

**YAZILIM KALİTE SİSTEMLERİNİN
KARŞILAŞTIRILMASI VE BİR ÖRNEK
MODELİN OLUŞTURULMASI**

Ali TÜZÜNKAN

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Eylül 2004

JÜRİ VE ENSTİTÜ ONAYI

Ali Tüzünkan'ın Yazılım Kalite Sistemlerinin Karşılaştırılması ve Bir Örnek Modelin Oluşturulması başlıklı Bilgisayar Mühendisliği Anabilim dalındaki, Yüksek Lisans tezi 16.08.2004. tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı Soyadı	İmza
Üye (Tez Danışmanı):	Yard. Doç. Dr. Yusuf Oysal	
Üye	: Prof. Dr. Ali Güneş	
Üye	: Yard. Doç. Dr. Celal H. Kağmıoğlu	

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 17.09.2004 tarih ve 30/14... sayılı kararıyla onaylanmıştır.

Enstitü Müdürü
Prof. Dr. Altuğ İFTAR
Fen Bilimleri Enstitüsü
Müdürü

ÖZET

YÜKSEK LİSANS TEZİ

YAZILIM KALİTE SİSTEMLERİNİN İNCELENMESİ VE KOBİ YAZILIM ŞİRKETLERİ İÇİN BİR ÖRNEK MODELİN OLUŞTURULMASI

ALİ TÜZÜNKAN

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Yusuf Oysal

2004, 121 SAYFA

KALDER tarafından 2001 yılında hazırlanan Yazılım Sektörü anketinin değerlendirmesinde, Türkiye'deki Bilişim sektörünün birçok sorun ve eksikliklerle karşı karşıya olduğu değerlendirilmiş ve bir dizi öneriler Türkiye Bilişim sektörüne aktarılmıştı. Başlıca öneriler: Kalite ve verimlilik için gerekli Yazılım Kalite Sistemlerinin referans alınması, eğitim eksiklerinin giderilmesi, uygun altyapı ve araçlar ile çalışma ortamının sağlanması, hız ve sistematik zaman planlama ve kontrolünün sağlanması idi.

Bu çalışma ile birlikte, Türkiye Bilişim sektöründeki Küçük ve Orta ölçekli yazılım firmaları için bu önerilere cevap verecek bir model, bir çerçeve hazırlandı. Bu amaçla, öncelikle günümüzde referans olarak kullanılan Yazılım Kalite Sistemleri incelenmiş ve bu incelemenin sonucunda, özellikle KOBİ'lere yönelik olarak örnek bir yazılım firması için günün koşullarında kaliteyi sağlamak için gerekli eğitimlerin kapsadığı eğitim paketi, süreç akış planları, yazılım kalite sistemlerinin otomasyonunu sağlayacak bir XML teknolojisinin yaygın olarak kullanıldığı portal şablonları hazırlanmıştır.

Anahtar Kelimeler: Yazılım Mühendisliği, Kalite Sistemleri, Otomasyon
Yazılımı, Proje Yönetimi, Web Portalları

ABSTRACT

Master of Science Thesis

ANALYSIS OF SOFTWARE QUALITY SYSTEMS AND DEVELOPING A MODEL FOR SMALL SOFTWARE COMPANIES

ALİ TÜZÜNKAN

**Anadolu University
Graduate School of Sciences
Computer Engineering Program**

Supervisor: Assoc. Prof. Yusuf OYSAL

2004, 121 pages

The 2001 KALDER survey indicates that the field of information technologies in Turkey is facing numerous setbacks. Several recommendations, therefore, were made in order to overcome these problems. These suggestions include; (1) utilizing Software Quality Systems for overall quality and productivity, (2) quenching the problematic elements of education in software engineering, (3) providing necessary infrastructure and tools, and (4) maintaining and controlling speed and suitable project management. In this context, a model that utilizes these suggestions was developed to help small and mid-sized software companies in Turkey improve their operations. After examining the Software Quality Systems, which are used as reference, an educational package was prepared for small software companies containing necessary elements to achieve quality under particular circumstances. Additionally, process workflow diagrams were created, and several portal templates that can be used to ensure the automation of Software Quality Systems were built.

**Keywords: Software Engineering, Software Quality, Automation Software,
Project Management, Web Portal**

TEŐEKKÜR

Bu tezin hazırlanmasında danıřmanım olarak bana yol gsteren Anadolu Üniversitesi'nden Sayın Yard. Doç. Dr. Yusuf OYSAL'a, desteklerini esirgemeyen Savronik alıřanlarına, Sayın Arř. Gör. Alper K. UYSAL'a ve Sayın Arř. Gör. Ali Yavuz AKIR'a ve her trl alıřmamda bana destek olan aileme teőekkr ederim.

Ali TZNKAN
atuzun@ciu.edu.tr

İÇİNDEKİLER

ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER.....	v
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ	x
KISALTMALAR.....	xi
1. GİRİŞ	1
1.1. Kalite Kavramı.....	1
1.2. Kalite'nin Tarihçesi.....	2
1.2.1 Yazılım Krizi	3
1.3. Yazılım Kalite Sistemleri	5
1.3.1 Yazılım Kalite Faktörleri	6
1.4. Türkiye'deki Firmaların Özellikleri.....	9
1.4.1 Süreçlerin Durumu.....	9
1.4.2 İnsan Kaynakları	10
1.4.3 Altyapı	10
2. ÖRNEK MODEL'İN PLANLANMASI	13
2.1. Eğitim Eksiklikleri	13
2.2. Belgelendirme Eksiklikleri	14
2.3. Yardımcı Araçların Kullanımındaki Eksiklikler.....	16
2.4. Yönetim, Verimlilik ve Kalite Bilincinin Eksikliği.....	20
2.5. Yazılım Geliştirme Süreçlerinin Belirsizliği	21
2.5.1 Olgunluk ve Yetenek Modeli (Capability Maturity Model) Çözümü.....	23
2.5.2 IEEE-12207 Kalite Sistemi Çözümü.....	24
2.6. Bilgi Portalı.....	31
2.6.1 Proje Yönetimi.....	33
2.6.2 Doküman Yönetimi.....	34
2.6.3 Uygulamalara Arayüz	34

3. ÖRNEK FİRMA ÖZELLİKLERİ.....	35
3.1. Altyapı Özellikleri.....	35
3.2. Gelecek Hedefleri	36
3.3. Sistemdeki Sorunlar	37
4. OLUŞTURULAN ÖRNEK MODEL.....	38
4.1. Eğitim Paketi	39
4.2. Prosedürler ve İş Akış Süreçleri	40
4.3. Yazılım Mühendisliği Araçları.....	40
4.3.1 Test Aracı	40
4.3.2 REM (Requirement Management Tool).....	41
4.3.3 Visio 2003.....	43
4.3.4 Infopath 2003.....	44
4.3.5 Sharepoint Team Services.....	46
4.3.6 Microsoft Project 2003.....	47
4.4. Örnek Uygulama.....	48
5. OLUŞTURULAN MODELİN DEĞERLENDİRİLMESİ VE SONUÇ.....	55
EK-A EĞİTİM PAKETİ.....	59
EK-B SÜREÇLER VE TALİMATLAR.....	93

ŞEKİLLER DİZİNİ

1-1. Kalite Elemanları.....	2
1-2. Yazılım arz ve talep artışları	4
2-1. BDYM araçları ve kullanım evreleri (Demirörs 2001)	18
2-2. Ön ve arka kullanım cephelerine BDYM araçları (Demirörs 2001)	18
2-3. Yazılım Mühendisliği Ortamı modeli (Demirörs 2001).....	19
4-1. Örnek Model'in Çerçevesi (UML Dağıtım Şekli).....	39
4-2. Test işlemlerinin hedefleri (Demirörs 2001).....	41
4-3. Senaryo	49
4-4. Ayrıntılı form otomasyonu.....	50
4-5. SharePoint Portal sayfası	51
4-6. Satınalma talep formu sayfası	51
4-7. Satınalma talep formu.....	52
4-8. Yeni eklenen formun form sayfasında görüntüsü	53
4-9. Satınalma talep formu onay görüntüsü	53
4-10. Satınalma talep formu nihai onay görüntüsü	53
4-11. Satınalma talep formu talep sonucu görüntüsü	54
A-1. Takımda İletişim Sunusu 1.....	59
A-2. Planlama ve Kontrol Sunusu 1	60
A-3. Planlama ve Kontrol Sunusu 2	61
A-4. Planlama ve Kontrol Sunusu 3	62
A-5. Planlama ve Kontrol Sunusu 4	63
A-6. Planlama ve Kontrol Sunusu 5	64
A-7. Planlama ve Kontrol Sunusu 6	65
A-8. Proje Yönetimi 1	66
A-9. Proje Yönetimi 2.....	67
A-10. Proje Yönetimi 3.....	68
A-11. Proje Yönetimi 4.....	69
A-12. Proje Yönetimi 5.....	70
A-13. Proje Yönetimi 6.....	71
A-14. Proje Yönetimi 7.....	72

A-15. Proje Yönetimi 8.....	73
A-16. Proje Yönetimi 9.....	74
A-17. Proje Yönetimi 10.....	75
A-18. Proje Yönetimi 11.....	76
A-19. Problem Çözme Teknikleri 1	77
A-20. Problem Çözme Teknikleri 2	78
A-21. Problem Çözme Teknikleri 3	79
A-22. Problem Çözme Teknikleri 4	80
A-23. Problem Çözme Teknikleri 5	81
A-24. Problem Çözme Teknikleri 6	82
A-25. Problem Çözme Teknikleri 7	83
A-26. Problem Çözme Teknikleri 8	84
A-27. Problem Çözme Teknikleri 9	85
A-28. Toplantı Yönetimi 1.....	86
A-29. Toplantı Yönetimi 2.....	87
A-30. Toplantı Yönetimi 3.....	88
A-31. Toplantı Yönetimi 4.....	89
A-32. Toplantı Yönetimi 5.....	90
A-33. Toplantı Yönetimi 6.....	91
A-34. Toplantı Yönetimi 7.....	92
B-1. Konfigürasyon Yönetim Planı Kontrol Listesi.....	93
B-2. Kalite Planı Kontrol Listesi 1	94
B-3. Kalite Planı Kontrol Listesi 2	95
B-4. Yazılım İsterleri Kontrol Listesi.....	96
B-5. Yazılım İsterleri Kontrol Listesi 2.....	97
B-6. Kişisel Değerlendirme Formu 1.....	98
B-7. Kişisel Değerlendirme Formu 2.....	99
B-8. Kişisel Değerlendirme Formu 3.....	100
B-9. Kişisel Değerlendirme Formu 4.....	101
B-10. Sürüm Kontrol Listesi	102
B-11. Yazılım Tasarım Süreci İş Akışı.....	103
B-12. Yazılım İsterleri Süreci İş Akışı	104

B-13. Kod Geliştirme Süreci İş Akışı.....	105
B-14. Yazılım İşletim Süreci İş Akışı	106
B-15. Yazılım Test Süreci.....	106
B-16. Yazılım Bakım Süreci İş Akışı.....	107
B-17. Belgelendirme Süreci İş Akışı.....	107
B-18. Problem Çözme Süreci.....	108
B-19. Konfigürasyon Yönetimi Süreci İş Akışı.....	109

ÇİZELGELER DİZİNİ

2-1. CMM Seviyeleri.....	24
--------------------------	----

KISALTMALAR

BDYM:	Bilgisayar Destekli Yazılım Mühendisliği
CAD:	“Computer Aided Design”
CRM:	“Customer Relationship Management”
CMM:	Capability Maturity Model
ERP:	“Enterprise Resource Planning”
ISO :	“International Standards Organization”
KALDER:	Kalite Derneği
KOBİ:	Küçük ve Orta Büyüklükteki İşletme
MERNİS:	Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü Projesi
MPM:	Milli Prodüktivite Merkezi
NASA:	“National Space Association”
PUKÖ:	Planla-Uygula-Kontrol et-Önlem Al
SPPS:	“Sharepoint Portal Server”
SCM:	Tedarik Zinciri Yönetimi
SEI:	Software Engineering Institute
SQL	“Structured Query Language”
SWEBOK:	“Software Engineering Body of Knowledge”
TS:	Türk Standartları
UML:	“Unified Modelling Language”
VEDOP:	Vergi Dairelerinin Otomasyonu Projesi
XML:	“eXtended Markup Language”

1. GİRİŞ

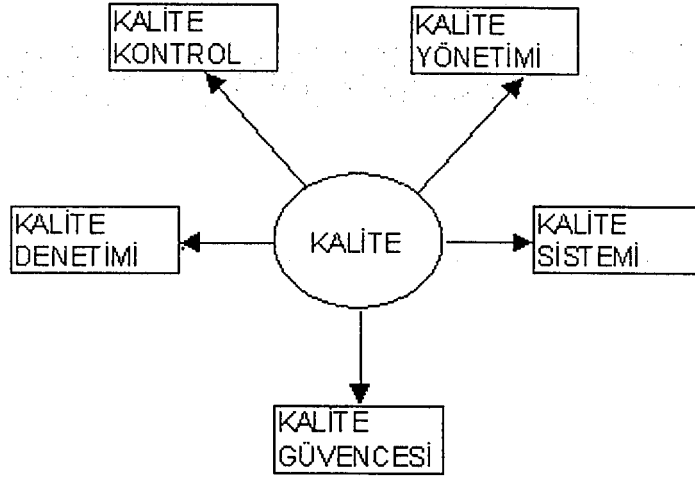
KALDER tarafından 2001 yılında hazırlanan Yazılım Sektörü anketinin değerlendirmesinde, Türkiye'deki Bilişim sektörünün birçok sorun ve eksikliklerle karşı karşıya olduğu değerlendirilmiş ve bir dizi öneriler Türkiye Bilişim sektörüne aktarılmıştır. Başlıca öneriler: Kalite ve verimlilik için gerekli Yazılım Kalite Sistemlerinin referans alınması, eğitim eksiklerinin giderilmesi, uygun altyapı ve araçlar ile çalışma ortamının sağlanması, hız ve sistematik zaman planlama ve kontrolünün sağlanmasıdır.

Bu tez çalışması ile birlikte, Türkiye Bilişim sektöründeki Küçük ve Orta ölçekli yazılım firmaları için bu önerilere cevap verecek bir model, bir çerçeve hazırlanması amaçlanmaktadır. Bu çalışma kapsamında, öncelikle günümüzde referans olarak kullanılan Yazılım Kalite Sistemleri incelenmiş ve bu incelemelerin sonucunda, özellikle Küçük ve Orta Büyüklükteki İşletmelere (KOBİ) yönelik olarak örnek bir yazılım firması için, günün koşullarında kaliteyi sağlamak için gerekli eğitimlerin kapsandığı eğitim paketi, süreç akış planları, yazılım kalite sistemlerinin otomasyonunu sağlayacak XML teknolojisinin yaygın olarak kullanıldığı portal şablonları hazırlanacaktır.

1.1. Kalite Kavramı

TS 9005'e göre kalite denince "bir ürün veya hizmetin belirlenen veya olabilecek ihtiyaçları karşılama kabiliyetine dayanan özelliklerinin toplamını", kalite kontrol denilince "kalite isteklerini sağlamak için kullanılan uygulama teknikleri ve faaliyetlerini", kalite güvencesi denilince "ürün veya hizmetin kalite için belirlenen istekleri karşılamak amacıyla yeterli güveni sağlaması için gereken planlı ve sistematik faaliyetlerin bütünü", kalite yönetimi denilince "genel yönetim fonksiyonunun kalite politikasını tespit eden ve uygulayan bölümünü", kalite sistemi denilince "kalite yönetiminin uygulanması için gerekli olan kuruluş yapısı, sorumluluklar, prosedürler, işlemler ve kaynakları", kalite denetimi denilince "kalite ile ilgili faaliyetlerin ve sonuçlarının, planlanan düzenlemelere uyup uymadığının, bu düzenlemelerin etkili olarak uygulanıp

uygulanmadığının ve amaca ulaşmak için uygun olup olmadığının sistematik ve tarafsız olarak incelenmesi” anlaşılmaktadır (Şekil 1-1) (Özkan 2001).



Şekil 1-1. Kalite Elemanları

1.2. Kalite'nin Tarihçesi

Dünya endüstri tarihinde 1950 ve 1960'lı yılların kütle üretimi ve kütle tüketimi devri olarak adlandırılması gibi, 1970'li ve 1980'li yıllar da Kalite yılları olarak anılacaktır. Bu on yılda, Amerikan tüketicileri tercihlerinde güvenilirlik, kalitede süreklilik ve kişisel tatminlerini asgari ölçüde karşılayan standartlardaki ürünleri talep etmeye başladılar. Yüksek kaliteli firmalar pazar paylarını yükseltti. Buna karşın bir çok kötü kaliteli firma ise pazar paylarını kaybetti ve iş hayatından silindi.

Eski politikalara sıkı sıkıya bağlı olan ve ne yaparlarsa yapsınlar müşterinin satın almasını bekleyen bir çok Amerikan şirketinin lideri, geriye baktıklarında, yakın geçmişin “daima yarının bir başlangıcı” olduğunu ve daha önceki üretim uygulamalarının yanlış olduklarını kavradılar.

Birleşik devletlerde (hepsi Japon ve hepsi yüksek kaliteli ürün olarak bilinen) Honda, Panasonic, Hitachi, Seiko, Nikon, Mazda ve Bridgestone kelimeleri günlük hayatın içinde girdiler ve bu sırada birden bire Amerikan üst yönetim yetkilileri Deming, Juran ve Geigenbaum gibi eski kalite liderlerine başvurdular. Çok geçmeden Amerikan firmaları Japon Kalite Rönesansının büyüünün, prensiplerde ve uygulamada, ilk önce birleşik devletlerde

keşfedildiğini ve kullanıldığını anladılar. Örneğin, II. Dünya savaşı sırasında Amerikan malzeme fabrikalarında uygulanan ve Walter Shewhart'ın öncüsü olduğu İstatistiksel Kalite Kontrol, W. Edward Deming ve J.M. Juran tarafından 1950'lerin başında Japonya'da uygulanmaya başlanmıştı. Oysa Birleşik Devletler bu sırada kütle halinde ürettiği nisbeten az kaliteli ürünleriyle sürekli ve hızlı bir büyüme içindeydi.

Deming'in felsefesindeki ana başlıklar aşağıdaki gibidir.

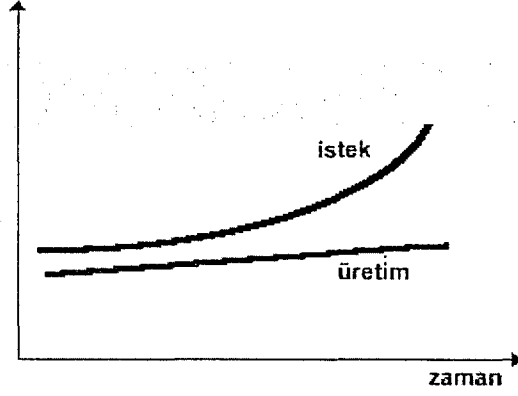
- Müşteriler ile satıcılar arasında yakın ilişki kurmak.
- Hataları azaltmak ve sapmaları daraltmak için İstatistiksel Proses Kontrolü kullanmak.
- Son ürün muayenesine dayanan çalışma şeklini ortadan kaldırmaya çalışmak, bunun yerine ürün ve hizmet kalitesini güvence altına alacak proses kontrolü uygulamak.
- Bölümler arasındaki engelleri ortadan kaldırmak, yönetim ile işçiler arasındaki güveni arttırmak ve insanları eğitmek için sürekli çaba vermek.
- Sürekli iyileşmeyi sağlamak
- Üst yönetimin destek, katkı ve liderliğini sadece aylar veya birkaç sene değil çok uzun bir süre sürdürmek.

Deming'in bu kavramları daha sonra dünyadan da geniş destek buldu. Bu kavramlar zinciri daha da gelişerek ISO9000 ile üretim ve hizmet endüstrilerinde hayat buldu. ISO9000 ve IEEE standartlarına göre kalite kavramı: "Bir ürünün veya hizmetin belirlenen ihtiyaçları karşılayabilme yeteneğine yönelik özelliklerinin bütünü" şeklinde tanımlanmıştır (Sanders D. ve ark. 1994).

1.2.1 Yazılım Krizi

Bu terimin esas açıklayıcısı, üstel olarak artan talebe karşılık doğrusal diyebileceğimiz bir hızla artış gösteren üretim kapasitesidir ve Şekil 1-2'de görüntülenmektedir. Durum, arz ve talebin arasının gittikçe açılacağını ve dünya yazılım ihtiyacının gittikçe daha ümitsiz bir duruma geleceğini ima etmektedir. Diğer yandan ele alınan yazılım projelerindeki başarısızlık oranı da şaşırtıcı derecede yüksek olup bu krizi beslemektedir. Şu an vardığımız sonuç, yazılım

sürecinin zor, anlaşılammış ve tam olarak nasıl yapılacağını bilmediğimiz bir saha olduğudur.



Şekil 1-2. Yazılım arz ve talep artışları

Zorluklarına rağmen bu sahaya hakim olanlar ödüllendirmesi çok yüksek sonuçlar alarak her sahada önderliği ellerinde tutacaklardır. Artık yazılımın girmediği, kontrol etmediği bir saha veya bir problem çözüm yolu kalmayacaktır. Eğitim ileri teknolojiler ile yapılmakta, savaşlar bu yollarla kazanılmakta, her türlü üretim yine benzer şekilde rekabet edebilecek hale gelmektedir. İleri teknolojilerin ise donanım bileşenleri bugün için çözülmüştür. Haberleşme ile ilgili alt yapı hızla gelişmekte, fiyatlar düşerken kapasite artmaktadır. Bilgisayar donanımı ile ilgili gelişmeler benzer şekilde ve şaşırtıcı boyutlarla önümüzdedir. İki seneden kısa bir süre içerisinde işlemcilerin hızı katlanmakta, belleklerin kapasitesi katlanmakta ve her ikisinin de fiyatları düşmektedir. Yalnız bu donanımların çözüm üretmesi kendi kendilerine yapabilecekleri bir iş değildir. Eldeki donanım kapasitesini kullanabilecek yazılım erginliğine dünya hiç bir zaman ulaşamamıştır. Yazılım krizi deyimi bu açıdan da tanımlanmıştır. Sonuç olarak yazılım kabiliyetini elinde tutanlar için ufuklar açıktır, ve her konuda en önemli kilit teknoloji, uzun bir süre için yazılım olarak devam edecektir.

1.3. Yazılım Kalite Sistemleri

Kalite kavramı, doğumundan bu yana içinde bulunduğu koşullardan da etkilenererek değişik tanımlara sahip olmuştur. Giderek artan ölçüde ilgi görmesinin (popülarite) getirdiği olumlu yönlerin yanısıra, kalite ile ilgili kavramları tekrar gözden geçirmekte fayda görülmektedir.

ISO tanımlarına göre yazılım, bir veri işleme sistemi operasyonuna bağlı olan programlar, yordamlar ve dokümantasyonudur. Yazılım ürünü ise kullanıcıya ulaştırılmak üzere tanımlanmış bilgisayar programları kümesi, yordamları ve dokümanlarıdır.

İşletme yazılımları diye tabir edebileceğimiz yazılımlar genelde 3 temel gruba ayrılır. 1000\$'lar civarında maliyete sahip olan en alt kategoride küçük çaplı kullanıcıların gereksinimlerini karşılayan, çok fazla destek istemeyen, kullanımı kolay yazılımları kabul edilmektedir. 100.000\$'lar civarında maliyete sahip olan orta kategoride kullanıcı sayısı 50'lere varmaktadır. En son kategoride ise milyon \$ civarında maliyete ve yüzlerce kullanıcıya sahip olan yazılımlar vardır. Bunların içinde en hızlı gelişme sırası en alt kategoriden, üst kategoriye doğru bir sıra izlemektedir. Öte yandan ofis (workgroup) yazılımları, kişisel (eğitim, oyun vb.), endüstriyel ve finansal fonksiyonlara odaklanmış yazılımlar (cad, cam, cae, istatistik, bankalar), yüksek teknoloji yazılımları (telekom, uydu, füze, nükleer santraller vb.) gibi çeşitli sınıflandırmalar da yapılabilir.

Yazılım, içinde yaşadığımız "bilgi toplumu" çağının en önemli mekanizması olmak yolundadır. Ülkemizde kamu ve özel sektöre ait her boyuttaki yazılım çalışmalarının yanısıra, toplumsal boyuta sahip bilişim projelerinden BDE (Milli Eğitim Bakanlığı Temel Eğitim Projesi), Sağlık Enformasyon Sistemi (Sağlık Bakanlığı) gibi projeler, konunun önemini anlamak için yeterlidir.

Kalite kavramına tekrar geri dönülürse, kalite olgusunun üretim alanında, bitmiş ürünün kontrolundan, giderek üretim aşamasına ve tasarımına doğru ilerlerken, hizmet alanına da bir çok cepheden girdiği görülmektedir. Durum böyle iken, bünyesinde ürün olma özelliklerini, süreç olma özelliklerini ve ayrıca yaratım aşamasında birbirinden farklı bir çok disiplini (teknoloji yönetimi, proje yönetimi, süreç yönetimi, kalite yönetimi, insan kaynakları yönetimi vb.) kullanan yazılım sektöründe de kalite arayışlarının olmaması mümkün değildir.

Geçmiş dönemlerde bir ürünün son (final) kontrolunda aldığı onay, onun kalitesini belirlerken, yazılımın kod satırlarında program geliştiriciye yönelik olarak yer alan ve “*rem” diye anılan program açıklamalarının varlığı da yazılımın kaliteli olduğunun belirtisiydi. Ardından yazılımın bitiminde uygulanan inceleme (review), teftiş (inspection), gözlem (walkthrough), denetim (audit) diye tanımlanan değişik kalite çalışmaları geldi. Bugün ise yazılım sektörü, yazılımın her aşamasında kalite olgusunu tam anlamıyla gerçekleştirebilmek için değişik modelleri, metodolojileri, resmi ve gayri resmi standartları kullanmaya çalışıyor, bunların dünya çapında kullanılabilir olması için çaba veriyor (Özkan 2001).

1.3.1 Yazılım Kalite Faktörleri

Yazılımın ürün olma özelliği açısından yazılım sınıflandırmalarından bağımsız olarak kalitesini belirleyen temel özellikler 9126 standardında tanımlanan aşağıda belirtilen altı yazılım kalite faktörü ile ifade edilmektedir:

- *Fonksiyonellik*: Tanımlanan ihtiyaçları karşılama ölçüsüdür.
- *Güvenilirlik*: Yazılımın, kullanıma hazır olma süresinin ölçüsüdür.
- *Kullanılabilirlik*: Öğrenilebilirlik, anlaşılabilirlik ve işletilebilirlik kavramları açısından yazılımların kolay kullanma ölçüsüdür.
- *Verimlilik*: Zaman ve kaynak kullanımı açısından sistem kaynaklarının en uygun şekilde kullanma ölçüsüdür.
- *Bakılabilirlik*: Yazılımda ortaya çıkan problemlerin; analiz edilebilirlik, değişebilirlik, sağlamlık ve test edilebilirlik açılarından, kolaylıkla tamir edilebilme ölçüsüdür.
- *Taşınabilirlik*: Yazılımın; uyarlanabilirlik, kurulabilirlik, uygunluk ve değiştirilebilirlik özellikleri açısından bir ortamdan başka bir ortama aktarılabilme ölçüsüdür (Pressman 2001).

Tekrar kullanılabilirlik (reuse) ise üzerinde dikkatle durulması gereken bir konudur. Tekrar kullanım sadece kodun tekrar kullanımını değil, tasarımın, test ortamının, dokümanların ve benzeri bilgilerin kullanımını kapsamaktadır. Yazılım, esnek bir ürün olma özelliği kullanılarak, önceden belli bir disiplin ile üretilirse, bir formdan diğerine geçebilir. Bir başka deyişle yeniden

kullanılabilirlik özelliği ile önceden belli bir gereksinime göre üretilmiş olan bir yazılım, farklı bir gereksinime göre yeniden düzenlenebilir.

Diğer bir önemli konu ise; her sektörde tüm ürünlerin ortak sorunu olan “müşteri tatmini nedir, nasıldır?” sorusunu da gerçekçi bir şekilde çözebilmek gerekmektedir. Oldukça soyut ve değişken bir olgu olan “müşteri tatmini” konusunda ilerleme sağlamanın ön koşulu, müşteriyi yazılım ve kalite unsurları hakkında üründen bağımsız olarak- bilgilendirmektir.

Sonuçta yukarıda sayılan özelliklerin tümü, yazılımın bir ürün olarak (süreç değil!) giriş (input), işlem (proses), çıkış (output) ve çıktılarının çıkış kurallarına uygunluğunu kontrol eden kalite kontrol kriterlerine uyumunu sağlarlar.

Ancak yazılımlar, türlerine göre sınıflandırıldıklarında yukarıda sayılan genel özellikler daha detaylı özelliklere dönüşmektedirler. Örneğin iletişim ve eşgüdüm odaklı “groupware” çalışmalarında yazılımın dış ortamlarla entegrasyonu, platform bağımsız çalışabilme özelliklerini desteklemesi önemli kalite kriterleridir. Bir başka örnek olarak, yüksek teknoloji alanında (nükleer santral, savunma sanayi, uydu vb.) çalıştırılan yazılımlardaki “aksaklığa dayanıklılık” özelliklerini verebiliriz. Çeşitli denetim metotlarıyla hata bulma, giderek hatayı sınırlama, ileriye ve geriye dönük olarak hatayı kurtarma, aksaklığı giderme vb. özellikler, ilgili yazılımın kalite göstergeleridir. Yine bir başka örnek olarak, akıllı yazılımları verilebilir. Bu tür yazılımlardaki, kullanıcı denetimi altında olmadan çalışabilme, başka sistemler ve kullanıcılarla diyalog kurabilme, dış ortam değişikliklerine tepki verebilme, işin o durumda en iyi nasıl yapılacağına karar verebilme, “on-line” ya da “off-line” çalışabilme vb. özellikleri ise yazılımın türüne bağlı olan farklı kalite göstergeleridir. Uygulama yazılımlarında ise genellikle, “açık sistem” anlayışına uygun olarak üretilmiş yazılımlar, önemli kalite kriterlerini yerine getirmiş sayılmaktadırlar. Bu tür yazılımların özellikleri, en az değişiklikle taşınabilirlik, verilerin paylaşılabilirliği ve kullanım kolaylığı amaçlı, açık destek yeteneklerine sahip olmalarıdır.

Yazılım projelerinde kaliteyi sağlamak; yazılım kalite faktörlerinin karşılanmasıyla meydana gelecektir. Yazılım kalite güvence sistemleri ve süreçleri ise yazılım kalite faktörlerinin sağlanmasını garanti altına alma çabaları sonucunda ortaya çıkmıştır.

