler, bir anahtar süreç alanının uygulanmasının ve kurumsallaşmasının etkin, tekrara edilebilir, ve kalıcı olup olmadığını belirten niteliklerdir. Ortak özellikler(common feature);

- i) **Yerine getirme taahhüdü** (*commitment to perform*): Sürecin yerleşmesini ve kalıcı olmasını güvenceye alıcı hareketler; tipik olarak organizasyon politikalarının yerleşmesi örneklenebilir.
- ii) **Yerine getirme kabiliyeti** (*ability to perform*): Süreci başarıyla uygulamak için ön şartlar; tipik olarak kaynaklar, eğitim gösterilebilir.
- iii) **Aktiviteler** (*activities performed*): Bir anahtar süreç alanının uygulanması için gerekli roller ve prosedürler,



iv) **Ölçüm ve analiz** : Süreci ölçme ve analiz etme gereksinimleri

v) **Uygulamanın doğrulanması**: Aktivitelerin sürece uyumlu yerine getirildiğini güvenceye alan adımlardır, gözden geçirme örneği verilebilir.

Yukarıdaki ortak özellikleri süreç olgunluk modeli 2.seviyesi anahtar süreç alanlarından yazılım konfigürasyon yönetimi için uygulayacak olursak;

Amaç: Yazılım yaşam döngüsü süresince ürünlerin bütünlüğünü tesis etmek ve korumak.

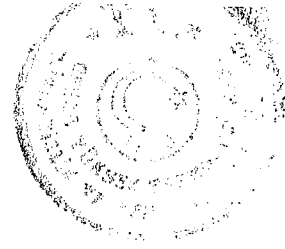
Hedefler: Yazılım konfigürasyon yönetimi aktivitelerini planlamak, seçilen yazılım iş ürünlerini belirleme ve kontrol etme, değişiklikleri izleme ve etkilenen gruplar, yazılım dayanaklarının (baseline) statüleri ve içerikleriyle bilgilendirmenin sağlanması.

Yerine getirme taahhüdü: Proje için yazılım konfigürasyon yönetimi ile sorumlu bir grup vardır ve korunur. Yazılım konfigürasyon yönetimi aktiviteleri için yeterli kaynak sağlanıyordur.

Aktiviteler: Her proje için dokümente edilmiş bir prosedüre göre yazılım konfigürasyon yönetimi planı hazırlanır, konfigürasyon yönetim kütüphanesi kurulur ve konfigürasyon yönetimine alınacak iş ürünleri belirlenir.

Ölçme ve analiz: Konfigürasyon yönetimi planı aktivitelerinin statüsünü belirlemek için ölçmeler yapılır.

Uygulamanın doğrulanması: Konfigürasyon yönetimi planı aktiviteleri düzenli olarak gözden geçirilir. Konfigürasyon yönetimi planı grubu düzenli olarak yazılım dayanaklarını, tanımlama dokümanları ile uyumlu olup olmadığını kontrol etmek için denetler. Yazılım kalite güvencesi grubu ise konfigürasyon yönetimi planı aktivite ve iş ürünlerini gözden geçirir/denetler ve sonuçları rapor eder.



2.5.10. CMM Değerlendirme Sonuçları

Yazılım firmalarının yetenek ve olgunluk seviyelerini ölçmeye ve karşılaştırmaya yarayan CMM, yazılım geliştirmede beş olgunluk / yetenek düzeyi tanımlar. Yazılım geliştiren bir birim, şu olgunluk / yetenek düzeylerinde bulunabilir:

- i) **İlk (initial) Düzey:** Yazılım geliştirme gelişigüzel yollarla veya yetersiz tanımlanmış süreçlerle yapılmaktadır. Başarı kişilere bağlıdır.
- ii) **Tekrarlanabilir (repeatable) Düzey:** Temel proje yönetim süreçleri işlemektedir. Benzer projelerde başarının tekrarı beklenir.
- iii) **Tanımlanmış (defined) Düzey:** Yönetim ve mühendislik açısından yazılım geliştirme süreçleri tanımlanmış, standartlaşmış, entegre edilmiş olarak uygulanır.
- iv) **Yönetilen (managed) Düzey:** Süreç ve ürün kalitesi ile ilgili nicel ölçütler toplanmakta, analiz edilmektedir. Süreçler denetim altındadır.
- v) **Eniyileşen (optimizing) Düzey:** Nicel ölçümlere dayanarak, yeni teknolojilere, fikirlere ve süreçlere girilmekte, kalite ve verim yükseltilmektedir.

Birinci düzey dışında, her düzey için anahtar süreç alanları (key process areas) tanımlanmıştır. Örneğin, CMM ikinci düzeyi için anahtar süreç alanları, İsterler Yönetimi, Yazılım Proje Planlaması, Proje Takibi, Yazılım Kalite Güvencesi, Taşarın Yönetimi ve Yazılım Konfigürasyon Yönetimi alanlarını içerir.

Süreç iyileştirmesi, anahtar süreç alanları üzerinde yoğunlaşmayı ve bunların amaçlarına ulaşmayı gerektirir. Her süreç alanının amaçlarına ulaşmak, "anahtar pratikler" (key practices) aracılığı ile gerçekleşir. Anahtar pratikler, süreç alanlarının temel aktivitelerini ve prosedürlerini oluştururlar. Her anahtar süreç alanı için, uygulama ve kurumsallaşma açısından ne ölçüde etkin, tekrarlanabilir ve sürekli olduklarını gösteren, ortak özellikler (common feautres) altında gruplandırılır.



3. YAZILIM SÜREÇ DEĞERLENDİRMESİ ve SPICE MODELİ

3.1. Giriş

Haziran 1991’de ISO/IEC JTC1’in dördüncü genel toplantısında, yazılım süreç değerlendirmesi için bir uluslar arası standardın ihtiyaç ve gerekliliklerini araştırmak için bir çalışma başlatılması kararı alınır.

11 Haziran 1992’de JTC1/SC 7 N944R çalışmasının sonucunda; Süreç değerlendirmesi için bir standardın ihtiyaç ve gerekliliklerinde uluslar arası bir fikir birliği olduğuna, kullanıcı ihtiyaçlarını tam karşılayabilen bir standart sağlamak ve kabul edilebilir bir zaman aralığında kullanışlı çıktılar sağlayabilmek için deneme ve geliştirmeyi belli bir yola oturtmada uluslar arası bir uzlaşmaya ihtiyaç olduğuna, daha önceki tüm uluslar arası standartlarda yayınlandığı gibi kullanıcı deneme veya tecrübelerinin bir süresince değişmeyen standartların geliştirilmesine olanak vermesi için standart başlangıçta yayınlanması kararları verilmiştir.

JTC1 tarafından 1993’de SPICE (Software Process Improvement and Capability dEtermination) projesi JTC1/SC7 tarafından şu emirlerle bir Teknik Rapor yayınlanmıştır:

- i) Hazırlayıcı aşama olan ilk taslak çalışmasının geliştirilmesinde proje ölçümlendirmesine yardımcı olmak;
- ii) Tüm uluslar arası standartlardaki gibi yayınlanmadan önce yayınlanmış olan standardın gözden geçirilmesinin temelinde önceki kazanılmış deneyimlerden kazanç elde etmek için kullanıcı tecrübeleri göz önüne alınmalıdır;
- iii) Standardın değişimi tekrar başlatılmalı ve pazar isteklerinden habersiz oluşturulmalıdır.



SPICE tasarısı buradaki görevlerden ilkinin tamamlanmıştır. ISO/IEC çalışmaları grubu (ISO/IEC SC7/WG10) Haziran 1995’de PDTR’ ye kayıtlı SC7’nin gizli oylaması sonucunda Teknik Rapordan sonraki yayınlanmış taslak çalışmasını geliştirmeden sorumlu tutulmuştur. SPICE taslağının birinci evresinden geriye dönen raporlar deneyim ve kurumun üye yorumları olan tecrübeler kullanılır.

Yayınlanan Teknik Rapor2’de Yazılım Süreç Değerlendirme dokuz ana bölüm altında toplanmıştır. Bu Teknik Rapor yazılım süreçlerinin değerlendirmesi için bir yapı sağlar. Bu yapı mevcut, işletilen, geliştirilen yazılımın desteklenmesi ve kontrol edilen, denetlenen, yönetilen ve planlanan ilgili organizasyonlar tarafından kullanılacak olmalıdır.

Bir süreç değerlendirme çerçevesi şu şekillerde olmalıdır:

- i) Kuruluşun kendi değerlendirmesine yardım etmeli,
- ii) Değerlendirilen süreçlerde yönetimin yeterliliğini belirtmeli,
- iii) Değerlendirilen süreçlerin işletilmesindeki genel durum raporunu vermeli,
- iv) Başarısızlıktan ziyade süreç bir süreç profilini set eden üretim yapmalı.

Bir kuruluşun kaliteli ürün geliştirebilmesi için, uygun geliştirilen bir program parçası gibi kullanılan sonuçların anlamına sahip olan ve süreçlerin durumunu değerlendirmek için tutarlı, güvenilir bir yöntem ve bazı edinilmiş deneyimlere sahip olması gerekir.

Organizasyonun içindeki süreç değerlendirmesinin kullanımı şunlara yardım etmelidir:

- i) Doğru mekanizmaların oluşturulması ve sürekli iyileştirmeleri kültür olarak benimsemek ve kültürün bakımını yapmak,
- ii) İş gereksinimlerini karşılamada süreç mühendisliği,
- iii) Kaynakların optimizasyonu sağlama.



Bu en son kullanıcı tatminini maksimuma çıkarttığı gibi onların ürünlerinin tüm maliyet yaşam süreçlerini en aza indirir ve pazar ihtiyaçları, müşterilerine yanıt verememe en fazla olan yetenekli organizasyonları doğuracaktır.

Müşteriler süreç değerlendirmelerinin kullanımından yararlanacaklardır. Onun yetenek belirlemede kullanımı;

- i) Anlaşma hükmünden önce tasvir edilmiş olan anlaşmanın yeteneği ile birleştirilmiş olan risklerin olanaklı kılınmasıyla yoğun sistemlerde, yazılım sağlayıcıları seçmede kararsızlığa zorlayacak,
- ii) Risk içeriği (countainment) yerine konmuş olan uygun kontrolleri olanaklı kılma,
- iii) Boy ölçüşen (münakaşaya giren) sağlayıcıların yeteneklerine karşı proje maliyet tahminleri göz önünde bulundurmanın seçimi için temel nitelikleri sağlamak.

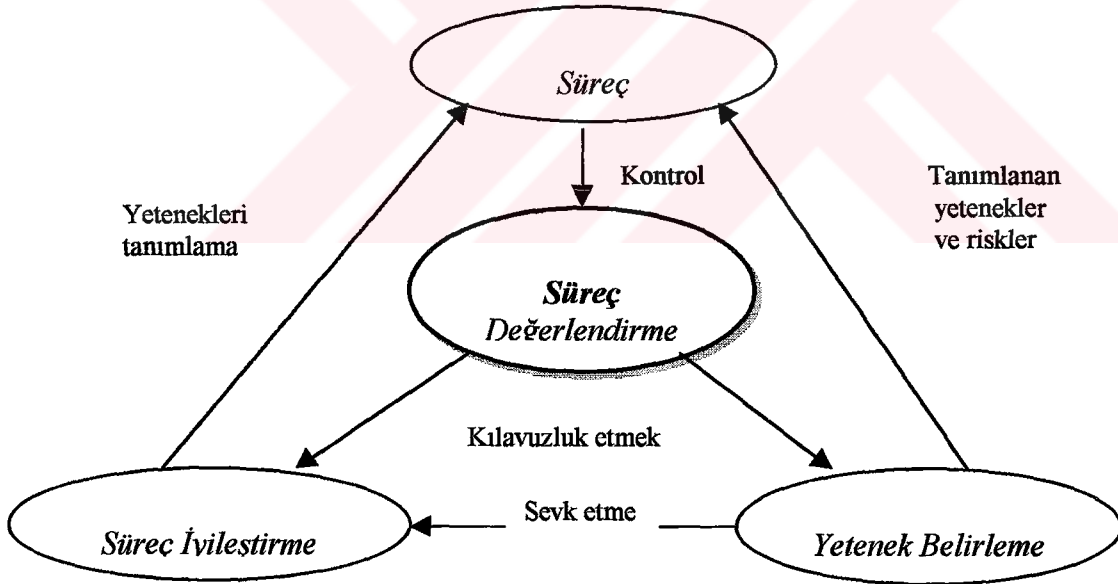
Süreç değerlendirmede standartlaştırılan bir yöntemin daha fazla yararları aşağıdakilerin yerine getirilmesiyle sağlanacaktır.

- i) Süreç değerlendirme model paylaşımına bir alan sağlama,
- ii) Yetenek değerlendirme ve süreç iyileştirmesi için süreç değerlendirmesinin kullanımının yaygınlaşmasına liderlik etmek,
- iii) Yetenek belirlemeyi herkese açık temin etmeyi kolaylaştırmak,
- iv) Kullanıcının deneyimi dikkate alarak düzenli olarak gözden geçirme ve kontrol etmiş olmak,
- v) Yalnız uluslar arası anlaşmayla değiştirebilmek,
- vi) Varolan tasarımın birbirine uymasına yardım etmek.

Teknik raporda tanımlanan süreç değerlendirme yöntemi süreç değerlendirmesinin sonucunda tasvir edilen yaygın yöntemin temelini oluşturmak için tasarlanmıştır. Süreç isteklerinin karışık ve kullanıcı tecrübelerine bağlı olması onun genel durumundan kaynaklanmaktadır. Örneğin beş kişilik bir proje takımı için planlama gereksinimleri elli kişilik bir takımdan daha azdır. Bu genel durum süreç profilleri arasındaki karşılaştırılabilirlik derecesi etki ve uygunluğunun değerlendirilmesi sırasında bir pratiğe kalite değerlendiricilerin nasıl karar vereceğine etki eder.

Süreç değerlendirme Şekil 3.1'deki diyagramda gösterildiği gibi kullanımında başlıca iki uygulama alanına sahiptir. Süreç iyileştirme boyutu; süreç değerlendirmede seçilmiş olan süreçlerin yeteneği süresince bir organizasyon birimi içerisinde geçerli olan pratiklerin karakteristiklerini anlatmayı sağlar. Kuruluş yöneticilerinin verecekleri tecrübelerden elde edilecek sonuçların analizi süreçlerin doğasında var olan sertlik, güçsüzlük ve riskleri tanımlar. Bu bize süreçlerin amaçlarını gerçekleştirmede etkin olup olmadıklarını belirleyebilmemizi, kötü kalitenin önemli sebeplerini tanıyabilmemizi yada zaman ve maliyet farklılıkları bulabilmemize olanak sağlar. Bunlar süreçlerde öncelikli geliştirmelerde sürücülerini sağlar.

Süreç yetenek belirleme boyutu, seçilen sürecin bir projede kullanımını üzerine almada risk gerektirmeyi tanımlayabilmek için bir hedef süreç yetenek profiline karşı süreçlerin seçiminde önerilen yetenek analizinden etkilenmiştir. Önerilen yetenek önceki süreç tahminlerinin sonuçlarına dayanabilir yada önerilen yeteneği kurma amacı için yapılmış bir değerlendirmeye dayanabilir.



Şekil 3.1 – Yazılım Süreç Değerlendirme

Yayınlanan teknik rapor müşterilerin, sağlayıcıların ve denetleyicilerin gereksinimlerini ve tek bir kaynak içerisinde oluşan bireysel gereksinimlerini karşılamak amacıyla tasarlanmıştır. Bu doküman takımının kullanımıyla müşteriler, sağlayıcılar ve denetleyiciler daha fazla fayda elde edebilirler.

Müşteriler:

- yazılım süreç sağlayıcıların mevcut ve potansiyel yeteneğini belirleme kabiliyeti.

Sağlayıcılar:

- kendi yazılım süreçlerinin potansiyel ve mevcut yazılım kabiliyetlerini belirleme yeteneği,
- yazılım süreç iyileştirmesi için öncelikleri ve alanları tanımlama yeteneği,
- yazılım süreç iyileştirmesi için yol haritası tanımlayan bir çerçeve sağlama.

Denetleyiciler:

- yönettiği değerlendirmelerin tüm yönlerini tanımlayan bir çerçeve sağlama.