Kalitenin uygulanabilir olduđu yazılım geliştirme süreçleri hakkında, değişik görüşler vardır. Bir görüşe göre iki aşamalı bir süreç tanımı yeterlidir:

Geliştirme (analiz/tasarım/program/test) ile geçiş ve sonrası (geçiş/işletim/bakım) süreçleri; öte yandan daha detaylandırılmış süreçler de söz konusu olabilir: İhtiyaç analizi, sistem tasarımı, kodlama, test ve bakım süreçleri gibi. Ya da hedeflenen ürünün tanımı, yöntem geliştirme, uygun teknoloji araştırması ve seçimi, analiz, görsel tasarım, teknik tasarım, geliştirme, iç testler, dış testler ve pilot proje süreçleri sayılabilir. Proje yönetimi açısından ele alındığında bunlara maliyet tahmini, risk analizi, kaynak ayırma ve koordinasyon, süreç planlaması ve zamanlama süreçleri de eklenebilir.

Yazılımı, süreç olma özelliği içerisinde etkileyen en önemli kriterlerden birisi, onun entellektüel bir ürün olmasından ileri gelir. Onun bu özelliği, kullanıcıya yansıyan üst yapı (görsel arayüz) ile performans ve programcıya yansıyan alt yapıyı etkileyebilecek güçtedir. Dolayısı ile yazılım geliştiren şirketlerde insan kaynakları politikasının sürekli eğitim, iş tatmini vb. unsurlarla beraber dikkatle planlanması ve yürütülmesi gerekmektedir.

Yazılım süreçlerinde kalite güvencenin nasıl sağlanması gerektiği hakkındaki genel düşünceleri toplarsak:

- Kalite güvence, organizasyona bir fonksiyon olarak yerleştirilir (kalite güvence grubu belirlenir, kaliteyle ilgili politika ve prosedürler saptanır, kalite güvence yönetimi seçilir, kalite kontrol sonuçları toplanır ve değerlendirilir, kalite kontrol sonuçlarına göre iyileştirme çalışmaları yapılır, iyileştirme sonuçlarına göre standartlar revize edilir)
- Yazılım çevrim süresi azaltılır (müşteri gereksinimleri önceden ve somut bir şekilde tespit edilir, tekrar kullanım artırılır, değişiklik yapmak azaltılır, süreçler iyileştirilir-basitleştirilir, çalışma ortamı ve ergonomide iyileştirmeler sağlanır, iş için en iyi personel kullanılır, sorunlara proaktif bir şekilde yaklaşılır, standartlar kullanılır, kod ve algoritma karmaşıklığı engellenir).
- Yazılımlarda sık kullanılan ve standart (popüler) fonksiyonlar kullanılır, bilgi, uyarı ve hata iletişimi, güvenlik (yetki), performans,

diğer platformlarla ilişki kurabilme ve uyarlanabilme yeteneđi (customizing) için uygun alt yapı oluşturulur.

1.4. Türkiye'deki Firmaların Özellikleri

Kalder (Kalite Derneđi) Uzmanlık Grubu¹, Türkiye'de yazılım mühendisliđi ve yazılım yönetimi uygulamalarının bir fotoğrafını çekmek; ülke ekonomisi için stratejik önem taşıyan bu alanda iyileştirme önceliklerini belirlemek; varılan sonuçları önce katılımcılarla tartışmak, sonra da kamuoyu ile paylaşmak amacıyla 12 Kasım - 31 Aralık 2001 tarihleri arasında web ortamında KalDer 2001 Yazılım Sektörü Anketi'ni düzenlemiştir.

Anket konuları Yazılım Mühendisliđi ve Yazılım Yönetimi olarak iki ana başlık altında toplanan Software Engineering Body of Knowledge'in konu sınıflandırmasına dayanan toplam 10 kategoriden oluşmaktadır.

Sonuçları üretmek için gereken girdiler

- süreçler
- insan kaynakları
- altyapı

olarak değerlendirilmiştir.

Çıktılar ise öneriler ile birlikte ele alınmıştır.

1.4.1 Süreçlerin Durumu

Yazılım yönetimi konularının anket katılımcıları tarafından daha az bilindiđi, kurumsallaşmanın yazılım mühendisliđi konularına kıyasla daha az olduđu ortaya konuldu. Anket katılımcılarının süreç yeterliliđini “yönetilen” (CMMI 2) beyan etmesine rağmen, verilerin “yapılan” (CMMI 1) düzeyini işaret ettiđi, iyimser bir görüşün hakim olduđu değerlendirildi.

¹ Kalder Yazılımda TKY Uzmanlık Grubu; yazılım sektöründe kalite bilincini geliştirmek, süreç iyileştirme yöntem ve modellerini tanıtmak, ürün kalitesi ve süreç yetkinliklerini artırmak isteyen kuruluşları bilgilendirmek ve desteklemek misyonu ile kuruldu

1.4.2 İnsan Kaynakları

Yazılım şirketlerinde proje ihtiyaçları ve alınan işe göre geçici yapılar oluşturulduğu gözlemlenmiştir. Performans ve kariyer yönetimi için katedilmesi gereken aşamalar olduğu, eğitim, altyapı ve eleman nitelik ve niceliğinde sıkıntı söz konusu olduğu tespit edilmiştir. Yazılım kadrosu 1-9 veya 50-99 arasında olduğunda yönetimin daha etkin olduğu, küçük organizasyonlarda kodlama dışındaki süreçlerde zaman ve eleman sorunlarının baş gösterdiği, rol paylaşımı yönteminin kullanıldığı ele alınmıştır.

1.4.3 Altyapı

Temel altyapı göstergesi olan araç kullanım oranının, verimlilik için zorunlu olduğu düşünülmesine rağmen, düşük olduğu, araçların oldukça pahalı olduğu, seçim ve temin konusunda güçlükler yaşandığı, yeterli düzeyde ölçüm yapılamadığı ve iyileştirme için girdilerde sıkıntı olduğu işaret edilmiştir.

KalDer tarafından yapılan anketin sonuç bölümünde aşağıdaki öneriler yer almaktadır;

- kalite ve verimlilik adına süreç standartları/modellerinin referans alınması,
- uygun altyapı ve araçlar ile çalışma gerekliliği,
- eğitim eksikliklerinin giderilmesi
- maliyet ve kaynak yönetimi için organizasyonlarda uzmanlık alanlarının çeşitlendirilmesi,
- rol esaslı kariyer geliştirme modellerinin uygulanması,
- yazılımlarda kullanım ve bakım kolaylığının, tekrar kullanılabilirliğin daha çok öneme sahip olması gerektiği,
- sistematik maliyet kestirim ve kontrolünün önemi,
- rekabet gücü adına kaynak ve altyapı kullanımının planlanması,
- hız ve sistematik zaman planlama ve kontrolünün önemi değerlendirildi.

“KalDer 2001 Yazılım Sektörü Anketinin” sonuçları süreç standartlarının/modellerinin bizim yazılım sektörümüzde de referans alınması, uygun

altyapı/araçlarla çalışma ve eğitim eksikliklerinin giderilmesi, organizasyonlardaki uzmanlık alanlarının çeşitlendirilmesi, mühendisler için rol esaslı kariyer geliştirme modellerinin uygulanması gereklerini ortaya koymuştur.

Yaklaşık 60 yerli şirketin katıldığı anketin çarpıcı sonuçlarına göre ülkemizde yazılım projelerinin profesyonel bir şekilde planlanıp yönetilmediği ortaya çıkmıştır. Ankete katılan şirketlerin beyanlarına göre yazılım yönetimi çok zayıf bir şekilde uygulanmakta ve yönetimde yeterince araç desteği kullanılmamakta, yazılım projelerinde ölçme metodları kullanılmamakta, düzenli iş akışı takip edilmemekte ve yönetsel zaafılar yazılım kalitesini olumsuz yönde etkilemektedir.

Kuruluşlarda yazılım mühendisliği bilgi birikimi ağırlıklı olarak “usta-çırak” ilişkisi ile kurumsallaştırılıyor. Usta çırak ilişkisi yöntemi ise en çok yazılım yüklenicisi firmalarda görülüyor.

Yine anket sonuçlarına göre yazılım üreten firmalarda süreç yaklaşımı olmadığı, özellikle iç kullanım amaçlı yazılım geliştiren firmaların herhangi bir standardı temel almadığı ortaya çıkmaktadır.

Yazılım mühendisliği ve yönetim araçlarının seçiminin çoğunlukla proje yöneticisi tarafından yapılmakta olduğu ortaya çıkmıştır. Süreçlere verilen önem, eldeki kaynaklarla orantılıdır. Yazılım tasarımı da kaynak varsa “tasarımcı”, kaynak yoksa “en yetkin proje elemanı” olarak genellikle proje yöneticisi tarafından yapılmaktadır. Kodun tümleştirilmesi de “uzman!” ellere teslim edilmekte, projelerde yazılım, proje yöneticisi tarafından tümleştirilmektedir.

Araçları kullanacak mühendislerin araç seçimlerinde etkili olmadığı ve teknik danışmanların araç seçimine katılımının olmadığı ortaya çıkmıştır.

Yazılım proje planlarının; %52’inde proje süre/emek tahminleri, %28’inde proje hedefleri, %25’inde proje görev/zaman çizelgesi yer almamaktadır.

Risk yönetimi süreci; firmaların sadece %25’inde tanımlanmakta ve bu firmalardan sadece %8’inde risk yönetimi araçları kullanılmaktadır. Yazılım proje planları içinde; proje riskleri ve risk yönetim metodu %30 oranında yer almaktadır.

Risk yönetimi uygulamaları 2001 yılının 2. çeyreğinde bir araştırma grubu tarafından dünya çapında gerçekleştirilen, aralarında Xerox, Motorola, Nasa’nın

da bulunduđu 268 yazılım firması üzerinde yapılan “Risk Yönetimi” uygulamaları konusundaki bir araştırmanın sonuçlarına göre; katılımcı firmaların %97’inde riskleri tanımlamak ve değerlendirmek için prosedürler kullanıldıđı, %80’inde problemlerin tahmin edilebildiđi ve önlem alınabildiđi, %60’ında sürprizlerin önlendiđi ortaya çıkmıştır. Katılan kuruluşların %64’ünde proje yönetimi ofisinin bulunduđu, %68 oranında proje yönetim süreçleri tanımlandıđı, %70 oranında yazılım geliştirme süreçlerinin tanımlandıđı belirlenmiştir. Araştırma sonuçları; yazılım projelerinde küreselleşme ve gerektirdiđi rekabet koşulları göz önüne alındıđında, standartlaşma ve evrensel metotların kullanımı konusunda üç kritik boyutu net olarak ortaya koymaktadır.

Birinci boyutta yazılım üreten firmaların konuya gereken önemi vermesi ve gerekli kaynađı sağlayarak yazılım projelerinin tanımlı metot/standartlar ile dünya standartlarında geliştirilmesi gerekliliđi ortaya konmuştur.

İkinci boyutta üniversitelerin eğitim kapsamlarını, yazılım mühendisliđi kapsamı ile kıyaslayarak gözden geçirme gereksinimi ortaya çıkmıştır. Benzer şekilde yazılım geliştirme konusunda tanımlı metot/standartların, eğitim kapsamında yer alması gerekliliđi de değerlendirilmiştir.

Üçüncü boyutta ise yazılım alımı yapan firmaların bilgi ve bilinç düzeylerinin yükseltilerek, yazılım/yazılım firması seçiminde dođru tercihleri yapabilmelerine olanak sağlanması gerekliliđi belirtilmiştir (Aykol 2002).

Bahsedilen bu faktörleri dikkate alacak bir örnek modelin varlıđı, bilişim sektörünün yaşadıđı sorunları aşmasında büyük yardımları olacaktır.

2. ÖRNEK MODEL'İN PLANLANMASI

Türkiye'deki bilişim sektöründeki firmaların büyük bir çoğunluğunun KOBİ sınıfına girmesine rağmen, üretilen yazılım ürünlerinin kalitesinin yeterli olmadığı değerlendirilmektedir.

Kalder tarafından yapılan anketlerde KOBİ'ler için tespit edilen eksiklik ve sorunları, (örnek firmada da benzer bir şekilde ortaya çıkmaktadır) aşağıdaki şekilde sınıflandırılabilir:

- Eğitim eksiklikleri
- Belgelendirme eksiklikleri
- Yardımcı araç kullanımının yetersizliği ve eksikliği
- Yazılım geliştirme süreçlerinin eksikliği, belirsizliği ve yetersizliği

2.1. Eğitim Eksiklikleri

Eğitim eksiklerinin nedenlerinden bir tanesi örnek firmada olduğu gibi firmalarda yeterli bilgisayar mühendisinin çalıştırılmaması olmaktadır. Bilgisayar Mühendisliği bilimi, firmalar tarafından sadece programcılık olarak algılanmaktadır. Bilgisayar ve yazılım mühendisliğinin sadece programcılıktan ibaret olduğu düşünüldüğü takdirde teknik ve süreçlere ilişkin birçok sorun ile karşılaşmak son derece doğaldır. Ülkemizde son yıllarda ardı ardına açılan Bilgisayar Mühendisliği bölümlerinin mezun vermeye başlaması ile birlikte bu sorunun ortadan kalkabileceği düşünülmektedir.

Yazılım üreten şirketler, bu ihtiyacı ancak yeterli altyapı sağlanırsa karşılayabilmektedirler. Bir yazılım organizasyonu, başarılı olabilmek için herşeyden fazla yazılım mühendislerine ihtiyaç duyar. Sermaye ve makine gibi diğer faktörler, yazılım endüstrisi için ikincil sırada kalmaktadır. Yazılım sektörü, gelişmekte olan ülkelerde göreceli olarak küçük olduğundan, nitelikli iş gücünü sağlamak için gereken eğitim fonksiyonunu verebilecek kurumların başında, üniversiteler gelmektedir.

Türkiye'deki bilgisayar mühendisliği bölümlerinin çoğu Yazılım Uygulamaları bilgi kategorisinde yeterli kapsamı sağlamaktadır. Ancak, Yazılım Ürün Mühendisliği ve Yazılım Yönetimi bilgi kategorilerinde maalesef yetersiz

kalındığı anlaşılmaktadır. Üniversitelerden bazıları Yazılım Ürün Geliştirme bilgi kategorisinde bulunan Yazılım Gereksinimleri, Yazılım Tasarımı bilgi alanlarını içeren bazı dersler verselerde, bu dersler bu bilgi alanlarının sadece bazı bilgi ünitelerini kapsamaktadır. Bazı bilgi alanlarında ise hiçbir ders verilmemekte yada hiçbir derste bu alanlar anlatılmamaktadır. Bu bilgi alanlarına örnek olarak: Örneğin: Yazılım Operasyonları ve Yazılım Güncelleştirme bilgi alanlarında hiçbir ders verilmemektedir (Demirörs 2002).

Eğitim alanındaki tüm bu eksikler, kurumlardaki yetişmiş, kalifiye insan gücünü yetersiz kılmaktadır. Bunun sonucunda, kurumların rekabet ve kalite konularında geri düşmeleri kaçınılmaz olmaktadır. Tüm bu açıklar kurum içi eğitimlerin önemi bir kat daha artırmaktadır.

Kurum içi eğitim konularında, mali kaynak problemi ile de karşılaşıldığından özellikle süreç iyileştirme çabalarının en başında çalışanların öğrenme süreçlerini aktif olarak kendilerinin yönlendirmelerinin zorunluluğu ortaya çıkmaktadır. Aktif öğrenme, İnternet'teki bilgi kaynaklarının etkin kullanımı, sinerjik etkilerin kullanılması gerekmektedir.

2.2. Belgelendirme Eksiklikleri

Belgelendirme, tüm dünyada kurumsallaşmanın ve kalitenin olmazsa olmazları arasında değerlendirilmektedir. Ancak Kalder'in yaptığı anket sonucunda belirlenen belgelendirme eksikliği (MPM'nin yaptığı araştırmaya göre de kalite sistemlerine sahip firmaların karşılaştıkları en büyük sorun %44 ile belgelendirme), ülkemizdeki bilişim sektörünün geri kalmışlığının en temel nedenleri arasında görülebilir. Belgelendirme eksikliklerinin nedenleri aşağıdaki gibi değerlendirilmiştir.

- Eğitim eksikliği: Türkiye'deki daha önceden de bahsedilen üniversitelerdeki Yazılım Ürün Mühendisliği konularındaki eğitim eksikliği belgelendirme sırasında problem yaratmaktadır.
- İlave iş gücü gerektirdiğinin düşünülmesi: Birçok firmanın kalite belgesine ya da kalite sistemine araç yerine amaç olarak bakmasından dolayı, belgelendirme işini işin özü kaynağı yerine, kalite belgesinin gereği olarak bir zorunluluk olarak bakmaktadır.

Bunun sonucunda da belgelendirme için ayrılan zaman çok az olmakta ve bunun sonucunda ortaya çıkan belgelerin de ciddi bir şekilde yetersiz olabileceği değerlendirilmektedir.

- Katılımcılık, motivasyon ve ekip çalışmasındaki yetersizlikler: Belge hazırlamak genelde bir işin özüne hitap etmediğinde kaynaklanan nedenlerle bir belgeyi hazırlamak bir kişinin sorumluluğunda yapılmaktadır. Sinerjik etki ve takım çalışmasından yoksun olan belgenin yetersiz olabileceği çok açıktır.
- Belge şablonlarının eksikliği: Bir şablonun varolmadığı bir dokümantasyon, içerik açıdan isteneni sağlaması tamamen kişisel çabaları bağlı kalacaktır. Böyle bir ortamda, kurumsal olarak belgelendirmenin kalitesinin güvence altında olduğundan bahsedilemez.
- Yardımcı araçların eksikliği: Yazılım mühendislerinin çoğunda yazı yazma konusunda bir direnç kolaylıkla görülebilmektedir. Başka ortamlarda hali hazırda olan verilerin dokümantasyona aktarılabilmesi, işin tekrar etmemesi, verimlilik ve veri bütünlüğü açısından büyük önem taşımaktadır. Bu nedenle, yardımcı araç kullanımındaki eksiklik önemli bir sorun olarak bilişim sektörünün önünde durmaktadır.
- Hazırlanan belgelere ulaşmaktaki pratik sorunlar (Dosyalama sorunları, indeksleme sorunları, vs.): Belgelere ulaşmakta yaşanan bazı pratik sorunlar, hazırlanacak belgeleri işin dinamik, her an değişen proje şartlarına uyum sağlaması için engel teşkil etmektedir.
- Bilgilerin, kurumsal bilgi portal eksikliği nedeniyle kayıt altına alınmadan dolaşması (email, telefon vs): Belgelendirmenin ana amaçlarında bir tanesi kurum içerisinde üretilen bilgilerin kayıt altına alınarak, öğrenen organizasyon yaratmaktadır. Özellikle küçük işletmelerde görülen olgulardan bir tanesi üretilen bilgilerin email, telefon, sözlü vs. gibi yollardan paylaşıldığıdır. Bu da belgelendirme zaten paylaşılan bilginin bir daha paylaşılması anlamına gelebilmektedir (örnek: email ile önceden gönderilen hata kodlarının

ve yaşanan sorun ve tartışmaların tekrar belgelendirmenin içerisinde yer alması ilave iş gücü gerektirdiği düşünülmektedir).

- Hazırlanan belgelerin gözden geçirme süreçlerinin yapılmaması ve dolayısıyla belgelendirme, sağlanması gereken faydalarının şirkete yansımaması: Ciddi şekilde gözden geçirilmeyen belgelendirmenin yanlış ve hataları barındıracağı açıktır. Bu sebeple, belgelendirme işi gereken önem ve kaynak verilmemektedir. Gözden geçirilmeyen doküman, test edilmemiş yazılıma olan benzerliği büyüktür.

Yukarıda bahsedilen nedenlerin çözülmesi için uygulamada aşağıdaki önlemler alınmıştır.

- Uluslararası standartlar (MIL-STD-498) referans alınarak belge şablonları hazırlanması,
 - belgelerin doğruluğunun anlaşılabilmesi için kontrol listelerinin hazırlanması,
 - belgelerin gözden geçirilmesi amacıyla süreçlerin oluşturulması,
 - kurumsal bilgi portalının oluşturulması,
 - bazı belge tipleri için kullanılacak yardımcı araçlar temini,
- belgelendirme konusunda yaşanan bir çok önemli sorunu çözecektir.

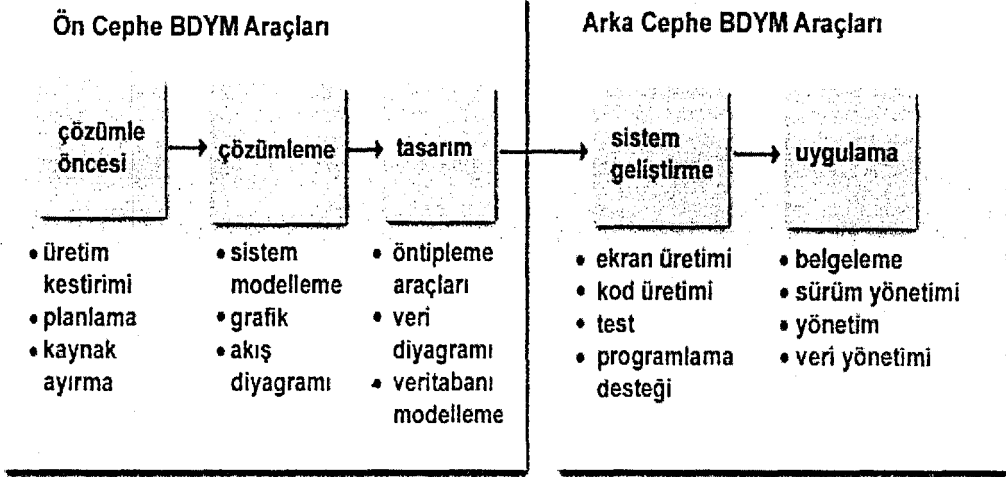
2.3. Yardımcı Araçların Kullanımındaki Eksiklikler

Yardımcı araçların etkin bir şekilde kullanımı, yazılım üreten firmalar için önemi, sektörün doğası gereği hızlı, sürekli değişen istekler ve koşullar içermesinden gelmektedir. Yardımcı araçların etkin kullanımı, yazılım geliştirme süreçlerinin, hem etkinliğine hem hızına hem de kalitesine olumlu katkılar yapacaktır. Bu amaçla, küçük yazılım firmaları için en gerekli araçlar yazılım gereksinimleri ve yazılım test araçlarının kullanımını sağlanmalıdır.

Yazılım krizinin işaret ettiği gibi yazılım projelerinin geliştirmesinde hem isteğin doğru ve kaliteli olarak üretilmesi hem de yeterli hızda üretime erişilip isteğin karşılanabilmesi zorlukları vardır. Otomasyon yardımı ile bu eksiklikleri biraz da olsa gidermek mümkündür. Araçlar insanın yerini almazlar, ancak geliştirme sürecinde yardımcı olurlar.

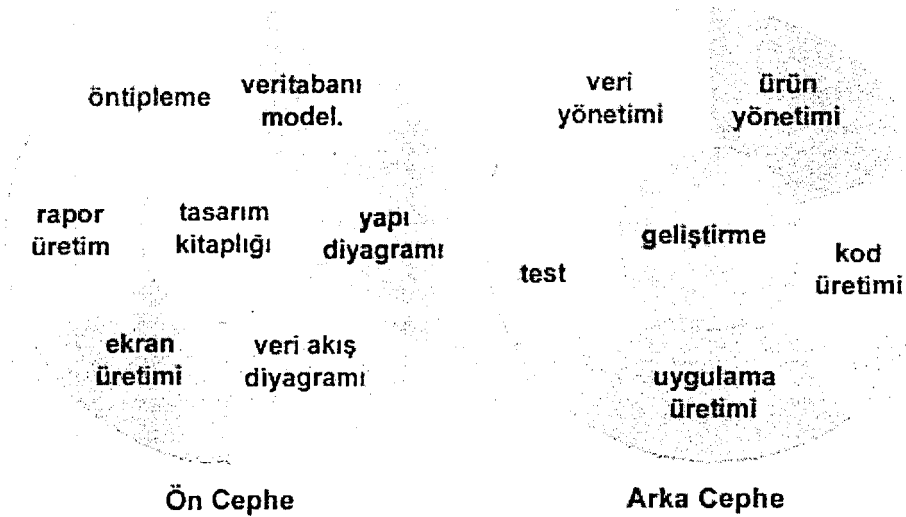
Bu tür araçlar önce makine mühendislerine yönelik olarak Bilgisayar Destekli Tasarım (Computer Aided Design : CAD) adı ile yaygınlaştı. Genellikle 3 boyutlu katı maddelerin çizimi, değişik açılar ve kesitler tanımlanarak görüntünün oluşturulması temelinde başarılı bir geçmişe sahip oldular. Bunu elektronik konusundaki araçlar izledi ve günümüzde bir çok mühendislik alanında tasarım artık büyük ölçüde bir yazılım çabası şekline döndü. Bu araçlar üzerinde, geliştirilecek sistemler yazılan programlar ile tanımlanmaktadır. Gerekli yerde grafik ortamda yapılan girdiler araç tarafından otomatik olarak bir programa çevrilmektedir. 'Mum dibine ışık vermez' sözünü doğrularcasına yazılım mühendisleri bu gibi bir araç desteğine diğer mühendisliklerden daha geç kavuştular. Ancak hızla yetişip değişik boyutlarda gelişen BDYM teknolojisi, diğer alanlardaki benzer teknolojiler ile bir karşılıklı etkileşime girdi ve sonra yazılım konusunda bulunan teknikler (örneğin UML) diğer alanlara da uygulanmaya başlandı.

Bilgisayar Destekli Yazılım Araçları (BDYM) değişik geliştirme safhalarına yönelik olabilirler. İlk başlarda kodlama düzeyinde destek veren araçlar ortaya çıkmıştır. C gibi bir dilde yazılan bir programda fonksiyonların hiyerarşisi, bunun grafik gösterimi, aranan bir program biriminin bulunması gibi konularda programcılara destek olmuşlardır. Daha sonra bu destek, kodlama düzeyinden, daha önceki ve sonraki çaba türlerine doğru gelişmiştir. 80'li ve 90'lı yıllarda BDYM desteğinin en çok çözümlene ve tasarım evrelerinde kullanımının önemsendiğini görüyoruz. Şekil 2-1, değişik evrelerde kullanılacak BDYM araçlarını ardışıl evrelere göre göstermektedir. Giderek önem kazanan bir özellik de değişik evrelere yönelik desteğin (veya araçların) ne derecede bütünlük oluşturduğudur. Örneğin, geleneksel çözümlene yapılırken veri akış diyagramındaki bir veri biriminin ayrıntılarını görmek için kullanıcı, ortam değiştirmeden, az çaba ile bu birimin veri kitaplığındaki girişine ulaşabilmektedir.



Şekil 2-1. BDYM araçları ve kullanım evreleri (Demirörs 2001)

Şekil 2-2'de ise değişik evreleri, ön ve arka cephe olarak ayırarak BDYM araçlarını bu yeni gruplandırma içerisinde göstermektedir. Bu iki şekil de bütün olabilecek BDYM araçlarını kapsamamaktadır. Evreler de Çağlayan modelinde bahsedildiği gibi biraz değişik isimlendirilmiş olabilir. Amaç örnekler ile araçların gruplama perspektiflerine oturtulmasıdır. Yeni araçlarda bir çok destek, tümleşik olarak, tek araç içinde sunulmaktadır.

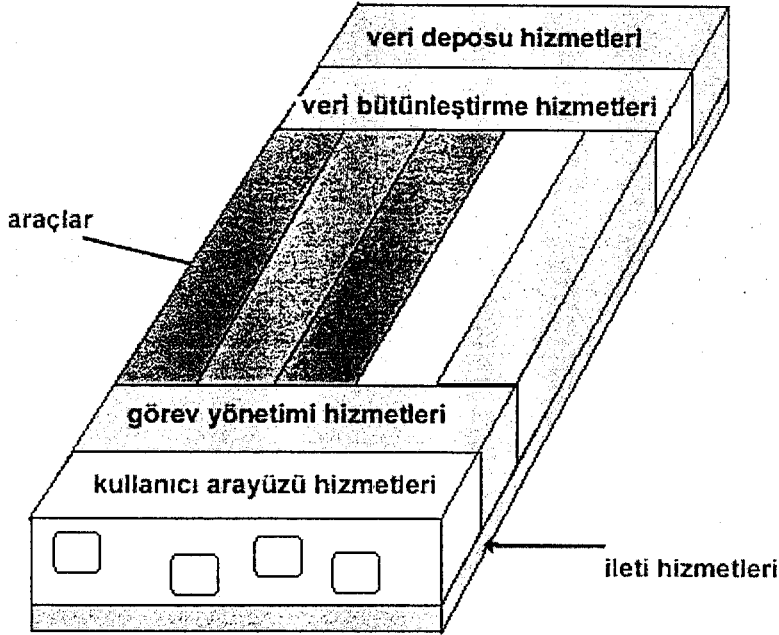


Şekil 2-2. Ön ve arka kullanım cephelerine BDYM araçları (Demirörs 2001)

Bir çaba türünün (çözümleme gibi) her işlemine destek verecek BDYM araçları olduğu gibi değişik işlemler için ayrı araçların da kullanılması söz konusudur. Bir evreye destek verecek araçlar kümesi, BDYM Çalışma Tezgahı olarak da tanımlanır. Örneğin kodlama işlemi için kelime işlemci, derleyici, hata

bulucu gibi araçlar kodlama için çalışma tezgahı oluşturabilir. Ancak bu araçların ne denli bütünleştirildiği önemlidir. Bu yüzden BDYM araçlarının daha genel bir anlayış içerisinde Yazılım Mühendisliği Ortamları (YMO) olarak örgütlenmeleri de savunulmuştur. Değişik araçlar, işletim sistemi (operating system) ve hatta donanım platformu da gözönüne alınarak bütüncül bir yaklaşımla bir YMO oluşturmak üzere değerlendirilirler.

BDYM araçlarının tümleştirilmesinde donanımdan başlayıp kullanıcı arayüzüne kadar ulaşan seviyeler ele alınmalıdır. Bu tümleştirme çerçevesi Şekil 2-3'de gösterilmektedir. Ayrıca BDYM tümleşmesi veri, kontrol ve kullanıcı arayüzü gibi açılardan ele alınır.



Şekil 2-3. Yazılım Mühendisliği Ortamı modeli (Demirörs 2001)

Yardımcı araçların kullanımı, özellikle yazılım sektörü için olmazsa olmazlar arasındadır. Ancak, özellikle kalite sistemleri söz konusu olduğu takdirde sadece programlama yönelik değil, belgelendirme, tasarım, test ve problem takip konularında yardımcı araçların yaygın olarak kullanılması bir zorunluluk halini almaktadır. Türkiye'deki yazılım sektörünün yardımcı araçların kullanımındaki yetersizliklerin nedeni mali olabileceği gibi kullanımının gerekliliği konusunda da tam bir görüş oluşturulamadığı gerçeği yatmaktadır.

KOBİ boyutundaki yazılım firmaları için tam entegreli ve her şeyi yapan bir araç ihtiyacı olmamasına rağmen, bazı araçların temini ve kullanımı gereklidir. Bunların başında yazılım gereksinimlerinin takip araçları, birim test araçları, problem takip araçları gelmektedir. Aslında, bu tip araçları temin etmek bir KOBİ için çok pahalıya mal olabileceği gibi, endüstride yaygın olarak kullanılan bedava araçlarla da belirli bir seviye tutturmak mümkün. Gereksinimlerin takibi için REM, Word, Test Yazılımları için xUnit, Planlama için hemen hemen tüm firmalarda kullanılan Microsoft Office'in bir parçası olan Microsoft Project pekala kullanılabilir. Web tabanlı "bug tracking" sayfaları da oldukça yaygın olarak kullanılabilir.

Yardımcı araç kullanımının bütün boyuttaki firmalar için gerekliliği açıktır. Yardımcı araç kullanımı için başlangıç olarak, elde ne varsa veya en ucuz ne varsa kullanmak bile amaçlan kaliteye ulaşmakta önemli birer basamak olacaktır.

2.4. Yönetim, Verimlilik ve Kalite Bilincinin Eksikliği

Kalite sistemlerinin temel özelliklerinden üçünü; sürekli gelişme, verilerle düşünme ve müşteri odaklı olmayı sayabiliriz.

Sürekli gelişen bir örgütün oluşabilmesi için tüm süreçlerin sistematik olarak izlenmesi - veriye dönüştürülerek, geri bildirimlerin alınmasını gerekli kılmaktadır. Dolayısıyla tüm çalışanlar verilerle düşünme alışkanlığı edinmeli ve süreçlerini izleyerek sürekli iyileştirebilme yeterliliğini gösterebilmelidirler. "Ancak bizler toplum olarak hafızası oldukça zayıf ve rakamları sevmeyen daha çok duyguları ile karar verme alışkanlığına sahip bir insan karmasına sahibiz. Dolayısıyla sistemin sürekli gelişebilmesi için gerekli verilerin toplanması, analiz edilmesi ve çözüm önerilerinin bireysel ve departmansal düzeyde organize edilerek yönetilmesi gereklidir." (Şahin 2003).

Ne yazık ki, ülkemizdeki durum bu açıdan, pek iç açıcı değildir. Kalite güvence sistemlerini sadece ticari kaygılarla destekleyen ancak sistemin felsefesini tam anlayamayan üst yönetici sayısı az değildir (Şahin 2003).

Kurum ve projelerindeki statükodan memnun olmayan çalışanlar varsa kalite sistemleri gerçekten doğru değişikliğe rehberlik edebilir. Çalıştıkları

koşullardan memnun çalışanlar için kalite sistemlerinin fazla bir değişiklik sağlaması beklenemez.

Yazılım süreç iyileştirme tartışmalarının açılış sorusu şu olmalıdır: “Kurum niye bir kalite sisteminin rehberliğine ihtiyaç duymaktadır?”. Eğer ihtiyaç duyulan şey kurum profesyonel iş amaçları ile süreç iyileştirme istekleri arasında sıkı bir ilişki varsa; o kurum için kalite sistemleri çok kullanışlı ve yetenekli olacaktır. Bir kurum için kalite sistemleri rehber niteliğinde olmalıdır. Kalite sistemleri yol göstericidir. Profesyonel iş hedefleri ana amaç olarak benimsenmeyen kurumlarda kalite sisteminin kalite sisteminden çıkıp, bürokrasi halini aldığı, çalışanların ise birer bürokrat halini aldığı gözlenebilmektedir (Humprey, 2001).

Yazılım süreçleri, firmaların boyutuna uygun bir şekilde tasarlanmalıdır. Kalite sisteminin amaç değil de iş hedeflerine ulaşmakta bir araç olduğu gerçeğinin özellikle yöneticiler tarafından kabul edilmesi gerekmektedir. Aksi takdirde zaten Türk toplumunun genel karakteristik özelliklerinin dışında özellikler barındıran kalite sistemlerinin uygulanamamasını, dolayısıyla iş hedeflerinden iyice uzaklaşmayı kaçınılmaz kılacaktır.

Tüm bunları başarmak için kurumsal değişim en üst yönetici tarafından yönlendirilmelidir. Çünkü; liderlerin hareketleri gözlenir ve takip edilir. Vizyonu oluşturacak kişi liderlerdir. Süreç iyileştirme çabaları için danışmanlık faaliyetlerini liderler yürütür. Süreç iyileştirme çabalarının ödüllendirilmesi veya değişimin önündeki engellerin cezalandırılması için üst yöneticilerinin yönlendirmesi olmazsa olmazlar arasındadır. Bu yönlendirme olmazsa; süreç iyileştirme çabaları kısa ömürlü kişisel çabalardan öteye gidemez (SEI 2002).

2.5. Yazılım Geliştirme Süreçlerinin Belirsizliği

Üretilen yazılımların maliyet etkin olması, zamanında üretilmesi ve yazılım kalite faktörlerini uygun seviyede karşılamaının kurumsal güvence altına alınabilmesi için dünyada yayılan yararlılığı kanıtlanmış yazılım kalite sistemlerinin referans alınması gerekmektedir.

Küçük projeler, çalışan sayısının 5 veya daha az projenin toplam planlanan süresinin ise 6 aydan kısa olduğu projelerdir (Johnson 1996).

Kalite sistemlerini uygularken en önemli şart; sistemi uygularken zeka ve sağduyuyu kullanmaktır (Paulk 1996). Bu zeka ve sağduyuyu kalite sistemlerinin gereklerini amaç olarak değil de araç olarak kullanırken kullanmak gerekmektedir. KOBİ'ler için öncelikle gerekli süreçler tanımlanmalıdır. Her boyuttaki yazılım firmaları için şu olguların varlığı kaçınılmazdır: belgelendirilmiş müşteri gereksinimleri, müşteri iletişimi, alınmış kararlara uygun hareket etme, planlama, belgelendirilmiş süreçler, iş kırınım ağacı.

Küçük firmalar için en büyük zorluk, bu tip firmaların birincil amacının “ayakta kalmak” olmasıdır. Mevcut durumun yanlış olduğuna inanılan kurumlarda bile kaynak bulmak, süreç iyileştirmek için sorumluluk atamak ve daha sonrasında süreçleri tanımlamak ve uygulamaya almak, küçük firmalar için oldukça zordur. Küçük firmaların yanlış düşünceleri arasında şunlar yer alır:

- “Hepimiz ustayız” – dolayısıyla aramıza yeni katılan arkadaşları eğitmek için zaman ve para harcayamayız.
- “Küçük firma olduğumuz için bizde herkes herkes ile iletişim kurar”
- “Hepimiz kahramanız” – ne yapılması gerekiyorsa onu yaparız, dolayısıyla bize kurallar işlemez (sadece işi yapmaya konsantre oluruz.), kısa zamanlı ve stres altında yaşayabiliriz.

Doğal olarak, küçük firmalarda da belgelendirilmemiş gereksinimler, tecrübesiz yöneticilerin hataları, kaynak planlaması, eğitim, gözden geçirme ve ürünü belgelendirmek konusunda büyük kurumlarda olduğu gibi sorunlar yaşamaktadırlar.

Verilerle düşünmek ve karar vermek hemen hemen tüm organizasyonlar için toplam bir felsefe anlayışlarında bir değişiklik gerekmektedir. Bu kavramın başarılı olabilmesi için gerçekten faydalı ve proje için kritik olan değerlerin ölçümünü sağlamak gerekmektedir. Kolaylıkla toplanan bilen verilerin değil, proje için gerçekten kritik olan (toplaması zor olsa bile) parametrelerin toplanması gerekmektedir.

Süreçler tanımlanırken çok katı kurallar halinde tanımlanması halinde kurum içi direnç noktaları oluşacaktır. Bu yüzden süreçler tanımlanırken katı

kurallar yerine esnek yapılar (örneğin prensipler) kullanılması çalışanların yaratıcı katkılarını sürece yansıtacaktır.

Süreçleri belgelendirme nedenlerini; iletişim, süreçlerin durumunun algılanması, tutarlılığın desteklenmesi olarak sayabiliriz. Belgelendirilmiş süreçler öğrenen organizasyonlar yaratır. Buna karşın, belgelendirmenin karışık ve uzun olmasına gerek yoktur. Amacı karşılayan 1-2 sayfalık dokümanlar başlangıç için yeterli olabilmektedir. Süreçlerin tasarımını yaparken, yazılım tasarımının bilinen prensiplerini (modüller arası bağımsızlık, soyutlama vb) uygulamak süreçlerin karmaşık olmasını engelleyecektir. Belgelendirilmiş süreçlerin, herkesin kolayca ulaşabileceği bir yerde olması süreçler kadar önemlidir.

Süreçlerin üst yönetim tarafından desteklenmesi, birçok zorluğu aşmakta kurum çalışanlarına ekstra motivasyon sağlayacaktır. Proje yürütücülerinin hazırlayacakları planların hazırlanırken önemli ve öncelikli olanların göz önüne alınması gerekmektedir. Birden fazla seviyede plan hazırlanması planların etkinliğini arttırabilecektir.

Gözden geçirme ve inceleme tüm boydaki kurumlar için maliyetleri kazandırdıklarına oranla çok daha düşüktür. Bu yüzden gözden geçirme ve inceleme faaliyetlerinin desteklenmesi ve kararlılıkla uygulanmasında büyük yarar vardır (Ackerman 1989).

2.5.1 Olgunluk ve Yetenek Modeli (Capability Maturity Model) Çözümü

Olgunluk anketini kullanarak yazılım sürecini değerlendiren CMM, Amerikan Savunma Bakanlığı'nın 1970-80'lerde yaşanan yazılım krizine çözüm bulması amacıyla Carnegie Mellon Üniversitesi'nden yardım istemesi üzerine, üniversite bünyesinde kurulan Yazılım Mühendisliği Enstitüsü (Software Engineering Institute-SEI) tarafından geliştirildi. Genellikle büyük boyutlu firmalara hitap eden CMM'de, hem dışa karşı belgelendirme, hem de iç süreçlerin detaylı bir şekilde değerlendirilmesi söz konusudur.

CMM 5 seviye (1-5 arası) sunar. Parantez içerisinde her seviyeye ait anahtar süreç alanları Çizelge 2-1'de verilmiştir:

Çizelge 2-1. CMM Seviyeleri

Seviye	Odak	Anahtar Süreç Alanları
5 En iyilenen	Sürekli Süreç İyileştirme	Hata Önleme Teknoloji Değişim Yönetimi Süreç Değişim Yönetimi
4 Yönetilen	Ürün ve Süreç Kalitesi	Analitik Süreç Yönetimi Yazılım Kalite Yönetimi
3 Tanımlı	Mühendislik Süreçleri ve Kurumsal Destek	Kurumsal Süreç Tanımlama Eğitim Programları Tümleşik Yazılım Yönetimi Yazılım Ürün Mühendisliği İletişim Gözden Geçirmeler
2 Tekrarlanabilir	Proje Yönetim Süreçleri	Gereksinimlerin Yönetimi Proje Planlama Proje İzlenmesi ve Takibi Alt Yüklenici Yönetimi Kalite Güvence Konfigürasyon Yönetimi
1 Başlangıç	Kahraman Çalışanlar	—

CMM uygulaması için hiyerarşik olarak, seviye belirleme, bir sonraki seviyeye geçmeden önce eksiklikleri belirleme, eksiklikleri hiyerarşik sıraya dizme, eksikliklerin giderilmesi için plan yapma, planı hayata geçirmek için kaynak ayırma ve uygulama, döngüye yeni baştan başlama aşamaları uygulanır.

2.5.2 IEEE-12207 Kalite Sistemi Çözümü

Her süreç kendi öğelerinin aktivitesi açısından daha ileri tasarlanmıştır. Bunların her biri öge görevlerinden de daha ileri tasarlanmıştır. Süreç altındaki aktivite birtakım tutarlı görevlerdir.

Görev bir grup basit veya atomik aksiyondur. Görev girdileri tüketir (data, bilgi, kontrol) ve çıktıları üretir (data, bilgi, kontrol). Durum, gereksinim, öneri ve izin verilebilir hareket biçiminde ifade edilir. Bu amaç için, standart farklı görevin farklı biçimlerini ayırmak için bazı yardımcı eylemleri (-ecek, -acak, -malı ve -ebilmek) kullanır. “-ecek” bir durumun amacını ya da bir grubun amacını açıklamak için , “-acak” iki ya da daha çok grup arasındaki bağlayıcı koşulu açıklamak için, “-malı” diğer olasılıklar arasındaki öneriyi açıklamak için ve “-ebilmek” standardın limitleri içinde bir dizi izin verilebilir hareketi göstermek

için kullanılır. Zorunlu bir hareketi belirten “ –meli, -malı” standart içerisinde hiçbir zaman kullanılmaz. “-ecek, -acak” içermeyen başka hareketler de vardır. Bu hareketler gereksinim değıllerdir; bunlar bağlamı tamamlayan veya eklenen giriş, farz veya sözlerdir.

Standart toplam nitelik yönetiminin prensiplerini tamamlar.

Başlangıç olarak, standart Software hareket devrinin gerekli bölümlerine, nitelik ile ilgili olan tüm aktiviteleri ele alır. Böylece, başlangıçtan itibaren nitelik otomatik olarak hesaba katılır.

Bir sonraki adım olarak, hareket devrindeki sürece nitelik ile ilgili aktiviteler ayırır. Her süreç yerinden oynatılamayan devir ile “Planla-Uygula-Kontrol Et-Önlem AI” (PUKÖ) ile donatılmıştır. Her bir süreç ve personel kendilerinin başlangıç iç nitelik süreçleri ile ilgili olan değıerlendirmeler içeren aktiviteleri yapmak için sorumludurlar.

Ek olarak, iki özel süreç, kanıtlama ve onaylama daha önceden söz edilen süreç iç değıerlendirme arzu edilen derecede bağımsız ve objektif olmak için ekleme ve tamamlayıcı olarak sağılanmışlardır.

Özel bir süreç, nitelik garantisi ürünlerin ve servislerin uyumunu kendi gereksinimlerini açıkça garanti etmeye adanmıştır. Bu süreçten sorumlu olan kişilere örgütsel özgürlük ile uyumu etkilemek için otorite sağılanmışdır. Örgütsel özgürlük servis sağılayan ve ürün üreten direk yönetim özgürlüğü olan birinden özgür olmak demektir. Bu arada otorite ilgili değıerlendirmelere izin veren ve ilgili düzeltici hareketlere başlayan otoritedir. Bu süreç düzeltici hareketlere “kuvvet” için otorite vermez.

Sonuç olarak, standart yönetim, kontrol ve kurulu süreci geliştiren ve onun başarısı – bunun ötesinde sözleşmeli zorunluluk için gelişim sağılar.

Standartta “organizasyon” ve “grup” terimleri nerdeyse eşanlamlıdır. Organizasyon belirli amaçlar için organize edilen kişilerin iskeletidir; kulüp, dernek, kurum veya halk gibi. Organizasyon bütün olarak veya bir bölüm olarak antlaşmaya girdiğı zaman gruptur. Organizasyonlar ayrı iskeletlerdir fakat gruplar aynı ya da ayrı organizasyonlardan oluşabilir.

Bir organizasyon ya da grup adını sorumlu olduđu süreçten alır; örneğin, kazanç süreci yapıldığı zaman bu organizasyon kazanım olarak anılır. Bir grubun adı fonksiyonel anlamdadır ve bu organizasyon için bir yapı ima etmez.

Standart esnektir ve herhangi bir hareket devri modeliyle (Şelale, artan, evrimsel, helezon veya diğerleri), herhangi bir software mühendislik metodu (objektif yönlendirme tasarımı, yapısal kodlama, alt üst deneme ve diğerleri) ya da herhangi bir programlama dili (Ada, Assembler, Makine dili ve diğerleri) ile birlikte kullanılabilir. Bunlar software projesine ve teknolojinin durumuna çok bağlıdır ve seçimleri standardın kullanıcılarına bırakıldı.

Standart herhangi bir iş sektörü (askeri, ticari, otomobil sanayi, havayolları ve diğerleri veya herhangi bir ulusal, organize kültür tarafından uygulanabilir.

Standart yazılım sisteminin tüm hareket devrini kapsar ve hareket devrinin içerisindeki bağımsız grupları çeşitlendirmek için görevi üzerine alır. Standart, objektif hareket devrinin bölümlerini ve grupları farklı ve bazen karmaşık olarak bağdaştırır ve uzlaştırır. Sonuçta bu doğal olarak karmaşıktır. Standart organizasyonun nesnelere ve projenin bağlamı dışında okumak standardı yorumlamayı olumsuz etkiler. Standardı etkili ve üretici kılmak için şu önceden gereksinimler karşılanmalıdır:

- a) Eğitilmiş personel
- b) Organizasyonun kurallarına aşinalık
- c) Projenin çevresine aşinalık
- d) Standardı anlama

Uluslararası standart standardın temel sebeplerini içeren bir girişle başlar. Teknik materyaller 7 bölüm ve 4 eke ayrılır:

Bölüm 1-Faaliyet alanı ve uygulama sahası.

Bölüm 2-Normal referanslar

Bölüm 3-Açıklamalar

Bölüm 4-Hareket devir sürecinin en üst düzeyde gözden geçirilmesi

Bölüm 5-Beş öncelikli süreci aktiviteleri ve görevleri

Bölüm 6-Sekiz yardımcı sürecin aktiviteleri ve görevleri

Bölüm 7-Dört organizasyon sürecinin aktivite ve görevleri

Ek A-Standardı yazılım projesine uydurmak için gerekli aktiviteler ve görevler

Ek B-Uyum standartları üzerine kısa bir kılavuz

Ek C-Organizasyon süreçler ve ilişkileri üzerine genel bilgi

Ek D-Bibliyografya.

2.5.2.1 Ana Süreçler

Uluslararası standart bir periyot veya yazılım projesinin hareket devrinin kendi kavramından genişleyerek bitimine kadar olan meydana gelmiş bir dizi ilk süreci tarif eder. İlk süreçler yazılımın içinde olan kazanç, sağlama, geliştirme, operasyon ve bakım gibi anahtar gruplarına hizmet eder. Her ilk süreç öğelerinin aktiviteleri ve görevleri açısından açıklanır ve tarif edilir.

Her ilk öğe giriş ile başlar, bir dizi düzeyleşmiş aktivite onu takip eder ve software ürünlerine ve servislerine bir dizi aktivite birleşmiş görevler sağlayan düzenekle devam eder.

- **Yazılım Temin Süreci:** Bu hareket devir süreci kazançlının sözleşmeli olarak yazılım ürün ve servislerinden kazanç elde eden aktivite ve görevleri açıklar. Ürüne ve servise ihtiyacı olan organizasyon, bunun sahibi olabilir. Sahip olan kişi tüm kazanım görevlerinin etmen olan tüm kazanım görevlerine kontrat yapabilir. Kazançlı olan kişi kullanıcının ihtiyaçlarını ve gereksinimlerini gösterir.
- **Kazanım süreci:** Yazılım ürün ve servislerinin kazancı için gerekli olan tanımlama ile başlar. Süreç önerinin hazırlanması, müteahhidin seçimi ve kabul etme sistemi içerisindeki kazanım süreci yönetimi ile devam eder. Süreç kendi özel başlangıç görevleri ile birlikte şu aktiviteleri içerir: Başlangıç, öneri hazırlığı için talep, kontrat hazırlığı ve yenileme, müteahhit denetleme, kabul etme ve tamamlama. İlk 3 aktivite anlaşmadan önce son 2 aktivite anlaşmadan sonra meydana çıkar.
- **Yazılım Sağlama Süreci:** Hareket devir süreci müteahhidin aktivite ve görevlerini içerir. Bu süreç ya öneriyi hazırlamak için kazançlıya

cevap olarak talep edilen kararıyla ya da anlaşma kontrat imzalayarak kazançlıya yazılım servisi sağlayarak başlar. Bu hizmet yazılım ürünlerinin, yazılım içeren sistemlerin, yazılım sistemlerinin operasyonunu ve yazılım ürünlerinin bakımını geliştirir. Süreç tanımlama prosedürü ve yönetilmesi gereken kaynakların sağlama servisi, kazançlının plan boyunca taşıma hizmetini geliştirmesi ve yapması ile devam eder. Sağlama süreci kendi aktivitelerinin yanında şu aktiviteleri de içerir: Başlama, cevap vermeye hazırlık, kontrat, planlama, yapma ve kontrol etme, gözden geçirme ve değerlendirme, taşıma ve tamamlama. İlk 2 aktivite anlaşmadan önce son 5 aktivite ise anlaşmadan sonra meydana gelir.

- Geliştirme Süreci: Bu hareket devir süreci geliştiricinin görev aktivitelerini içerir. Geliştirme terimi hem yeni yazılımın gelişimini hem de var olan yazılımın değişimini belirtir. Gelişim süreci en az 2 yolda uygulanmak için tasarlanmıştır.
 - İlk örnekleri geliştirmek için metodbilimi veya ürünün tasarımı ve gereksinimlerinin çalışması.
 - Ürün üretmek için süreç: Bu süreç yazılım gelişiminin tüm sistemde içsel bölümün daha büyüğü olarak tek başına bir varlık olmasını sağlar.

Gelişim süreci kendi aktiviteleri dışında şu özel görevlerden oluşur: Süreç gerçekleştirme, sistem için gerekli analiz, sistem tasarımı, yazılım için gerekli analiz, yazılım mimari tasarımı, yazılım detaylı tasarım, yazılım kodlama ve denetleme, yazılım bütünleşme, yazılım nitelik denemesi, sistem tamamlama, sistem nitelik testi, yazılım donanımı ve yazılım kabul destekleme. Bu aktivitelerin durumsal düzeni, zaman düzeni göstermez. Bu aktiviteler tekrarlanabilir veya aktivite şelale düzenini ima etmek için yeniden tekrarlanabilir. Tüm bu görevler içerisinde aktivitelerin ilk veya verilen herhangi bir tekrarda tamamlanması gerekmez. Ancak bu aktiviteler son tekrara gelindiğinde tamamlanmış olmalıdır. Bu aktiviteler ve görevler bir yada daha fazla geliştirme modeli (şelale evrimsel helezoni ve diğerleri ya da bunların birleşimi) proje veya organizasyon inşa etmek için kullanılır.

Standart temel gereksinimler için önceden ürün gelişimi sırasında saptanmış tasarım ve kodlara gelişim sürecinin kontrolü içerisinde izin verir. Zamanında engellenen erken veya planlanmamış gerekli değişiklikler ve değişimin kontrol ettiği etkili temeller engellenir. Standart aynı zamanda temel içerisinde olmak için ilgilene gruplara forumlar (ortak gözden geçirme ve bilanço süreci) sağlar. Gelişim sürecinin biçim yönetim sürecini ve temel görevleri yönettiği not edilmelidir.

- Operasyon Süreci: Bu hareket devir süreci yazılım sistem operatörünün görev ve aktivitelerini içerir. Yazılım operasyonunun tüm sistem operasyon içinde tamamlanır. Bu süreç yazılım operasyonunun ve kullanıcılar için operasyonel yardımı kapsar. Bu süreç kendi aktiviteleri dışında şu aktiviteleri de içerir: Uygulama süreci, operasyonel deneme, sistem operasyonu ve kullanıcı yardımı.
- Bakım Süreci: Bakıcının görev ve aktivitelerini içerir. Bu süreç bir sistem kodlama ve bununla ilgili dokümantasyonlarda hata, eksiklik olduğunda yada gelişime ve adaptasyona ihtiyaç olduğunda çalıştırılır. Amaç bütünlük korunurken var olan sistemi değiştirmez. Ne zaman yazılım ürünün değiştirilmesi gerekirse gelişim süreci etkili koruma ile değişimleri düzgün bir şekilde tamamlar. Bu süreç sistemin kapanması ile biter. Bu süreç kendi aktivitelerinden başka şu aktiviteleri de içerir: Süreç tamamlama, sorun ve değişim analizi, değişim tamamlama, bakım kabulü/gözden geçirme, göç etme ve yazılımın yürürlükten kaldırılması.

2.5.2.2 Destek Süreçler

Bu standart bir dizi sekiz yardımcı süreç içerir. Yardımcı süreç herhangi bir diğer süreçleri içsel bölüm olarak farklı amaç ve projenin başarısı ve niteliğine katkıda bulunur. Yardımcı süreç istendiğinde kazanım, sağlama, gelişim, operasyon veya bakım süreci veya diğer yardımcı süreçler tarafından istenir.

Yardımcı süreç giriş ile başlar, birtakım birleşmiş düzey hareketleri ile takip edilebilir. Diğer hareket devir süreçlerini tutan bir dizi aktivite ve uygun görevlerle devam eder.

- **Dokümantasyon Süreci:** Bu süreç hareket devri tarafından üretilen bilginin kaydedilmesi sürecidir. Bu süreç aktiviteleri, planı, tasarımı, görünümü, dağıtımı ve bu dokümanların bakımını gerektiğinde ilgililerle; yöneticiler, mühendisler ve sistem kullanıcıları açıklar. Görevleri ile birlikte dört aktivite şunlardır: tamamlama süreci, tasarım ve geliştirme, üretim ve bakım.
- **Konfigürasyon Yönetim Süreci:** Bu süreç sistem içindeki temel yazılım öğelerini ayırt etmek, tanımlamak için, tamamları kontrol etmek ve öğeleri serbest bırakmak için, öğeleri kaydetmek ve öğelerin durumunu ve tamamlama taleplerini rapor etmek için, öğelerin doğruluk ve tamamlamalarını doğrulamak için ve öğelerin depolarını kontrol etmek ve de öğeleri idare etmek ve taşımak için görevlendirilmiştir. Bu süreç şunları içerir: tamamlama, biçim tanımlama, biçim kontrol, biçim durum hesaplaması, biçim değerlendirmesi ve serbest yönetim ile taşıma.
- **Kalite Güvence Süreci:** Bu süreç; ürün ve hizmetlerin, belirlenmiş plan ve kontratsal gereksinimlere uygunluğunun bağımsız ve objektif olarak uygunluğunun garanti altına alacak çerçeve sağlar.
- **Sınama (Verification) Süreci:** Bir ürün veya hizmetin gerekli tüm şartları karşıladığının değerlendirmesini yapma sürecidir.
- **Sağlama (Validation) Süreci:** Sağlama süreci; en son üretilen sistemin istenen amaca uygunluğunu belirleme sürecidir.
- **Gözden Geçirme Süreci:** Bu süreç; değerlendirme yapan ve değerlendirilenler arasındaki etkileşimlerin çerçevesinin hazırlandığı süreçtir.
- **Denetim Süreci:** Bu süreç; Kontratla belirlenmiş resmi denetimlerin çerçevesinin belirlemektir.
- **Problem Çözme Süreci:** Problemleri çözme ve tespit edilen problemleri gidermek için kapalı döngü mekanizmanın sağlanması sürecidir.

2.5.2.3 Organizasyonel Süreçler

Organizasyonel süreçler, projelerden çok kurumsal süreçlerin tanımlamasını yapar. Bu süreçler, diğer süreçlerin kurulmasını, izlenmesini ve geliştirilmesini sağlarlar.

- **Yönetim Süreci:** Bu süreç, yazılım yaşam döngüsünün proje yöneticisinin genel faaliyet ve görevlerini tanımlar.
- **Altyapı Süreci:** Bu süreç; tüm yaşam döngüsünü destekleyen altyapının oluşturulmasını ve idame edilmesini sağlayan faaliyetleri tanımlar.
- **Süreç İyileştirme Süreci:** Bu süreç, yazılım yaşam döngü süreçleri ölçmek, seviyesinin belirlemek, kontrol ve iyileştirmek için gerekli temel ve en üst derecedeki faaliyetleri tanımlar.
- **Eğitim Süreci:** Bu süreç; çalışan kaynaklarının teknik ve yönetsel becerilerinin geliştirilmesi için gerekli temel faaliyetleri tanımlar.

2.6. Bilgi Portalı

Öğrenen Organizasyonlar² kavramıyla beraber kurum içi bilgi paylaşımının önemi vurgulanmıştır. Bu kavramının tartışıldığı ilk yıllarda tüm işletme verilerinin depolandığı, çalışanların bilgi birikimlerini diğer çalışanlar ile paylaşabildikleri platformlar bugünkü kur-çalıştır rahatlığından oldukça uzaktaydılar.

Kurumsal bilgi portalları; kurum içi haberleşme ve bilgi paylaşımını en verimli şekilde gerçekleştirmeyi hedeflemektedir.

Bugün “Kurumsal Bilgi Portalları”nın işleyiş mantığına baktığımızda portalların ERP, CRM, Tedarik Zinciri Yönetimi gibi birebir iş süreçlerini iyileştirmeyi ve veri depolayıp raporlamayı hedefleyen uygulamaların yetersiz kaldıkları iç ve dış müşteri iletişimi arayüzünü sağladığını görürüz.

Bu arayüz son yıllarda tartışılan ve işletme verimliliğini arttırdığı kanıtlanan Toplam Kalite Yönetimi, Öğrenen Organizasyonlar gibi pek çok yönetim

² Kurum faaliyetlerinin her aşamasında bilgiyi üreten ve bunu kurum kültürü haline getiren kurum.

kavramının da işletme bünyesinde daha etkin olarak kullanılmasını sağlayacağı açıktır. Hatta ISO2000 standardı kurum içi haberleşme ve iletişimi ana maddelerinin içerisine yerleştirmiştir ve aktif ve doğru olarak kullanılması durumunda portal arayüzü bu ihtiyacı tartışmasız bir şekilde karşılayabilmektedir.

Kurumsal bilgi portallarının ana amacı çalışanların, kişiselleştirdikleri bir platformdan yetkileri dahilinde iç ve dış verilere ulaşmalarını sağlamak ve bu sayede iletişimi arttırırken iletişim maliyetlerini azaltmak, herkesin verimli çalışmasını sağlamak ve en önemlisi hızlı hareket ederek rakiplerden bir adım önde olabilmektir.