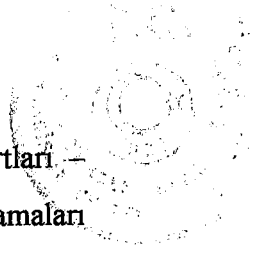
Teknik Rapor organizasyonlar ve süreçlerin etkisizliği ve yetenek değerlendirmede diğer modeller ve yaygın diğer uluslar arası standartlar için tamamlayıcıdır.

Teknik raporda diğer ihtiyaçları karşılamada yeteneğe sahip sağlayıcıların güçlü olup olmadığını değerlendirmede bir yapı elde edinmeyi sağlama, sağlayıcıların kalite yönetiminde güven sağlamada ISO 9001'i temel alan kalite bilançolarının başarılı/başarısız karakteristiklerinin kullanımından ziyade tekrarlanabilen, karşılaştırılabilen, sürekli ölçümlemeyle süreç yetenek kabiliyetini belirlemede kullanıcılara bir yol sağlar. Bununla birlikte bu Teknik raporun tasvir ettiği yapı bir organizasyon biriminde kullanılmış olan tüm süreçlerden ziyade ilgili özel süreçleri maskelemede değerlendirmenin faaliyet alanını ayarlamak için elverişli zamanı ayarlar.

Teknik rapor ISO 9000 serilerinin takip edilen belirli kısımlarıyla ilgilidir.

Bunlar:

- ISO 9001 – 1994, Kalite güvencesi tasarımı, geliştirmesi, üretimi, kurulumu ve servisi için modeldir.

- 
- ISO 9000-3 – 1991, Kalite yönetimi ve kalite güvencesi standartları – Yazılımın geliştirilmesi, bakımı ve tedarikinde ISO 9001'in uygulamaları için rehber.
 - ISO 9004 1993, Kalite yönetimi ve kalite sistem elemanları – Kalite geliştirme için rehber.
 - ISO/IEC 12207-1-1994, Yazılım yaşam çevrim süreçleri kuvvetlice ilişkilidir.
 - ISO/IEC 10011-2:1991, Kalite kontrol sistemi için rehber – Kalite sistem denetleyicileri için Kalite kriteri.

Teknik rapor, kullanıcının süreç iyileştirme ve süreç yetenek belirlemesi olan iki genel durumunu ve yazılım süreç değerlendirmesi ile ilgili tüm genel kavram bilgilerini verir. O uygun takımlarının birlikteliğinin bölümleri tasviri ve onların seçim ve kullanımı için rehber sağlar.

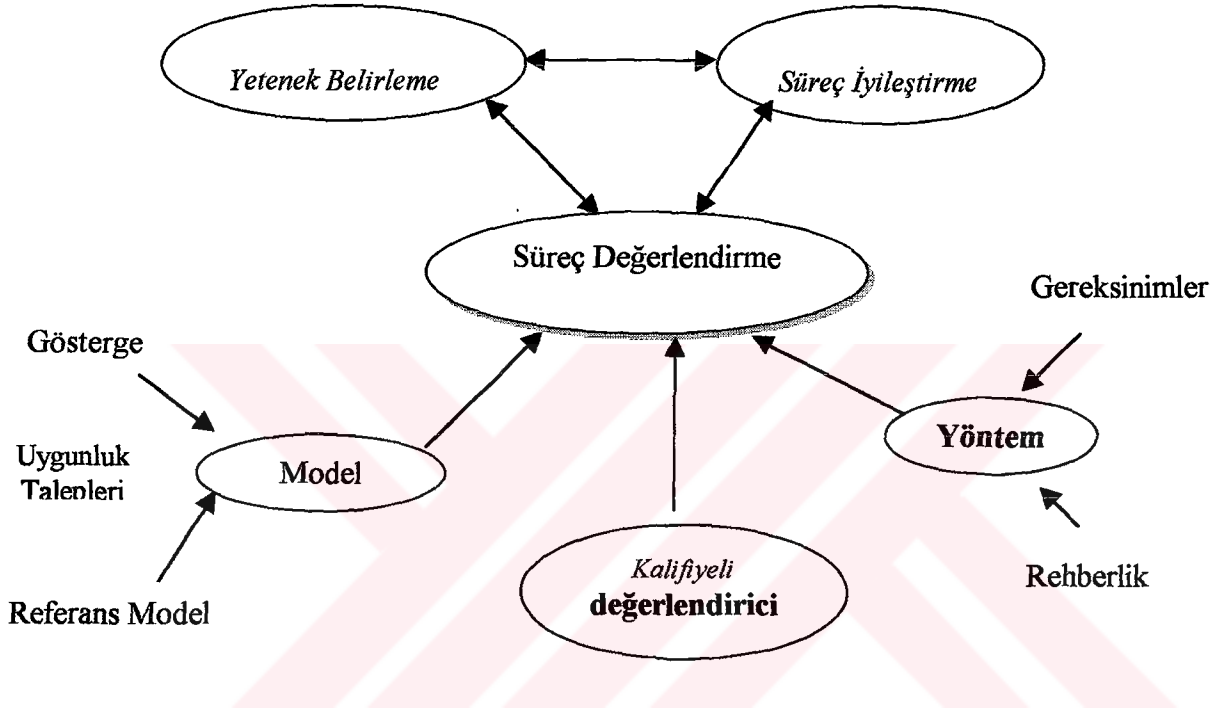
3.2. Genel Kavramlar

ISO/IEC JTC1/SC7'nin yayınladığı Teknik Rapor, yazılım süreçlerinin değerlendirilmesinde bir yapı (çerçeve) sağlar. Bu yapı yazılımın destek ve gelişimi, işletilmesi, geliştirilmesi, desteklenmesi, kazancın artırılması ve kontrolü, izlenmesi, yönetilmesi ve planlanmasıyla ilgilenen organizasyonlar tarafından kullanılacak olmalıdır.

Süreç değerlendirme, organizasyonların amaçlarını başarmaları için kuruluş tarafından kullanılmakta olan süreçleri gözden geçirir. Değerlendirme seçilmiş olan süreçlerin yeteneği açısından bir organizasyon içerisinde ki geçerli pratiği nitelendirir. Sonuçlar, süreçlerin asıl risk ve zaafları, tanımlanan güçlükleri, kuruluşların iş ihtiyaçlarının içeriğinin sonuçları analiziyle süreç yetenek belirleme veya süreç iyileştirme aktiviteleri yürütmede kullanılabilecek olmalıdır.

Standart ya süreç yetenek belirleme yada süreç iyileştirmeleri için kullanılabilen ve benzer genel durumlar içinde karşılaştırılabilen, nesnel ve tekrarlanabilen değerlendirme sonuçları sağlamak için tasarlanmıştır.

Süreç değerlendirme, süreç iyileştirme ve süreç yeteneklerinin belirlenmesi arasındaki ilişkiler Şekil 3.2’de belirtilmiştir.



Şekil 3.2 – Standart Parçalar Arasındaki İlişkilerin Genel Görünümü

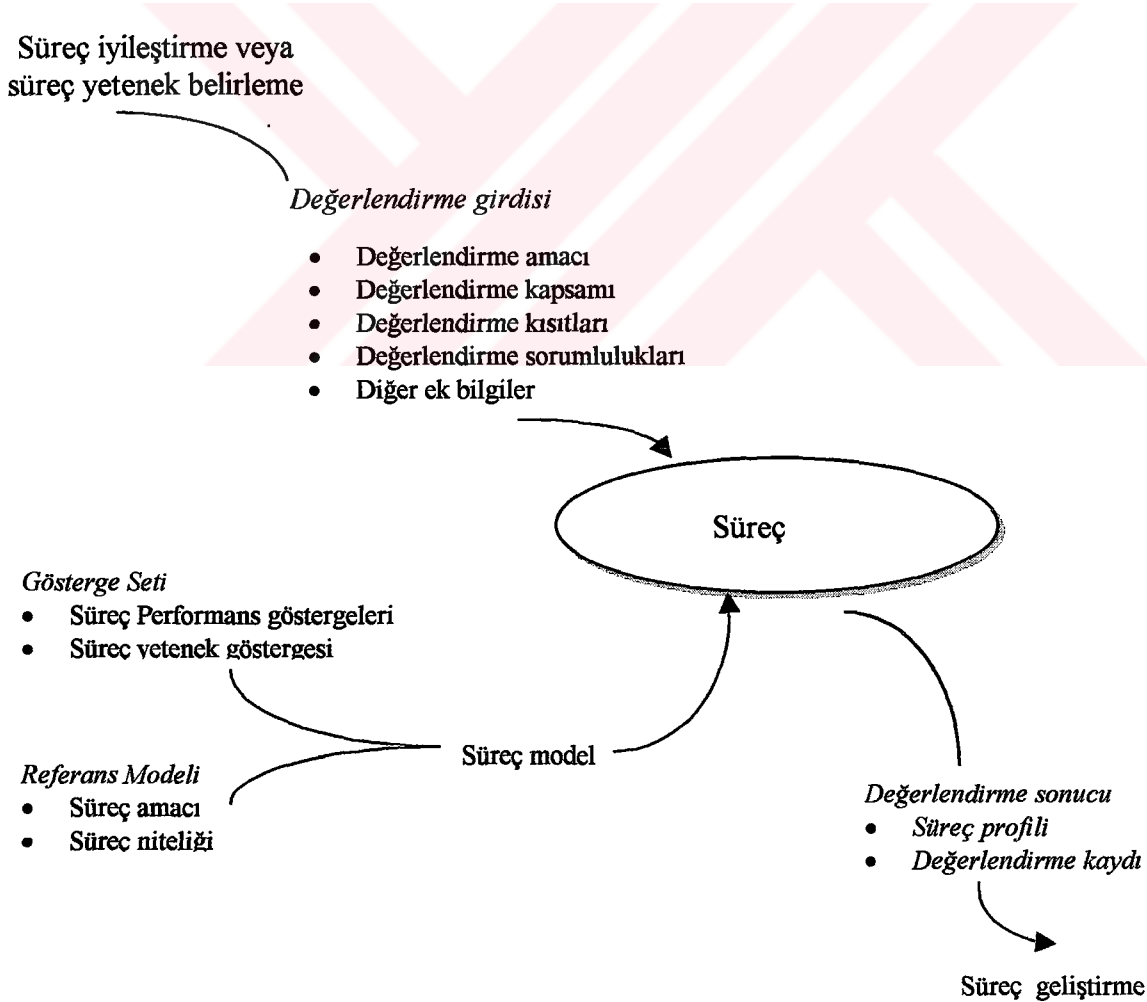
3.2.1. Süreç Değerlendirme İçeriği

Bir süreç değerlendirmesinin içeriği Şekil 3.3’de özetlenmiştir. Teknik Raporda tanımlanan referans model süreç niteliklerinin başarısının temelinde süreçlerin yetenek tahmini için bir temel sağlar. Teknik Rapor, değerlendirme performansı için olan gereksinimleri tanımlar ve gereksinimlerinin yorumlanması ile değerlendirmelerin yerine getirilmesinde rehberlik sağlar. Çeşitli metotlar, teknikler ve araçlar kullanılarak yapılacak olan bu rehberlik bütün kuruluşlar ve idari değerlendirmelere uygulanmak için yeterli derecede verimlidir.



Süreç değerlendirmeleri için resmi giriş değerlendirme amaçları belirlenen değerlendirmeler, değerlendirme sahası ve değerlendirme esnasındaki her hangi bir zorlamaların karşılaştırmasıyla olur. Değerlendirme girdileri değerlendirmelerin gerçekleştirilebilme sorumluluğunu tanımlamalıdır. Bir değerlendirme; değerlendirme- de seçilen süreç modeline karşı seçilmiş olan süreçlerin değerlendirmesiyle olur. Bu süreç model Teknik Raporda tanımlanan referans model ile uyum içinde olmalıdır. Bu iki boyutlu model süreçlerin ve süreç niteliklerinin bir kümesinden oluşur. Yetenek nitelikleri tüm süreçlerin çapraz uygulanmasıyla olur. Değerlendirme sonucu değerlendirilen her bir örnek süreç için yetenek not seviyesi seçenekleri ve set edilen süreç profillerini içerir.

Bir değerlendirme, hedeflenen başarı notlarını doğrulamada kullanılan yetenek veya performansını nesnel olarak gösteren kayıtlarla belgelendirilmiş olmalıdır.



Şekil 3.3 – Süreç Geliştirme Genel Durumu



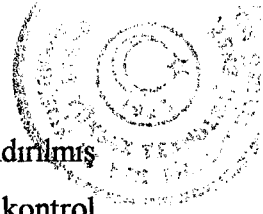
3.2.2. Süreç Değerlendirme Mimarisi

Süreç değerlendirmesinin amaçları için kullanılmış olan her bir model için temel formları olan süreçler ve süreç yetenekleri için referans bir modeli tarif eder. Referans model süreç yeteneğinin değerlendirilmesinde iki boyutlu bir yaklaşımı içerir. Birinci boyutta değerlendirilmekte olan süreçler tanımlanır, diğerinde ise yeteneğin ölçümlendirilmesi için ölçüyü tarif eder. Her bir model değerlendirme kullanılmakta olan referans model ile uyumlu olmalıdır ve her bir uyumlu değerlendirme sonuçları yaygın bir kullanın alanına sokulmuş olabilecektir.

Her bir süreç, süreci ifade etme gayesi ile tasvir edilmiştir ki bu süreç gösterimlerinin istenilen sonuçlarının bir taslağını içerir. Süreçler Tablo 3.1’de gösterildiği gibi beş süreç kategorisine dönüşen gruplara ayrılır.

Süreç kategorisi	Kısa tanımlama
Müşteri-Tedarikçi	Müşteriye doğrudan etki eden süreçler, müşteri yazılımlarının geçişi ve geliştirilmesinin desteklenmesi ve bunların doğru işletim ve kullanımının sağlanması.
Mühendislik	Doğrudan belirtilen süreçler, yerine getirme veya bir sistemin bakımı ve yazılım ürünü ve bu ürünün kullanıcı belgeleri
Destekleme	Yazılım yaşam çevriminin değişik noktalarında diğer süreçlerin (diğer desteklenen süreçleri içeren) birinde kullanılmış olan.
Yönetim	Yazılım yaşam çevriminde projenin herhangi bir aşmasının yönetilmesini biri tarafından kullanılmış olmada türe ait olan pratikleri içeren süreçler.
Kuruluş	Organizasyonun projelendirilmesinde kullanıldığı zaman kuruluşun hedeflerini başarmasına yardım edecek olan kaynak aktifler, ürün, gelişen süreç ve kuruluşun iş amaçlarını kuran süreçler.

Tablo 3.1 – Süreç Kategori Grupları



Süreç yetenek geliştirme yetenek seviyelerinin bir serisi halinde gruplandırılmış olan süreç niteliklerine göre ifade edilmiştir. Her bir yetenek seviyesi sürecin kontrol ve yönetimindeki artımsal değerlendirmeyi sunar böylece değerlendirme modelleri artımsal yetenekler için bir yol gösterir.

3.2.3. Süreç Değerlendirme Göstergeleri

Süreç yeteneği ve başarımlar oranlarını artırmada değerlendirme olaylarını belgelendirerek tutarlılığı, güvenilirliği ve tekrarlanabilirliğini maksimum yapma için bu olaylar kaydedilmeli ve saklanmalıdır. Bu olaylar süreç performans ve yeteneğini gösteren form göstergelerindedir ki bunların değerlendirme adımlarının nesnel olarak karakteristikleri gösterilmiştir. Süreç değerlendirmelerinin tam bir modeli detayları gösteren göstergeleri içerir.

Bu değerlendirme göstergeleri bazı değerlendirme formları kullanılarak basit bir şekilde belgelendirilebilir ve bu belgeler el kitabı olarak tasarlanabilir. Örneğin kontrol listeleri (örneğin, soru ve ya kontrol liste formları) veya işlemleri otomatizme edilebilir. Örnek olarak veri tabanı veya uzman sistem teknoloji aletleri kullanılması verilebilir.

3.2.4. Değerlendiricilerin Kalitesi

Kalite değerlendiricilerinin en önemli rolü diğer takım üyelerinin değerlendirme ve bu konudaki bilgilerin doğru bir şekilde yönlendirilmesine sahip olanları bir arada tutmayı sağlamaktır. Kalite değerlendiricisi, takıma gerekli olan öncülüğü sağlar ve yorumların tutarlılığını sağlamada diğer takım üyelerinin oluşturduğu tahmin notların modernize edilmesini sağlar.

Değerlendirmelerin yapılmasında kalite değerlendiricileri ve hazırlayıcılarına sağlanan rehberlik, özellikle onun amaçları, başlangıç ve değerlendiricilerin sonraki

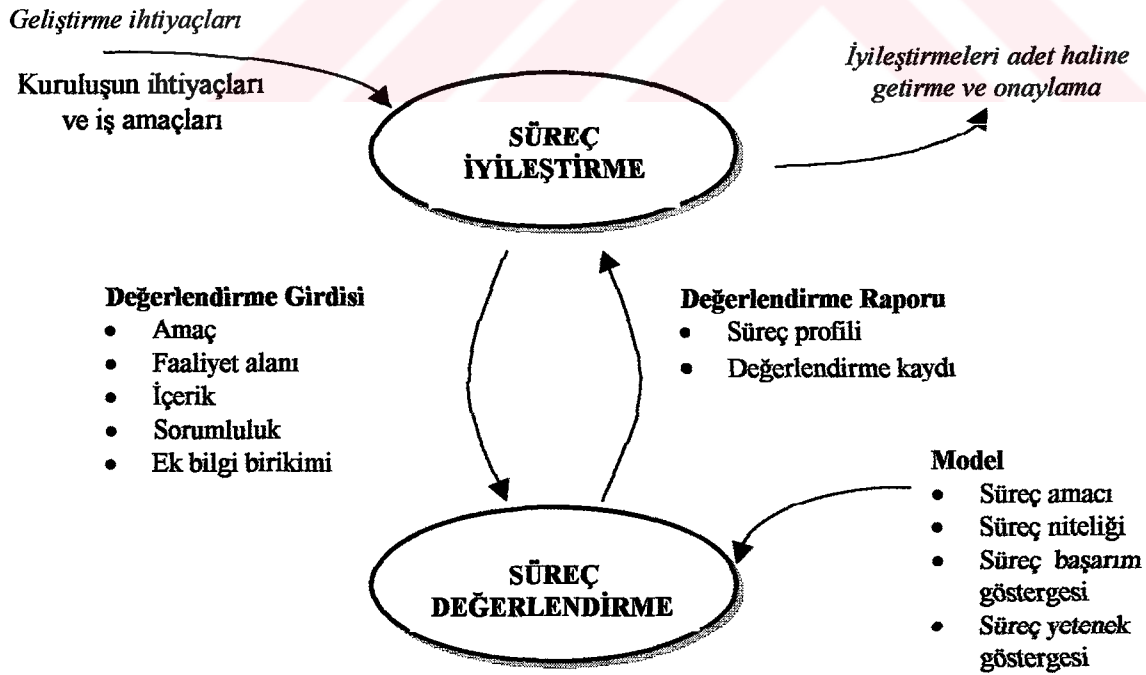
değerlendirmelerine rehberlik sağlamalıdır. Bu, değerlendiricilerin bu konudaki yeterince yetkisi ve almış olduğu eğitimiyle ilgilidir.

Bir değerlendirme sağlanıyorken (yerine getiriliyorken) kalite denetleyicileri belgelendirilen ve izlenen sürecin gereksinimlerini sağlamadan sorumludur. Belgeleme belli gereksinimleri kapsamaktadır:

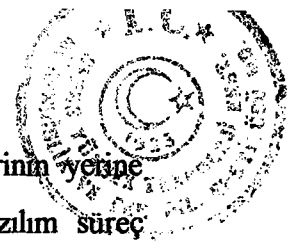
- değerlendirme amacı
- değerlendirme kapsamı
- değerlendirme içeriği
- değerlendirme sorumlulukları
- süreç iyileştirme ve yetenek belirlemede ek bilgi birikimi

3.2.5. Süreç İyileştirme Kavramı

Başarılı bir yazılım süreç iyileştirme, organizasyonunun açıkça belirtilen ve anlatılan amaçları ile özel ihtiyaçlarını karşılayabilen bir görev içeriğine sahip olmalıdır.



Şekil 3.4 – Süreç İyileştirme



Teknik Raporda sürekli bir çevrimde yazılım süreç iyileştirmelerinin yetine getirilebilmesi için metot ve tamamlanmış bir yapının bölümü gibi yazılım süreç değerlendirmelerinin kullanımında rehberlik sağlar. Bununla birlikte tek çevrimler için iyileştirme aktivitelerini kapsamaktadır. Süreç iyileştirme içerik ayrıntıları şekil 3.4’de gösterilmektedir.

Rehber kılıflar;

- bir yazılım süreç değerlendirmeyi çağırma,
- bir yazılım süreç değerlendirme sonuçlarını kullanmayı,
- yazılım süreç etkisizliği ve geliştirme etkisizliğini ölçümleme,
- iş amaçları için iyileştirme hareketlerini tanımlama,
- iyileştirmeler için referans modeli kullanma,
- yazılım süreç iyileştirmelerinin içeriğinde kültürel yayınları kullanma,
- idari yayınlarla ilgili iş ilişkilerinin yazılım süreç iyileştirmeleri üzerindeki etkisi.

Rehber sağlama özel organizasyonların yapıları, yönetimi, felsefeleri ve yazılım yaşam çevrim modelleri veya yazılım geliştirme metotlarını göz önüne almaz. İçerikleri ve prensipler farklı iş gereksinimlerinin her şeyi için uygundur (uygulama alanları ve organizasyonun büyüklüğü) böylece onlar gelişim aktivitelerine öncülük etmek için bütün yazılım organizasyonlarında kullanılabilir.

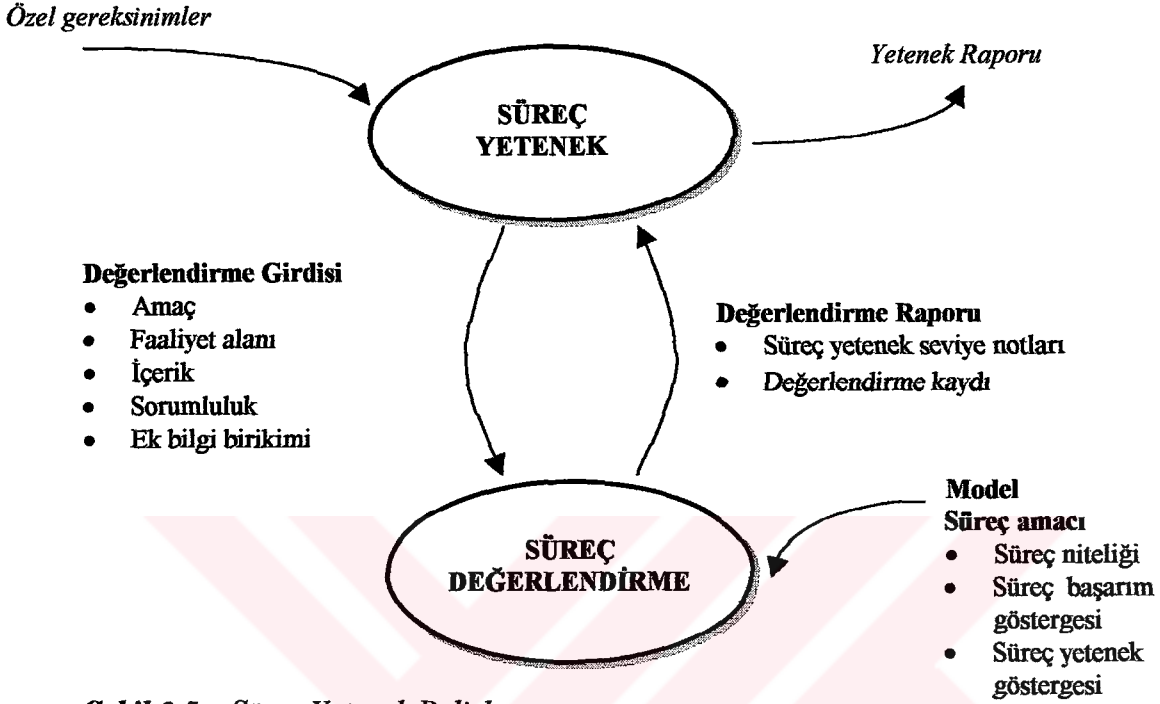
3.2.6. Süreç Yetenek Belirleme Kavramı

Süreç yetenek belirleme standartta tasvir edildiği gibi esasında süreç değerlendirmelerin üzerine inşa edilmiştir. Süreçler tanımlanan süreç modellerine karşı ölçümlendirmeyi kullanarak puanlandırılır. Süreç yetenek belirleme kavramı Şekil 3.5’de gösterilmiştir.

Yazılım ürünleri veya servislerinin kazanılması belirlenen ihtiyaçlarda açıklandığı gibi teknik ve diğer bazı ihtiyaçlara sahiptir. Öncelikle bir yazılım tedarikçisi, süreç yetenek belirlemede yazılımı elde edeceklerin ihtiyaçlarını karşılayan



bir sözleşme yapmalı veya yazılım isteğini karşılamadan önce yazılımı sağlayanlar kendi süreç yeteneklerini araştırmalıdır. Süreç yetenek belirlemede teknik veya diğer ihtiyaçların açıkça belirtilen gereksinimleri belgelendirilmiştir.



Şekil 3.5 – Süreç Yetenek Belirleme

Açıkça belirtilen gereksinim süreç değerlendirmedeki amaç girdi ve süreç yetenek gereksinimi şeklinde sunulan bir hedef yeteneğe dönüştürülmüştür. Tedarikçi bir organizasyon biriminin ilgili birimleri ile teklif edilen süreç-süreç yetenek seviye notlarının set edilmesi gibi ileride önerilen bir süreç yeteneği koymalıdır. Basit bir haliyle teklif edilen süreç yeteneği son zamanlardaki yarı değerlendirme veya diğer anlamlarını temel almış olmalıdır. Daha çok karışık seçimlerde iyileştirme planlarıyla ilgili ve tedarikçinin geçerli profillerini temel alan gelecekte tamamlanmış bir süreç yeteneğini tedarikçiler önermelidirler. Önerilmiş olan süreç yeteneğinin inanırılığı sunulan bir risk içermesiyle birlikte analiz edilmiş ve süreç yetenek raporunda belirtilmiştir.

Değerlendirme girdilerinin belgelendirilmesi, izlenebilirlikte gerekli gözden geçirimlerin yerine getirilmiş olmasını sağlıyor olabilmeye emin olmayı ve değerlendirme kayıtlarının da muhafaza edilmiş olmasını sağlamalıdır.



3.3. SPICE Modeli

Yazılım Süreç İyileştirme ve Yetenek Belirleme (*SPICE-Software Process Improvement and Capability dEtermination*) modeli; yazılım mühendisliğinin temelini teşkil eden evrensel yazılım süreçlerinin belgelendirilmesi, en iyi pratik (*best practice*) yapma aktivitelerini kapsayan, süreçlerin yeteneklerini karakterize eden, yazılımın; değişimi, işletilmesi, geliştirilmesi, tedariki ve elde edilmesi ile organizasyon süreçlerini kapsayan *referans* modeldir. Süreç değerlendirilmesinin yapılmasında değerlendirici bu referans model ile uyumluluğu kullanır. Böylece görevlendirilen karar vericiler ile temel ortaklıklar sağlanmış olur.

Referans modelin amacı, yazılım süreç değerlendirmedeki farklı modeller ve metotlar için temel bir yaygınlığı sağlamaktır. Bu ise değerlendirme sonuçlarının genel bir içerik raporlanmasını sağlayacaktır. Referans model ile uyumlu modellerin kullanımı, değerlendirme notlarının raporlanmasında yaygın kavramları güvence altına alacaktır. Yaygın bir referans model formlarının kullanım esası olan karşılaştırmalar esas teşkil edebilir.

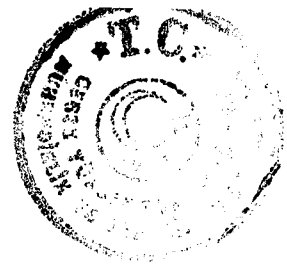
Referans model mimarisinin iki boyutu vardır. Birinci boyutu *süreç boyutu* olup, hedeflenen ifadelerin bir setiyle karakterize edilmiştir. Süreç amaç ifadeleri tanımlanan süreç amacını elde etmek için başarılmış olan terimlerin ölçümlenmesiyle tanımlanır. Süreçler, ISO12207 (1995, Yazılım yaşam çevrim süreçleri) ile sıraya konularak oluşturulmuştur. Diğer boyutu ise bir hedefe ulaşmada kuruluş tarafından kullanılmış olabilen veya belirli bir süreçte kazanılmış olan organizasyondaki yetenek seviyelerini karakterize eden *süreç yetenek boyutudur*.

3.3.1. Referans Modelin Kullanımı

Yazılım süreç iyileştirme ve yetenek belirleme referans modelin kimler tarafından niçin, ne zaman ve nasıl kullanılacağına dair bilgiler Tablo 3.2’de özetlenmiştir.

<i>Kim</i>	<i>Niçin</i>	<i>Nasıl</i>	<i>Ne zaman</i>
Değerlendirme Model Geliştiricileri	Referans model ile uyumlu modeller geliştirme Referans model ile geliştirilen modelin uyumluluğu	Model yapısı için bir referans model gibi Yetenek gösterimindeki bir kriterin set edilmesi gibi	Bir model geliştirilmesi esnasında Değerlendirmelerin yönetiminde kullanım önceliği ve izlenen gelişme olduğunda
Yazılım organizasyonu	Yazılım süreçlerinin geliştirilmesinin nasıl yapılacağını anlamada Müşterilerin gösterdiği süreçlerin yeteneklerini belirlemede Süreçleri ve değerlendiricinin değerlendirdiği yetenekleri anlamada	Yazılım süreçleri ve yetenek seviyelerinin gösterimi sırasında yönetime bir çalışma rehberi gibi Süreç yeteneği göz önüne alındığında en önemli süreç için bir referans rehberliği şeklinde Bir çalışma dokümanı gibi Karşılaştırma için esas geçerli bir izin vermede bir referans yapısı gibi Bir süreç veya süreç yetenek kategorisi gibi	Organizasyon yazılım süreçlerinin uygulanması esnasında Organizasyonun yazılım süreçleri ve sürekli iyileştirme aktivitelerinin geliştirilmesi veya yeniden gözden geçirilmesi esnasında Yukarıdaki gibi Pazarlama durumunda iç bir başlangıç anında Teklif isteklerine cevap verme esnasında Bir değerlendirme önceliğinde
Yazılım Süreç Değerlendiriciler	Bir organizasyonun yazılım süreçleri ve diğer sonuçlarını yönetilmesine karar vermede Bir değerlendirme modeline uyumluluk kurmada	Bir süreç veya süreç yetenek kategorisi gibi Model uyumluluğunda bir referans model gibi	Yazılım süreç değerlendirme esnasında ve öncelik durumlarında Bir değerlendirme programı veya değerlendirme için öncelik durumunda

Tablo 3.2 - Referans Modelin Kullanımı



3.3.2. Referans Modelin Yapısı

Referans model mimarisi iki boyutta tasarlanmıştır.

- *Süreç boyutu*, bir sürecin temel ölçülebilir nesnellikleri olan süreç amaçları nitelendirilmiştir.
- *Süreç yetenek boyutu* ise süreç geliştirmesinin uygulanması ve bir sürecin yönetiminde gerekli ölçümlendirilebilen karakteristikleri sunan her hangi bir süreçte uygulanabilen süreç niteliklerinin birisiyle karakterize edilmiştir.

Referans model süreçleri sahip oldukları aktivitelere göre süreç boyutunda beş kategoriye ayrılır.

- i) **Müşteri-Tedarikçi** süreç kategorisi, müşteriye doğrudan etki eden müşteriler için yazılımın değişimi ve geliştirilmesini destekleyen ve yazılımın doğru işletilmesi ve kullanımını sağlayan süreçlerden oluşur.
- ii) **Mühendislik** süreç kategorisi açıkça belirtilen, tamamlanan veya bir sistem ve yazılım ürünü ve bunların kullanıcı dokümanlarının tutulduğu süreçlerden oluşur.
- iii) **Destek** süreç kategorisi, yazılım yaşam çevriminin bir çok noktasında diğer süreçlerin (diğer desteklene süreçleri içeren) herhangi biriyle kullanılmış olabilen süreçlerden oluşur.
- iv) **Yönetim** süreç kategorisi bir yazılım yaşam çevriminin içerisinde projenin her bir işlem sırasının yönetilmesini her hangi birinin kullandığı genel doğal pratikleri içeren süreçlerden oluşur.
- v) **Organizasyon** süreç kategorisi, bir kuruluşun projesinde kullanıldığı zaman kuruluşun iş hedeflerini ve başarmada yardımcı olacak kaynaklar, ürün, süreç geliştirme ve organizasyonun iş ihtiyaçlarını yayımlayan süreçlerden oluşur.