İdeal Kurumsal Bilgi Portallarının temel bileşenlerine bakacak olursak aşağıdaki 6 önemli kavramı görürüz:

- Kişiselleştirme: Kişiselleştirme portalın; iç ve dış veri kaynaklarıyla entegre edilebilmesi, kuruma özel sunum şeklinin düzenlenebilmesi (kurum logosu, sayfa düzeni...vb), kullanıcı rollerinin tanımlanması ve güvenlik politikasının kurulabilmesi demektir. Portal çalışanların hizmetine açıldıktan sonra ise her çalışan çalışma sayfasını kendi iş ihtiyaçları ile ilgili içeriklere göre düzenleyebilir.
- İçerik Dağıtımı: Kurumsal bilgiler, portala kurum tarafından belirlenen grup ve kategorilere göre yerleştirilmelidir. Oluşturulan bu bilgi kümeleri yetkilendirme ve arama süreçlerinde oldukça yararlı olacaklardır. Çalışanlar ilgi alanlarına hitap eden kategorilere üye olabilmeli ve böylece o kategorilere ait portala yerleşen her bilgiden otomatik olarak haberdar edilebilmeliler.
- Uygulama ve Veri Entegrasyonu: Çalışanlar işlerini gerçekleştirmek için gereken tüm uygulamalara ve verilere portal üzerinden erişebilmeliler. Diğer bir deyişle kullanıcılar e-mail, takvim yönetimi gibi basit masaüstü uygulamalarının yanı sıra ERP, CRM, SCM gibi kurumsal uygulamalara portal arayüzüyle ulaşabilmeliler.
- Bilgi Paylaşımı: Kurum içerisinde bilginin tüm araçlarla portal üzerinden paylaşılması sağlanır: veri yayınlama, bilgi gruplarına üyelik, tartışma grupları, mesajlaşma, web konferansı araçları gibi. Böylece kurum içerisinde dolaşan bilgi yapısal olmasa da kayıt

altına alınabilmesini sağlar. Eldeki veriler analiz edildikten sonra kurum için önemli bir bilgi haline getirilebilmektedir.

- Arama: Kurumsal bilgi portallarındaki arama mekanizmaları, kurum içinde birden fazla bilgi kümesi üzerinde, yapılandırılmış veya yapılandırılmamış veri üzerinde ve de iç ve dış kaynaklarda yer alan tüm bilgi kümeleri üzerinde arama yapmaya imkan tanır ve arama sonuçlarını kullanıcıya bütünlük bir biçimde sunabilmelidir.
- Güvenlik: Güvenlik mekanizmasının amacı, portal içerisindeki bilgiyi yetkisiz kişilerden korumak ve gelecek saldırıları önlemektir. Özellikle hem kurum çalışanlarının hem de iş ortaklarının veya müşterilerin aynı portal kapısından girerek verilere ulaştığı düşünüldüğünde bu güvenlik yapısının ne kadar sağlam olmasının gerektiği bir kez daha anlaşılır.

Portalların günümüz işletmelerinde hangi süreçlerde nasıl kullanılabileceği aşağıdaki gibi değerlendirilmiştir:

2.6.1 Proje Yönetimi

Günümüzde pek çok işletmede proje bazlı çalışmalar yürütülmektedir. Projeler genellikle farklı birimlerde yer alan çalışanlardan oluşan ekipler tarafından yürütülmektedir. Bu da veri paylaşımını güçleştirmektedir. Proje yönetimi içerisinde;

- Düzenli toplantıların yapılması,
- Toplantı yerinin ve zamanının ayarlanması,
- Toplantı notlarının düzenlenmesi ve yayınlanması,
- Ortak dosyalar üzerinde çalışılması,
- Düzenli gelişme raporlarının yönetime sunulması,
- Görevlerin dağıtılması, görev raporlarının yazılması ve takip edilmesi,
- Proje ile ilgili konular takip edilebilmesi,
- Proje ile ilgili riskler takip edilebilmesi,

gibi pek çok zaman alan süreç yer almaktadır. Kurumsal bilgi portalı sayesinde gerekli tanımlamalar yapıldığında toplantı odasının uygun olup olmamasından, ekip elemanlarının iş planına kadar her bilgi yetki verilen kişiler

tarafından paylaşımaya açılabilir, yazılan raporlar otomatik olarak ilgili kişilere iletebilir, gerekirse toplantılar sanal ortamda bile gerçekleştirilebilir. Yapılan tüm yazışmalar, tartışmalar ve hazırlanan dokümanların da kurumda güvenli bir yerde saklanması sağlanır.

Bu sayede işletme, çalışanlarını daha etkin kullanmış olur ve zaman alıcı süreçleri azalttığından daha hızlı hareket etme ve karar verme yetisine kavuşarak rekabet avantajı sağlar.

2.6.2 Doküman Yönetimi

İç yazışmalar her kurum için önemlidir. Sipariş formlarının onay mekanizmalarından, gelen tekliflerin taratılıp sisteme kaydedilmesine, kalite prosedürlerinin hazırlanıp yayınlanmasına kadar pek çok süreç doküman yönetimi içerisinde yer alır.

Bu çerçevede portal bir doküman hazırlama, onay ve yayın mekanizmasının tanımlanmasına izin verir. Gerekli doküman (sipariş formu, duyuru, haber... vb) yetki verilen kişi tarafından hazırlanır. Onay mekanizmasında tanımlanan kişi/kişiler tarafından belirlenen süreçte onaylanır ve sonra da ilgili gruba bağlı çalışanın/çalışanların kullanımına açılır.

2.6.3 Uygulamalara Arayüz

Kişiselleştirilmiş arayüz ile her kullanıcı sadece kendi işiyle ilgili çalışma sayfasını oluşturarak gerektiğinde ihtiyacı olan verilere eskisinden çok daha kısa sürede ulaşabilir.

Bu çalışmalar şu anda pek çok kullanıcı tarafından bilgisayarın masaüstüne tanıtılan kısayollar ile yapılmakta ancak çalışılan dosya sayısının fazla olması durumunda bu da pek kolay bir çözüm olmamaktadır. Ayrıca pek çok durumda her farklı uygulamaya girişinde kullanıcı, ilgili programa ait kullanıcı adı ve şifresini girmektedir. Hangi bilginin nerede olduğunun bulunması ise başlı başına bir sorundur. Kurumsal bilgi portalı sayesinde bu sorunlar ortadan kalkmakta, çalışanların daha kolay çalışabilecekleri bir çalışma arayüzü sağlanmaktadır.

3. ÖRNEK FİRMA ÖZELLİKLERİ

Bu tez çalışmasında örnek alınan firma, 1986'da, Eskişehir de büyük bir holding ve grup şirketlerinin iştiraki ve tamamı milli sermaye ile kurulmuştur. Kuruluş amacı Türk Savunma Sanayiinin milli ve kritik kategorilerinde ele alınan Elektronik Savunma Sistemleri gereksinimlerine çözüm üretmektir.

Bu anlayışla hareket eden firma, Modernizasyon ve Araştırma Geliştirme Teknoloji Projeleri olmak üzere iki temel başlık altında toplanabilen Türk Savunma Sanayi ihtiyaçlarına Sistem Geliştirme ve Teknolojik Alt Yapı Oluşturma kapsamında çözümler üretmiştir.

Firma içi süreçlerin sürekliliği, denetimi ve iyileştirilebilirliği 1998 yılından itibaren AQAP-110 NATO Kalite Standardı ile garanti altına alınmıştır.

Örnek firmanın geçmişinde yapılan donanım ağırlıklı projeler nedeniyle, firmada çalışan mühendislerin tamamı elektronik mühendisidir. Yeni iş hedefleri doğrultusunda yazılım grubunun önemi son zamanlarda giderek artmaktadır. Projelerin yazılım tarafını hazırlayan grup ise 5 kişiden oluşmaktadır.

3.1. Altyapı Özellikleri

Örnek firmanın uzun geçmişine dayanarak iyi bir altyapısı olduğu söylenebilir. Bazı altyapı özellikleri şöyledir.

- Kişi başına en az bir bilgisayar düşmektedir. Bilgisayar yan ürünlerinde (yazıcı, tarayıcı vs.) de eksikliği yoktur.
- Firma içerisinde yerel bir ağ bulunmaktadır.
- Firma içerisinde Microsoft ürünleri yaygın olarak kullanılmaktadır ve firma Microsoft "partner"idir.
- Firmanın oldukça yaygın olarak kullandığı leased line Internet hattı mevcuttur.
- Firmanın Ankara, İstanbul ve İzmir birimleri de bulunmaktadır. Merkez ofis ve diğer birimler arasında bilgi akışı telefon ve elektronik posta aracılığıyla yapılmaktadır.

3.2. Gelecek Hedefleri

Örnek firmanın stratejik hedefleri;

- Ülkenin birlik ve güvenliğinin teminatı milli bir sanayi kuruluşu olarak barışta ve savaşta TSK'nin görevlendirmesine hazır olabilmek için gerekli birikim ve kabiliyetleri bünyesinde oluşturmak.
- Müşteri gereksinimleri doğrultusunda, üstün teknolojik özellikleri esnek, taşınabilir, tümleştirilebilir maliyet-performans etkin sistem çözümlerine dönüştürmek ve müşteri odaklı yapısını koruyarak ve güçlendirerek mutlak müşteri memnuniyetini sağlamak.
- Teknolojik gelişmeleri yakından takip ederek çözümlere temel oluşturacak teknolojileri özünde birikime dönüştürmek; güçlenen yapısı ve edindiği birikim ile teknolojilere şekil veren, Türkiye'de yerleşik bir uluslar arası şirket olmak.
- Yurt içindeki mevcut kabiliyetlerden azami yararlanmaya devam etmek, yatırımını yeni ve ülkede var olmayan kabiliyetlerin kazanımı amaçlı odaklamak.
- Yurt içindeki gereksinimlere daha geniş spektrumda çözümler oluştururken, uluslararası pazarlara açılmak.
- Savunma Sanayiinde elde ettiği birikimi, uzmanlık alanlarından uzaklaşmadan, sivil alanlarda da açılımlara dönüştürerek ülke ekonomisini katkıda bulunmak.
- Çevreye ve toplumsal değerlere duyduğu saygı, ticari ahlakı, takım ruhu ve işbirliği anlayışının yanı sıra sunduğu çözümlere, yarattığı fark ve kattığı değer ile, müşterisinin, rakiplerinin ve iş ortaklarının saygın ve güvenilir dostu kimliğini korumak.
- Tüm kaynaklarını verimli kullanarak dahili süreçlerini sürekli iyileştirmek ve gelişen kalite standartlarını yakından takip ederek ekonomik ve kusursuz çözümlere ulaşmayı garanti altına almak.
- Çalışanlar arasında yaratılan katılımcı ve yaratıcı ortamı korumak, çalışanlarına tatminkar, güvenli ve üretken bir ortam sağlamak.

3.3. Sistemdeki Sorunlar

Sistemde karşılaşılan bazı sorunlar aşağıdaki gibidir.

- Kalite sistem dokümantasyonunun hazırlanması ve kullanımı
- Personelin bilgi ve bilinç düzeyinin düşüklüğü
- ISO9000 standartlarının anlaşılabilmesi ve yanlış yorumlanması
- İlave işyükü ve harcama ihtiyacının ortaya çıkması
- Üst yönetimin yanlış tutumu ve organizasyonel bozukluklar
- ISO9000 kalite güvencesi modellerinin temel şartlarının sağlanması
- Katılımcılık, motivasyon ve ekip çalışmasındaki yetersizlikler
- Değişime ve yeniliklere karşı direnç, alışkanlıklardan kopamama
- Süreç yaklaşımındaki yetersizlikler
- Belgelendirme eksiklikler

4. OLUŐTURULAN ÖRNEK MODEL

Türkiye’deki biliőim sektöründeki firmaların büyük bir çoğunluğunun KOBİ sınıfına girmesine rağmen, üretilen yazılım ürünlerinin kalitesinin yeterli olmadığı değerlendirilmektedir.

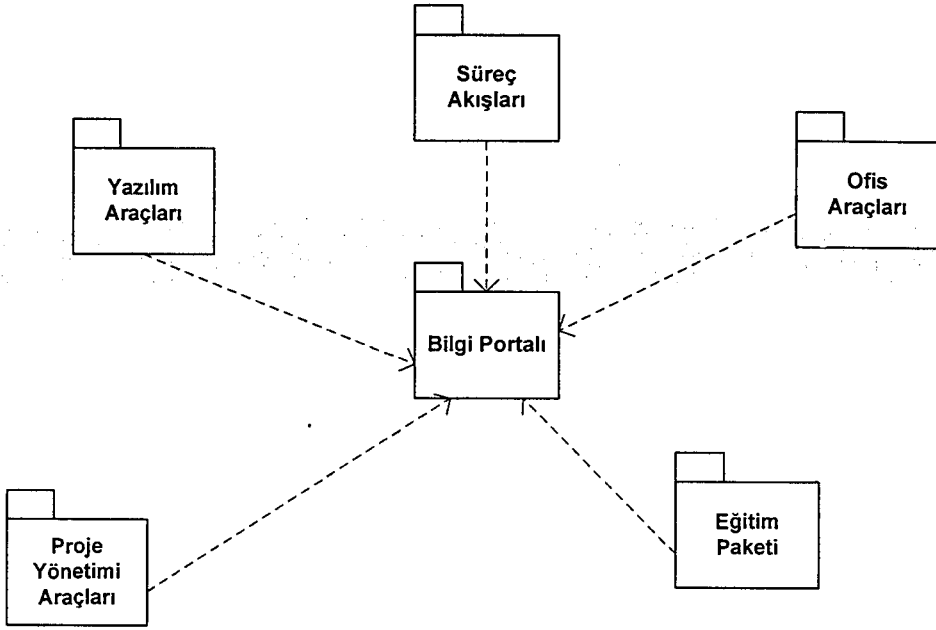
Kalder tarafından yapılan anketlerde KOBİ’ler için tespit edilen eksiklik ve sorunları, (örnek firmada da benzer bir şekilde ortaya çıkmaktadır) aőağıdaki şekilde sınıflandırılabilmektedir:

- Eğitim eksiklikleri
- Belgelendirme eksiklikleri
- Yardımcı araç kullanımının yetersizliğı ve eksikliğı
- Yazılım Geliőtirme süreçlerinin eksikliğı, belirsizliğı ve yetersizliğı

Bu çalışmada, İkinci bölümde bu eksikliklere karşılık olarak,

- Kurum içi eğitim kanallarının işlevsel hale getirilmesi ve de Internet üzerindeki bilgi kaynaklarının taranarak, kurum içi eğitimde etkin olarak kullanılabilmesi
- Uluslar arası belge şablonlarının (Örnek: MIL-STD-498) kurum içerisinde kullanımının sağlanması
- Bazı ekonomik yardımcı araçların kullanımının araştırılması ve kurum içerisinde kullanımının sağlanması
- Kurum içi bir bilgi portalının oluşturulması,
- Kurum içi etkin bir proje yönetimi için gerekli altyapının kurulması,
- Yazılım geliştirme süreçleri için ise CMMI sistemini rehber olarak kullanılması,
- Kurum içerisindeki süreçlerin anlaşılabilir ve çalışanlar tarafından savunulabilmesi için hafifleőtirilmesi ve pratik çalışma koşullarına uyumlu hale getirilmesi

çalışmaları planlanmıőtı. Buna göre; örnek firma içerisinde yapılan çalışmalar sonucunda örnek model’in çerçevesi Őekil 4-1’deki oluőtmuőtur.



Şekil 4-1. Örnek Model'in Çerçevesi (UML Dağıtım Şekli)

4.1. Eğitim Paketi

Kullanılan programlama dillerinin daha etkin kullanımı ve bu konudaki teknik eksiklerin giderilebilmesi için İnternet ortamında eğitim, teknik bilgi, makale, örnek olay veren web sitelerinin tamamı taranmıştır. Sürekli takip edilmesi gereken web siteleri belirlenmiş, bu sitelerin takip edilerek, çevrimiçi eğitimlere katılım sağlanması, makale ve örnek olayların uygulanarak zorlu bir eğitim sürecinin tamamlanması sağlanmıştır. Bu kapsamda takip edilen bazı web siteleri şöyledir:

- www.msdn.microsoft.com ve birçok alt siteleri (eğitim siteleri, teknik makaleler, öğretici sayfalar, tasarım köşeleri vs.)
- www.rsqa.com/spi (Yazılım mühendisliği üzerine bir çok alt başlık ve alt bağlantı içermekte)
- www.embedded.com (Gömülü yazılım geliştirme konularına ilişkin bir çok makale, teknik bilgi vs)
- www.uml.org (UML modelleme dili üzerinde bir çok makale ve teknik bilgi içermektedir.)

Örnek firmadaki eğitim eksiklerini gidermek, yeni oluşacak sisteme uyum sağlamak için aşağıdaki eğitimler planlanmıştır:

- Yazılımda Kalite
- Yazılım Gereksinimlerinin Yönetimi
- Yazılım Test Stratejileri
- ISO9001:2000 Eğitimleri
- Yazılım Mühendisliği
- Problem Çözme Teknikleri
- Toplam Kalite Yönetimi
- Visual Studio.Net
- XML Teknolojileri
- Proje Yönetimi
- Takım içi İletişim
- Bilgi Portallarının Kullanımı
- Nesneye Dayalı Programlama
- Tasarım Şablonları
- Yardımcı Araçların Kullanımı

Bazı eğitimler ile ilgili hazırlanan sunular EK-A'da verilmiştir.

4.2. Prosedürler ve İş Akış Süreçleri

Hazırlanan bazı iş akış süreci, prosedür ve talimatlar EK-B'de verilmiştir.

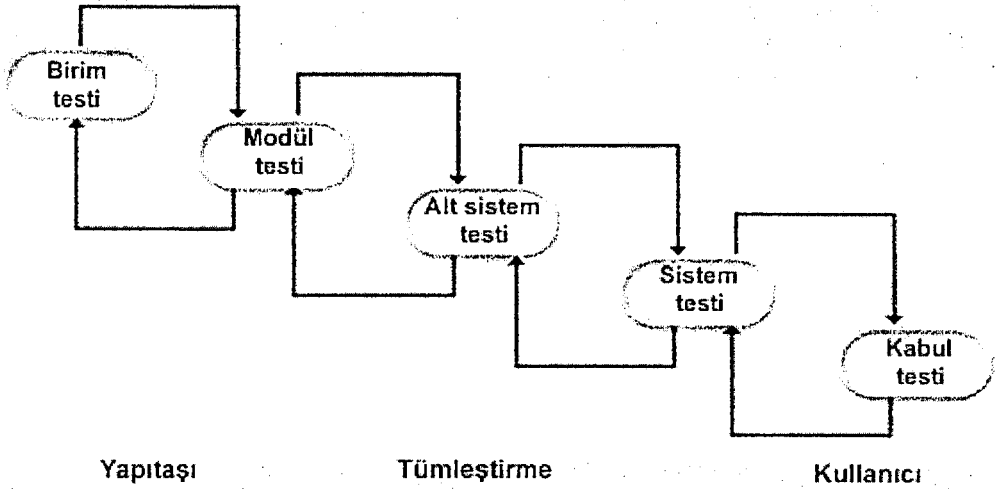
4.3. Yazılım Mühendisliği Araçları

4.3.1 Test Aracı

Test ve bütünleştirme işlemlerinin bir strateji içinde gerçekleştirilmesi, planlanması ve tekniklerin seçimi gerekmektedir. Bütünleştirme işleminde en küçük birimlerden başlanarak sistem seviyesine çıkılmaktadır. Bu değişik düzeylere hitap edecek test yöntemleri olmalıdır. Şekil 4-2'de uygulandığı hedef açısından test yöntemleri gruplandırılması gösterilmektedir. Buradaki test işlemleri:

- Birim testi: Diğer sistem unsurları olmadan, tamamen soyutlanmış olarak yapıtaşlarının doğru çalışmalarının elde edilmesi içindir

- Modül testi: Modüller, ilişkili yapı taşlarının bütünleştirilmesinden oluşurlar. Yine diğer unsurlardan bağımsız olarak sınanmalıdırlar.
- Alt-sistem testi: Alt sistemler ise modüllerin bütünleşmesi ile ortaya çıkar. Yine bağımsız olarak testleri yapılmalıdır. Bu düzeyde en çok hata arayüzlerde bulunmaktadır, arayüz hatalarına yönelik testlere yoğunlaşılmalıdır.
- Sistem testi: Üst düzeyde bileşenlerin sistem ile olan etkileşimlerinde çıkacak hatalar aranmaktadır. Ayrıca belirtilen ihtiyaçların doğru yorumlandıkları da sınanmalıdır.
- Kabul testi: Çalıştırılmadan önce sistemin son testidir. Artık yapay veri yerine gerçek veriler kullanılır. Bu test türüne alfa testi de denir. Alfa testi tanımında, testin geliştirici organizasyonun yerleşkesinde, kullanıcıların da gelecek katkıda bulunmasını içerir. Daha sonra ürünün pazarlama işlemi sırasında beta testi denilen, kullanıcının kendi mekanında yaptığı test de önemli bir uygulamadır.



Şekil 4-2. Test işlemlerinin hedefleri (Demirörs 2001)

4.3.2 REM (Requirement Management Tool)

REM deneysel bir gereksinim yönetimi aracı olup ücretsiz olarak kullanılmaktadır. Tasarım amacı yazılım geliştirme yaşam döngüsü içerisindeki gereksinim mühendisliği fazına yardımcı olmaktır.

REM' in gereksinim mühendisliği projeleri 4 adet doküman'dan oluşur

1. Müşteri odaklı gereksinim dokümanı
2. Geliştirici odaklı ister dokümanı
3. Çakışma ve sorun giderme isterleri
4. Değişiklik isteği isterleri

REM mevcut isterlerinin altında OOP baz alınarak modellenmiştir.

- Her REM nesnesinin bir içerik menüsü vardır.
- REM içinde isterler önceden tanımlı şablon ve özelleştirilebilen dil şablonları ifade edilir.
- Kullanıcı arayüzü çalışma anında seçilebilir. Şu anda İspanyolca ve İngilizce mevcuttur. (Bu çalışma kapsamında REM için Türkçe metinler oluşturulmuştur.)
- REM isterleri, bir ilişkisel veritabanında tutar böylece diğer uygulamalarda bu bilgilere SQL kullanarak erişebilir.
- REM isterler dokümanı üretirken XML ve XSLT kullanır. XSLT stil kağıtları dışardan ayarlanabilir Eğer REM 'in varsayılan WYSIWYG amaçlı HTML ayarlarının yanısıra özel stil kağıtlar oluşturulabilir.
- REM ayrıca isteklerin tekrar kullanılabilmesine olanak tanır. Herhangi bir REM dosyasını "temel" doküman olarak kullanabilirsiniz. Böylece geniş bölümleri her zaman tekrardan tanımlanması gerekmez. Diğer bir yanda bir projeden diğerine istekleri sürükleyip bırakabilir yada aynı projede istekleri kopyalanabilir.
- Kullanıcı arabirimi dış kaynaklı DLL'lerden okunur. Böylece REM kullanıcı arabiriminin dili dinamik olarak değiştirilebilir.
- Dış kaynaklı taban dokümanlar yeni bir REM projesi için gereken güncel nesnelere depolar. Farklı standartlar için farklı temel dokümanlar bulundurabilir (Örnek olarak IEEE-830 için farklı ve MIL-STD-498 için farklı dokümanlar kullanılabilir).
- Dış kaynaklı XSLT stil kağıtları sunum hakkında bilgi depolarlar. Örnek olarak REM projesindeki nesnelere gösterimi verilebilir.

XSLT stil kağıtları yazmak zor değildir, REM'in varsayılan stil kağıdı rehber olarak kullanılabilir.

4.3.3 Visio 2003

Visio 2003, çalışanların karmaşık iş süreçlerini ve sistemlerini belgelendirmesine, tasarlamasına ve geliştirmesine yardımcı olarak birlikte çalışma verimini artırmalarını ve daha etkin kararlar alabilmelerini sağlayan bir yazılım aracıdır.

Önceden tanımlanmış Microsoft SmartShapes® simgelerini sürükleme özelliğinin yanı sıra doğru şekli yerel makinede ya da Web'de bulunmasını sağlayan güçlü arama becerileriyle Visio 2003 sayesinde çizimler dokümanlara kolayca eklenebilir. Visio, uzman çalışanların, proje yöneticilerinin, satış ve pazarlama ekibinin, işletim yönetiminin, BT uzmanlarının, yazılım geliştiricilerinin, veritabanı ve Web sitesi yöneticilerinin, uygulama yöneticilerinin ve mühendislerin çizim gereksinimlerini karşılayan özel araçlar içermektedir.

- Blok çizimler: Bu çizim şablonlarıyla, uzman çalışanlarınız hiyerarşik yapıda olan, alt ve üst ağ ilişkileri içeren, basit veri akışları ve etkileşimlerinin olduğu, veri yapılarının betimlendiği kavramların, sistemlerin ve süreçlerin anlaşılmasına ve planlanmasına yardımcı olan çeşitli çizimler oluşturabilirler.
- İş Süreci: İş Süreci şemaları, kavramsal şemalar, karar ağaçları, akış şemaları, süreç ve yordam şemaları, zaman ve etkinlik çizelgeleri gibi karmaşık sistemlerin analiz edilmesi ve konuyla ilgili olarak iletişim sağlanması amacıyla kullanılan genel iş grafiklerinin oluşturulmasına yardımcı olacak şablonlar ve araçlar içeriyor. Altı Sigma kalite geliştirme kriterlerinin ve ISO 9000 belgelerinin çoğu, bu kategorideki çizimlerle gösterilebilir.
- Akış Şeması: Akış şeması şablonları, karmaşık iş süreçlerini göstermek ya da çizmek için kullanılabilir. Yukarıdan aşağıya çizimler, bilgi izleme, süreç planlaması ve yapı belirleme çizimlerinde temel akış şemaları kullanılabilir. Muhasebe

işlemleri, finans yönetimi, mali durum bilgileri, para yönetimi ve karar alma şemaları için d<fenetim çizimleri kullanılabilir. Akış şeması şablonları arasında Sebep ve Sonuç, İşlevsel, Toplam Kalite Yönetimi, İş Akışı ve Veri Akışı gibi seçenekler sunuluyor.

- Organizasyon Şeması: Kuruluş hiyerarşisindeki kişilerin, işlemlerin, işlevlerin ve etkinliklerin arasındaki ilişkiyi göstermek için organizasyon şemaları kullanılabilir.
- Yazılım: Visio 2003, Unified Modeling Language (UML) gibi birkaç yazılım çizimini destekliyor. UML, kavramların, otomatik işlemlerin, insan ve kuruluş ilişkilerinin nasıl gösterildiğini belirleyen bir gösterim sistemidir. Çeşitli insan ve program işlemlerini tanımlayarak tasarım fikirlerini tartışmak amacıyla standart gösterim kullanarak, yazılım geliştirme sürecinde tasarım ilkelerini açıkça belirtilebiliyor.
- Web şeması: Visio 2003, site haritası bağlantıları için çeşitli metin görüntüleme seçeneklerinden birini seçme becerisi gibi Web çizimi özelliklerini içeriyor. Özel şekiller kullanma ve Web sitesindeki bağlantıları otomatik olarak bulma becerilerini de içeriyor.

Paylaşılan Çalışma Alanı işlevi, Microsoft Windows® SharePoint™ Services Beta 2 sürümüyle birlikte çalışma olanaklarını destekliyor. Windows SharePoint Services çalışma alanına kaydedilen Visio çizimleri, çalışma alanından (bir açılır liste kutusundan) doğrudan Visio ile açılabilir. Bir çizim Windows SharePoint Services çalışma alanından açıldığında, Visio'da diğer dosyaları, üyeleri, görevleri ve bağlantıları da içeren tüm ilişkili bilginin görüntülediği bir Paylaşılan Çalışma Alanı görev bölmesi açılıyor.

4.3.4 Infopath 2003

Bugünün organizasyonları satış durumlarından satın alma performans değerlerine birçok bilgi kullanmakta ve yaratmaktadır. Bilgiyi toplama ve yönetme süreçlerinin bir parçası olarak Amerika'da 70 milyon çalışan veya %59'u işlerindeki sorumluluklarının bir parçası olarak düzenli bir şekilde bu bilgileri tamamlamaktadırlar.

Bugün birçok çalışan bu bilgilere kağıt yada elektronik formlarla, elektronik tablolaştırma araçları (Excel vs.) ve elektronik posta gibi verimsiz yöntemlerle ulaşmaya çalışıyor. Diğerleri de Web tabanlı veya patentli form ayarları gibi bilindik uygulamaları kullanıyor. Bilgi işçileri bazen başlıca veri giriş hatalarını ve verimsiz iş metodlarını yeniden yazmak zorunda kalabiliyorlar çünkü bilgileri uygulamalardan yada ayarlardan çıkarmak ve yeniden kullanmak güç olabilmektedir.

Bundan başka, yöneticiler genellikle bilgilendirme kararlarının yapımına elverişli değildirler çünkü ihtiyaçları olan bilgiler organizasyonların içinde başka yerlerdeki dokümanların ve veritabanlarının içinde saklıdır. Teknoloji “Extensible Markup Language” (XML) Web gibi servisler sayesinde bir sunucudan diğerine iş süreçlerinin geliştirilmesine yardımcı oluyor. Fakat başka bir gün, bilgi işçileri XML Web servisi ile direkt olarak iletişim kurabilecekleri ve ihtiyaçları olan girişim bilgilerine ulaşabilecekleri bir yol bulamıyorlar.

InfoPath³ bütün bu müşterilerin ihtiyaçlarını adreslemek için tasarlanmıştır. Bu tasarım sayesinde organizasyonların ihtiyaçları olan bilgilere ulaşmak ve bu bilgileri yeniden kullanmak daha kolay ve daha verimlidir. InfoPath XML Web servisini ilk kullanan müşteriler için kullanımı kolay bir uygulamadır. Ayrıca bu yöntem ile bilgi işçileri kendi bilgisayarlarından XML Web servislerine etkin bir şekilde ulaşma yolu buluyorlar.

Bilgi işçileri InfoPath’in onlara sağlayacağı ayarlama özelliklerinden yararlanarak dinamik formlardaki resmi ve resmi olmayan iş süreçleri bilgilerini geniş bir yelpaze ile bir araya getirme ve paylaşma imkanına sahiptir. Geleneksel form paketlerinden farklı olarak InfoPath sayesinde kişiler yüksek kalitede bilgilere rahatlıkla ulaşıyorlar. Bu kişiler ihtiyaçları olan güncel bilgileri kolaylıkla elde ediyorlar ve bu konularda sağlıklı kararlar verebiliyorlar.