Süreç kategorileri ve süreç aktivitelerinin bir çeşidiyle gruplandırma sağlanır. Referans modeldeki her bir süreç; amaçlanan durumlara göre tasvir edilir. Buradaki ifadeler uygun bir ortam anında bir sürecin tek bir fonksiyonel objektifliğinden oluşur.

Süreç amacında belirtilen öğeleri başarmış olmak için bunların nasıl veya ne yoluyla tanımlanması gerektiğini referans model vermez. Süreç amaçları, çalışan ürünlerin üretimini gerçekleştirmekteki pratikler, görevler ve bir çok düşük seviyeli aktiviteleri bir organizasyon sayesinde başarmış olabileceklerdir. Buradaki görevler, aktiviteler ile pratikler ve çalışan ürünlerin üretilmesinin karakteristikleri olan göstergelerin başarılan belirli bir ürün için gösterilip gösterilemeyeceği belirtilir.

Yavaş yavaş geliştirilen süreç yeteneği, yetenek seviyeleri şeklinde gruplandırılan süreç nitelik koşullarını ifade eder. Süreç nitelikleri bir süreç niteliğinin ölçümlenmesini sağlayan bir başarımlar cetvelinde değerlendirilebilen bir sürecin belirleyici nitelikleridir. Onlar tüm süreçler için uygulanabilir. Her bir süreç niteliği, organizasyonun iş amaçlarına katkıda bulunmayı ve onun hedefini başarımındaki yetenek yönetimini çeşitli yönleriyle tasvir eder.

Bir yetenek seviyesi bir sürecin başarımları için yeteneğin daha fazla artırılmasını sağlamakla birlikte iş nitelik veya niteliklerinin set edilmesiyle olur. Her bir seviye sürecin başarımlarının da yeteneğin daha fazla geliştirilmesini sağlar. Seviyelerde her bir sürecin yeteneğinin geliştirilmesiyle rasyonel bir ilerleme sağlanmış olur.

Dokuz süreç niteliklerinin birleştirilmesiyle referans modelde altı yetenek seviyesi vardır.

Seviye 0; Eksik: süreç amacına ulaşmasında genel bir başarısızlık vardır. Kolayca tanımlanabilen iş ürünleri süreç çıktıları yoktur.

Seviye 1; Varolan: Süreç amaçları genellikle başarılmıştır. Başarımlar sıkıca planlanmış ve izlenmiş olmayabilir. Organizasyonun bireyliği ile yerine getirilebilen bir



etkiyi onaylar ve bir gereksinim olduğu zamanki gibi yerine getirilebilir bu hareketlerin bir anlaşması vardır. Süreçler için tanımlanabilen iş ürünleri vardır ve bunlar hedeflenen amaçların başarılmasını gösterir.

Seviye 2; Yönetilen: Yönetilen: Süreç zaman ölçeğinde tanımlana kabul edilebilir kaliteli iş ürünlerini alıcıya teslim eder. Belirlenen prosedürlerdeki başarımlar uygunluğu planlanmış ve izlenmiştir. İş ürünleri belirli standartlar ve gereksinimlere uyar. Varolan seviyeden temel farkı süreç başarımlarıdır. Bunlar ise ilerleme yönünde tanımlanan, yönetilen ve planlanan süreç başarımlarıdır.

Seviye 3; Yerleşmiş: İyi bir yazılım mühendisliğinin temel ilkesi tanımlanan bir sürecin kullanım yönetimi ve başarılmasını sağlamaktır. Süreç kullanımı bireysel uygulamaları onaylar, standart sürümlere yeni biçimler verir ve süreçleri belgelendirir. Süreç tanımlamalarını kurmak için gerekli olan kaynaklar aynı zamanda yerindedir. yönetilen seviyeden en önemli farkı; standart bir sürecin kullanım yönetimi ve planlanması kurulmuş olan süreç seviyesidir.

Seviye 4; Kestirilebilir: Tanımlana süreç, amaçların başarılması için tanımlanan kontrol limitleriyle birlikte sürekli olarak pratiklerin yerine getirilmesidir. Performansın ayrıntılı ölçümlemesi analiz edilmiş ve birleştirilmiştir. Bu önderlik süreç yeteneğinin nicel anlaşılması için ve kestirilebilir performans için iyileştirme yeteneğidir. Başarımlar tarafsızca yönetilir. Çalışılan ürünlerin kalitesi nicel olarak tanınır. Yerleşmiş seviyeden en önemli farkı süreç tanımlamalarının nicel olarak anlaşılması ve kontrol edilmesidir.

Seviye 5; En iyileşen: Süreçlerin performansı geçerli ve gelecek iş ihtiyaçlarını karşılamayı en iyi şekilde kullanır ve süreç iş amaçlarının tanımlanmasında tekrarlanabilen süreç başarımlarıdır. Nicel süreç etkisizliği ve başarımı için iyi çalışan amaçlar organizasyonun iş amaçları esasında kurulmuştur. Buradaki amaçlara karşı süreç izlenimlerinin sürekliliği analiz sonuçlarıyla başarılı olan iyileştirmeler ve nicel geri beslemelerin elde edilmesiyle sağlanır. Süreç optimizasyonu yeni fikirlerin teknoloji planı ve nesnellik veya planlanan süreç için süreçlerin etkisindeki değişimi kapsar. Kestirilebilir süreçten en önemli farkı süreç sürekli zariflik ve gelişime uğrayan

standart süreçleri tanımlar. Temelinde ise buradaki süreçler için değişimlerin etkilerinin nicel olarak anlaşılmasıdır.

Referans model yalnız başına yeterli olmayan detay seviyeden bir birini tutan ve güvenilir süreç yetenek değerlendirmelerini yönetmedeki esas gibi kullanılışlı olmayabilir. Referans model süreç performans ve yetenek göstergelerinin kapsamlı olarak set edilmesiyle desteklenmiş olan bir ihtiyaçtır. Diğer bir yol ise bir birini tutan süreç yetenek notlarının mümkün olabilmesidir.

3.3.3. Süreç Boyutu

Bu bölüm yazılımın işletilmesi, tedariki, kazanımı, bakımı ve geliştirilmesiyle ilgili olan organizasyon tarafından normalde üzerine alınan süreçlerin bir sınıflandırılmasını sağlar. Sınıflandırma bir takım süreçleri içeren her bir süreç kategorisinin bir kaç tanesini tanıtır. Süreç kategorileri ve süreçler ISO 12207 (Yazılım yaşam çevrim süreçleri) de tanımlamayı kuvvetlice düzene sokmuştur.

Beş süreç kategorisi şunlardır:

CUS Müşteri – Satıcı (Tedarikçi) süreçleri

ENG Mühendislik süreçleri

SUP Destek süreçleri

MAN Yönetim süreçleri

ORG Organizasyon süreçleri

Her bir süreç kategorisinin tanımlanması süreç isimlerinin bir listesiyle takip edilmeyi kapsayan bir süreç karakteristiğini içerir.

Her bir süreç tanımlaması iki kısımdan oluşur:

- Süreç amaç ifadesi, süreç başarımlarının her taraftaki nesnelliğini yüksek seviyede tanımlama ve sürecin etkili olan sonuçlarının gösterimi gibi genel terimlerin tasvirinin genişletilmesinde, ve

- Süreç hakkında daha fazla bilgi ve ISO 12207'deki süreç tanımlanması ile bu referans modeldeki diğer süreçler için onların ilişkilerini sağlamaya daha çok önem vermede.

3.3.3.1 Müşteri-Satıcı Süreç Kategorisi (CUS)

Müşteri-Satıcı süreç kategorisi müşteriler için yazılımın çevrimi, genişletilmesi ve onların doğru kullanım ve işletilmelerinin sağlanması ile müşteriye doğrudan etki eden süreçlerden oluşur.

Müşteri-Satıcı süreçlerine ait olan süreçler aşağıdadır;

CUS.1 Yazılım edinme

CUS.2 Müşteri ihtiyaçlarını yönetme

CUS.3 Yazılım sağla (tedarik et)

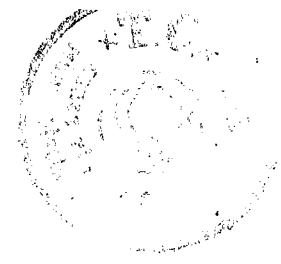
CUS.4 Yazılım işlet

CUS.5 Müşteri hizmeti sağla

CUS.1 Yazılım edinme

Yazılım edinme sürecinin amacı müşteriler tarafından ifade edilen ihtiyaçları karşılayacak servis ve/veya ürünün elde edilmesidir. Süreç kazanımı tedarikçiler tarafından çıkartılır. Süreç, bir müşterinin ihtiyacını belirlemesiyle başlar ve müşteriler tarafından gereksimlenen servis ve/veya ürünün kabulüyle son bulur. Sürecin başarılı bir şekilde yerine getirilmesiyle;

- Satıcı ve müşterinin her ikisinin beklentileri, sorumlulukları ve yükümlülüklerini açıkça ifade eden bir anlaşma geliştirmiş olacak,
- Bir ürün veya servis müşteri ihtiyacını belirleyen bir prosedüre sahip olacak,
- Kazanım belirli zorlamalarla (maliyet, liste ve kalite gibi) yönetilmiş olacaktır.



CUS.2 Müşteri ihtiyaçlarını yönetme

Müşteri ihtiyaçlarını yöneten sürecin amacı yazılımın işletilen yaşamı boyunca müşteri gereksinim ve ihtiyaçlarını başarmasının izlenmesi, işletilmesi ve yönetilmesi içindir. Yazılım iş ürün ve aktivitelerinde projelerin esasları gibi servis sağlayan bu temel hattı bir yazılım gereksinimlerini kurmada ve bu temel hat değişimlerinin yönetilmesidir. Sürecin başarılı bir şekilde yerine getirilmesiyle;

- Müşteriyle iletişimi açık ve devamlı olabilecek şekilde kurma,
- Tanımlanmış olan müşteri ihtiyaçlarını kabul etme ve belgelendirme (yönetim değişikliğiyle),
- Müşteri ihtiyaçlarını proje kullanımı için bir temel hat gibi kurmuş olma,
- Müşteri ihtiyaçlarının izlenmesinin devamı için bir mekanizma kurmak,
- Müşteri isteklerini düzenleme ve müşterilerin durumlarına kolayca karar vermelerini sağlamak için bir mekanizma kurmuş olmak.

CUS.3 Yazılım sağla (tedarik et)

Yazılım sağla sürecinin amacı müşterinin yerinde yazılımı paketleme, teslim etme ve kurmadır. Kaliteli bir yazılımın sağlanması, tanımlanan gereksinimlerin teslim edilmesiyle olur. Sürecin başarılı bir şekilde yerine getirilmesiyle;

- Karar verilen bir yazılımın paketlenmesi, teslim edilmesi ve yüklenmesi gereksinimleri,
- Yazılımı paketlemiş olmayla bir bakıma yazılımın verimli ve etkili bir şekilde yüklenmesi işletilmesini kolaylaştırır.
- Gereksinimleriyle tanımlanmış olan kaliteli bir yazılımı müşteriye teslim etme ve yüklenmesinde tanımlanan gereksinimlerle uygun bir şekilde başarılmış olma.



CUS.4 Yazılımı işlet

Yazılımı işle sürecinin amacı yazılımın ortama yüklenmesinde yazılımın kullanımı süresince yazılımın doğru ve etkili bir şekilde işletilmesini desteklemektir. Sürecin başarılı bir şekilde yerine getirilmesiyle;

- Yazılım takdiminde işletme tarafından tanımlanan işlevsel riskler ve bunların yönetilmesi sağlanır,
- Belgelendirilen prosedürde yazılım için tasarlanmış iş uyumluluğu yazılımdan istenilir,
- İşlevsel destekte kullanıcı sorgu ve isteklerini ele alma ve işlevsel problemlerin yeniden çözümlenmesiyle sağlanmış olunur,
- Güvence, kullanıcı ihtiyaçlarını karşılamak için yeterli olan yazılım ve ana sistem yeteneklerini sağlamış olur.

CUS.5 Müşteri hizmeti (servisi) sağlama

Müşteri hizmeti sağla sürecinin amacı yazılımın etkili kullanımını desteklemede müşteri için hizmetin kabul edilebilir seviyesinin kurulması ve bakımının yapılmasını sağlamaktır. Sürecin başarılı bir şekilde yerine getirilmesiyle;

- Müşteri destek ve servis ihtiyaçlarının devamlılığını sağlamış olmak,
- Müşteriye sağlanan servis destekleri ve ürünün kendisi hakkındaki müşteri memnuniyetini sürekli bir şekilde değerlendirmeye tabi tutacaktır.
- Müşteri ihtiyaçları uygun servislerin verilmesi sonrasında karşılanır.

3.3.3.2. Mühendislik Süreç Kategorisi

Mühendislik süreç kategorisi bir yazılım ürünü veya sistemin doğrudan belirli uygulanması veya bakımında onun kullanıcı dokümanlarından olan süreçlerden meydana gelir. Sistem durumunda mühendislik süreci yazılım gibi yalnız yapı ve bakım ile uğraşır.



Mühendislik süreç kategorisine ait olan süreçler:

ENG.1 Sistem isterlerini ve tasarımını geliştir.

ENG.2 Yazılım isterlerini geliştir

ENG.3 Yazılım tasarımını geliştir

ENG.4 Yazılım tasarımını uygula

ENG.5 Yazılım entegrasyonu ve testini yap

ENG.6 Sistem entegrasyonu ve testini yap

ENG.7 Sistemin ve yazılımın bakımını yap

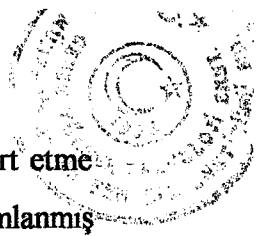
ENG.1 Sistem isterlerini ve tasarımını geliştir.

Sistem isterlerini ve tasarımını geliştir sürecin amacı sistem isterlerini ve mimarisini kurmak, sistem elemanları ve sonraki sürümleri için tahsis edilmiş olan sistem isterlerinin tanımlanmasıdır. Bu süreç kullanıcılar, operatörler, donanım, yazılım vb. sistemin çeşitli parçalarının sunulması bir grup insan tarafından başarılmış olabilecektir. Sürecin başarıyla uygulanması sonucu olarak;

- Müşterilerin belirttiği ve gösterdiği ihtiyaçları eşleştirme ile sistemin isterlerini geliştirmiş olur,
- Sistemin elemanlarının tanımlanmasında etkili bir çözüm sağlanmış olur,
- Esas elemanların her biri için tanımlanan gereksinimlere yer tahsis edilmiş olur,
- Öncelikli tanımlanan sistem isterlerinin yerine getirilmesi için uygun bir sürüm stratejisi geliştirilir,
- Gereksinimler, öneriler ve onlar arasındaki ilişkiler tüm etkilenen gruplar için bildirilmiş olur.

ENG.2 Yazılım isterlerini geliştir

Yazılım isterlerini gerçekleştirmede sürecin amacı sistem yazılım unsurları gereksinimini kurmaktır. Sürecin başarıyla uygulanması sonucu olarak;

- 
- Sistemin yazılım parçaları ve diğer arabirimleri için istelere yer ayırt etme müşterilerin belirttiği ve gösterdiği ihtiyaçları eşleştirmede tanımlanmış olacaktır,
 - Analizi yapılan, doğru ve test edilebilir yazılım isteleri geliştirilmiş olur,
 - İşletilen çevre aygıtlarının yazılım gereksinimlerine etkisi anlaşılmış olur,
 - Uygun bir yazılım sürüm stratejisi geliştirilmiş olur ki bu, yazılım gereksinimlerinin yerine getirilmesindeki önceliği tanımlar,
 - Yazılım isteleri gereklilikteki gibi beğenilmiş ve güncelleştirilmiş olur,
 - Yazılım isteleri tüm etkilenen gruplara bildirilmiş olur.

ENG.3 Yazılım tasarım geliştirme

Yazılım tasarım geliştirme sürecinin amacı isteleri yerine getiren ve isteleri yerine getirilmesini test edebilen yazılımın tasarımını tarif eder. Sürecin başarıyla uygulanması sonucu olarak;

- Yazılım istelerini yerine getiren ana yazılım parçalarının tasvirini yapan yapısal bir tasarım gerçekleştirilir,
- Her bir yazılım parçasının iç ve dış ara birimlerinin tanımlanması sağlanır,
- Yazılım birimlerinin inşası ve test edilebilirliğinin tasvir edilmesinde ayrıntılı bir tasarım gerçekleştirilmiş olur,
- İzlenebilirlik, yazılım isteleri ile yazılım tasarımı arasında kurulmuş olur.

ENG.4 Yazılım tasarımını uygulama

Yazılım tasarımını uygulama sürecinin amacı yerine getirilebilir yazılım birimleri üretmek ve bunların yazılım tasarımına yansıyan uygunluğunu araştırmaktır. Sürecin başarıyla uygulanması sonucu olarak;

- Yazılım istelerine karşı tüm yazılım birimleri için kriter doğruluğunu kanıtlamayı tanımlamış olur,
- Tüm yazılım üniteleri, tanımlanan tasarımla üretilmiş olur,



- Yazılım birimlerinin tasarlanmasına karşı doğruluğunu araştırma başarılmış olur.

ENG.5 Yazılım entegrasyon ve testi yapma

Yazılım entegrasyon ve test yap sürecinin amacı yazılım isterlerini karşılayacak olan diğer yazılım ürünlerinin her biriyle yazılım entegrasyonunu sağlamaktır. Bu süreç bireyler veya takımlar halinde adım adım başanılır. Sürecin başarıyla uygulanması sonucu olarak;

- Sürüm stratejisine uygun yazılım birimleri için bütünleşen bir strateji geliştirilmiş olur,
- Bir birim için ayrılan yazılım isterleriyle uyumu araştırma, kabul edilebilir kriterlerin bütünü için geliş olur,
- Tüm yazılım tanımlanan bütün kriterlerin kullanımını araştırmış olur,
- Test sonuçları kaydedilmiş olur,
- Entegre bir yazılımın yapılmış olan parçalarından birinin değişimi veya tanımının yeniden testi için bir vazgeçme stratejisi geliştirilmiş olur.

ENG.6 Sistem entegrasyonu ve testi

Sistem entegrasyon ve test sürecinin amacı, kullanıcıların beklediklerini açıkladıkları sistem isterlerini karşılayacak veya onları tatmin edecek komple bir yazılım ürünü ve işletim veya donanım el kitabı gibi diğer parçalarla birlikte yazılım parçasının entegrasyonunu sağlamaktır. Bu süreç, bir yazılım uzmanı içeren bir grup insan tarafından adım adım yönetilmelidir. Sürecin başarıyla uygulanması sonucu olarak;

- Sürüm stratejisine göre inşa edilen sistem birimlerinin tümü için bir entegrasyon planı geliştirilmiş olur,
- Bir birim için ayrılan sistem isteriyle ayrılan sistem isterleriyle uyumluluğu araştırma kabul edilebilir kriterin her bir bütünü için tanımlanmış olur.

- Bütün sistemler için tanımlanan tüm kriter uygulanıp kontrol edilmiş olur,
- Sistem isterleri (fonksiyonel, fonksiyonel olmayan, işletimi,ve bakımı) ile uyum entegre bir sistemi gösterme sağlanmış olur,
- Test sonuçları kaydedilmiş olur,
- Entegre edilen bir sistemin yapılmış olan parçalarından birinin değişimi veya tamamının yeniden testi için bir vazgeçme stratejisi geliştirilmiş olur.

ENG.7 Sistemin ve yazılımın bakımı

Sistem ve yazılım bakım sürecinin amacı kullanıcı isterlerine karşılık vermede sistem parçaları (bir ağ, işletim el kitabı, yazılım ve donanım gibi)'nın atıl hale gelmesi, bozulması, değişikliği ve yönetilmesi içindir. İsterlerin merkezi geliştirme veya adaptasyonunda bir problem veya ihtiyacı keşfetmiş olmalıdır. Nesne, kurumsal işlemlerin bütünlüğünü koruyorken var olan sistem ve/veya yazılım değiştirme ve/veya kullanımını kaldırmak içindir. Sürecin başarıyla uygulanması sonucu olarak;

- İşletmede var olan sistem arabirimleri için işletmelerin ve kuruluşların etkisi tanımlanmış olur,
- Şartnameler, tasarım belgeleri ve test planları güncelleştirilmiş olur,
- Uzlaşma sağlanmış olan sistem gereksinimlerinin gösterdiği test ve belgelerin birleştirilmesiyle değiştirilen sistem parçaları geliştirilir,
- Sistem ve yazılım güncellemeleri kullanıcının aygıtlarının değiştirilecek olmasını gerektirir,
- Yazılım ve sistem isteği kullanıcılar için en az karışıklığın olduğu kontrol edilen bir yolun kullanılmasını sağlar.

3.3.3.3. Destek Süreç Kategorisi (SUP)

Destek süreç kategorisi, yazılım yaşam çevriminin çeşitli noktalarında diğer süreçlerin (desteklenen diğer süreçleri içeren) her biriyle çalıştırılmış olan süreçlerden oluşur.



Destek süreç kategorisine ait olan süreçler:

SUP.1 Dokümantasyon geliştir

SUP.2 Konfigürasyon yönetimi yap

SUP.3 Kalite güvencesi sağla

SUP.4 İş ürünlerini doğrula

SUP.5 İş ürünleri tedarik et

SUP.6 Ortak gözden geçirmeler yap

SUP.7 Denetimler yap

SUP.8 Problemleri çözümle

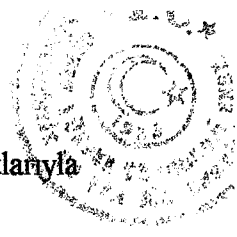
SUP.1 Dokümantasyon geliştir

Dokümantasyon geliştir sürecinin amacı bir süreç içindeki süreç veya aktivite tarafından üretilen kaydedilmiş bilgilerin geliştirilmesi veya bakımının yapılmasını sağlamaktır. Sürecin başarıyla uygulanması sonucu olarak;

- Süreç veya proje tarafından üretilmiş olan tüm dokümantasyonlar tanımlanmış olur,
- Tüm dokümanların içerik ve amacı belirtilmiş olur ve onların üretilmesi planlanmış ve programlanmış olur,
- Dokümanların geliştirilmesi için başvuru standartları tanımlanmış olur,
- Tüm dokümanlar görevlendirilen planlar ile uyumluluğu ve tanımlanan standartlarla uyumluluğunu yayınlamış ve geliştirmiş olur,
- Tüm dokümanlar belirlene kriterlerle uyumluluğu korumuş olur.

SUP.2 Konfigürasyon yönetimi yap

Konfigürasyon yönetimi yap sürecinin amacı bir süreç veya projenin tüm iş ürünlerinin bütünlüğünü korumak ve kurmaktır. Sürecin başarıyla uygulanması sonucu olarak;



- Süreç veya proje tarafından oluşturulan tüm uygun parçaların ana hatlarıyla tanımlanan kimliği saptanmış olur,
- Parçaların değişiklikleri ve sürümleri kontrol edilmiş olur,
- Parçaların durumu ve değişiklik istekleri kaydedilmiş ve raporlanmış olur,
- Bir birini tutan parçaların tanımlanması sağlanmış olur,
- Parçaların arşive kaldırılması, kullanımı ve teslim edilmesi sağlanmış olur.

SUP.3 Kalite güvencesi sağla

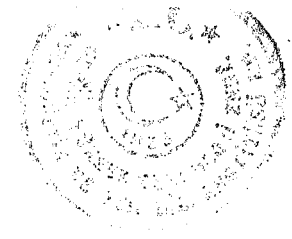
Kalite güvencesi sağla sürecinin amacı tüm kabul edilebilir standartlar, prosedürler ve isterlere razı olan bir süreç veya projenin aktiviteleri ve iş ürünlerini sağlamak içindir. Sürecin başarıyla uygulanması sonucu olarak;

- Süreç veya proje için kalite güvence aktiviteleri tanımlanmış, planlanmış ve programlanmış olur,
- Kalite güvence aktivitelerini sağlamada araçları, prosedürleri, metodolojileri ve kalite standartları tanımlanmış olur.
- Kalite güvence aktivitelerinin sağlanması için kaynaklar ve sorumluluklar tespit edilmiş olur,
- Kalite güvence aktivitelerinin sağlanmasındaki buradaki sorumluluk, süreç temininin sağlanması ve yönetimin görev bağımlılığını sağlamış olacaktır,
- Tanımlanan kalite güvence aktiviteleri uygun plan ve programlara mutabık uygulanmış olur.

SUP.4 İş ürünlerini doğrula

İş ürünlerini doğrula sürecinin amacı bir süreç veya projenin her bir iş ürününün yapısındaki isterlerin uygun bir biçimde gereği gibi yapılmasını teyit eder. Sürecin başarıyla uygulanması sonucu olarak;

- İstenilen bütün iş ürünleri için doğruluğu kanıtlanmış kriter tanımlanır,
- Zorunlu olan doğruluğu kanıtlama aktiviteleri sağlanmış olur,
- Tasarım geliştirmede sağladığı rahatlık gibi iş ürünleri içinde kanıt sağlar.



SUP.6 Ortak gözden geçirme yap

Ortak gözden geçirme yap sürecinin amacı müşteri tatminini sağlayan bir ürünün geliştirilmesini sağlamaya yardım eden ve kontratın nesnelliğine karşı geliştirmede müşteriyle ortak bir anlaşmayı sürdürmelidir. Sürecin başarıyla uygulanması sonucu olarak;

- Sürecin bir aktivitesinin ürünleri ve durumu müşteriler, tedarikçiler ve diğer çıkar sahipleri arasında aktivitelerin ortak gözden geçirilmesiyle değerlendirilmiş olur,
- Uygulamada ortak gözden geçirme planlanmış ve programlanmış olur,
- Yeniden gözden geçirmenin sonucu tüm hareket parçaları son olarak izlenmiştir.

SUP.7 Denetimler yap

Denetimler yap sürecinin amacı belirli isterlerle uyan kullanılan süreç ve ürünlerin bağımlılığını teyit etmektir. Sürecin başarıyla uygulanması sonucu olarak;

- Gereksinimlere uyan, münasip planlar ve anlaşmalar sağlar,
- Bağımlı uygun bir grup ile iş ürünleri veya süreç performansını denetleme yönetimini düzenler.

SUP.8 Problemleri çözümü

Problemleri çözümü sürecinin amacı bulunan bütün problemlerin analiz ve çözümlenmesini ve tüm yönlerinin tanımlanmasını sağlamaktır. Sürecin başarıyla uygulanması sonucu olarak;

- Zamanla bulunan tüm problemlerin analiz ve çözümlenmesini sağlamak için belgelendirme ve çözümleme sağlanacaktır.

- Problemlerin tanımlanmasındaki eğilim için etki ve seçimin sağlanmasında bir mekanizma sağlar.



3.3.3.4. Yönetim Süreç Kategorisi

Yönetim süreç kategorisi, bir yazılım yaşam çevrimi içerisinde proje veya sürecin her hangi bir aşamasının yönetimini yapmış biri tarafından kullanılmış olması onun genel doğasında bulunan pratikleri içeren süreçlerden oluşur.

Yönetim süreç kategorisine ait olan süreçler:

MAN.1 Projeyi yönet

MAN.2 Kaliteyi yönet

MAN.3 Risk yönetimi yap

MAN.4 Taşaronları yönet

MAN.1 Projeyi yönet

Projeyi yönet sürecinin amacı bir ürün yönetiminde gerekli olan kaynakları ve bir projenin yönetimi, koordinesi ve kurulumu için gerekli olan süreçler tanımlamaktır. Sürecin başarıyla uygulanması sonucu olarak;

- Projedeki çalışmanın kapsamı tanımlanmış olur,
- Tamamlanmış bir çalışma için gerekli görev ve kaynaklar boyutlandırılmış, tahmin edilmiş, planlanmış, izlenmiş ve ölçümlenmiş olacaktır,
- Organizasyon birimleri ve diğer projelerle birlikte proje elemanları arasındaki arabirimler tanımlanmış ve yönetilmiş olur,
- Proje hedefleri başarılmadığı zaman doğru yol gösterilmiş olacaktır.

MAN.2 Kaliteyi yönet

Kaliteyi yönet sürecinin amacı müşteri tatmininin sağlamak ve projenin ürünleri ve servislerinde kaliteyi yönetmek içindir. Proje ve organizasyon seviyesinin her



ikisindeki süreç ve ürünün kalitesini yönetmede bu süreç bir odak kurmayı amaçlar
Sürecin başarıyla uygulanması sonucu olarak;

- Müşterinin kalite isterlerinin temelindeki kalite hedefleri projenin yazılım yaşam çevrimi içerisinde çeşitli kontrol noktalarıyla kurulmuş olur,
- Proje aktiviteleri sonuçların ölçülmesindeki metrikler, kalite amaçlarını başarıp başarmayacağını değerlendirmede projenin yaşam çevrimi içerisindeki kontrol noktalarının kurulmasını ve tanımlanmasını sağlar,
- Yazılım mühendisliğindeki iyi pratikler, kullanılan yazılım yaşam çevrim modelleri içinde sistematik olarak tanımlanmış ve entegre edilmiş olacaktır,
- Tanımlanan kalite aktiviteleri uygulanmış olacak ve onların teyit edilen uygulaması sağlanmış olur,
- Kalite hedefleri başarılmadığı zaman doru yol verilmiş olur.

MAN.3 Risk yönetimi yap

Risk yönetimi yap sürecinin amacı bir projenin yaşam çevrimi boyunca proje risklerini hafifletmek ve enine boyuna tanımlanmasını yapmaktır. Proje ve organizasyon seviyesinin her ikisindeki risklerin yönetilmesinde bir merkez kurulması süreç tarafından sağlanır. Sürecin başarıyla uygulanması sonucu olarak;

- Projenin uygulanmasında risk yönetimi kapsamı tespit edilmiş olur,
- Projedeki riskler, geliştirmede olduğu gibi tanımlanmıştır,
- Riskler analiz edilmiş ve burada tespit edilen risklerin yönetimi için başvuru kaynaklarındaki öncelikler belirlenmiş olur,
- Beklenen ilerleme sağlandığı zaman doru yol belirtilmiş olur.

MAN.4 Taşeronları yönet

Taşeronları yönet sürecinin amacı kaliteli yazılım tedarik eden taşeronları ve onların performansının seçimini yönetmektir. Sürecin başarıyla uygulanması sonucu olarak;



- Taşeron altında uygulanan bir çalışma raporu yayınlanmış olur,
- Potansiyel taşeronlar yazılım fonksiyonlarının yerine getirilmesinde onların yeteneklerinin bir değerlendirmesi süresince kaliteli olacaktır.
- Kaliteli taşeronlar, tanımlanan kontratın (anlaşmanın) parçalarını sağlamak için seçilmiş olacaktır,
- Standartlar ve prosedürlerin üzerinde anlaşılmışlığa karşı taşeronları uyup uymadıkları değerlendirilmiş olacaktır,
- Taşeronlara teslim edilen ürün ve servislerin kalitesi değerlendirilmiş olacaktır.

3.3.3.5 Organizasyon Süreç Kategorisi

Organizasyon süreç kategorisi kuruluşun iş hedeflerine ulaşmasına yardım edecek, organizasyon süreçleriyle kullanıldığında kaynak değerlendirmesi, üretimi, süreç geliştirilmesi ve organizasyon iş amacının kurulmasını sağlayan süreçlerden oluşur. Bununla birlikte organizasyon işlerinin geneli yazılım süreçlerinden oluşan bir kaç kardeş amaca sahiptir. Bu yazılım süreçleri etkili bir şekilde olan kuruluş ortam gereksinimi ve iş içeriğini yerine getirmiştir.

Buradaki organizasyon süreçleri;

- Gerekli organizasyon alt yapısının inşası,
- Her hangi bir organizasyon bölümünde en iyi ilerlemeleri sağlamak,
- Tüm geçerlilikleri yerine getirmeyi sağlar.

Organizasyon süreç kategorisine ait olan süreçler:

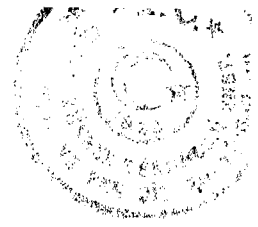
ORG.1 İşin mühendisliğini yap

ORG.2 Süreci tanımla

ORG.3 Süreci iyileştir

ORG.4 Kaliteli iş gücü temin et

ORG.5 Yazılım mühendisliği alt yapısını sağla



ORG.1 İşin mühendisliğini yap

İş mühendisliği sürecinin amacı organizasyon ve projelerin bireyselliği ile onlara fonksiyon etkinliği sağlamada yetki veren bir kültür ve vizyon sağlar.