Dahası, InfoPath’in temel aldığı dosya formatı XML olduğu ve herhangi kullanıcı tarafından tanımlanmış olan XML şemalarından desteklenebildiğinden dolayı; InfoPath’te toplanan veri kurumsal veritabanı ile kolayca entegre olabilmektedir. Bu demek oluyor ki InfoPath ile organizasyon içindeki herhangi

³ InfoPath, Microsoft tarafından geliştirilmiş elektronik form uygulamalarını destekleyen bir programdır.

bir bilgiye ulaşmak ve planlamak herkes için ve her yöntem için mümkün, böylece verimliliği arttırmak ve bilgileri güçlendirmek mümkün.

Microsoft Office Sisteminin yeni üyesi InfoPath üç aktiviteyi destekliyor:

- dinamik formların yaratılması,
- bu formların bütünlenmesi ve
- bu formlarda ihtiyaçları olan bilgileri sistemlere ve metotlara bağlı olarak sunma.

InfoPath; bilgi işçilerinden topladıkları bilgileri kullanmak zorunda oldukları dinamik formları sağlıklı, rahat ve kolay tasarıma sahip araçlarla kullanmalarını sağlar. Ayrıca resmi iş geliştirme metotlarını veya resmi olmayan takım projelerini de bu şekilde kullanmalarını sağlar.

InfoPath kişilere (individual) ve organizasyonlara mevcut digital formatlardan çok daha fazla bir esneklik ile daha önceden hiç görülmemiş şekilde kolay olarak uygulamalar ve süreçler arası entegrasyonunu sağlar.

4.3.5 Sharepoint Team Services

SharePoint Portal Server 2003, işletmelere kullanıcıları, takımları ve bilgileri sürekli ve düzgün şekilde bağlayan akıllı portallar geliştirme imkanı sağlar. Böylelikle insanlar iş süreçlerindeki iş süreçlerindeki konuyla ilgili bilgidan yararlanarak daha verimli bir çalışma gerçekleştirebilirler. SharePoint Portal Server 2003 birçok farklı sistemden alınan/temin edilen bilgiyi tek bir çözüm altında bütünleştirerek işletme iş çözümlerini, esnek düzenlemeleri ve yönetim araçlarıyla bezenmiş uygulama yeteneğini sağlar. Portal; insanlar, takımlar ve bilgi için toplanma, organizasyon ve arama yeteneklerini mümkün kılarak işbirliğini kolaylaştırır. Kullanıcılar, portal içeriğini, bilgi akışını kişiselleştirmek ve özgün olarak şekillendirmek suretiyle ilgili bilgiye daha çabuk ve hızlı ulaşabilirler. Organizasyonlar; kullanıcılara, onların organizasyonel rollerini, takım üyeliğini, ilgili alanlarını, güvenlik grubunu tanımlayabileceği, güncelleyebileceği bir bilgi akışı, program hedeflerler.

SharePoint Portal Server 2003; insan, bilgi ve organizasyonlar için portal sayfaları oluştururken "Sharepoint Team Services" sitelerini kullanır. Portal ayrıca organizasyon ve yönetim araçları ile sitelerinin yetkilerini genişletir ve

takımların sitelerinde yer alan bilgiyi tüm organizasyon içine yayılması imkanını sağlar.

SPPS 2003 MSOffice sistem programları, kurumsal zeka ve proje yönetim sistemleri gibi muhtelif sistemlerle, yan üreticiler ve endüstriye özgün mevcut iş uygulamalarını içeren programları tek nokta ulaşımı temin eder. Yaygın mimari yapısı, ölçeklendirebilirliği ile portal, sizin büyüme ve genişleme için organizasyon ihtiyaçlarını belirleyen esnek gelişim ve yönetim araçlarını temin eder. Bu bütünleştirici yönleri size firman kaynaklarını etkin şekilde kullanmanız için mevcut bilgiyi düzgün şekilde size aktarır. Kullanıcılar sistemlerden ve raporlardan tam zamanında ve doğru bilgiyi alıp, yeniden kullanıp, firma dokümanlarını, proje ve iş uygulamalarını çabuk bir şekilde oluşturabilirler. Bu portal Microsoft Araştırma Laboratuvarı tarafından geliştirilmiş olan ve dosya paylaşımlarının, Web Server'larının, Exchange Server'ın herkese açık olan dosyalarının, Lotus notlarının ve WSSP Servis sitelerinin aranmasını sağlayan bir arama teknolojisine de sahiptir. Bunun yanı sıra dokümanlar ve bilgiler konuya göre organize edilebilir ve konuyla alakalı tarama yapılabilir. Yeni bir bilgi eklendiğinde ya da mevcut bilgi değiştiğinde; mevcut bilgilerin daha iyi kullanılabilmesi için bir takım uyarılarda kullanıcının dikkatini çeker.

4.3.6 Microsoft Project 2003

Proje ilgili bilgiler nedeniyle ihtiyaç duyulan kurumsal istihbarat sistemi, firmanızın projelerdeki performansını değerlendirebilmesine, kaynak ve proje seçimi konusunda bilinçli kararlar alabilmesine yardımcı olmakta ve dolayısıyla firmanızın performansını artırmaktadır.

Daha etkin kurumsal kararlar almak ve planlar yapma sırasında size yardımcı olan proje ilgili firma bilgilerini araştırmak her zaman kolay olmayabilir. Konu bilgi paylaşımına, ekipler halinde çalışmaya ve kaynakların tüm organizasyonda kullanılmasına gelince firmanız, diğer pek çok firma gibi zorluklarla karşılaşabilir.

Proje verileriniz (görevler, çizelgeler, kaynaklar ve geçmiş performans istatistikleri), başarı ve kar elde etmeniz açısından son derece güçlü bir kaynak haline gelebilir. Bu verileri bulmanıza, bunlara erişim sağlamanıza ve hatta

bunları performans analizlerinizde diğer kurumsal konulara taşıyabilmenize yardımcı olacak doğru araçlar ile proje verileriniz, her düzeydeki çalışanınıza bilinçli kurumsal kararlar alma ve hedeflere ulaşmak için harekete geçme konusunda bir bakış açısı sağlayacaktır.

Kurum performansını her yönüyle gözlemleyebilmek için önceliklerinizi değerlendirmenize ve sıraya koymanıza yardımcı olan Microsoft teknolojilerinden faydalanabilirsiniz. Birlikte çalışma ve iletişim kurma imkanı sağlayan bir işletim ortamı kullanarak proje ile ilgili bilgilerinizin organizasyonun bünyesindeki her kişiye açık olmasını sağlayıp, kurumsal faydalarınızın katlanarak çoğalmasını mümkün kılınmaktadır.

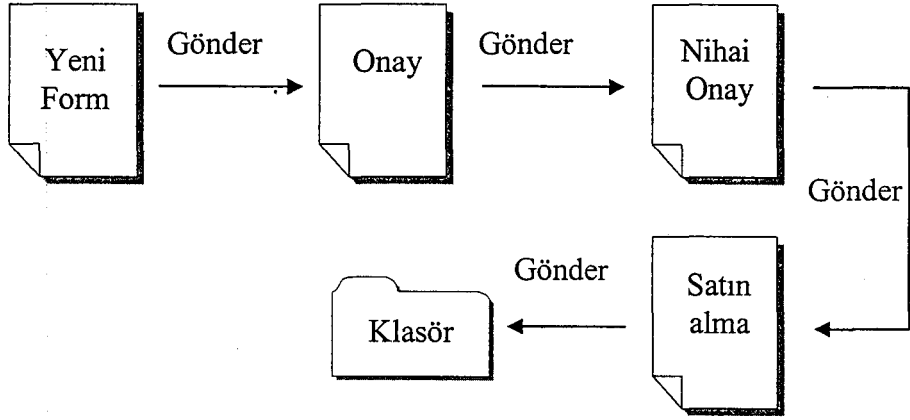
Microsoft teknolojisi esasına dayanarak oluşturulan proje istihbarat çözümü, gerekli bilgilerin organizasyonunun bünyesindeki her kişiye ulaşmasını sağlamaktadır.

- Sürekli değişim ortamına karşı kurumsal öncelikleri belirlemesi gereken bir yönetici;firma çapında, üst düzey bir performans ve kaynak kullanımı değerlendirmesi yapabilir ya da sadece belli başlı projeleri mercek altına tutabilir.
- Yöneticiler, projenin ne şekilde ilerlediğini ve kaynakların en iyi nasıl kullanılabileceğini değerlendirme için kolay anlaşılır, “öncelik sırasına göre renklendirilmiş” görüntüler şeklindeki (kırmızı, sarı ve yeşil renkte konu başlıkları sıralaması ile) proje verisi sunumlarından faydalanabilirler.
- Bir proje yöneticisi yürürlükteki projelerle ilgili bir bakış açısı kazanarak gerekli şekilde harekete geçebilir.
- Birden fazla projeyi gidişat, kaynak veya proje düzeyinde araştırmak ve projeler hakkında veri elde etmek üzere Microsoft Project Professional ve Microsoft Project Server’i kullanılabilir.

4.4. Örnek Uygulama

Bu örnek uygulamada şirket içerisindeki satınalma departmanı ele alınmıştır. Bu departmana satınalma taleplerini bildirmek isteyen çalışanların doldurması için “Satınalma Talep Formu” oluşturuldu. Bu form satınalma

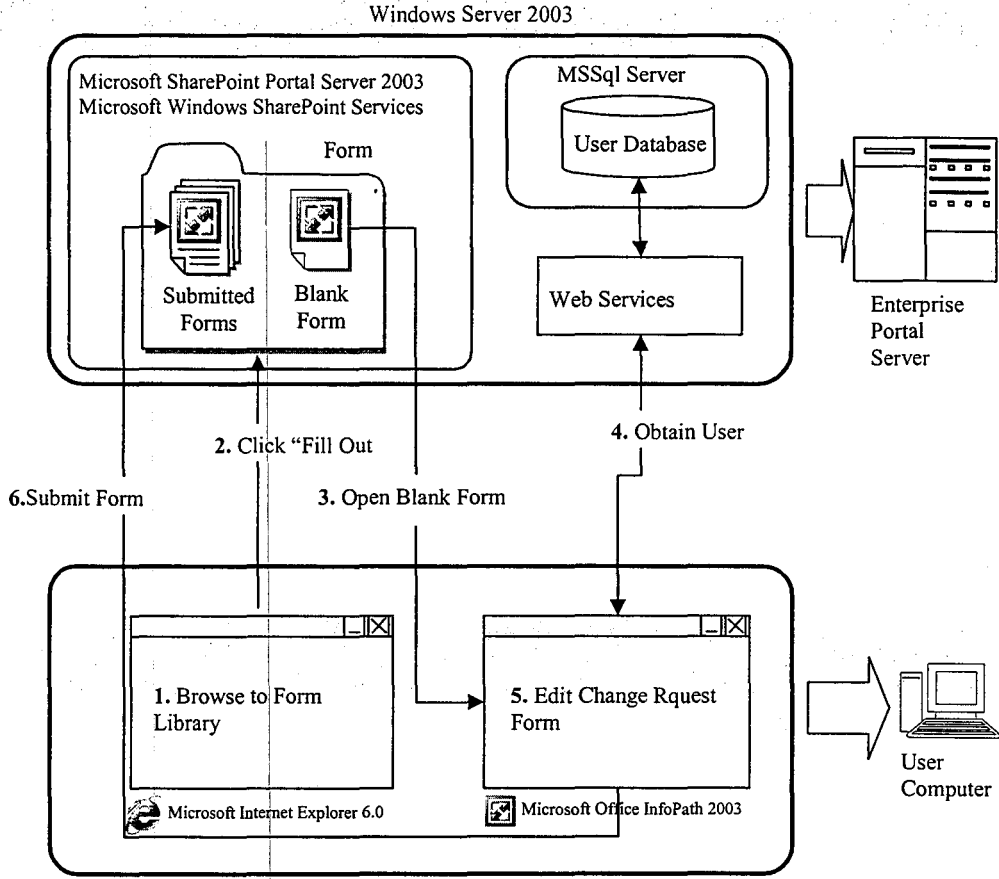
departmanı tarafından işleme konulmadan önce bazı onaylar gerekmektedir. Aksi halde işleme koyulmayacaktır. Bu onay operasyonunu daha ayrıntılı açıklarsak (Şekil 4-3), önce yeni bir formun doldurulması ve gönderilmesi gerekir daha sonra onaylayacak ve nihai onay verecek yöneticiler tarafından onaylanarak satınalma departmanı tarafından işleme konulması sağlanır. Satınalma departmanı daha sonra talebi karşılar ve formu tamamlayarak kaydeder.



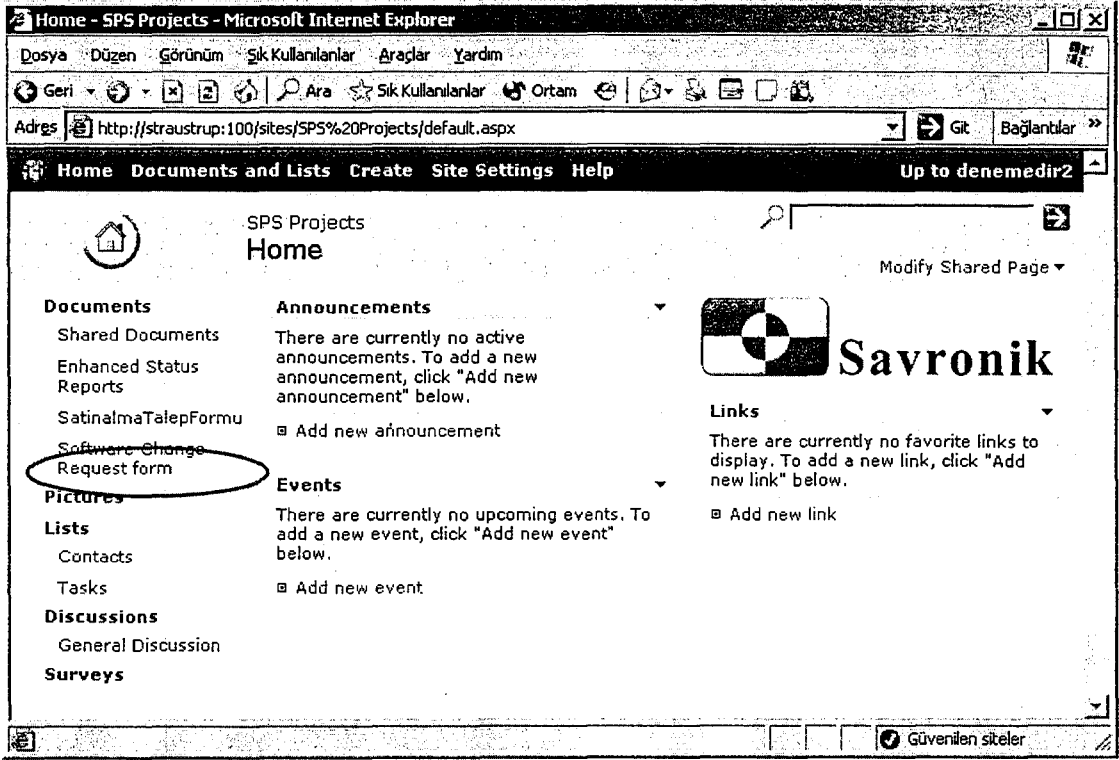
Şekil 4-3. Senaryo

Şekil 4-4'te bir kullanıcının şirketin SharePoint Portal Server sayfasına girip bir formu doldurup kaydetmesi işlemi ayrıntılı olarak anlatılmaktadır. Öncelikle form doldurmak isteyen çalışan şirketin bu tür formlar için ayırdığı sayfaya girmesi gerekmektedir (Bkz. Şekil 4-5). Bu sayfa SharePoint Portal Server 2003 tarafından oluşturulur ve belirli iş akışları, duyurular ve görevler gibi bu sayfadan takip edilebilir. Daha sonra bu sayfadan doldurmak istediği formun bağlantısını takip ederek doldurmak istediği formun sayfasına gider (Bkz. Şekil 4-6). Bu sayfa SharePoint Servisleri tarafından şablon formun portalda yayınlama işlemiyle birlikte oluşturulur. Bu sistemde, kullanıcı adı sistem tarafından otomatik olarak algılandığından kullanıcının ayrıca kullanıcı adı ve şifre girmesi gerekmez. Bu sayfada formlar onay durumuna yada diğer özelliklerine göre filtre edilerek farklı görünümde görüntülenmesi sağlanır. Kullanıcı önceden doldurulmuş formları eğer izin verilmişse görebilir ve kendiyile ilişkili formları görüntüleyebilir. Bu sayfada ayrıca doldurulmuş formların bazı bilgileri sıralanmış olarak görülebilir. Bu bilgilere göre yeniden sıralama değiştirilebilir yada formlar eğer üyenin yetkisi varsa silinebilir. Eğer kullanıcı yeni bir form

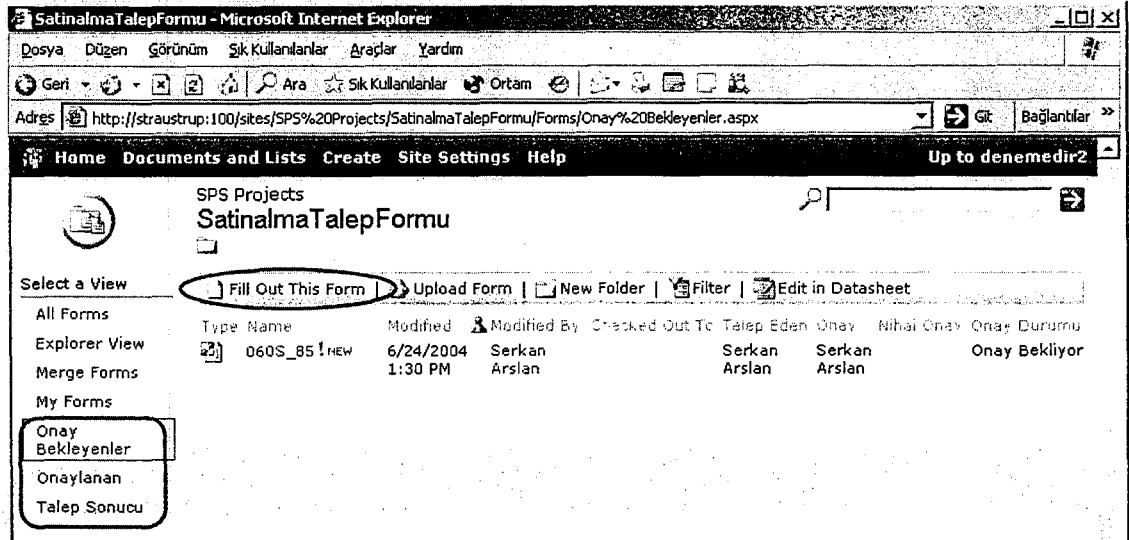
doldurup sisteme kaydetmek isterse “Fill Out This Form” bağlantısını seçerek boş bir formun açılmasını sağlar



Şekil 4-4. Ayrıntılı form otomasyonu



Şekil 4-5. SharePoint Portal sayfası



Şekil 4-6. Satınalma talep formu sayfası

Form, açıldığında (Bkz. Şekil 4-7) ilk önce ikincil veri kaynaklarına bağlanarak gerekli bilgileri edinir. Bu formda ürün (web) servislerinden kullanıcının adı, soyadı, e-posta adresi, formun numarası ve onay verebilecek çalışanların listesi alınır. Portal sunucusu Active Directory ile bağlantılı olarak

tüm kullanıcılarla ilgili tüm bilgileri adı, soyadı, e-posta adresi, yöneticisinin ismi, çalıştığı departman ve benzer bilgileri saklar. Örün servisleri bu bilgilerden gerekli olanları alarak Infopath yada başka bir programa döndürebilir. Bunun için örün servislere gerekli olan tek şey kullanıcı adıdır ve bu bilgi sistem tarafından sağlanır. Kullanıcı sayfaya girdiğinde tanınır. Infopath açıldığında örün servisleri ile bağlantı kurar ve kullanıcın adını, soyadını, e-posta adresini form numarasını ve onay verecek çalışanların listesini alır ve formda gerekli boşluklara yerleştirir. Bu listenin amacı onay vermesini istediği kullanıcıyı seçebilmesini ve hata yapmasını engellemektir.

Form numarası her form için özeldir ve sıralıdır. Bu işlemi manuel olarak doğru bir şekilde sağlamak güçtür. Bunu engellemek için de örün servisleri kullanılmıştır. Önce SQL sunucusu da form numarasını tutan bir veritabanı oluşturuldu. Örün servisleri bu veritabanı ile bağlantı kurarak form numarasını alır ve gerekli işlemleri yaparak Infopath'e döndürür. Infopath boş olarak açıldığında örün servisinden gelen numarayı otomatik olarak gerekli yere yerleştirir.

Şekil 4-7. Satınalma talep formu

Form da gerekli yerleri doldurduktan sonra “Gönder” butonuna bastığımızda form sunucuya kaydolur (Bkz. Şekil 4-8) ve onay için bekler. Form listeye eklendiğinde ayrıntılı bilgilerinde kim tarafından gönderildiği ve onaylaması gereken kişinin isimleri de gözükür. Böylece formu kimin onaylaması gerektiği bilinir. Formun işleme konulması için onay listesinden seçilen kişi

tarafından onaylanması gerekmektedir. Bu form gönderildikten sonra başka biri tarafından görüntülenemez. Form ikinci kez açıldığında (Bkz. Şekil 4-9) ilk kez açılan formdan daha farklı bir görünüme sahiptir. Bu Infopath'in içinde değişik görüntüler saklayabilmesi özelliğiyle sağlanmıştır. Doldurulmuş bir form açılırken formun kimin tarafından doldurulmuş olduğuna, onaylaması gereken kişi tarafından açılıp açılmadığına ve onay seviyesine bakarak hangi görüntüyle açması gerektiğine karar verir. Form tekrar açıldığında onay beklediği ve açan kişinin onaylaması gereken kişi olduğu için Şekil 4-9'deki görüntüyle açılır. Bundan sonraki işlemde bir öncekilerine benzer. Kullanıcı sadece onay verir ve gerekirse değerleri değiştirebilir. Bu formda da bir sonraki onaylanacak kişi seçilir ve gönderilir.

Type Name	Modified	Modified By	Checked Out To	Talep Eden	Onay	Nihai Onay	Onay Durumu
060S_85	6/24/2004 1:30 PM	Serkan Arslan		Serkan Arslan	Serkan Arslan		Onay Bekliyor
060S_86	6/25/2004 12:52 PM	Administrator		Administrator	Serkan Arslan		Onay Bekliyor

Şekil 4-8. Yeni eklenen formun form sayfasında görüntüsü

Talep Eden: Serkan Arslan	Onay : Serkan Arslan	Nihai Onay: Serkan Arslan
		Serkan Arslan
Kapat		Onay

Şekil 4-9. Satınalma talep formu onay görüntüsü

Form nihai onay'ı verecek üye tarafından açıldığında formun alt kısmı Şekil 4-10'daki gibi gözükür. Bu nihai onaydan sonra form portalın "Onaylandı" sayfası altında gözükür ve bundan sonra işleme konur. Form bundan sonra sadece gerekli departman tarafından açılabilir.

Talep Eden: Administrator	Onay : Serkan Arslan	Nihai Onay: Serkan Arslan
Kapat		Onay

Şekil 4-10. Satınalma talep formu nihai onay görüntüsü

Satınalma talebinde bulunanın talepleri karşılandıktan sonra, form tekrar açılabilir. Daha farklı bir görüntüye sahip formda (Şekil 4-11) ayrıca bir kaç yeni sütun bulunur. Bu sütunlara gerekli bilgiler girilir ve onay yapılır. Bu onayı yapacak kişi belli ve tek olduğundan ayrıca bir önceki kullanıcı tarafından seçilmesi gerekmez.

Sıra No	Parça No	Malzemenin Tanımı	Parçanın No	Üretici / Firma	Birim	İstenecek Miktar	Alınan Miktar	Fatura Tarihi	Fatura No	Birim Fiyat
1	1	ABCDEFGF	12	Savronik	Adet	10				
2	7	XYZWN	30	Microsoft	Adet	5				

Şekil 4-11. Satınalma talep formu talep sonucu görüntüsü

Her Infopath formu kendi içinde belirli scriptleri çalıştırabilir. Infopath Jscript ve VBScript olmak üzere iki script dili desteği vermektedir. Bu çalışmada form içinde Jscript kullanıldı.

5. OLUŞTURULAN MODELİN DEĞERLENDİRİLMESİ VE SONUÇ

Bu çalışmada, öncelikle kalite konusunun önemi vurgulanmıştır. Kalite'nin iş hedefleriyle ayrılmaz bir bütün olduğu dolayısıyla kalite sistemlerinin amaç değil araç olması gerektiği üzerinde durulmuştur. Buradan hareketle, kalite sistemlerinin bilindik sıkıcı ve bürokratik sorunlarının yerine sağduyunun tüm süreçlere hakim olması gerekmektedir. Kalite sistemlerinin kurum kültürüne katılması konusunda en önemli şart; üst yönetimin bu işe inanmasıdır. Üst yönetiminin liderliğinde kurum içerisindeki, bu çalışmada değinilen birçok olumsuzluğun giderilmesi kolaylaşmaktadır. Özellikle kurum içerisinde bilgi ve motivasyon akışını sağlayacak bilgi portalları aynı zamanda kalite sistem gereksinimlerinin karşılanmasını da sağlayabilecektir.

Bu çalışma sonucunda, yapılan araştırmalar sonucunda kalite sistemlerinin, Türkiye'deki bilişim sektörünün temel vazgeçilmezleri arasında olması gerektiği vurgulandı. Ancak ne yazık ki, bilişim sektörünün hiç de istenilen noktada olmadığı görüldü. Bu tez kapsamında oluşturulan bu örnek model'in ve kurumsal portal yapısının yaşanan sorunların çözümünde önemli oranda katkı yapacağı öngörülmektedir. Kurumlar içerisinde kalite sisteminin özellikle kalite formlarının otomasyonu ve süreçlerin yetkinliğini ve performansını ölçmek amacıyla önemli altyapı sağladığı değerlendirildi. Kurum içerisinde gerçekleşen hemen hemen tüm faaliyetlere tek noktadan gözlem sağlamak önceden kestirilemeyecek, üst yöneticinin hayalleri oranda katkı yapabileceği değerlendirildi.

Bu çalışmada, oluşturulan portal yapısının sadece kurumsal portal için aşağıdaki alanlarda da kullanılabilmesi değerlendirildi.

- kurumlar arası konsorsiyumların çalışma sahası
- üniversitelerin ders portalları
- sivil toplum örgütü portalları
- arkadaş gruplarının portalları

Bu çalışmanın ötesinde, kalite süreçlerinin etkinlik ve verimliliğini ölçebilecek, yardımcı araçlar ile tam entegrasyonu sağlayacak, Microsoft Office ürün ailesi ile tam bir entegrasyon sağlanması durumunda, bu modelden elde edilecek verimin çok daha büyük boyutlara arttırılabileceği düşünülmektedir.

Son olarak, kalite çağdaş dünyanın belki en vazgeçilmez olgularındandır. Bu çağdaş dünyada yer almak için kalite sistemlerinin tüm sektörlerde yerleştirilmesi planlanmalı gereği şekilde kullanılmalı yaygınlaştırılmalı ve iyileştirilmelidir. Tüm bunların yapıldığı takdirde, Türkiye'nin dünyada hakkettiği yere ulaşacağı değerlendirilmektedir.

KAYNAKLAR

ACKERMAN A.F, BUCHWALD L.S., ve LEWSKI F.H., "Software Inspections: An Effective Verification Process," IEEE Software, 6, 3, 31-36 (1989).

BRUEGGE B. ve DUTOIT A., *Object-Oriented Software Engineering Conquering Complex and Changing Systems*, Prentice Hall (2000).

DEMİRÖRS O., TANRIÖVER Ö. ve HOŞVER C., *Yüksek Lisans Programlarının Yazılım Mühendisliği Kapsamı*, Enformatik Enstitüsü, Ortadoğu Teknik Üniversitesi, Ankara, (2002).

HILBURN B., HIRMANPOUR I., KHAJENOORI S., TURNER R. ve QUASEM A., 'A Software Engineering Body of Knowledge Version 1.0,' SEI (1999).

HUMPHREY W. *The Process Bureaucracy*, Carnegie Mellon University, Software Engineering Institute, Web Adresi: <http://www.sei.cmu.edu/publications/articles/watts-humphrey/process-bureaucracy.html> (2004).

IEEE, *Information Technology, Software Life Cycle Processes, International Standard Technical ISO/IEC 12207*, (1995).

JOHNSON D. L. ve BRODMAN J.G., *The LOGOS Tailored Version of the CMM for Small Businesses, Small Organizations, and Small Projects*, Version 1.0 (1996).

ÖZKAN M. *Yazılımda Kalite*, Computer Life Dergisi 53. sayı, İstanbul (2001).

MPM, *Kalite Yönetiminde Karşılaşılan Sorunlar*, Milli Prodüktivite Merkezi, Anahtar Dergisi 9/108, 4, (2000).

PAULK Mark C., "Effective CMM-Based Process Improvement," Proceedings of the 6th International Conference on Software Quality, Ottawa, Canada, 226-237 (1996).

PAULK M., CURTIS B., CHRISSIS M. ve WEBER C., *Capability Maturity Model for Software*, Version 1.1, Software Engineering Institute, Carnegie Mellon University, CMU/SEI-93-TR-24 (1993).

PRESSMAN R., *Software Engineering, A Practitioner's Approach, 5th Edition*, McGraw-Hill Int. Edition (2001).

SANDERS D. PIERRE B., ROBERT D., ALAIN A., JAMES W., LEONARD T. ve DENNIS F., '*Guide to Software Engineering Body of Knowledge*', <http://www.swebok.org/>, A Stone Man Version (1999).

SANDERS J., JOHNSON R. ve SCOTT C., *Çev. ISO9000 Nedir? Niçin? Nasıl?*, Rota Yayın Yapım (1994).

ŞAHİN Ü., *Toplam Kalite Yönetimi ve Türkiye*, İnsan Kaynakları Dergisi, 6, İstanbul (2003)

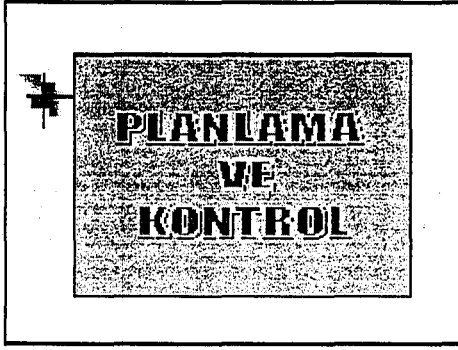
EK-A EĞİTİM PAKETİ



TAKIMDA İLETİŞİM SORUNLARI	
AŞLAYICI Küçük düşürmeyi hedefler.	■ "Böyle hayalci öneriler tamda sana uygun yani..."
KAPAYICI Gerilimi yükseltmeyi hedefler.	■ "Hermalde dalga geçiyorsun" ■ "Bu konuşmaların hiçbir anlamı yok..." ■ "Senin yerinde olmak istemezdim..."