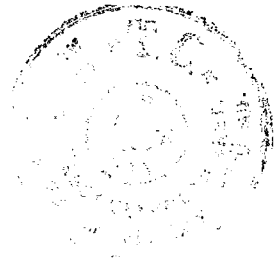
Bununla birlikte yeniden iş mühendisliği ile toplam kalite yönetiminin yazılım süreçlerinde çok daha geniş bir kapsama sahip olmasına rağmen yazılım süreci geliştirme ticari bir bağlamda ortaya çıkar ve şayet başarılı olacaksa ticari hedeflere yönelmelidir. Sürecin başarıyla uygulanması sonucu olarak;

- İşler için bir vizyon, misyon, nesneler ve hedefler tanımlanmış olur ve tüm müşterilere bilgi üretilir,
- Her bir birey, kendi iş tanımlamasını sağlamak için çağrılır ve işin vizyonunun gerçekleşmesi için etkilice yazı verilmesi sağlanmış olur.

ORG.2 Süreci tanımla

Süreci tanımla sürecinin amacı bu referans modeldeki tüm modelleri kapsayan yönetim süreçleri ve yazılım mühendisliğinin uygulamasını, kararlı ve tekrarlanabilir şekilde destekleyecek süreç tanımlamaları (standartlar, prosedürler ve modelleri içeren)nın yeniden kullanılabilir bir kütüphanesinin oluşturulması içindir. Sürecin başarıyla uygulanması sonucu olarak;

- İyi tanımlanan süreçlerin bakımı yapılmış ve standart seti her bir sürecin uygulanabilirlik göstergeleriyle beraber var olacaktır,
- Standart sürecin yardımcı iş öğeleri, aktiviteleri ve ayrıntılı görevleri beklenen performans karakteristikleriyle birlikte tanımlanacaktır,
- Her bir proje için yerleştirilmiş özel süreçler olacaktır ki bu projenin ihtiyaçlarıyla uyumlu, standart süreçle yeni bir biçim verilmiştir.
- Belirli projelerin standart süreçlerinde kullanılmak için veriyle ilişkili ve bir bilgi kütüphanesi olacak ve bakımı sağlanacaktır.



ORG.3 Süreç iyileştir

Süreç iyileştir sürecinin amacı iş ihtiyacıyla mutabık organizasyon tarafından kullanılan süreçlerin yeterliliğini ve etkinliğini sürekli olarak gerçekleştirmektir. Sürecin başarıyla uygulanması sonucu olarak;

- Organizasyon standart yazılım süreçlerine bağlı yeterlilik veya yetersizlikleri anlaşılmış olacaktır,
- Standart ve tanımlanan süreçlerin değişimleri önceden kestirilebilen sonuçlarıyla bir çeşit kontrol edilmişlik sağlanır,
- Planlanan ve kontrol edilen yazılım süreç geliştirme aktiviteleri kontrol edilmiş bir tarzda sağlanmış olur.

ORG.4 Kaliteli iş gücü temin et (İnsan kaynakları sağla)

Kaliteli iş gücü temin et sürecinin amacı, seçilen bir grupta birlikte çalışan ve onların rollerindeki yetenekliliğini sağlamak için süreç bilgi ve becerilerine sahip olan bireyler ile projeler ve organizasyonlar tedarik edilir. Sürecin başarıyla uygulanması sonucu olarak;

- Organizasyon ve projelerin işletilmesinde gerekli rol ve beceriler tanımlanmış olacaktır,
- Organizasyon içerisindeki geçişler, seçimler ve resmi prosedür atamaları yapılır,
- Değerlendirmelerin yerine getirilmesinde ihtiyaç duyulan becerilere sahip tüm bireyleri sağlamada eğitim planlanmış ve yönetilmiştir,
- İhtiyaç duyulan beceri ve yetenekteki bireyler tanımlanmış ve proje rolleri ile organizasyonu sağlamada uygun eğitim sağlanır,
- Bireyler ve gruplar arasındaki etkili etkileşim sağlanmış olur,
- İş amacı aktivitelerinin etkinliğini koordine etmede ve bilgi paylaşımı yeteneklerine sahiptir,



- Geri beslenen performansı sağlamak ve performans devamlılığı için ölçümlendirilmiş olabilen grup ve birey performanslarına karşı nesnel kriterler tanımlanır.

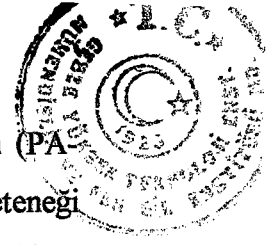
ORG.5 Yazılım mühendisliği alt yapısı sağla

Yazılım mühendisliği alt yapısını sağla sürecinin amacı tanımlanan süreçleri destekleme ile birbirlerini tutan organizasyon projelerinde kullanımı için yazılım geliştirme metotları ve araçlarından oluşan entegre bir set ile kararlı ve rahat bir ortam sağlayacaktır. Sürecin başarıyla uygulanması sonucu olarak;

- İyi tanımlanmış, kurulmuş ve bakımı yapılmış bir yazılım mühendisliği ortamı; proje veri bütünlüğü, geçerliliği ve güvenilirliği ile temin edilen organizasyonu içeren yeniden kullanma, süreç standardını sağlama ile uygun bir şekilde var olacaktır,
- Yazılım mühendisliği ortamı, proje gereksinimleri ve proje takımı ile birlikte tüm proje aktivitelerinin nereye uygulandığında verimli bir şekilde kullanılabileceği biçimlendirilir,
- Tekrar kullanım için tanımlanmış ve yayılmış bir strateji belirtilir.

3.3.4. Yetenek Boyutu

Herhangi bir sürecin yetenek notlandırmasının nasıl olacağını tanımlanır. Süreç yeteneği altı noktalı sıralı skalada tanımlanmıştır. Bu skala ile yetenek değerlendirmesi yapmak mümkündür. Skala performans için hedefleri göstermede tamamen doldurulma yeteneğine sahip olmayan performansından yerine getirilmiş süreçleri artırılan yeteneğini belirtir. Ek olarak söz konusu performans tatmin edici bir şekilde sürekli süreç gelişimini ve hedeflerini buluşturma yeteneğine sahiptir. Skala her bir bireysel süreç için artırımlardan iyi tanımlanan yolu böylece belirtir.



Yetenek modeli içerisinde yeteneğin ölçümü dokuz süreç niteliklerinin (PA Process Attribute) bir kurulumu esas alınmıştır. Süreç nitelikleri verilen bir yeteneği bulan bir süreci tanımlamakta kullanılır. Her bir nitelik süreç yeteneklerinin özellikle bir yönünü ölçer. Nitelikler dört noktalı sıralı bir skala üzerinde ölçümlenir ve böylece süreç yeteneklerinin özel bir yönüne doğru daha ayrıntılı bir bakış atmamızı sağlar. Süreç yeteneği, yetenek tanımlamaları ve süreç gelişimlerini desteklemeyi gerektirir.

3.3.4.1. Seviye 0: Tamamlanmamış (Eksik) Süreç

Süreç uygulanmamış veya amaçlarını başaramamıştır. Bu seviyede nitelik yoktur.

3.3.4.2. Seviye 1: Uygulanan (Varolan) Süreç

Uygulanmış süreç tamamlanan amaçlarını başarır. Sürecin aşağıdaki nitelikleri bu seviyenin başarısını gösterir.

PA 1.1 Süreç Performans niteliği

Sürecin uygulama uzunluğu pratiklerin bir takımını kullanır. Pratiklerin bir seti, kullanılan tanımlanabilir girdileri izler ve başlatır. Tanımlanabilir iş girdileri, tanımlanabilir çıkışları üretmede kullanılır. Tanımlanabilir çıkış iş üretimi sürecin amaçlarını yerine getirmede tatmin için yeterlidir.

3.3.4.3. Seviye 2: Yönetilen Süreç

Performe edilen süreç, tanımlanan zaman skalaları ve yeniden kaynaklandırılmış ihtiyaçların içerisinde kabul edilebilir kalitede iş üretimini dağıtır.

Süreci aşağıdaki nitelikleri bu seviyenin başarısını gösterir:

PA 2.1 Performans yönetim niteliği



Sürecin icra edilme uzunluğu belirlenen zaman ve kaynak gereksinimleri içerisinde iş ürünlerini üretmek için yönetilir.

PA 2.2 İş ürünü yönetim niteliği

Sürecin icra edilme uzunluğu iş ürünlerini üretmek için kullanılır. Söz konusu iş ürünleri dokümanite ve kontrol edilerek fonksiyonel ve fonksiyonel olmayan ihtiyaçları karşılar. Sürecin amacı kaliteli iş ürünleri üretmektir.

3.3.4.4. Seviye 3: Kurulmuş (Yerleşmiş) Süreç

Yönetilen süreç tanımlanan bir sürecin kullanımını sağlar, öyle ki tanımlanan süreç iyi yazılım mühendislik prensiplerini esas alır.

Sürecin aşağıdaki nitelikleri bu seviyenin başarısını gösterir:

PA 3.1 Süreç tanım niteliği

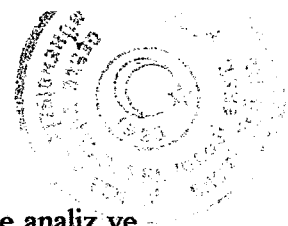
Sürecin uygulama uzunluğu standart bir süreci temel alan süreç tanımını esas alır. Standart süreç organizasyonun tanımlanan iş özelliklerine katkıda bulunması için süreci mümkün kılar.

PA 3.2 Süreç yeniden kaynaklandırma niteliği

Sürecin uygulanması sırasında uygun kabiliyetli insan kaynakları kullanılır. Böylece organizasyonun tanımlanan iş özelliklerine katkıda bulunulur.

3.3.4.5. Seviye 4: Kestirilebilen (Tahmin edilebilen) Süreç

Kurulmuş süreç amaçlarını başarılması tanımlanan kontrol limitleri içinde sağlanır. Sürecin aşağıdaki nitelikleri bu seviyenin başarısını gösterir.



PA 4.1 Süreç ölçüm niteliği

Sürecin icra edilme özelliğini kontrol ve doğrulama için ölçümlenme analiz ve kontrollerini sağlar. Gerekli her yerde tanımlana süreç hedeflerini güvenilir bir biçimde süreç performansının ölçümlenmesini sağlar.

3.3.4.6. Seviye 5: En iyileşen (Uygun hale getirilmiş) Süreç

Tahmin edilebilir süreç gelecek iş ihtiyaçlarını ve tanımlanan iş özelliklerini buluşturmada ki başarıları onun kendi performansını artırmasını optimize eder. Sürecin aşağıdaki nitelikleri bu seviyenin başarısını gösterir.

PA 5.1 Süreç değişim niteliği

Tanımlamayı değiştiren uzunluk sürecin performans ve yeteneğini organizasyonun iş amaçlarını başarmada daha iyi kontrol eder.

PA 5.2 sürekli iyileştirme niteliği

Süreci değiştiren uzunluk organizasyonun tanımlanan iş amaçlarının yerine getirilmesinde sürekli bir geliştirmeyi tanımlar ve gösterir.

3.3.4.7. Süreç Nitelikleri Notlandırılması

3.3.4.7.1. Süreç Nitelik Notlandırma Skalası

Bir süreç niteliği yukarıda tanımlandığı gibi her hangi bir sürecin ölçümlenebilir karakteristiğini gösterir. Notlandırılan skala aşağıda tanımlanmıştır ve süreç niteliklerinin tanımlanan yeteneklerinin başarıları seviyelerini tanımlamada kullanılmalıdır.

N	Başarılmamış	Tanımlanan niteliğin başarısının kanıtı yoktur.
P	Kısmen başarılı	Tanımlanan niteliğin bazı başarıları vardır.
L	Büyük oranda başarılı	Tanımlanan niteliğin önemli ölçüde başarı vardır.
F	Tamamen başarılı	Tanımlanan nitelik tamamen başarılmıştır.

3.3.4.7.2 Süreç Nitelik Notlandırması

Örnek değerlendirilmiş olan her bir süreç için, her süreç niteliği, değerlendirme konusunda kapsanmış olan en yüksek yetenek seviyesini kapsar. Bu durum 6.7.1’de tanımlanan bir notlama nitelik skalasına uygun olmalıdır.

3.3.4.7.3 . Başvurulan Notlandırmalar

Bu Teknik Rapordaki referans modele uygun bir nitelik notlandırmasını anlatmak için, süreç örneklerini tanımlayan nitelik notlandırmalı her bir süreç için tek bir referans model üretmek mümkündür. Süreç kategorisi içerisinde ki süreç yetenek seviyesi içerisindeki süreç yeteneğini de kapsar.

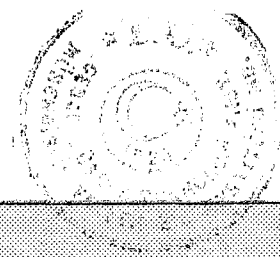
3.3.4.7.4. Toplam Notlandırma

Bir süreç örneği bir çok yönden değerlendirilir. Örneklerin süreç nitelik notlamalarının sıklık dağılımını göstermek için toplanılmış olmalıdır. Bir yetenek seviyesi içerisindeki süreç nitelik notlamaları toplanılmamalıdır. Notlar toplandığı zaman ise yeniden sunma; süreç tanımlamalarını, dağılım içerisindeki notlamaların düzenlenmesi ve örneklerin sayısı toplam içerisinde tanımlanmış ve kapsanmıştır.

3.3.4.8. Süreç Yetenek Seviye Modeli

3.3.4.8.1 Süreç Yetenek Seviyelerinin Başarısı

Burada bir süreç örneği tarafından yetenek seviye başarısı tanımlamak veya rapor edilmek istenir. Süreç yeteneği tablo 3.3’deki süreç yetenek seviye modeline göre bir süreç örneği için nitelik notlandırmaları dağıtılmalıdır.

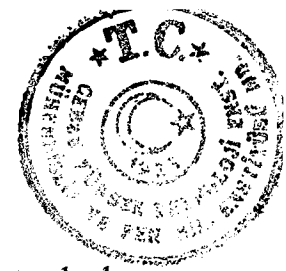


Skala	Süreç Niteliği	Notlama
Seviye 1	Süreci Yerine Getirebilme	Büyük oranda veya Tam (<i>L veya F</i>)
Seviye 2	Süreci Yerine Getirebilme Başarım Yönetimi İş Ürün Yönetimi	Tam (<i>F</i>) Büyük oranda veya Tam (<i>L veya F</i>) Büyük oranda veya Tam (<i>L veya F</i>)
Seviye 3	Süreci Yerine Getirebilme Başarım Yönetimi İş Ürün Yönetimi Süreç Tanımlama ve Biçimlendirme Süreç Kaynak	Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Büyük oranda veya Tam (<i>L veya F</i>) Büyük oranda veya Tam (<i>L veya F</i>)
Seviye 4	Süreci Yerine Getirebilme Başarım Yönetimi İş Ürün Yönetimi Süreç Tanımlama ve Biçimlendirme Süreç Kaynak Süreç Ölçümleme Süreç Kontrol	Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Büyük oranda veya Tam (<i>L veya F</i>) Büyük oranda veya Tam (<i>L veya F</i>)
Seviye 5	Süreci Yerine Getirebilme Başarım Yönetimi İş Ürün Yönetimi Süreç Tanımlama ve Biçimlendirme Süreç Kaynak Süreç Ölçümleme Süreç Kontrol Süreç Değişim Sürekli İyileştirme	Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Tam (<i>F</i>) Büyük oranda veya Tam (<i>L veya F</i>) Büyük oranda veya Tam (<i>L veya F</i>)

Tablo 3.3 – Yetenek seviye notlandırması

3.3.4.8.2 Yetenek seviyeleri toplamı

Bir süreç örneğinden daha fazlasının değerlendirildiği yerlerde bir süreç örneği tarafından süreç yetenek seviyeleri başarılmış süreç yetenek seviyelerinin sıklık dağılımını göstermek için toplanmış olabilir. Yetenek seviyeleri toplanıldığı zaman yeniden sunum sürecini tanımlamada kullanılır ve yetenek seviyeleri dağılımındaki notların düzenlenmesini, örneklerin sayısını ve örnekteki gibi toplamı içerir.