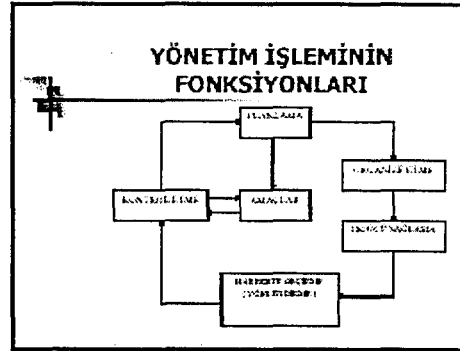
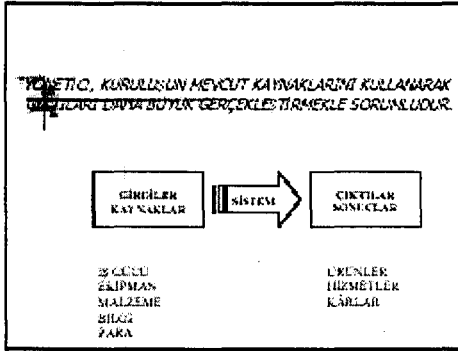
TAKIMDA İLETİŞİM SORUNLARI	
SINIK Sorumluluktan kaçmayı hedefler.	"Bunları daha öncece gördük. Anlat anlat heyecanlı oluyor.
KUSUR ARAYICI Üstünlük kurmayı Hedefler.	"Öyle demek istemedin herhalde" "Ha... Böyle diyeceksin ama kazın ayağı çy'e değil..."

Şekil A-1. Takımda İletişim Sunusu 1



PLANLAMA 5 N + 1 K

Neyi, Neden, Nerede, Nasıl, Ne Zaman ve Kim tarafından yapılacağına karar verme sürecine denir



LİDER

İnsanları peşinden koşturan, istenenleri kendi rızasıyla yaptıran ve doğru iş yapan insandır

LİDER

Tarzı durum belirler. Tarzımızın iyi olup olmadığının sonuca bakarak anlarız. Herkes geminin iskeleye yanaşmış ya da yanaşmadığına bakar

Şekil A-2. Planlama ve Kontrol Sunusu 1

YÖNETİM TARZI

1. Demokratik tarz : katılımcı yönetim tarzı
2. Despot- otokrat: baskı ve korku ile yönetim tarzı
3. Liberal (esnek): riske girmeyen, kültürünü korumaya çalışan eyyamcı tarz

ÇALIŞMA HAYATINDAKİ İNSAN GRUPLARI

1. Aşırı duygusal (melankolik) tipler
2. Ağır kanlı tipler
3. Öfkeli ve sinirli tipler
4. Canlı tipler

KİŞİLİK : Fiziksel+ zihinsel

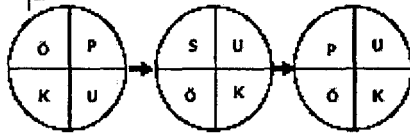
AİLE : Soyçekim, irsiyet, kalıbm, kan bağı, sütale(armut Dabine düşer)

ÇEVRE : Din,dil,ırk,okul,tabaya koşulları iş ve mahalle

PLANLAMANIN GETİRDİKLERİ

- Vakit kazandırır
- Düzenli iş akışını sağlar
- Disiplin sağlayarak karmaşayı önler
- Verimliliği yükselir
- Üretimdeki aksaklıkları giderir
- Çalışanlara sorumluluk bilinci kazandırır

PLANLAMADA PUKÖ DÖNGÜSÜ



Planla
Uygula
Kontrol et
Önlem al

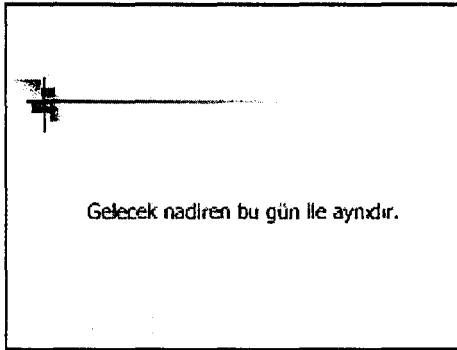
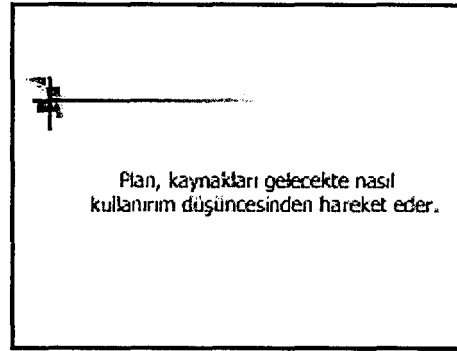
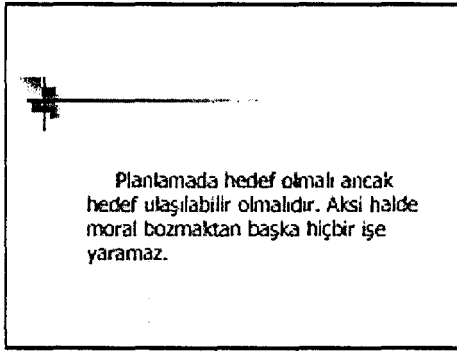
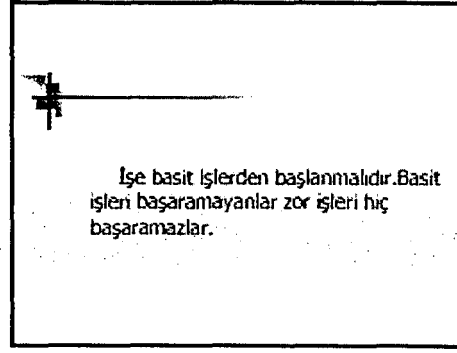
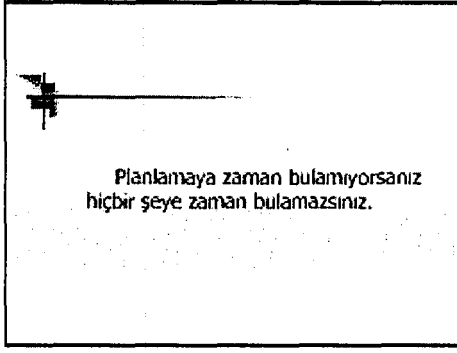
Standardlaştır
Uygula
Kontrol et
Önlem al

Planla Uygula
Kontrol et
Önlem al

PLANLAMANIN ADIMLARI

- Hedef
- Yöntem (metot)
- Koordinasyon

Şekil A-3. Planlama ve Kontrol Sunusu 2



Şekil A-4. Planlama ve Kontrol Sunusu 3

PLANLAMA

Değişimle baş etmenin denenmiş bir yöntemidir.

PLANLAMADA AMAÇ

Varmak istediğimiz yerlerin tespiti.

Sonuç kendiliğinden meydana gelmez. Nedenleri vardır

AMAÇLARI BELİRLEME AŞAMASINDAN SONRA PLANLAMA İŞLEMİ İLE NE YAPILMALIDIR?

- 1- Ana plan geliştirdiniz, bu plan ana hedef doğrultusunda olmalıdır.
- 2- Ana hedefe katkıda bulunacak destekleyici planlar yapınız.

AMAÇLARI BELİRLEME AŞAMASINDAN SONRA PALANLAMA İŞLEMİ İLE NE YAPILMALIDIR

- 3- Olabilecek her şeye sayı ve tarih koyunuz, çalışanlar kendilerinden neyin ne kadar istendiğini bilirlerse plan o kadar iyi yürür
- 4- Çalışanlara planın yürütümünü yüklenecek sorumluluk dağılımı yapınız.

AMAÇLARI BELİRLEME AŞAMASINDAN SONRA PALANLAMA İŞLEMİ İLE NE YAPILMALIDIR

- 5- Planları çalışanlar ile paylaşınız, plana açıklık getirerek bilgi vererek vs
- 6- planınızı değişime açık tutunuz.

Şekil A-5. Planlama ve Kontrol Sunusu 4

UYGULAMA SÜRELERİNE VE AMAÇLARINA GÖRE PLANLAR

- 1- **UZUN VADELİ PLANLAR:**
- 2- **KISA VADELİ PLANLAR:**

UYGULAMA SÜRELERİNE VE AMAÇLARINA GÖRE PLANLAR

- 3- **KALICI PLANLAR:**
- 4- **BİR KEREYE MAHSUS PLANLAR:**

ZAMAN ÇOĞALTILABİLİR Mİ?

Çoğaltılamaz ancak planlama yaparak çoğaltılabilir.

ZAMAN

- Ünlversal bir kaynaktır.
- Yeri doldurulamaz
- Geniştirilemez
- Bekletilip daha sonra kullanılmaz

ZAMAN KULLANIMI

- Zamanı çoğaltmanın yolu planlama yapmaktır
- Zamanı kontrol etmelisiniz
- Plan yapmazsan hiçbir zaman zamanın olmaz
- Zamanı bir bütçe malzemesi gibi kullanmalıyız

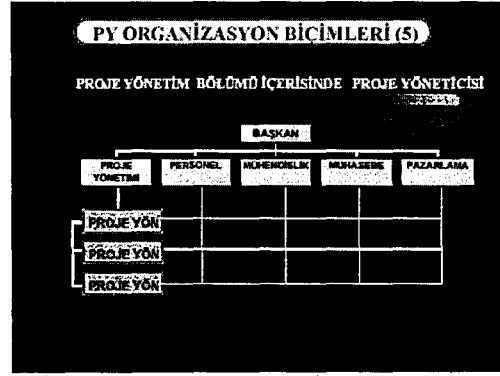
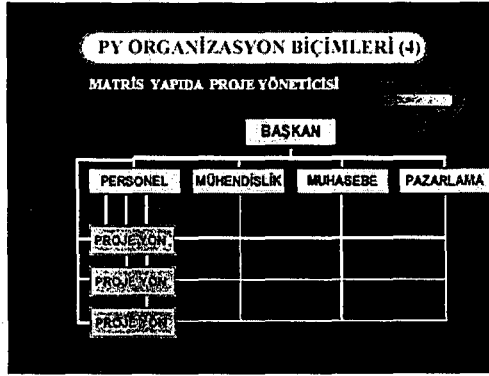
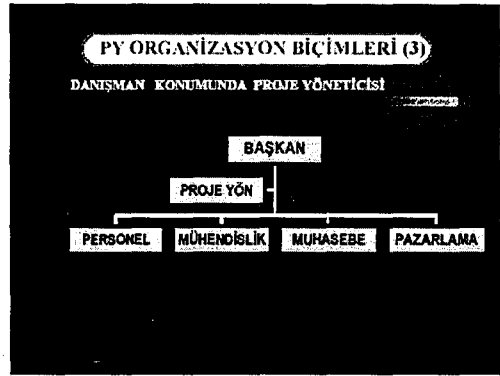
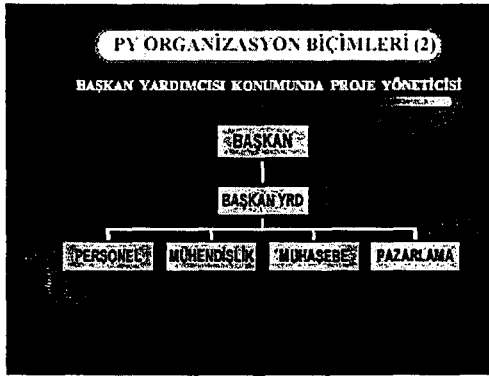
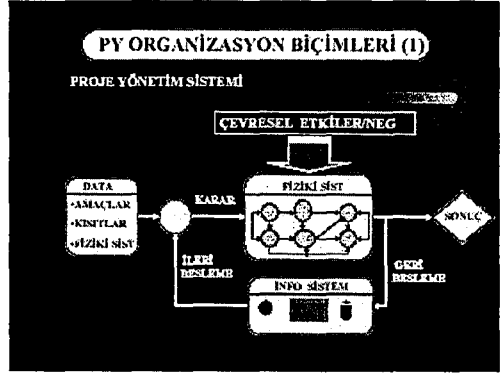
BEY-NOKTA PLANLAMA KONTROL ŞEMASI

NE? NEREDE? NE ZAMAN? NEREDE? ADI? NEREDE?	HİÇBİR YER GİÇEN ZAMAN TAKTİKLERİ KURALLIK MÜKEMMELİK	YANILAR KARAR VERİLEBİLİR BİRLİKTE TESTİN YERİ BAŞLAMA TARİHİ BİRİN TARİHİ HİÇBİR YER GİÇEN ZAMAN TAKTİKLERİ KURALLIK MÜKEMMELİK YERİNE KONTROL GÖRÜLEBİLİR
NE? NEREDE? NE ZAMAN? NEREDE? ADI? NEREDE?	HİÇBİR YER GİÇEN ZAMAN TAKTİKLERİ KURALLIK MÜKEMMELİK	YANILAR KARAR VERİLEBİLİR BİRLİKTE TESTİN YERİ BAŞLAMA TARİHİ BİRİN TARİHİ HİÇBİR YER GİÇEN ZAMAN TAKTİKLERİ KURALLIK MÜKEMMELİK YERİNE KONTROL GÖRÜLEBİLİR

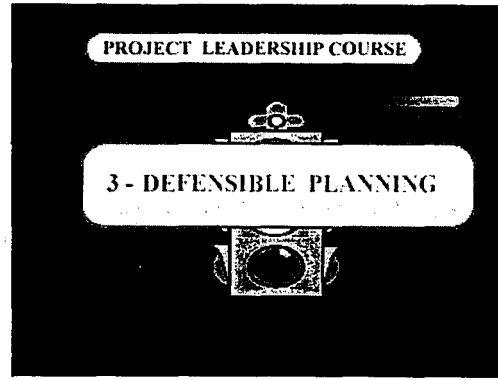
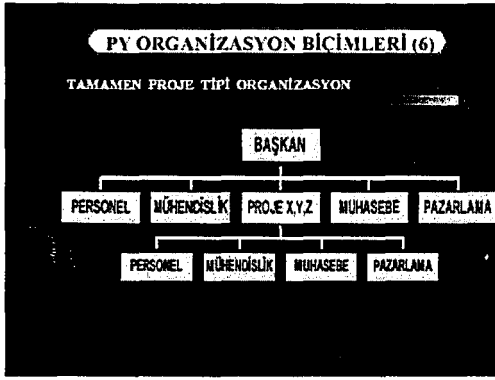
Şekil A-6. Planlama ve Kontrol Sunusu 5

PROJECT LEADERSHIP COURSE

2 - PY ORGANİZASYON BİÇİMLERİ



Şekil A-8. Proje Yönetimi 1



DEFENSIBLE PLANNING (1)

PLANLAMA NEDİR?

- ☐ YÖNETİM TEMELİDİR.
- ☐ YÖNETİM İŞLEVİNİN BAŞLANGICIDIR.
- ☐ BİR İLETİŞİM ARACIDIR.
- ☐ BİR YOL HARİTASIDIR.
- ☐ BİR TAHHÜTTÜR.
- ☐ NE?, NE ZAMAN?, NASIL?, KİM? KARARLARIDIR.

DEFENSIBLE PLANNING (2)

SAVUNULABİLİR BİR PLAN NASIL OLMALIDIR ?

- ☑ KAPSAMLI, SİSTEMATİK VE ANLAŞILABİLİR OLMALIDIR.
- ☑ KABUL EDİLEBİLİR SAVUNULABİLİR OLMALIDIR.
- ☑ ESNEK/ÇALIŞILABİLİR OLMALIDIR.
- ☑ AÇIK/ERİŞİLEBİLİR OLMALIDIR.

DEFENSIBLE PLANNING (3)

PLANLAMADA DÖRT ADIM

- ↓ AMAÇ/AMAÇLARI BELİRLEME
- ↓ MEVCUT DURUM ANALİZİ
- ↓ POSİTİF VE NEGATİF FAKTÖRLER
- ↓ PLAN OLUŞTURMA

DEFENSIBLE PLANNING (4)

PROJE PLANLAMA BİLEŞENLERİ

- ↗ PROJECT DEFINATION
- ↗ SCHEDULE
- ↗ RESOURCES
- ↗ RISK MANAGEMENT
- ↗ QUALITY
- ↗ BUDGET
- ↗ CONTROL

Şekil A-9. Proje Yönetimi 2

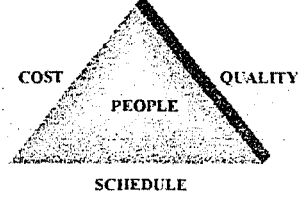
DEFENSIBLE PLANNING (5)

PROJENİN TANIMI(PD) - KAPSAMI

- κ PROJENİN AMAÇ VE İHTİYAÇLARINI BELİRLEME
 - κ İLGİLİLERLE KOORDİNE VE MUTABAKAT
- κ KABULLER VE SINIRLAR/KISITLARIN NET TANIMI
 - κ ENİ SEVİYELİ İHTİYAÇ VE SPEŞİFİKASYONLARI BELİRLEME
 - κ PROJENİN ETKİLEYECEK VARSAYIM VE KISITLARIN TANIMLANMASI
 - κ PROJE HEDEF/KISITLARININ TAYINI
- κ DEĞİŞİKLİKLERİN NASIL YAPILACAKININ TANIMLANMASI
- κ PROJE GÖREV DEKLARASYONUNUN HAZIRLANMASI

DEFENSIBLE PLANNING (6)

PROJENİN TANIMI(PD) - SINIRLAR VE OPTİMİZASYON



DEFENSIBLE PLANNING (7)

PROGRAMLAMA TANIM VE KAPSAMI - SCHEDULING - I

- PROJENİN NE ZAMAN BAŞLAYACAĞINI/BİTECEĞİNİ BELİRLEME İŞLEVİDİR. KAPSAMI;
- ⊕ TÛM İŞLER NASIL TANIMLANACAK ?
 - ⊕ HERBİR İŞ NE KADAR SÛRE ALACAK ?
 - ⊕ HANGİSİ ÖNCE YAPILACAK ?
 - ⊕ HERBİR İŞ NE ZAMAN BAŞLAYACAK ?
 - ⊕ PROJE İMADINI DAHA ÇOK ETKİLEYECEK İŞLER HANGİLERDİR ? - (KRİTİK YOL)

DEFENSIBLE PLANNING (8)

PROGRAMLAMA İŞLEVINİN ÖZELLİKLERİ - SCHEDULING - II

- ⊕ PROJE TANIMINDAN SONRAKİ EN YOĞUN İŞTİR.
- ⊕ PLANLAMA FAALİYETLERİNİN TEMELİDİR.
- ⊕ TANIMLI VE ÖLÇÜLEBİLİRDİR.
- ⊕ İŞ AKIŞ SİRASINI VE BAĞIMLILIKLARI GÖSTERİR.
- ⊕ STANDART TERMİNOLOJİYE SAHİPTİR. (HENRY GANTT)

DEFENSIBLE PLANNING (9)

STANDART HAZI TERİMLER - SCHEDULING - III

- ⊕ PROJECT
- ⊕ TASK
- ⊕ PREDECESSOR TASK
- ⊕ SUCCESSOR TASK
- ⊕ MILESTONE
- ⊕ NETWORK DIAGRAM
- ⊕ TYPICAL HOURS
- ⊕ START DATES (EARLY, LATE, PLANNED, ACTUAL START)
- ⊕ FINISH DATES (EARLY, LATE, PLANNED ACTUAL FINISH)
- ⊕ SLACKS
- ⊕ CRITICAL PATH
- ⊕ WORK BREAKDOWN STRUCTURE (WBS)

DEFENSIBLE PLANNING (10)

PROGRAMLAMA ARAÇLARI - SCHEDULING TOOLS - I

SUBJECT	TOOLS
WORK IDENTIFICATION	W B S
DURATION	ESTIMATING
RELATIONSHIPS/DEPENDENCIES	NETWORK
START/FINISH DATES	BAR CHART
SLACK	CRITICAL PATH

Şekil A-10. Proje Yönetimi 3

DEFENSIBLE PLANNING (11)

WORK BREAKDOWN STRUCTURE (WBS) - SCHEDULING TOOLS - II

TANIM

WBS, PROJEYİ OLUŞTURAN İŞLERİ SİSTEMATİK BİR BİÇİMDE TANIMLAMA, GRUPLANDIRMA VE HİYERARŞİK OLARAK ORGANİZE ETME ARACIDIR.

YARAR

PROJENİN DAHA KOLAY ANLAŞILMASI VE YÖNETİLMESİNİ SAĞLAR BU KAPSAMDA;

- 7 PLANLAMA , PROGRAMLAMA VE MALİYET BELİRLEMEDE DAHA DOĞRU TAJMİNLER YAPILMASINI,
- 7 PROJEDE SORUMLUKLUKLARIN DAHA NET TANIMLANMASINI VE
- 7 İŞLERİN TAKİPKONTROLÜNÜN DAHA KOLAY YAPILMASINI SAĞLAR.

DEFENSIBLE PLANNING (11)

WORK BREAKDOWN STRUCTURE (WBS) - SCHEDULING TOOLS - III

TANIM

WBS, PROJEYİ OLUŞTURAN İŞLERİ SİSTEMATİK BİR BİÇİMDE TANIMLAMA, GRUPLANDIRMA VE HİYERARŞİK OLARAK ORGANİZE ETME ARACIDIR.

YARAR

PROJENİN DAHA KOLAY ANLAŞILMASI VE YÖNETİLMESİNİ SAĞLAR BU KAPSAMDA;

- 7 PLANLAMA , PROGRAMLAMA VE MALİYET BELİRLEMEDE DAHA DOĞRU TAJMİNLER YAPILMASINI,
- 7 PROJEDE SORUMLUKLUKLARIN DAHA NET TANIMLANMASINI VE
- 7 İŞLERİN TAKİPKONTROLÜNÜN DAHA KOLAY YAPILMASINI SAĞLAR.

DEFENSIBLE PLANNING (12)

WBS ÖZELLİKLERİ - SCHEDULING TOOLS - IV

- 7 ANLAMLI VE TÜRDEŞ İŞLER OLMALI
- 7 DAHA FAZLA ALT GRUBA AYRILAMAYACAK ÇAFTA OLMALI
- 7 GİRİŞÇIKTILARI NET OLARAK TANIMLANABİLMELİ
- 7 RUTİN/TEKRARLI İŞLER OLMAMALI
- 7 PROJE SÜRESİ İLE ORANTILI BÜYÜKLÜKTE BİR SÜREYE SAHİP OLMALI (BİR WORK PKG İÇİN EN AZ İKİ HAFTA)
- 7 PROJENİN GEREKTİRDİĞİ TÜM İŞLER/TASKS KAPSAMALI

DEFENSIBLE PLANNING (13)

WBS ÖRNEK - SCHEDULING - V

SYSTEM TESTING	HARDWARE	CODING
1. SYS SPECS	2. HW SPECS	17. PRG SCREENS/REP.
3. DEFINE DATA DICTIONARY	4. SELECT TERMINALS	18. PRG DB DEFINITION
5. SCREEN & REPORT LAYOUTS	6. SELECT COMPUTERS	
7. DESIGN TEST PROCEDURE	10. TERMINAL ORDER & DELV.	
8. PREPARE TEST DATA	11. COMPUTER ORDER & DELV.	
9. SELECT DB MANAGER	12. SITE PREPARATION	
16. DEVELOP BACKUP/RESTORE	13. WRITE TERMINAL INTERF.	
19. WRITE USER'S MANUAL	14. INSTALL HW	
20. INTEGRATED TESTING	15. TEST TERMINALS	
21. TRAIN USERS		

DEFENSIBLE PLANNING (14)

GÖREV TANIM FORM ÖRNEĞİ (STD) - SCHEDULING TOOLS - VI

İŞ ID NO	İŞİN ADI
İŞ PAKETİ (WP)	
İHTİYAÇLAR(SİPARİS)	
BAGIMLI İŞ(PRED) ID NO	
SORUMLU PERSONEL	
SÜRE(GÜN/HAFTA...)	ADAMSAAT İHTİYACI
ÖZEL HUSUSLAR	
TECHİZAT İHTİYACI	
SERVİS İHTİYACI	
YETKİ İHTİYACI	
MALİYET	
DİĞER	

DEFENSIBLE PLANNING (15)

KRİTİK YOL METODU(CPM) - SCHEDULING TOOLS - VII

TANIM

PROJE NETWORK'ÜNDE EN UZUN PROJE SÜRESİNİ (YOLU) OLUŞTURAN BAĞIMLI İŞLER/TASKS BELİRLEME TEKNİĞİDİR.

ÖZELLİKLERİ

- 7 PROJENİN GEREKTİRDİĞİ EN UZUN SÜREYİ GÖSTERİR.
- 7 HİÇBİR SLACK (ÖLÜ ZAMAN) YOKTUR.
- 7 CPM'DE GEÇİKME PROJE BİTİŞ SÜRESİNİ DE GEÇİKTİRİR.

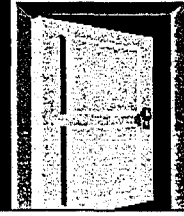
Şekil A-11. Proje Yönetimi 4

DEFENSIBLE PLANNING (16)

CPM - SCHEDULING TOOLS - VIII

TASK ID	TASK DESCRIPTION	DURATION WEEKS	NUM OF STAFF	PRED
A	DETERMINE SW & HW REQUIREMENTS	2	1	-
B	DEVELOP PURCHASE ORDER FOR HW	1	1	A
C	HW DELIVERY AND SET UP	2	1	B
D	SET UP WORK AREA	4	2	A
E	DEVELOP TRAINING DESIGN	4	2	A
F	DELIVER TRAINING	4	1	E
G	INSTALL SW	1	1	C,D,F

PROJECT LEADERSHIP COURSE



4 - PROJECT ORIENTATION

PROJECT ORIENTATION (1)

ÖNCELİKLERİN BELİRLENMESİ

- ROJE YÖNETİMİ, İŞLERİN BAŞLANGIÇTA HEDEFLERİNİ:
 - SÜRELER İÇTİRİSİNDE,
 - MALİYETTE VE
 - MÜŞTERİ TATMİNİNİ SAĞLAYACAK KALİTE DÜZEYİNDE GERÇEKLEŞMESİ İÇİN KAYNAKLARIN ETKİN KULLANIMINI GEREKTİRİR.
- KONULARIN ÖNCELİKLERİNİ BELİRLEYİN (#1, #2, #3...)
 - KALİTE
 - MALİYET
 - SÜRE
- HERBİR KONUNUN PERFORMANS DEĞERLENDİRMESİ İÇİN ÖLÇÜM METODLARININ BELİRLENİR.

PROJECT ORIENTATION (2)

İŞLERİ GRUPLANDIRMA - WORK BREAKDOWN STRUCTURE (WBS)

SYSTEM TESTING	HARDWARE	CODING
1. SYS SPECS	2. HW SPECS	17. PRG SCREENS/REP.
3. DEFINE DATA DICTIONARY	4. SELECT TERMINALS	18. PRG DB DEFINITION
5. SCREEN & REPORT LAYOUTS	6. SELECT COMPUTERS	
7. DESIGN TEST PROCEDURE	10. TERMINAL ORDER & DELV.	
8. PREPARE TEST DATA	11. COMPUTER ORDER & DELV.	
9. SELECT DB MANAGER	12. SITE PREPARATION	
13. DEVELOP BACKUP RESTORE	13. WRITE TERMINAL INTERF.	
13. WRITE USER'S MANUAL	14. INSTALL HW	
20. INTEGRATED TESTING	15. TEST TERMINALS	
21. TRAIN USERS		

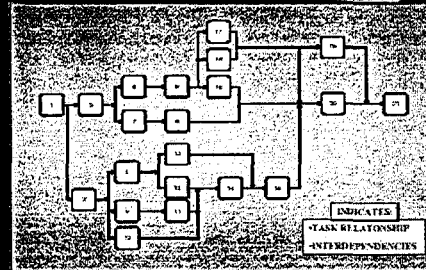
PROJECT ORIENTATION (3)

İŞ GRUPLARINDA - TASK DATA ÖRNEK

#	TASK DESCRIPTION	TYPE	PLAN	STAFF	SKILLS	PRED
1	SYS SPECS	170	2	SD	PR, IM	-
2	HW SPECS	180	1	HW	CC	1
3	DEFINE DATA DICTIONARY	190	2	PR	IM	1
4	SELECT TERMINALS	20	1	CC		2
5	SCREEN & REPORT LAYOUTS	210	2	HW		2
6	SCREEN & REPORT LAYOUTS	220	1	HW	IM	2
7	DESIGN TEST PROCEDURE	30	1	SD	PR, IM	3
8	PREPARE TEST DATA	140	2	IM		7
9	SELECT DB MANAGER	270	2	SD	PR, IM	6
10	TERMINAL ORDER AND DELIVERY	25 DAYS & 2 SHIP QUOTED BY VENDOR				4
11	COMPUTER ORDER AND DELIVERY	25 DAYS & 2 SHIP QUOTED BY VENDOR				5
12	SITE PREPARATION	25 DAYS & 2 SHIP QUOTED BY VENDOR				2
13	WRITE TERMINAL INTERFACE	280	2	CC		4
14	INSTALL HW	130	2	HW		10-11, 12
15	TEST TERMINALS	90	2	HW	CC	12-14
16	DEVELOP BACKUP RESTORE	290	2	HW	PR	9
17	PRG SCREENS & REPORTS	430	4	PR	SE	8
18	PRG DB DEFINITION	210	2	PR		9
19	WRITE USER'S MANUAL	150	2	PR	IM	12-20
20	INTEGRATED TESTING	120	2	HW	PR, CC, IM	13, 16, 21, 12
21	TRAIN USERS	140	2	PR	IM	12, 20

PROJECT ORIENTATION (4)

NETWORK DIAGRAMI ÖRNEK



Şekil A-12. Proje Yönetimi 5

PROJECT ORIENTATION (11)

PLANLAMADA BELİRLENECEK (KARARSAL) KONULAR

- 7 PERRFORMANCE
- 7 LABOR RATE
- 7 TECHNOLOGY RATE
- 7 ENHANCEMENT EFFORT
- 7 TASK DURATION
- 7 EARLY START/FINISH DATES
- 7 PLANNED HOURS
- 7 COSTS

PROJECT ORIENTATION (12)

TASLAK PROGRAMIN ANALİZİ

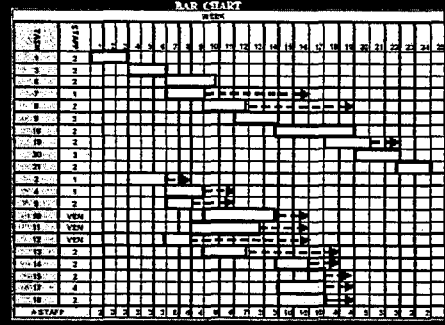
- CP FINISH DATE > REQUIRED FINISH DATE İSE :
 - CP'DEKİ GÖREV SÜRELERİNİ DÜŞÜR. (REDUCT CP)
 - SÜRELERİ DEĞİŞTİRMEK PLAN SAATLERİNİN MUTLAKA DEĞİŞTİRİLECEĞİ ANLAMINDA ALGILANMAMALIDIR.
- CP FINISH DATE < REQUIRED FINISH DATE İSE :
 - GÖREV SÜRELERİNİ İŞYÜKÜ SEVİYESİNE GÖRE ARTIR.
 - PLANLAMA KAHRAMAN GÖRÜNME SÜRECİ DEĞİLDİR.
 - PLANLAMASI ERKEN OLAN GÖREVLERİN GENELLİKLE UMULANDAN DAHA ÇOK ZAMAN ALDIĞINI UNUTMAYIN.
 - DAHA KOMPLEKS İŞLER İÇİN EKSTRA SÜRELER EKLEYİN.
- NON - CP TASKS WITH SLACKS :
 - PERSONEL SAYISINI MÜMKÜN OLDUĞU ORANDA AZALT.