3.3.5. Bölüm Değerlendirme Sonuçları

Yazılım süreç iyileştirme ve değerlendirme için bir uluslararası standardın gerekliliği konusunda, ISO/IEC JTC1/SC7 tarafından 1991 yılında alınan karar doğrultusunda 1993 yılında SPICE projesi başlatılmıştır. ISO içinde hızlı yoldan standart olmasına karar verilen SPICE (ISO15504 taslak standardı) halen 'kullanıcı denemeleri' aşamasındadır.

SPICE Modeli iyi bir yazılım mühendisliği ve yazılım süreç değerlendirme amacı için geliştirilmiş olan veya geliştirilecek süreç yönetim prensiplerini temel almaktadır. Referans model ile süreç değerlendirmesi için modellerin uyumunda bir yol gösterir. Değerlendirmelerden elde edilen çıktıların mümkün olan karşılaştırılabilir uyumluluk gereksinimleri için farklı model ve metotlar kullanır.

SPICE iki boyutlu bir model olup birinci boyutunda yazılım süreçleri yer alır, ikinci boyutunda ise yetenek düzeyleri bulunur. Süreçler belli amaçlara yönelik aktivite ve en iyi pratikleri içerir. SPICE süreç modelinde her sürecin amacı, ve bu amaca ulaşmak için gerekli temel pratikler (base practices) tanımlanır. SPICE versiyon 2'de beş kategoride gruplanmış 29 süreç tanımlanmıştır. Bir kalite modelinin SPICE uyumlu olabilmesi için model en az bir sürece uygulanabilmeli, sıfırdan başlayarak en az 2 yetenek düzeyini içermeli yani 0-1-2 yetenek düzeylerini içermesi zorunlu olup 3 ve 4'üncü seviyeler ise isteğe bağlı olarak sağlanmalıdır, zorunlu model ile bire bir izdüşümü olmalıdır.



4. YAZILIM SÜREÇ / KALİTE MODELLERİ KARŞILAŞTIRMASI

1980’li yıllardan bu günlere kadar yapılan çalışmalarda yazılımda kalite sertifikasyonu, süreç iyileştirme ve yetenek belirlemede çeşitli modeller ortaya konulup uygulanmaktadır. Çalışmalarım sırasında incelemeye tabi tuttuğum önemli üç model; TickIT (ISO 9001), CMM ve SPICE modelleri arasında kavram düzeyinde temel ilişkiler bulunsa da, başta yapı ve amaçları olmak üzere bu model ve standartlar arasında bire bir uyum ve iz düşüm olduğunu söylemek pek mümkün değildir. Temelde yazılım üreticileri ve müşteriler arasında tam bir görüş birliği bulunmamaktadır.

Yapılan çalışmalarda elde edilen bilgiler ışığında model ve standartları bir karşılaştırmaya tabi tuttuğumuzda ISO 9001’in yazılım endüstrisine uygulanmış şekli olan TickIT modeli, Amerika Birleşik Devletleri Savunma Bakanlığının destekleri ile geliştirilen ve Yetenek Olgunluk Modeli olan CMM ile Süreç İyileştirme ve Yetenek Belirleme modeli olan SPICE modelleri arasında özellikle çıkış hedefleri bakımından bazı farklılıklar bulunmaktadır.

TickIT (ISO 9001) yazılım üreten bir kuruluşun dışa yönelik özellikle müşterilere karşı kalite güvencesi verme amacı olan bir sertifikasyon sistemidir. Firmanın kendisine yönelik (içe) bir kalite/süreç iyileştirme modeli değildir. Buna karşılık SPICE modeli hem süreç iyileştirmeleri hem de yetenek düzeylerinin belirlenmesi olmak üzere iki ana amaç üzerine geliştirilmiştir. Burada süreçlerin iyileştirilmesi içe yönelik, yetenek düzeylerinin belirlenmesiyle de hem içe hem de dışa yönelik bir çalışma yapılmış olmaktadır. Yazılım üreten firma kendi kalitesini geliştirmek, süreçlerini kontrol altına alıp iyileştirmelerini sağlamak istiyorsa ve dışarıya karşı bir kanıtlama yapmayı *doğrudan amaçlamıyorsa* TickIT modeli yerine SPICE modelini seçmesi daha uygun olacaktır. CMM modeli ise TickIT ve SPICE modelleri arasında bir yer tutar. CMM’i geliştirenlerin ana amacı dışarıya karşı yetenek



düzeylerinin belirlenmesi olmakla birlikte model, içerisinde kalite/süreç iyileştirmeleri için bir yol göstermektedir.

Bir kalite sisteminde olması istenen minimum gereksinimleri TickIT modeli belirler. CMM ve SPICE modeli ise sürekli süreç iyileştirmelerini ön plana çıkartmaktadır. Bazı görüşlere göre aslında TickIt modeli de, bazı maddeleriyle süreç iyileştirme faaliyetleri yapar. Örneğin, ISO 9001'in düzeltici faaliyetleri (TS-ISO 9001 madde 4.14.1)'ni sürekli süreç iyileştirmeleri olarak yorumlamak mümkündür.

TickIT değerlendirmesinde çalışmaların standartlara uyup uymadığı dış bir denetleyici tarafından denetlenir. Denetleme sonucunda kabul yada red olmak üzere iki durum söz konusudur. CMM ve SPICE modellerinin temelinde ise bir değerlendirme söz konusudur. CMM'de sonuçlar beş değişik değer alabilen bir sonuca sahipken SPICE modelinde değerlendirme sonucu firmaya iki boyutlu bir profil verir. SPICE modeli ile her süreç ayrı ayrı değerlendirilip süreç iyileştirme yolları irdelenir. Esası sertifikasyon almaya dönük olan TickIT modelinde süreçlerin iyileştirilmesi amacıyla SPICE modeli rehber olarak kullanılabilir.

TickIT değerlendirmesinin sonucu yazılım üreten firma veya birimin belli bir standardın altında yada üstünde olduğunu bildirir. Ayrıntılı bir bilgi vermez. Bu sınırın neresinde olduğumuz konusuna açıklık getirmez. Sadece bir noktadaki uyum eksikliğinden dolayı sertifika alamamak ile hiçbir kalite veya süreç kavramına bile sahip olmamak eşdeğer gibidir. CMM kalite boyutunu beş düzeye çıkararak daha ayrıntılı bilgi vermekte, SPICE modeli ise iki boyutlu ve toplam 29 sürece sahip yapısıyla, her bir sürecin kendisi içerisinde değerlendirilmesiyle diğer modellere göre daha çok ayrıntılı bilgi verir.

SPICE modelini diğer modellerden ayıran en önemli özelliklerden biri SPICE'da değerlendirmelerin amaca ve ortama göre yapılabilmesidir. Değerlendirme tabi tutulacak süreçler amaca göre seçildiğinde sadece ilgili süreçler değerlendirilir. Örneğin tek müşterisi olan bir firma için Müşteri-Satıcı süreçleri önemli sayılabilir. Başka bir firma için ise Mühendislik süreçleri, Yönetim süreçleri veya Organizasyon süreçleri önemli görülüp değerlendirmeye tabi tutulabilir. SPICE modelinin bu özelliği

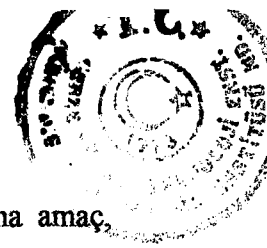
SPICE deęerlendirmesinden geirilen iki ayrı firmanın deęerlendirmeye alınan süreçleri aynı olmadığı sürece almış oldukları notlar bire bir karşılaştırılamaz. Ayrıca SPICE deęerlendirme sonuçlarını yorumlayabilmek için model hakkında ayrıntılı bilgilere sahip olmak gerekir, yorumlama dięer modeller kadar kolay olmaz.

CMM ve SPICE modellerinde firma için her zaman iyileştirme yolları açıkken TickIT modelinde sertifikasyon alan bir firmanın iyileştirmeler için gidebileceęi daha üst bir kalite düzeyi bulunmamaktadır.

Firmalar büyüklüklerine göre düşündüğümüzde tamamını bir bütün olarak deęerlendiren TickIT modeli ve beş yetenek düzeyi şeklinde deęerlendiren, deęerlendirmeyi daha küçük parçalara ayırarak yapmış olan CMM modeli daha çok büyük firmalara hitap eder. Küçük aplı firmalara uygulandığında deęerlendirme çok pahalı ve anlamsız olabilir. SPICE modeli ise esnek yapısı ve deęerlendirmelerin amaç ve ortama göre yapılmasında dolayı her boydaki firmalar için uydun bir deęerlendirme modeli olabilir.

TickIT (ISO 9001) modelinde sertifika alabilmek için gerekli olan tüm koşulların yerine getirilmesi şartı firmalara istemedięi ek maliyetleri beraberinde getirir. SPICE modelinde ise iç ve dış dönük bir durum deęerlendirmesi yapıldığından ve yine istenilen süreçlerin ele alınıp deęerlendirilmesinden dolayı ek bir maliyet gerektirmez. CMM'e bu açıdan bakıldığında, modelde bir üst seviyeye çıkabilmek için o seviyenin tüm şartlarını yerine getirme zorunluluğunda dolayı TickIT modeline benzetebiliriz.

TickIT deęerlendirmesi dış uzmanlar tarafından firmaların standart şartlarını sağlamadaki eksikliklerini ve hatalarını arayan, kanıt isteyen, savunmacı, negatif tutum sergileyen bir tavır içinde denetleme yapılırken SPICE modelinde; deęerlendirme iç ve dış uzmanlar tarafından gerçek arayan, mevcut durumu olumlu ve olumsuz yönlerini açığa çıkarmak için pozitif ve işbirlikçi tutum sergileyen, iyileştirmeler için yol gösterici sonuçları veren bir deęerlendirmeye sahiptir. CMM deęerlendirmesinde de denetleme içi ve dış uzmanlar tarafından yapılır. Dış uzmanlar her zaman denetleme sertifikasına sahip deęerlendiricilerden olmak zorundadır.



TickIT (ISO 9001), CMM ve SPICE modelleri arasındaki yapı, ana amaç, belgeler, değerlendirme ve iyileştirme amaçları bakımından öz karşılaştırmalarla birlikte Tablo 4.1.'de verilmiştir.

	TickIT (ISO 9001)	CMM	SPICE
Yapı	- İki Düzey	- İki Düzey	-İki boyut çok düzey
Ana Amaç	- Sertifikasyon (<i>Dışarıya kurum bildirme</i>) (<i>Dış</i>)	- Yetenek belirleme ve Sertifikasyon (<i>İç ve Dış</i>)	- İyileştirme ve Yetenek belirleme (<i>İç ve Dış</i>)
Belgeler	- Kısa - Soyut - ISO 9000 genel kökenli	- Orta - Somut bir şekilde - Yazılıma özgü ABD bağımsız gelişme	- Uzun - Somut ve ayrıntılı - Yazılıma özgü ISO üst modeli(15504)
Değerlendirme	- Denetleme - Soyut kriterler - Dış uzmanlar - Kısa ve öz - Hata arayan - Kanıt isteyen - Negatif tutum - Karşı güçler - Savunmacı	- Değerlendirme - Somut kriter - İç ve dış uzmanla - Uzun ve ayrıntılı	- Değerlendirme - Somut kriter - İç ve dış uzman - Uzun ayrıntılı - Gerçek arayan - Gösterge isteyen - Pozitif tutum - İşbirlikçi - Saydam
İyileştirme Amacı	- Çok sınırlı - Tüm organizasyon için tek bir not verilir	- İyileşme yolu var - Tüm organizasyon için tek değerlendirme yapılır	- Esnek iyileşme yolları var - Amaca göre iyileşme - Her süreç için - Her proje için - Her grup için ayrı değerlendirmeler yapılır

Tablo 4.1 - Değerlendirme Model Karşılaştırmaları

Modeller arasında bir izdüşüm alındığında TickIT (ISO 9001) modeli bir tarafta CMM ve SPICE modelleri diğer tarafta kalır. SPICE modeli CMM modeli için bir üst çerçeve oluşturur. CMM'in gelecek versiyonlarının SPICE uyumlu çıkartılması



beklenmektedir. Uyumlu hale geldiğinde CMM değerlendirmesinden SPICE izdüşümü elde edilebilecek ve SPICE değerlendirme sonucundan CMM sonucunu çıkartabilmek mümkün olabilecektir.

SPICE modeli, CMM modelinin “anahtar pratikleri” yanında bazı ek süreçleri de tanımlar. CMM ve SPICE modelleri arasındaki en önemli farklılık yapı bakımından karşımıza çıkmaktadır. CMM’in yapısında beş önemli olgunluk seviyesi ana rolü üstlenip pratikler belli olgunluk seviyelerine göre dağıtılarak tek boyutlu bir yapı oluşturulurken SPICE modelinde, süreçler ve yetenek seviyelerini belirleme olarak iki boyutlu bir yapıyı oluşturmaktadır.

TickIT modelinden SPICE modeline ise yapısal ve ayrıntılı seviyelerdeki temel farklılıklardan dolayı sadece üst düzey bir eşleştirme yapılabilir. Bu uyumluluk ise SPICE modelinin geliştirilmesinde ISO 9001 ve Yazılım Yaşam Çevrimi Süreçlerini tanımlayan ISO 12207 standardının girdi olarak alınmasından dolayıdır.

TickIT uyumlu bir firmanın CMM’in hangi düzeyinde olabileceği veya CMM’in üçüncü olgunluk seviyesinde olan firmanın ISO 9001 uyumlu olup olmayacağı gibi sorulara net cevapların verilmesi şu an pek mümkün gözükmemekle birlikte CMM kalite modelinin geliştirilmesinde önemli rol üstlenen Mark C.Paulk tarafından CMM’in anahtar uygulamaları ve ISO 9001’in maddeleri arasında bir eşleme yapılmasına yönelik çalışma yapılmış ve bazı eşleştirmeler yapılmaya çalışılmıştır. Örneğin ISO 9001’in Yönetim sorumluluğu (TS-ISO 9001: 1994, madde 4.1.) maddesi CMM’in ikinci seviyesine karşılık geldiği, yine Yazılım Proje Planlaması, Projenin İzlenmesi ve Yazılım Kalite Güvencesinin anahtar süreç alanlarıyla doğrudan ilişkili olduğunu belirtmiştir. Fakat bu eşlemeleri her bir ISO 9001 maddesi için yapmak çok zordur. Normalde ISO 9001 sertifikası olması için bir kuruluşun CMM’e göre üçüncü veya dördüncü seviyede olması gerekirken bazı firmaların CMM’in daha birinci seviyesinden bile sertifika alabildikleri gözlenmiştir. Mark C.Paulk’a göre bu farklılıklar ISO 9001’in üst düzey soyutlandırılmış yapısı ve bunun yol açtığı yorumlama hatalarından kaynaklanabileceğini belirtmektedir.



TickIT denetlemesi sonucunda firmaya ya sertifika verilir yada sertifika verilmez. Sertifika verilmediği durumlarda firmaya eksiklikleri bildirilir. Fakat bunları nasıl yerine getireceği konusu açıkça ifade edilmez. CMM’de ise belli olgunluk seviyelerini hedefleyen firmalar için hedefledikleri seviyeyi tutturabilmeleri için hangi noktalara odaklamaları gerektiğine dair açık bilgiler verir. SPICE modelinde ise amaçlanan tüm süreçler ve iyileştirme yolları için ayrıntılı yol gösterirken diğer yandan firmanın zaman içerisinde yaptığı süreç iyileştirmeleri ve geliştirmelerini belgelendirerek firmaya sunar.





5. TÜRKİYE'DE YAPILAN ÇALIŞMALAR

Yazılım kalite/süreç değerlendirme modelleri ile ilgili çalışmalar çok yenidir. Bu konuda faaliyet yapan kuruluşlara iki ayrı profilden bakmak daha doğru olacaktır. Birinci gruba kendi yazılım yeteneklerinin uluslararası bir modele göre değerlendirme yaptırarak sertifika almak isteyen kuruluşlar olarak kaşımıza çıkar. Diğer gruba ise istenen bu değerlendirme ve sertifikasyon hizmetini vermeyi hedefleyen kuruluşlar olarak bakabiliriz.

Türkiye'de ilk kez yazılım konusunda ISO 9001 sertifikasyonu 1995 yılında EBİ A.Ş. tarafından alınmış, fakat 1997 yılına gelindiğinde firma bu sertifikasyonun yararlarını sorgular duruma gelmiştir. Bir başka firma NETAŞ A.Ş., çalışmalarında ayrıntılı bilgi vermediğim telekomünikasyon yazılımı geliştirmeye özel Kanada'da geliştirilen TRILLIUM standart modele göre değerlendirme yapmakta. NETAŞ A.Ş. ARGE bölümü aynı zamanda son zamanlarda CMM model değerlendirmesi yaptırabilmek için çalışmalar yapmaktadır. ISO 9001 sertifikasyonu alma girişiminde bulunan yazılım firması sayısında artış olacağı beklenmektedir.

Remi sertifikasyon vermemekle birlikte SPICE modeline göre TÜBİTAK Marmara Araştırma Merkezi (MAM) ve İspanya'da bulunan Avrupa Yazılım Enstitüsü (ESI) arasındaki işbirliği çerçevesinde Türkiye'de bazı firmaların değerlendirmesi yapılmıştır. TÜBİTAK-MAM'ın 1988 yılından itibaren tamamen kendi başına SPICE değerlendirmesi yapma planları vardır.

Türkiye'de henüz TickIT sertifikasyonu alma girişiminde bulunan veya CMM temelli sertifikasyon veya değerlendirme yaptırmış bir yazılım firması bilinmemektedir.

Türkiye'de değerlendirme ve sertifikasyon yapabilecek kuruluşlar olarak TSE ve TÜBİTAK-MAM ön plana çıkmaktadır. ISO 9001 konusunda TSE genelde sertifikasyon hizmeti vermektedir. Yazılım konusunda ise henüz bir sertifikasyon



çalışması olmamakla birlikte yazılımdaki kalitenin uluslararası modellere göre değerlendirilmesi konusunda geleceğe yönelik planları bulunmaktadır.

TÜBİTAK Marmara Araştırma Merkezi (MAM), yazılımdaki kalitenin uluslararası modele göre değerlendirilmesi, ölçümlenmesi ve sertifikasyonu için yetenek geliştirme çalışmaları yapmakta. Bu amaçla Bilbao İspanya'daki Avrupa Yazılım Enstitüsü ile işbirliği anlaşması yapmış ve TÜBİTAK Marmara Araştırma Merkezinin planlarına göre ilk etapda SPICE (ISO 15504) modeline göre değerlendirmeler yapabilmek, daha sonra ise TickIT ve CMM modellerine göre değerlendirme/sertifikasyon yetenekleri kazanma planları vardır.



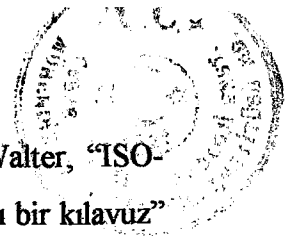


6. SONUÇLAR

Yazılım kalitesinin tanımı her zaman açık ve tutarlı olmadığı halde, çoğu proje yöneticisi yazılım kalitesini belirleyen dört etkende birleşiyor. Kullanılabilirlik, güvenilirlik, dokümantasyon ve bakım kolaylığı. Bu faktörlerin ağırlığı, şirketin kendi felsefesine dayanıyor. Bununla birlikte bugün, yazılımda kaliteden anlaşılan geliştirme sürecinin kontrol edilmesi, hataların ürün aşamasına gelmeden önlenmesi veya ortaya çıkarılıp düzeltilmesidir. Bu nedenle, tüm yazılım geliştirme faaliyetleri standartlar, yöntemler ve diğer yol gösterici dokümanlardan oluşan bir kalite güvencesi sistemi çerçevesinde yapılmalıdır. Yazılım sürecinin iyi kontrol edilmemesi sonucu ortaya çıkan hatalar, Amerika Birleşik Devletleri gibi büyük yazılım üreticisi ülkelerde birçok büyük yazılım projesinin ya planlanan maliyetin çok üzerinde yada tamamen projenin terk edilmesine neden olmaktadır. Bu tür başarısızlıklar bugün de devam etmektedir.

Kalite güvencesi yapılan hataların ve problemlerin sonradan ortaya çıkarılması olarak görülmemelidir. Kalite işlevi problemlerin daha çıkmadan standart, yöntem, teknik ve eğitim araçları kullanılarak önlenmesini amaçlar. Yeni geliştirilen metodolojiler ve yazılım geliştirme araçları kalite ve verimi şüphesiz artırmaktadır. Ancak, kuruluşların yeni teknolojilerden beklenen yararı sağlamaları, kendi kalite sistemlerini geliştirmiş ve geliştirme süreçlerini bu çerçevede tanımlamış olmalarını ve kaliteyi bir kültür olarak benimsemelerini gerektirir. Aksi taktirde, başarılar tesadüf olarak kalmaya mahkum kalacaktır.

ISO 9000 kalite güvence programı, imalat ve hizmet endüstrileri tarafından çok tutuldu. ISO 9000 onaylı firmaların istikrarlı bir yüksek kalite düzeyi yakalıyor, maliyetleri düşürmesi ve aynı zamanda müşterinin aradığı bir kriter olmasındandır. ISO 9000 kalite programı seti üç standarda ayrılır: ISO 9001, 9002 ve 9003 olup imalat ve üretim endüstrilerini hedefler, bu sebeple yazılım üretimiyle doğrudan bir ilgileri yoktur. ISO 9001 genel olarak tüm endüstrileri (yazılım üretimi dahil) hedefler. Bununla beraber 1990'ların başında ISO, özel olarak yazılım endüstrisinin ihtiyaçlarını karşılaya yönelik 9000-3 kaidelerini yayınladı.



ISO 9000 onaylı bir yazılım yayıncısı olan Softlab'den Frank Walter, "ISO-9000 kendi organizasyon ve işlemlerinize alacağınız kararlar için faydalı bir kılavuz" diyor. Walter, onun daha fazla dokümantasyona zorladığına ve amaca uygun hale getirmek için yeterince serbestlik tanıdığına inanıyor. Öte yandan bazen, küçük uygulamalar için, verimsiz bir harcama haline gelebilir. Walter, ISO-9000 uygunluğunun yüksek kaliteli bir ürün garantisi vermediğini, "müşteri memnuniyeti sağlamak" zorunda da olmadığını kabul ediyor.

ISO-9000 üzerine bir kaç kitabın yazarı olan Rob Kantner şuna dikkat çekiyor, "Sonuçta ISO şartlarında, sihirli bir şey ya da roket bilimi yok. Onlar, sadece herkesin akıl edebileceği basit kaideler". ISO-9000'in en önemli özelliklerinden bir tanesi firmaları kalite konusuna daha resmi şekillerde yaklaşmaları için teşvik ediyor. Bu ise daha olgun bir yazılım endüstrisine yol açabilir.

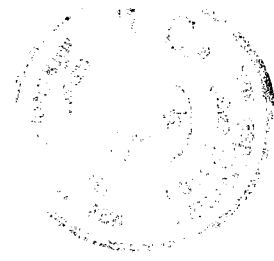
Yazılımda kalitenin ölçülmesi ve iyileştirilmesi amacıyla bugüne kadar bazı standart ve modeller geliştirilmiş olup modellerin çokluğu karşısında yazılımın kalitesiyle ilgilenen kuruluşların kesin bir yaklaşım sergilemeleri zor olmaktadır. Bu çalışma, yazılımda kaliteli ve güvenilir üretime önem veren kuruluşlar için öncelikle kendi mevcut yeteneklerini ve kalite sistemlerini değerlendirmede hangi model ve standardı seçmeleri gerektiğine karar vermelerine yardımcı olacaktır.

Kuruluşların hedefi ISO 9001 sertifikası almak ise TickIT modeli bir rehber olarak yardımcı olacaktır. Ancak kuruluşun hedefi içe dönük sürekli iyileştirmelere gitmek ise CMM ve SPICE modelini kendilerine örnek almaları uygun olacaktır. Özellikle SPICE projesi uluslararası standart olma yolundadır. Özellikle Avrupa Birliği yazılım pazarına girmek isteyen yazılım kuruluşları, bu modellerden birini kendilerine seçmek ve modelin getirdiği şartları yerine getirerek sertifikasyon almaları zorunlu olmaktadır. Ancak bugün yazılımın kalitesi ile ilgilenenler, modellerin çokluğu karşısında kesin bir yaklaşım ortaya koymaları zor olabilmektedir. Yazılım geliştirme süreçlerini daha iyi anlamak ve geliştirmek isteyen kuruluşlar bu tez çalışmasında incelenen standart ve modelleri inceleyerek işe başlayabilirler.

KAYNAKLAR

- [1]. Ercelebi S. ve Taral S., Yazılımda Kalite Güvencesi, TÜBİTAK-MAM, Gebze, Mayıs 1995.
- [2]. Andaç A., ISO-9001 Kalite Güvence Sistemi Standartlarının Yorumu ve Uygulama Örnekleri, Sayfa 1-2.
- [3]. TSE, TS-ISO 9001: Kalite Sistemleri-Tasarım, Geliştirme, Üretim, Tesis ve Serviste Kalite Güvencesi Modeli, Ankara, Aralık 1994.
- [4]. Ince F., An Introduction to Software Quality Assurance, 1994.
- [5]. Ince F., ISO 9001 and Software Quality Assurance, 1994.
- [6]. Yeh, Hsiag-Tao, Software Process Quality, 1993.
- [7]. Chow,T.S.,Software Quality Assurance, IEEE Computer Society Press, 1985.
- [8]. Mark C. Paulk, How ISO 9001 Compares with the CMM, IEEE Software, Ocak 1995.
- [9]. ESI, The SPICE Model of Best Practice, Bilbao Spain, Aralık 1995.
- [10]. TickIT Assessor/Lead Assessor-Delegate File, Presented by Lloyds Register Quality Assurance Limited, June 1997.
- [11]. ISO/IEC15504-1: Information Technology - Software Process Assessment Part 1: Concepts and Introductory Guide, Canada, February 1996.
- [12]. ISO/IEC15504-2: Information Technology - Software Process Assessment Part 2: A Reference Model For Processes And Process Capability, Canada, February 1996.
- [13]. SPICE (ISO 15504) Assessor Training Notes V2.0, ESI ,Bilbao Spain, January 1997.
- [14]. Scientific American, Eylül 1994.
- [15]. Ince F. ve Taral S., Yazılımda Kalite Sistemi Sertifikasyonu ve Süreç Değerlendirme/İyileştirme Modellerinin Karşılaştırmalı Değerlendirilmesi, 6.Ulusal Kalite Kongresi, Kasım 1997.
- [16]. Tober B., Quality Factor, McGraw-Hill Inc., Byte 1997.
- [17]. Ince F. ve Taral S., Yazılım Mühendisliğinde İleri Uygulamalar TÜBİTAK-MAM Yaz Okulu Ders Notları, Gebze, Eylül 1997.

ÖZGEÇMİŞ



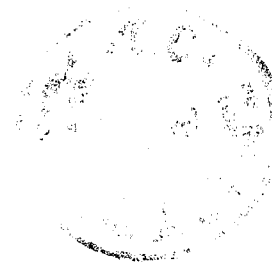
Ender Şahinaslan 27.03.1972 tarihinde Sivas ili Gürün ilçesinin Suçatı Kasabasında doğdu. Öğrenimlerini sırayla Suçatı ilkokulu ve ortaokulu, Gürün Lisesi'nde başlayıp Ankara Aydınlikevler İnönü Lisesi'nden 1989'da mezun oldu. 1986 Gürün ortaokullararası bilgi yarışmasında birincilik aldı. 1991 yılında Trakya Üniversitesi Müh.-Mim.Fak. Bilgisayar Mühendisliği Bölümünü kazandı ve 1995'de iyi derece ile mezun oldu. Yaz stajlarını Trakya Üniversitesi B.İ.M (Edirne, 1992 yaz, 20 iş günü), Adalet Bakanlığı Adli Sicil ve İstatistik Müdürlüğü B.İ.M (Ankara, 1993 yaz, 35 iş günü) ve Trakya Üniversitesi B.İ.M (Edirne, 1992 yaz, 35 iş günü) olarak yaptı. 1995 yılında Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü yüksek lisans programına kayıt yaptırıp bir yıl İngilizce hazırlık programına katıldı. Gazi Üniversitesi'nin 1996 yılında açmış olduğu araştırma görevlisi sınavını kazanarak Mart 1996-Ocak 1997 arasında Gazi Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Eğitimi Bölümünde araştırma görevlisi olarak çalıştı ve nakil yoluyla halen çalışmalarını sürdürmekte olduğu Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümüne atandı.

TÜBİTAK-MAM Bilişim Teknolojileri Enstitüsü tarafından 09-18 Eylül 1997 tarihleri arasında düzenlenen Yazılım Mühendisliğinde İleri Uygulamalar konulu eğitim programına katılarak başarı belgesi aldı. Eğitimi yoğun bir şekilde Yazılım Geliştirme ve Tasarımı konusunda aldı ve çalışmalarını bu yönde sürdürdü. Çalışmalarından bazıları, Edirne Valiliği İl Planlama Müdürlüğüne Yatırım Paket programını 1995 yılında yaparak Edirne valisi sayın Engin Kuru'dan teşekkür belgesi aldı, Adalet Bakanlığı A.S.B.İ.M'e MCP işletim sistemi altında Cobol-77 programlama dili ve SDF (Screen Desing Facility) paket programını kullanarak Kütüphane programı yaptı. Ayrıca Fakülteler Öğrenci İşleri, Bina Yöneticiliği ve Marmara Mobilya paket programları yaptı. Cobol-77, Pascal, C, Assembly, Delphi, C++ Builder ve SQL, büyük sistemler için Assembler ve Rexx, programlama dilleri, Dbase, Paradox ve Vista III veri tabanı programları, Dos, Widows ve Unix işletim sistemleri ile MS-Office paket programları üzerinde bilgi ve deneyim sahibi olduğu bazı bilgisayar yazılımları arasındadır.

EKLER

Süreç Kategorisi		Süreç	
ID	Başlığı	ID	Başlığı
CUS	Müşteri-Satıcı Süreç Kategorisi		
	CUS.1	Yazılım Edinme	
	CUS.2	Müşteri İhtiyaçlarını Giderme	
	CUS.3	Yazılım Tedarik Et	
	CUS.4	Yazılımı İşlet	
	CUS.5	Müşteri İhtiyacı Sağla	
ENG	Mühendislik Süreç Kategorisi		
	ENG.1	Sistem İsterlerini Sağla	
	ENG.2	Yazılım İsterlerini Geliştir	
	ENG.3	Yazılım Tasarımını Geliştir	
	ENG.4	Yazılım Tasarımını Uygula	
	ENG.5	Yazılım Entegrasyon ve Testi Yap	
	ENG.6	Sistem Entegrasyon ve Testi Yap	
	ENG.7	Sistemin ve Yazılımın Bakımını Yap	
SUP	Destek Süreç Kategorisi		
	SUP.1	Dokümantasyon Geliştir	
	SUP.2	Konfigürasyon Yönetimi Yap	
	SUP.3	Kalite Güvencesi Sağla	
	SUP.4	İş Ürünlerini Doğrula	
	SUP.5	İş Ürünleri Tedarik Et	
	SUP.6	Ortak Gözden Geçirmeler Yap	
	SUP.7	Denetimler Yap	
	SUP.8	Problemleri Çözümle	
MAN	Yönetim Süreç Kategorisi		
	MAN.1	Projeyi Yönet	
	MAN.2	Kaliteyi Yönet	
	MAN.3	Risk Yönetimi Yap	
	MAN.4	Taşeronları Yönet	
ORG	Organizasyon Süreç Kategorisi		
	ORG.1	İşin Mühendisliğini Yap	
	ORG.2	Süreci Tanımla	
	ORG.3	Süreci İyileştir	
	ORG.4	Kaliteli İş Gücü Temin Et	
	ORG.5	Yazılım Mühendisliği Alt Yapısını Sağla	

Tablo Ek-1 - SPICE Referans Modeli Süreçler ve Süreç Kategorileri



ID	Başlığı
Level 1	Uygulanan (Varolan) Süreç
1.1	Süreç Performans Niteliği
Level 2	Yönetilen Süreç
2.1	Performans Yönetim Niteliği
2.2	İş Ürünü Yönetim Niteliği
Level 3	Yerleşmiş Süreç
3.1	Süreç Tanım Niteliği
3.2	Süreç Yeniden Kaynaklandırma Niteliği
Level 4	Kestirilebilen süreç
4.1	Süreç Ölçüm Niteliği
4.2	Süreç Kontrol Niteliği
Level 5	En İyileşen Süreç
5.1	Süreç Değişim Niteliği
5.2	Sürekli İyileştirme Niteliği

Tablo Ek-2 - SPICE Referans Modeli Yetenek Seviyeleri ve Süreç Nitelikleri