PROJECT ORIENTATION (13)

PROGRAMIN SIKIŞTIRILMASI - CP SÜRESİNİN KISALTILMASI

- κ PERSONEL İLAVESİ
- κ FAZLA MESAİLİ ÇALIŞMA
- κ FAALİYET ALANI VE KALİTE SEVİYESİNİN GÖZDEN GEÇİRİLMESİ
- κ ALT KONTRAKTÖRLERE YÖNELİM
- κ GÖREVLERİN DAHA HIZLI VE İTRİN TAKİBİ
- κ PARALEL/AYUSULU MÜHENDİSLİK FAALİYETLERİ
- κ İŞLERİN DAHA ÇOK ALT PARÇALARA AYRILMASI (FWBD).

PROJECT ORIENTATION (14)



PROJECT LEADERSHIP COURSE

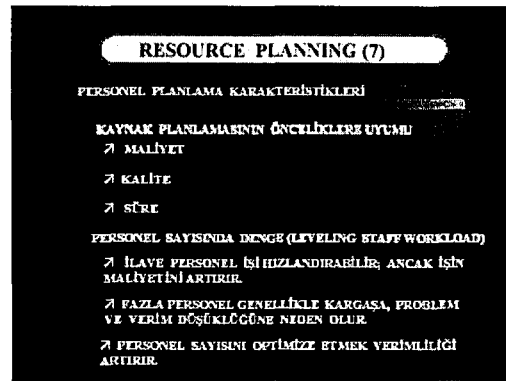
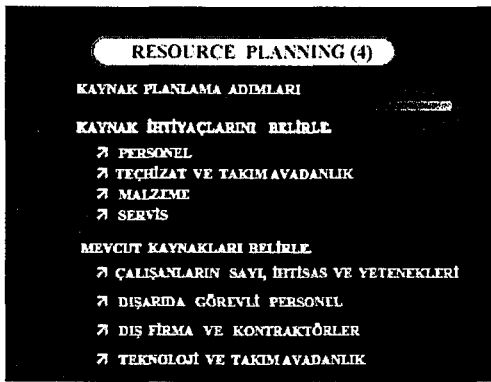
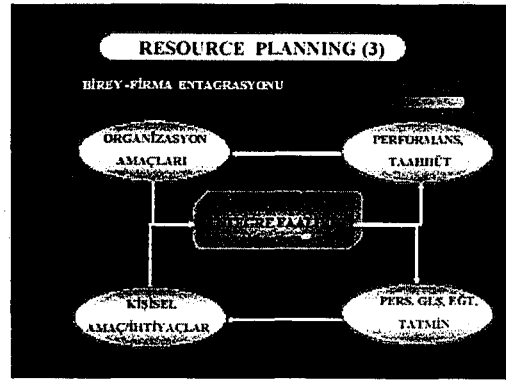


5 - RESOURCE PLANNING & RISK MANAGEMENT

RESOURCE PLANNING (1)

- KAYNAK PLANLAMA VE BİLEŞENLERİ
- KAYNAK;
İŞİN YAPILMASI İÇİN GEREKLİ OLAN TÜM GİRDİLERDİR.
- KAYNAK PLANLAMA BİLEŞENLERİ
- 7 KAYNAK ÇEŞİTLERİ NELER OLACAK ?
 - 7 NEREDEN KARŞILANACAK ?
 - 7 HANGİ MİKTARDA OLACAK ?
 - 7 NASIL ELDE EDİLECEK ?
 - 7 KİM OLACAK ?
 - 7 NE ZAMAN OLACAK ?

Şekil A-14. Proje Yönetimi 7



Şekil A-15. Proje Yönetimi 8

RESOURCE PLANNING (9)

PERSONEL PLANLAMA ÖRNEK (KİM-NE-NE ZAMAN)

WEEK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
GARY	1	1	1	3	3	3	6	6							
MAYER	1	1	1	3	3	3	6	6							
FRANZ	CC	CC	HW	2	2	2	4	4	4				V	V	
SEL #1						S	S	S							
SEL #2						S	S	S							
SEL #3															
SEL #4															
SEL #5															T, SK D

RISK MANAGEMENT(1)

RİSK YÖNETİM

RİSK NEDİR?
 PROJERİN HEDEFLERİNİ OLUMSUZ ETKİLEYEN BELİRSİZ OLAYLARIN YÜKÜMLÜLÜĞÜDÜR.

TEMEL DOĞRULAR.
 BELİRSİZLİK YOKSA RİSK YOKTUR.
 PROJENİN HEDEFLERİNİ BİREBİRİNDEN BAĞIMSIZ ETKİLEMEZLER.
 RİSK YÖNETİMİ ŞİRHİRLİ BİR DEĞERNEK DEĞİLDİR.
 TÜM RİSKLERİN OLUMSUZ YÖNELİMLERİYLE AŞIRI İLGİLENMEK MANTESİZ KARARLARA YOL AÇABİLİR.
 SORUN RİSKİN ALINIP ALINMAYACAĞI DEĞİL, MAKUL RİSKLERİN NASIL ALINACAĞIDIR.
 GÖREY RİSKTEN KAÇINMAK DEĞİL, ÖNÜMLÜMÜK, DEĞERLENDİRMEK VE YÖNETMEK OLMALIDIR.

RISK MANAGEMENT(2)

RİSK YÖNETİM PLANI (TAHMİN - ETKİ - KONTROL)

RİSK ID PROJEYE NELER OLABİLİR ?

RİSK QUANTIFICATION OLASI BANGİ RİSK/RİSKLER PROJENİN NE GRANDA ETKİLER ?

RİSK RESPONSE TEHDİTLERE KARŞI NELER YAPILABİLİR ?

RİSK CONTROL RİSK PLANI PROJENİN HERBİR FAZINI NE KADAR DEĞİŞTİREBİLİR ?

RISK MANAGEMENT(3)

RİSK TÜRLERİ

1. GENEL PROJENİN RİSKLERİ
 2. ÖZEL PROJENİN RİSKLERİ

DİŞ KAYNAKLI - KONTROL EDİLEMEZ
 ÖNGÖRÜLEMEYEN
 ÖNGÖRÜLEBİLİR
 İÇ KAYNAKLI - GENELLİKLE KONTROL EDİLEBİLİR (TEKNİK OLMAYAN)
 TEKNİK - GENELLİKLE KONTROL EDİLEBİLİR
 HUKUKİ - GENELLİKLE KONTROL EDİLEBİLİR

RISK MANAGEMENT(4)

OLABİLECEK OLAYLAR

PERSONEL DEVAMSIZLIĞI
 PERSONEL AYRILMA/İSTİFA
 PERSONELİN BAŞKA GÖREV İÇİN ÇIKILMA EMRİ
 İLAVE PERSONEL/BEÇERİ DÜZEYİNİN YOKLUĞU
 ÜRÜN GELİŞTİRME (ENHANCEMENT) SÜRECİNİN UMULANDAN DAHA ÇOK ZAMAN ALMASI
 EĞİTİMLER İSTENİLEN VERİMLİLİKTE OLMAMASI
 FİRMALARIN TAHHÜT ETTİKLERİ SÜRELERE UYMAMASI

RISK MANAGEMENT(5)

RİSK ANALİZ - (OLAY — PROJENİN FAZLI —)


<input type="checkbox"/> YÜKSEK OLASILIK	<input type="checkbox"/> YÜKSEK OLASILIK
<input type="checkbox"/> YÜKSEK ETKİ	<input type="checkbox"/> DÜŞÜK ETKİ
1. —	1. —
2. —	2. —
3. —	3. —
<input type="checkbox"/> DÜŞÜK OLASILIK	<input type="checkbox"/> DÜŞÜK OLASILIK
<input type="checkbox"/> YÜKSEK ETKİ	<input type="checkbox"/> DÜŞÜK ETKİ
1. —	1. —
2. —	2. —
3. —	3. —

Şekil A-16. Proje Yönetimi 9

RISK MANAGEMENT(6)	
RİSK YÖNETİMİNİN ÖNGÖRDÜĞÜ PLAN AYARLAMALARI	
OLAY	PLANA GÖRE AYARLAMALAR
DEVAMSIZLIK	
İSTİFA	
GÖREV DEĞİŞİKLİĞİ	

FİRMA TAAHÜT GEÇİCİME	

PROJECT LEADERSHIP COURSE



6 - QUALITY PLANNING & BUDGETING

QUALITY PLANNING (1)

TANIM

PROJENİN KALİTESİ

- ☑ MÜŞTERİ TARAFINDAN KABUL EDİLEBİLİR VE
- ☑ FİRMANIN MÜŞTERİ TATMİNİ İÇİN ÖNGÖRDÜĞÜ POLİTİKALARA UYGUN KALİTEDE ÜRÜN/HİZMET ELDE EDİLİNCEYE KADAR;

PLANLAMA VE İCRA SAHALARINDA GÖSTERİLEN HER TÜRLÜ FAALİYETİN DİREK BİR SONUCU OLARAK ORTAYA ÇIKAR.

QUALITY PLANNING (2)

KALİTE YÖNETİM ALANLARI

TEKNİK PERFORMANS / TECH PERE

- ☑ DİZAYN KALİTESİ/UYGUNLUĞU
- ☑ KARŞILANACAK ÖZELLİKLERİ BELİRLEME

SAGLAMLIK/CALISIRLILIK / DESIGN CAPABILITY

- ☑ ÜRÜN,
- ☑ SERVİS VE
- ☑ PROSESLERE YOĞUNLAŞMA.

İLAVE ÖZELLİK/DEĞER KATMA / ADDED VALUE

- ☑ ÜRÜNE MÜŞTERİ BEKLENTİSİ ÜZERİNDE DEĞER KATMA

QUALITY PLANNING (3)

PROJE - KALİTE YÖNETİM ALANLARI İLİŞKİSİ

KALİTE \neq MÜKEMMELLİK (M)

KALİTE $=$ MÜŞTERİ TATMİNİ (MT)

$M = MT + AV$ (ADDED VALUE)

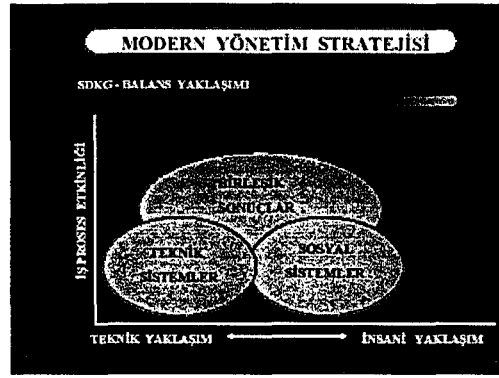
QUALITY PLANNING (4)

KALİTE MALİYETİ - COST OF QUALITY(COQ)

$COQ =$ KALİTESİZ ÜRÜN/HİZMET MALİYETİ

KALİTELİ ÜRÜN/HİZMET MALİYETİ COQ'NUN BİR BİLEŞENİ DEĞİLDİR.

Şekil A-17. Proje Yönetimi 10



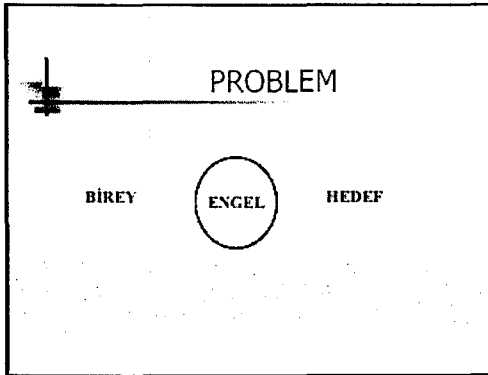
Şekil A-18. Proje Yönetimi 11

PROBLEM ÇÖZME TEKNİKLERİ

- ## PROBLEM ÇÖZME TEKNİKLERİ
- Problem kavramının açıklanması.
 - Problemlerin kaynağı
 - Problem çözmeyi zorlaştıran faktörler
 - Problem çözümünün dayanakları.
 - Problem çözme süreci
 - Problem çözümede kullanılan (istatistik) araçlar

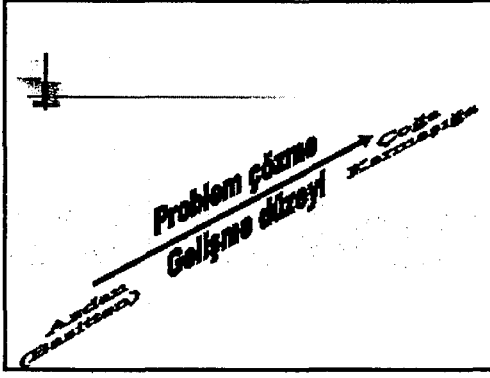
- ## PROBLEM NEDİR?
- Engeldir.
 - İstenmeyen durumdur.
 - Huzursuzluk yaratan durumdur.
 - Değişime tepkidir.
 - Üzerinde düşünülecek yada çözülecek bir sorundur.

- ## SORUNLAR NELERDİR?
- Dünyada
 - Türkiye' de
 - Öğretimde



- ## PROBLEM
- Problem hedefe ulaşamama durumudur.
 - Problem engele eşit değildir.
 - Engel aynı olduğu halde problem her kişi için değişir.

Şekil A-19. Problem Çözme Teknikleri 1



- Her çözüm kendi sorununu beraberinde getirir.
- Problemi çözmek problemi ortadan kaldırmaz.

- Gelişmiş ülkeler problemlerini çözmüş olan ülkelerdir.
- Gelişmekte olan ülkeler problemlerini çözmeye çalışmaktadır.
- Az gelişmiş ülkeler problemlerinin farkındadır.

- ### PROBLEM ÇEŞİTLERİ
- Yapılandırılmış problemler
 - Yapılandırılmamış problemler

PROBLEM ÇÖZME

Problemi çözenin ön şartı problemi çözülebilir olarak görmektir.

Ulaşılmaz olarak görülen problemler çözülemez.

- ### PROBLEM ÇÖZMEYİ ZORLAŞTIRAN FAKTÖRLER
- Hedeflerin iyi tanımlanmamış olması
 - Geçici çözümler
 - Problemlerin endişe kaynağı olarak algılanması

Şekil A-20. Problem Çözme Teknikleri 2

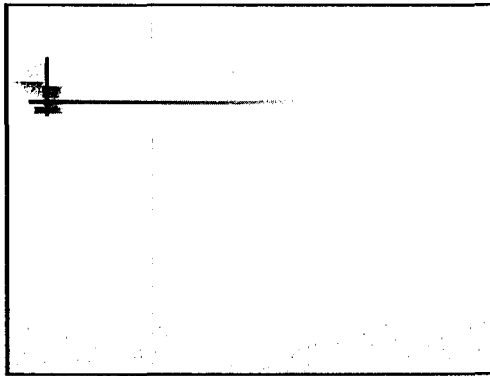
İŞ ÖNCELİKLERİ

	Önemli	Önemsiz
Acil	1	2
Acil Değil	3	4

- PROBLEM ÇÖZME YÖNTEMLERİ**
- Geleneksel yöntemler
 - Kişisel yöntemler
 - Uzmanlara dayalı yöntemler

- PROBLEM ÇÖZMENİN DAYANAKLARI**
- 1-Geleneksel yöntem: Çözümü geçmişte arayanlar.
- AVANTAJLARI:**
- Denenmiş olması
 - Ucuz olması
 - Alt yapısının hazır olması

- DEZAVANTAJLARI**
- Şartlara uymayabilir.
 - Verimsiz olabilir.
 - Yenileşmeye engel olur.
 - Zaman kaybına neden olur.
- Bu yöntem, gerekli ama tek başına yeterli değildir.



- PROBLEM ÇÖZMENİN DAYANAKLARI**
- 2-Kişisel deneyimler: Ben böyle yaparak başarılı oldum, herkes böyle yapmalı şeklindeki genellemeler, her zaman olumlu sonuç vermeyebilir.
- Tümevarım: Parçadan bütüne doğru yapılan genellemedir.

Şekil A-21. Problem Çözme Teknikleri 3

3-Uzmanlara dayalı problem çözümleri:
Konuyu uzmanlarına sorarak onların önerilerini uygularlar.

- Tumdengelim Bütünden parçaya doğru yapılan genellemedir.
- Bilimde mutlak doğru yoktur. Uzmanlar arasında görüş farklılıkları vardır
- Bu yöntemde tek başına yeterli değildir.

BİLİMSEL PROBLEM ÇÖZME SÜRECİ

Problem çözmeye başlamadan önceki sorgulama (kendimize)

1-Gerçek ile beklenti arasında fark var mı?

Hayır cevapınız ise problem yok
Evet cevapınız ise iki numaralı soru

2-Bu fark rahatsızlık veriyor mu?

Hayır cevapınız ise problem yok
Evet cevapınız ise üç numaralı soru

3-Etki alanınızda mı?

Evet cevapınız ise problemi çözmeye başlayın
Hayır cevapınız ise yetkili ile görüşün.

Tüm sorulara EVET diyorsak problem çözme süreci başlamış demektir. Problem çözme sürecinde problemin tanımlanması ve veri toplama aşamasında 5N+1K kuralı uygulanabilir. Bunu açacak olursak

5N + 1K

Niçin?
Nerede?
Nasıl? + Kim? } 5N+1K
Ne zaman?
Ne kadar?

Sorularını sıra ile problem üzerine uygulamamız gerekir.

Şekil A-22. Problem Çözme Teknikleri 4

Soru sorma veri toplama sürecidir.
Problemler çözmede sorular cevap odaklı değil soru sorma odaklı olmalıdır.
Doğru sorular sorularak doğru cevaplar alınmalıdır.
Veri sonuçlarına göre teşhis konmalıdır.

Problemlerin çözümünde başarısızlık:
Problemleri çözmek karışık bir süreçtir.
Problem çözmede etkili olamamanın nedenleri

- Metodik çalışmamak
- Yeterli istek ve kararlılığa sahip olmamak
- Problemi yanlış tanımlamak
- Gereken teknik ve bilginin olmaması

• Teknikleri etkili bir şekilde kullanılmamak

- Yanlış bir teknik kullanmak
- Yetersiz yada doğru olmayan bilgi
- Yaratıcı ve çözümlenici zihinsel yetenekleri koordine edememek
- Çözümü etkili bir şekilde uygulamaya sokamamak
- Çözümün önünde duran gizli faktörler

PROBLEM ÇÖZME SÜRECİ

- Problemin tanımlanması
- Olası nedenlerin tanımlanması
- En uygun nedenlerin belirlenerek araştırılması
- Olası çözüm alternatiflerinin belirlenmesi
- En uygun çözüm alternatiflerinin belirlenmesi
- Eylem planı
- Sonuçları değerlendirme

PROBLEM ÇÖZÜMÜNDE KULLANILAN BİLİMSEL KONULAR

- 1- Yöntem Bilimi
- 2- İstatistik
 - a) Verilerin toplanması
 - b) Verilerin gruplandırılması
 - c) Grafiğe aktarma

Şekil A-23. Problem Çözme Teknikleri 5

ÖLÇME

- Sınıflandırma
- Sıralama
- Eşit aralıklı ölçek
- Eşit orantılı ölçek

BEYİN FIRTINASI TEKNİĞİ

Beyin fırtınası 1930 yıllarından beri uygulanan bir tekniktir. Her insan sorunlar karşısında farklı düşüncelere sahiptir. Ancak insanlar düşüncelerini, ön yargıyla karşılanacağı, küçük görülme endişesi, söz hakkı sınırlaması ve eleştiriler endişesiyle söyleyemezler.

BEYİN FIRTINASININ KURALLARI

- 1-Toplantıya katılan her kişiye eşit söz hakkı verilmeli.
- 2- Zaman sınırlandırılması olmamalı.
- 3-Konuşmacıya müdahale edilmemelidir.
- 4-Konuşmacılar eleştirilmemelidir.

5-Değerlendirme yapılamaz

6-Sırasıyla söz alını. Sıra geldiğinde söyleyeceği olmayan pas geçer.

7-Konuşmacı söz aldığı anda sadece bir öneride bulunabilir.

8-Herkes pas geçmeye kadar devam edilir.

9-Pas geçenler çoğaldıktan sonra sıra takip edilmeden söz verilebilir.

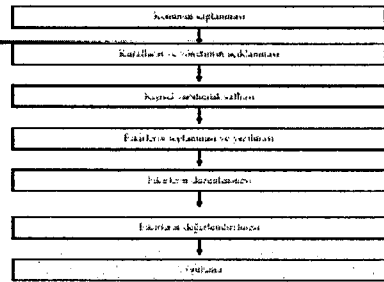
Beyin fırtınasından amaç çok sayıda fikir üretmektir.

YATAY DÜŞÜNME

Bir eleme, değerlendirme ölçütleri olarak düşünülürse elekten geçenler uygun çözüm önerileridir ve bir havuzda toplamı. Beyin fırtınasında nitelik değil, nicelik önemlidir. Öncelikli amaç kaliteli fikir değil, fikir sayısının artırılarak eleğin doldurulmasıdır. Beyin fırtınası tekniğinde yeşil şapka kullanılır.

Beyin fırtınası düşey geçişten yatay geçiş sağlayan bir tekniktir. Her çözüm önerisi dikey düşümedir. Her öneri değerlendirilir. Değerlendirme sonucu olumsuz olabilir.

BEYİN FIRTINASI ÖRNEĞİ



Şekil A-24. Problem Çözme Teknikleri 6

BEYİN FIRTINASININ KULLANILACAĞI YERLER

- 1- Bir problemin belirlenmesinde
- 2- Bir problemin olası nedenlerinin ortaya çıkarılmasında
- 3- Bir problemin olası çözümlerinin belirlenmesinde
- 4- Olası problemler için önlemlerin belirlenmesinde

FİKİR ÜRETİMİNİN DEĞERLENDİRİLMESİ

- 1- Sınıflandırma
 - a)- Benzer konular aynı başlıkta toplanır.
 - b)- Etki ve ilgi alanına göre sınıflandırılır.
 - c)- Grup çalışması

Bir'den fazla katılımcının grup oluşturarak önerileri değerlendirilmesidir. Beyin fırtınasında üretilen fikirler aynı toplantıda değerlendirilmez. Kılıçığı adı verilen diyagrama işlenir.

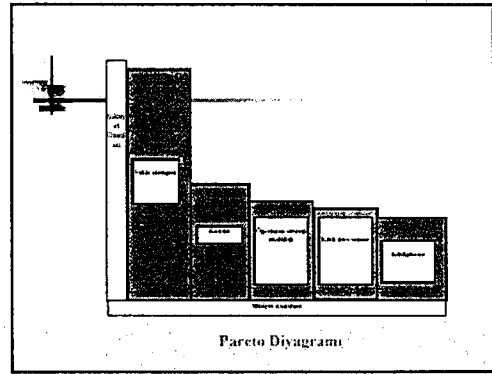
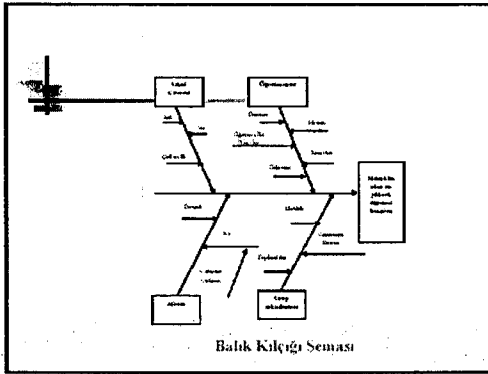
- d)- Oylama

Bir insanın seçenekler konusunda oy kullanması için problemin ayrıntısı hakkında fikir sahibi olması gerekir.
- e)- Neden sonuç diyagramı

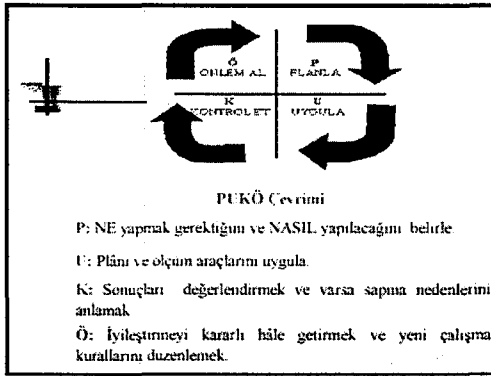
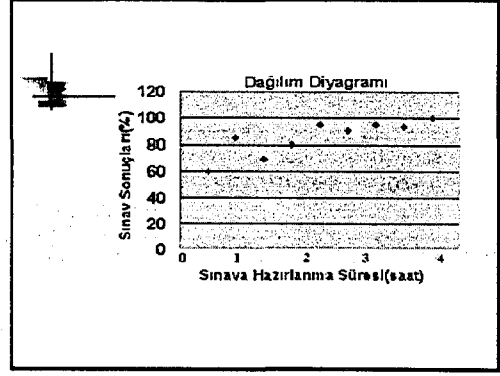
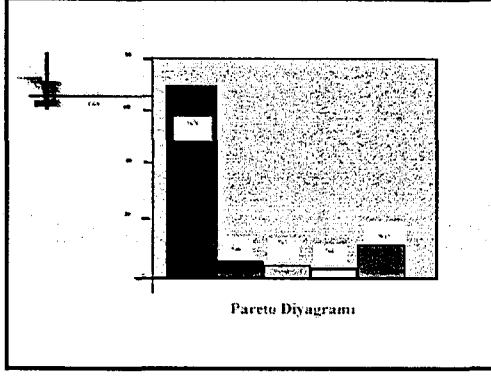
Bir probleme ait olası nedenleri tanımlama amacıyla kullanılır

- 2- Problemi etkileyen ana nedenler

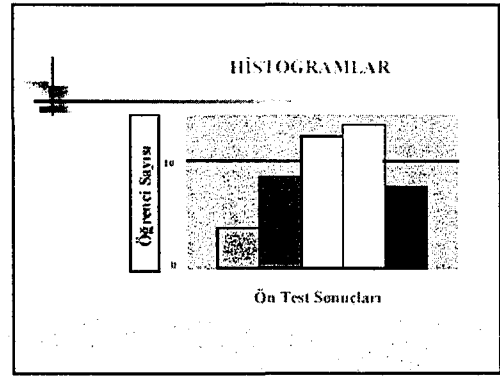
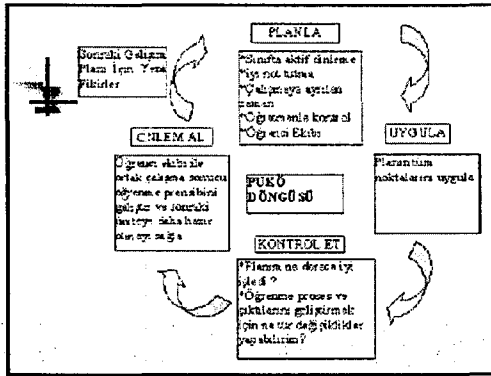
Süreci etkileyen ana nedenler ortaya konur. Sürekli neden diye sorularak ilgili yere yazılır. Her bir neden için "neden?" sorusu sorularak alt nedenler tanımlanır. Sonuçlar balık kılıçığı şemasına aktarılır.



Şekil A-25. Problem Çözme Teknikleri 7



- PUKÖ ÇEVİRİMİNİN TEMEL ÖZELLİKLERİ ŞUNLARDIR**
1. Katılımcı yönetim tarzidir. (Tüm çalışanlar katılır.)
 2. Fonksiyonlar arası faaliyetleri geliştirir.
 3. "İlk defada doğru yap" ilkesi geçerlidir.
 4. Kalite ve maliyet kriterlerine döndürür.



Şekil A-26. Problem Çözme Teknikleri 8

EDWARD DE BONDI' un 6 Şapkalı Düşünme
Tekniği

Renklerine göre her şapkaya farklı değerler ve bakış açıları yüklenmiştir. Şapkayı takan kişi o şapkanın yüklediği bakış açısı ve değerler ile olaylara bakarak karar verir

BEYAZ: Saflık, temizlik, masumiyet, kişisel yorum yapmadan objektif yaklaşım veriler ile konuşma değerleri yüklenmiştir. Örneğin "Bu yıl fire oranı %5, bu ay müşteri şikâyetinin %1 düzeyinde olduğunu söylemesi"

KIRMIZI: Şehvet, sinir, öfke duygularını yükler. Nedenini bilmeden, bence bu çözüm fire oranını düşürecek diye ifade kullanır. Sezgileriyle bunu ifade eder. Olaylara olumlu veya olumsuz yaklaşabilir.

SARI: Kışkırtıcı, habercı, güneş, Güneş, umut demektir. Bir şeyin olumlu taraflarını düşünür. Geçen yıl bunları yaptık sonuç olumlu oldu şeklinde bir ifade kullanılır.

SİYAH Kötümser

YEŞİL: Ferahlık, doğanın kendisi. Doğa yaratıcılık ve üreticiliğin simgesidir. "Tamam bunları yaptık ama daha yapabileceğimiz şunlar var" şeklinde bir anlayışın ifadesidir.

MAVİ: Gökyüzü, olayları tepeden bakan, ne zaman hangi şapkanın takılmasında, karar mekanizmasını işleten bir düşünce unajı yüklenmiştir.

Şekil A-27. Problem Çözme Teknikleri 9

TOPLANTI YÖNETİMİ

TOPLANTI YÖNETİMİ

- Eğitim üretim içindir.
- Yaşamayı kolaylaştırmak için öğrenmek gerekir.
- Zorla eğitim öğretim vermek hayvana yük yüklemeye benzer

EĞİTİM

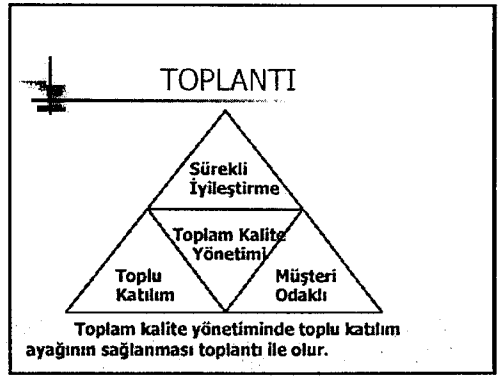
İstendik davranışların davranış haline getirilmesidir.
İstendik davranışın kaynağı ihtiyaç , bilim ve kültürdür.

LİDER

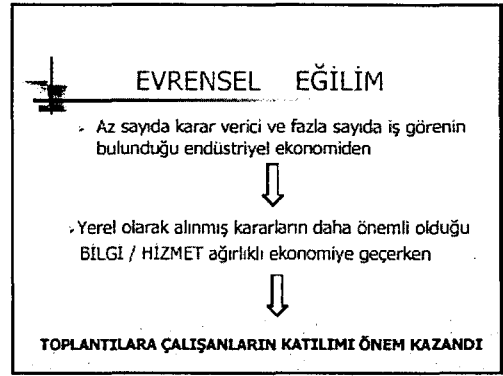
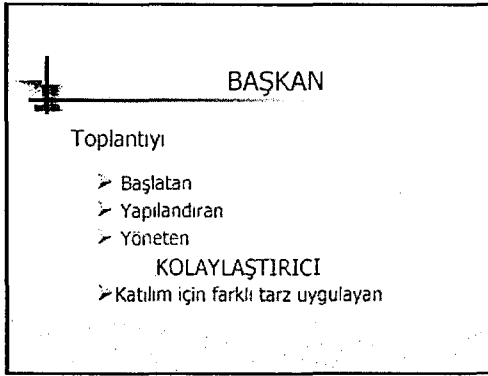
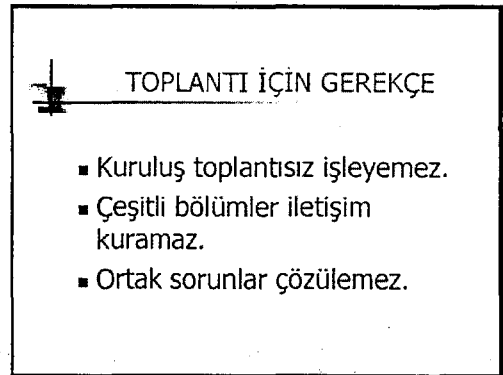
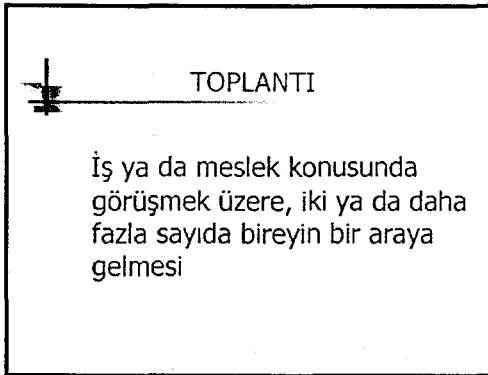
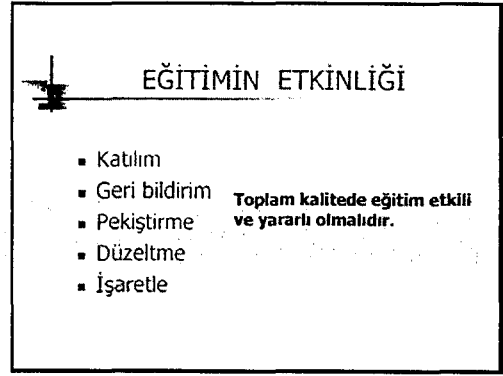
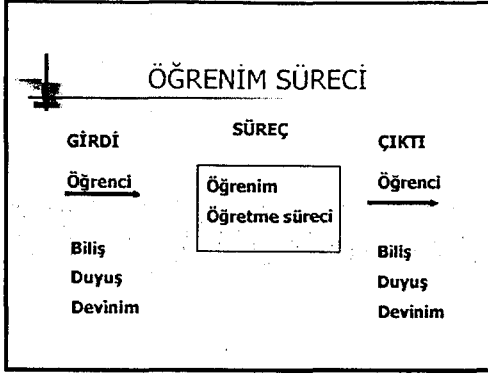
Teknik ve psikolojik yeterlilikler sağlandığı durumlarda lidere ihtiyaç yoktur.
Bu yetenekleri olan kişi olgun insandır.

HEDEF

- Lider olmadan toplantının yönetilmesidir.
- Karara katılma sürecinin sağlanmasıdır.
- Herkesi işin içine çekebilmektir.



Şekil A-28. Toplantı Yönetimi 1



Şekil A-29. Toplantı Yönetimi 2

"BU TOPLANTI GEREKSİZDİ"

- Düzenlenen toplantının ana amacı hakkında yeterince düşünülmemiş olabilir.
- Katılanlar konunun başka yöntemle daha etkili biçimde ele alınacağını düşünmektedirler. Tatmin olmazlar. Zaman boşa geçti diye düşünürler.

Cözüm:

- Bilginin telefon, mektup, faks... ile aktarılıp aktarılamayacağını düşünün

"TOPLANTI KÖTÜ PLANLANMIŞTI, İYİ HAZIRLANMAMIŞTI"

- Yapısal olarak eksikti, etkili bir gündemi yoktu.
- Gündem toplantı öncesi bildirilecek hazırlık için.

Cözüm:

- Gündemi planlayın, toplantıdan önce dağıtın.

"TOPLANTI KESİNTİYE UĞRAMIŞTI"

- SÜREKLİ KESİNTİ
- AŞIRI GÜRÜLTÜ

Cözüm:

- Acilen halledilmesi gereken işleri başkasına devredin
- Katılanların rahatsız edilmemesi gerektiğini belirtin.

"TOPLANTININ AMACI AÇIK DEĞİLDİ"

- Düzenleyen kişi amacı katılanlarla paylaşmamıştır. Ortak görüş yok. Önemsiz şikayete, konuyla ilgisiz noktalara saplanma.

Cözüm:

- Toplantıdan önce amacı saptayın, katılanlara açıklayıp paylaşın.

"TOPLANTI KÖTÜ BİR ORTAMDA YAPILMIŞTI"

- NEREDE, NE ZAMAN
- AŞIRI SICAK, SOĞUK
- AŞIRI KALABALIK

Cözüm:

- Toplantı yerini önceden ziyaret et. Koşulları gör.

"GEREKSİZ KİŞİLER KATILMIŞTI GEREKLİ OLANLAR YOKTU"

Cözüm:

- Gerekli kararların alınıp uygulanmasını sağlayacak anahtar kişiler saptanmalı ve toplantıya çağrılmalıdır.
- Kararları uygulama konumunda bulunmayanların etkili karar almaları olası değil.

Şekil A-30. Toplantı Yönetimi 3

"BAŞKAN YETERLİ DEĞİLDİ"

- Rehber olmalı
- Tartışmayı kabul etmeyen
- Katılanlar sindirilmiş, hayal kırıklığına uğramış
- Beklenen işbirliği sağlanamaz.

Cözüm:

- Katılanların duygularına, gereksinimlerine duyarlı olmalı
- Söz almayanları tartışmaya davet etmesi
- Aşırı konuşanları kısıtlaması

"HIÇBİR KARAR ALINMADI"

- Katılanlara aşırı hoşgörü göstermek onların en çok sevdikleri konularda işlerini dökmelerine neden olur.
- Gündemden uzaklaşılır.
- Karar alınmaz.

Cözüm:

- Gündeme uygun olarak sürdürülmeli
- Tartışılmasına önceden karar verilen konulardan uzaklaşılmalı

TOPLANTININ İŞLEVLERİ

HIZLI KARAR VERME

Değişim giderek hızlanıyor ⇒ Çalışanları bir araya getirin ⇒ Hızlı karar verme işlerlik kazanıyor

TOPLANTININ İŞLEVLERİ

İÇSEL DEĞİŞİKLİKLER

Kendilerini etkileyen değişiklikler kendilerine danışılmazsa karşı duruş ⇒ Çalışanları bir araya getirin ⇒ Yeni politikalar üzerine görüşmek değişim gereksinimlerini ortaya çıkarır.

TOPLANTININ İŞLEVLERİ

GÖRÜŞ VE DENEYİMLERİN DEĞİŞ TOKUŞU

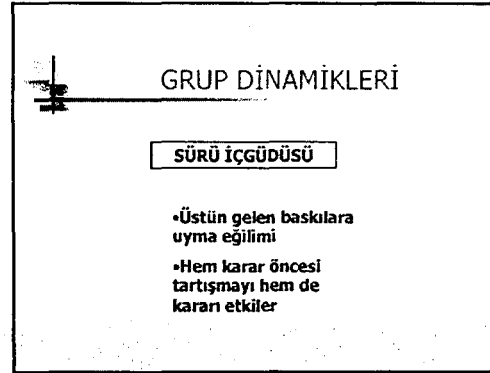
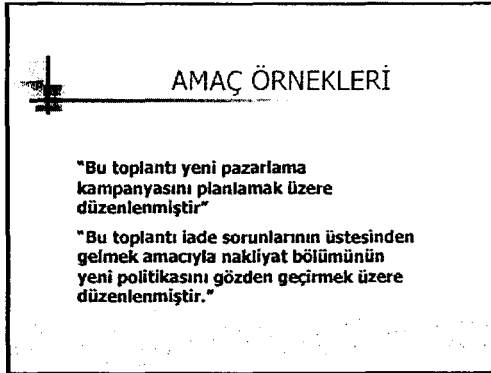
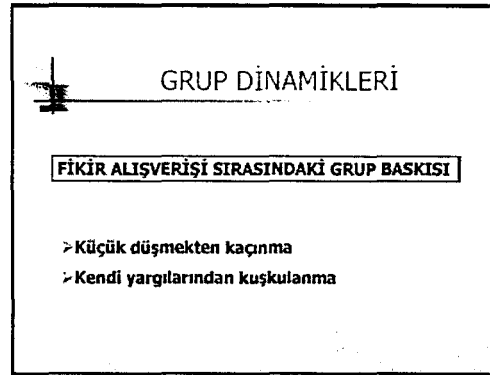
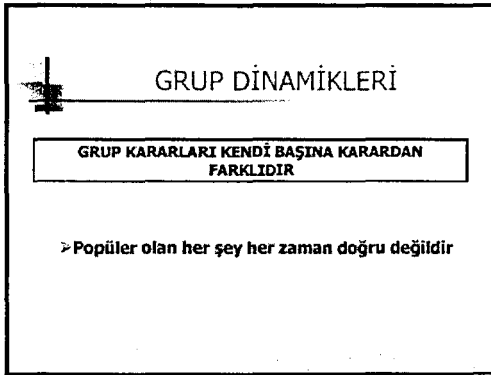
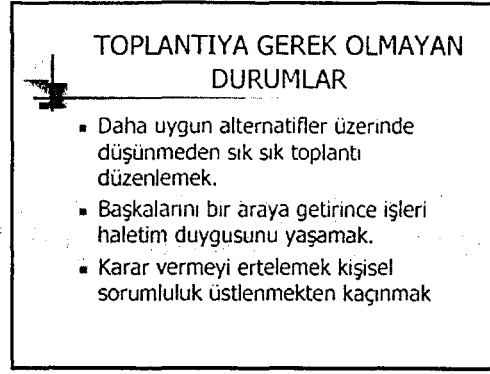
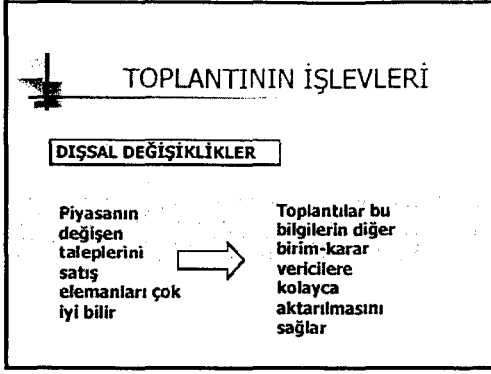
Yazılı not bilgi aktarır. ⇒ Toplantı, •Yorum sağlar •Yeni fikirler geliştirir. •Yeni çözümler yaratır

TOPLANTININ İŞLEVLERİ

BİLGİNİN YAYILMASI

Yazılı bilgi ilgi çekmiyor ⇒ Toplantıyla bilgi aktarmak daha etkili ⇒ İki yönlü iletişim

Şekil A-31. Toplantı Yönetimi 4



Şekil A-32. Toplantı Yönetimi 5

GRUP DİNAMİKLERİ

KARAR AŞAMASINDA GRUP BASKISI

- Toplantılarda insanlar konularını abartan kararlar alma eğilimindedirler
- Tek başlarına sorumlu tutulamayacakları düşüncesiyle daha riskli kararlar alınmasına neden olabilirler

TOPLANTI TUTANAĞI

- > Toplantıya katılanlar
- > Tartışılan konular
- > Anlaşmaya varılan konular
- > Eylem ve eylemelerden sorumlular

TOPLANTIYA KİM KATILACAK?

- > Başarmak istenen şeye bağlıdır
- > Diğerlerinin de görüşünü alın
- > Herkese ulaşarak haber verin, önem verildiklerini duyursunlar

HİKAYELEŞTİRME (Bilgi –Rapor-Plan)

1. SORUYU ORTAYA KOY
2. YANITLARI YAZDIR (10-15 DAKİKA)
3. YANITLARI TOPLA
4. İZİN VER YAZILANLARI OKUSUNLAR

İNSANLAR TOPLANTILARDA KONUYU NEDEN ÇARPITIRLAR

- > Diğerlerini etkilemek
- > İşlerinden kendilerini soyutlanmış hissetmek
- > Kendi seslerini duymaktan hoşlanmak

DÖRT TEMEL KURAL

- Katıl
- Odaklan
- Önem ver, konuya hakim ol , zamana uy
- Sonuçlandır

Şekil A-33. Toplantı Yönetimi 6

SAYMACA (NOMİNAL) GRUP TEKNIĐİ

1. DÜŐUNME
2. YAZMA
3. TARTIŐMA
4. SEŐME
5. BİRİNCİ PUANLAMA
6. İKİNCİ PUANLAMA
7. EYLEM

TEPKİ

"X konusunda sıkıntılarını anlayabiliyorum, ancak onunla bu toplantıda uğraşacak durumda değiliz. Onu Y konusundaki toplantıya getir burada üzerinde durulması gereken husus...."

"Seni burada durdurmam gerekiyor konumuza dönmeliyiz."

KARŐI TEPKİ

"Hemen bitireyim."

"Beni engellemeye mi çalışıyorsun?"


"Őimdiye kadar kimse bana böyle bir Őey söylememiŐti."

KARŐI TEPKİYE KARŐI

- İŐittiĐinizi belirtin
- Empati göstermeyin. yine konudan ayrılırsa,
- "Hayır konumuza devam etmeliyiz burada yoğunlaşmamız gereken konu..." diyerek
- Uyarıya devam edin
- otoriter bir ses tonu kullanın
- BaĐırmayın

Őekil A-34. Toplantı Yönetimi 7

EK-B SÜREÇLER VE TALİMATLAR

 YAZILIM KONFIGÜRASYON YÖNETİM PLANI KONTROL LİSTESİ	
Savronik	
I. Açıklık	Sonuç
<input type="checkbox"/> Bütün yazılım konfigürasyon ürünleri tanımlanmış mı?	
<input type="checkbox"/> Yazılım Ürünün her bir karakteristiği uygun terminoloji ile tanımlandı mı?	
II. Bütünlük	
<input type="checkbox"/> Kapak Sayfası var mı?	
<input type="checkbox"/> İçindekiler bölümü var mı?	
<input type="checkbox"/> Bütün şekiller, tablolar ve diyagramlar isimlendirilmiş mi?	
<input type="checkbox"/> Bütün şekil, tablo ve diyagramların referansları doğru mu?	
<input type="checkbox"/> Bütün terimler tanımlanmış mı?	
<input type="checkbox"/> Kapsam Bölümü var mı?	
<input type="checkbox"/> Yazılım Tanımı ve Konfigürasyon Bölümü var mı?	
<input type="checkbox"/> Konfigürasyon Denetimi ve Kontrollerinin nasıl yapılacağı belirtilmiş mi?	
<input type="checkbox"/> Konfigürasyon Yönetimi için organizasyondan bahsedilmiş mi?	
<input type="checkbox"/> Konfigürasyon faaliyetleri için yazılım araçları, tekniklerinden bahsedilmiş mi?	
<input type="checkbox"/> Konfigürasyon Değişikliklerinin nasıl uygulamaya konulacağı açıklanmış mı?	
III. İzlenebilirlik	
<input type="checkbox"/> Konfigürasyon kayıtlarının tutulması ve raporlanmasından bahsedilmiş mi?	
<input type="checkbox"/> Yukarıdaki tanımların tüm kullanıcı birimler tarafından izlenebilirliği sağlanmış mı?	
IV. Onaylanabilir	
<input type="checkbox"/> Gerçekleştirilmesi imkansız tanımlama var mı?	
<input type="checkbox"/> Her bir isterin test edilebilmesi için test stratejisi belirtilmiş mi?	
ONAYLAR:	
Bu bölüm, toplantıya katılanlar tarafından imzalanacaktır.	

Rev: 1.00

KAL.YF.005

Şekil B-1. Konfigürasyon Yönetim Planı Kontrol Listesi



YAZILIM KALİTE PLANI KONTROL LİSTESİ

	I. Açıklık	Sonuç
<input type="checkbox"/>	Bütün kalite gereksinimleri teknik olmayan bir dilde anlaşılır bir şekilde yazıldı mı?	
<input type="checkbox"/>	Yazılım Ürünün her bir karakteristiği uygun terminoloji ile tanımlandı mı?	
	II. Bütünlük	
<input type="checkbox"/>	Kapak Sayfası var mı?	
<input type="checkbox"/>	İçindekiler bölümü var mı?	
<input type="checkbox"/>	Bütün şekiller, tablolar ve diyagramlar isimlendirilmiş mi?	
<input type="checkbox"/>	Bütün şekil, tablo ve diyagramların referansları doğru mu?	
<input type="checkbox"/>	Bütün terimler tanımlanmış mı?	
<input type="checkbox"/>	Giriş Bölümü var mı?	
<input type="checkbox"/>	Giriş Bölümünde, Amaç, Kapsam, Kalite Planı Sürekliliği, Referans Dokümanlar, Diğer Planlara Referanslar ve Tanımlar ve Kısaltmalar yapılmış mı?	
<input type="checkbox"/>	Proje Tanımı Bölümü var mı?	
<input type="checkbox"/>	Bu bölümde, Projeye Genel Bakış ve Varsayımlar ve teslim edilecek ürünlerden bahsedilmiş mi?	
<input type="checkbox"/>	Yönetim tanımlarının yapıldığı Bölüm var mı? Bu bölümde,	
<input type="checkbox"/>	Yazılım Geliştirme İşlemi	
<input type="checkbox"/>	Organizasyon	
<input type="checkbox"/>	Düzeltilme faaliyetleri	
<input type="checkbox"/>	Yardımcı Firma Yönetimi	
<input type="checkbox"/>	Konfigürasyon Yönetimi	
<input type="checkbox"/>	Paket Yazılımlar	
<input type="checkbox"/>	Kayıtların Muhafazası	
<input type="checkbox"/>	Yazılım Ortamının Yönetimi ve Saklanması konuları var mı?	
<input type="checkbox"/>	Yazılım Mühendisliği Bölümü var mı? Bu bölümde,	
<input type="checkbox"/>	Yazılım Mühendislik Ortamı,	
<input type="checkbox"/>	Metotlar, Yöntemler, standartlar	
<input type="checkbox"/>	Geliştirme Dokümantasyonundan söz edilmiş mi?	
<input type="checkbox"/>	Değerlendirme ve Doğrulama bölümü var mı?	

Rev. 1.00

KAL.YF.004

Şekil B-2. Kalite Planı Kontrol Listesi 1



Savronik

YAZILIM KALİTE PLANI KONTROL LİSTESİ

<input type="checkbox"/>	Eksik bilgiler belirtilmiş mi?	
<input type="checkbox"/>	Tasarım kısıtları belirtilmiş mi?	
<input type="checkbox"/>	Donanım gereksinimleri belirtilmiş mi?	
<input type="checkbox"/>	Girdilere ilişkin gereksinimler belirtilmiş mi?	
<input type="checkbox"/>	Çıktılara ilişkin gereksinimler belirtilmiş mi?	
<input type="checkbox"/>	Raporlama gereksinimleri belirtilmiş mi?	
<input type="checkbox"/>	Yüklemeye ilişkin gereksinimler belirtilmiş mi?	
<input type="checkbox"/>	Kritik gereksinimler belirtilmiş mi?	
	III. İzlenebilirlik	
<input type="checkbox"/>	Yukarıdaki tanımların tüm kullanıcı birimler tarafından izlenebilirliği sağlanmış mıdır?	
	IV. Onaylanabilir	
<input type="checkbox"/>	Gerçekleştirilmesi imkansız gereksinim var mı?	
<input type="checkbox"/>	Her bir gereksinimin test edilebilmesi için test stratejisi belirtilmiş midir?	
ONAYLAR:		
Bu bölüm, toplantıya katılanlar tarafından imzalanacaktır.		

Şekil B-3. Kalite Planı Kontrol Listesi 2



Yazılım İsterleri Kontrol Listesi

I. Açıklık

- Bütün gereksinimler teknik olmayan bir dilde anlaşılır bir şekilde yazıldı mı?
- Bir'den fazla anlamı olan gereksinim açıklaması var mı?
- Yazılım Ürünün her bir karakteristiği uygun terminoloji ile tanımlandı mı?
- Özel terimlerin açıklandığı tanımlar bölümü var mı?
- Gereksinimlerin bağımsız bir grubun anlayacağı şekilde mi?

II. Bütünlük

- İçindekiler bölümü var mı?
- Bütün şekiller, tablolar ve diyagramlar isimlendirilmiş mi?
- Bütün şekil, tablo ve diyagramların referansları doğru mu?
- Bütün terimler tanımlanmış mı?
- Bütün terimlerin index'i var mı?
- Bütün ölçüm birimleri tanımlanmış mı?
- Geliştirme fazının başlamamasından dolayı tamamlanmamış gereksinim bilgileri belirtilmiş mi?
- Eksik bilgiler belirtilmiş mi?
- Detaylandırılması gereken gereksinim var mı?
- Fazla detaylandırılmış gereksinim var mı?
- Bütün gereksinimler tanımlanmış mı?
- Fonksiyonlile ilgili bütün gereksinimler tanımlanmış mı?
- Sizde zor olduğu hissi uyandıran gereksinim var mı?
- Performans gereksinimleri belirtilmiş mi?
- Tasarım kısıtları belirtilmiş mi?
- Bütün özellikler belirtilmiş mi?
- Dış arayüz noktaları belirtilmiş mi?
- Veritabanı gereksinimleri belirtilmiş mi?
- İletişim gereksinimleri belirtilmiş mi?
- Donanım gereksinimleri belirtilmiş mi?
- Girdilere ilişkin gereksinimler belirtilmiş mi?
- Çıktıları ilişkin gereksinimler belirtilmiş mi?

HAZIRLAYAN

ONAYLAYAN

KAL.F.075

Şekil B-4. Yazılım İsterleri Kontrol Listesi



Yazılım İsterleri Kontrol Listesi

- Raporlama gereksinimleri belirtilmiş mi?
- Güvenlik gereksinimleri belirtilmiş mi?
- Bakılabilirlik için gerekli gereksinimler belirtilmiş mi?
- Yüklemeyle ilişkin gereksinimler belirtilmiş mi?
- Kritik gereksinimler belirtilmiş mi?

III. Tutarlılık

- Gereksinimler arasında herhangi bir çakışma var mı?
- Mantık olarak çakışan gereksinim var mı?

IV. İzlenebilirlik

- Gereksinimlerin kullanıcı ihtiyaçları arasında izlenebilirliği sağlanmış mıdır?
- Gereksinimler ile gereksinimlerin kaynağı doküman veya kişi arasında izlenebilirlik sağlanmış mıdır?
- Gereksinimler tasarım dokümanında izlenebilecek midir?
- Gereksinimler, yazılım biriminde izlenebilecek midir?

V. Onaylanabilir

- Gerçekleştirilmesi imkansız gereksinim var mı?
- Her bir gereksinimin test edilebilmesi için test stratejisi belirtilmiş midir?

HAZIRLAYAN

ONAYLAYAN

KAI F 075

Şekil B-5. Yazılım İsterleri Kontrol Listesi 2



YAZILIM KİŞİSEL DEĞERLENDİRME FORMU

Lütfen uygun gördüğünüz seçeneği daire içine alınız

I. Problem Karakteristikleri

1. Yazılım Ekibinin belirttiği sorunların gerçek zorluk derecesini belirtiniz.

1 2 3 4 5
Kolay Normal Zor

2. Yazılım Geliştirme Takvimi ne kadar sıkıştıktı?

1 2 3 4 5
Rahat Normal Sıkışık

3. Yazılımdan istenenler ne sıklıkla değişiyor?

1 2 3 4 5
Az Normal Çok

4. Yazılım İsterleri dokümanlarının yeterlilik, tutarlılık ve doğruluk derecesini belirtiniz.

1 2 3 4 5
Az Normal Çok

5. Ne kadar dokümantasyon yapıyor?

1 2 3 4 5
Az Normal Çok

6. Gereksinimlerin gözden geçirme kalitesini belirtiniz.

1 2 3 4 5
Az Normal Çok

II. Kişisel Özellikler: Teknik Personel

7. Yazılım Ekibinin kalite ve yeterliliğini belirtiniz.

1 2 3 4 5
Az Normal Çok

8. Yazılım ekinin proje konusu üzerindeki deneyimini belirtiniz.

1 2 3 4 5
Az Normal Çok

9. Yazılım Ekibinin, yazılım geliştirme araçları üzerindeki deneyimini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

10. Proje boyunca yazılım ekibinin değişme oranını belirtiniz.

1 2 3 4 5
Az Normal Çok

Ali Tüzünkan
Proje Mühendisi
HAZIRLAYAN

Müveddet Başda İşık
Kalite Güvence Müdürü
ONAYLAYAN

KALF.075

Şekil B-6. Kişisel Değerlendirme Formu 1



YAZILIM KİŞİSEL DEĞERLENDİRME FORMU

III. Kişisel Özellikler: Teknik Yönetim

11. Proje yönetimini kalite ve yeterliliğini değerlendiriniz.

1 Az 2 3 Normal 4 5 Çok

12. Proje yönetiminin, proje konusu ile ilgili deneyimini değerlendiriniz.

1 Az 2 3 Normal 4 5 Çok

13. Proje süresince proje yönetimi ne kadar tutardı?

1 Az 2 3 Normal 4 5 Çok

14. Proje planlama ne derece tutardı?

1 Az 2 3 Normal 4 5 Çok

15. Proje planlarına ne derece uyuldu?

1 Az 2 3 Normal 4 5 Çok

IV. Süreç Özellikleri

16. Modern programlama teknikleri ne derece kullanıldı? (Akış şemaları, top-down geliştirme, yapısal programlama ve kod okuma)

1 Az 2 3 Normal 4 5 Çok

17. Yazılım Ekibi gereksinim değişiklik istekleri, gereksinim soruları ve arayüz tanımlamalarını kaydetmek için ne kadar disiplinli sistemleri var?

1 Az 2 3 Normal 4 5 Çok

18. Gereksinim analiz metodolojisi ne kadar yeterli?

1 Az 2 3 Normal 4 5 Çok

19. Tasarım metodolojisi ne kadar yeterli?

1 Az 2 3 Normal 4 5 Çok

20. Test metodolojisi ne kadar yeterli?

1 Az 2 3 Normal 4 5 Çok

Ali Tüzünkan
Proje Mühendisi
HAZIRLAYAN

Müveddet Başda İşik
Kalite Güvence Müdürü
ONAYLAYAN

KAL F 075

Şekil B-7. Kişisel Değerlendirme Formu 2



YAZILIM KİŞİSEL DEĞERLENDİRME FORMU

21. Hangi yazılım araçları kullanıldı?

1 2 3 4 5
Az Normal Çok

Compiler

Linker

Editor

Test Automation

Version Control

Simulator

22. Yazılım Ekibi ne derece test planları hazırlayıp, uyguladı?

1 2 3 4 5
Az Normal Çok

23. Yazılım Ekibi için Kalite Güvence prosedürü ne derece yeterliydi?

1 2 3 4 5
Az Normal Çok

24. Yazılım Ekibi için mevcut Konfigürasyon Yönetim Prosedürünü değerlendiriniz.

1 2 3 4 5
Az Normal Çok

V. Çevresel Özellikler

25. Yazılım Ekibinin, geliştirme araçlarına erişimini nasıl değerlendiriyorsunuz?

1 2 3 4 5
Kolay Normal Zor

26. Terminal/programcı oranı nedir?

1 2 3 4 5
8.1 4.1 2.1 1.1 1.2

27. Yazılım ekibinin donanım eksiklerini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

28. Proje boyunca yazılım ve donanım araçlarının değişim sıklığı nedir?

1 2 3 4 5
Az Normal Çok

Ali Tüzünkan
Proje Mühendisi
HAZIRLAYAN

Müveddet Başda Işık
Kalite Güvence Müdürü
ONAYLAYAN

KAI F 075

Şekil B-8. Kişisel Değerlendirme Formu 3



Savronik

KODU : SW.KAL.A.001
REV NO : 1.00
REV. TARİH : 28.6.2001

YAZILIM KİŞİSEL DEĞERLENDİRME FORMU

29. Yazılım Araçlarının yeterliliğini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

VI. Ürün Özellikleri

30. Yazılım Paketinin müşteri isterlerini karşıla oranını değerlendiriniz.

1 2 3 4 5
Az Normal Çok

31. Teslim edilmiş yazılım paketinin kalitesini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

32. Yazılımın Tasarım Kalitesini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

33. Dokümantasyonun yeterliliğini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

34. Yazılım ürünlerinin zamanında bitirilip bitirilmediğini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

35. Kabul testlerinin kolaylık derecesini değerlendiriniz.

1 2 3 4 5
Az Normal Çok

36. Yazılım Ürünlerinin kullanım kolaylıklarını değerlendiriniz.

1 2 3 4 5
Kolay Normal Zor

Ali Tüzüncan
Proje Mühendisi
HAZIRLAYAN

Müveddet Başda Işık
Kalite Güvence Müdürü
ONAYLAYAN

KAL.F.075

Şekil B-9. Kişisel Değerlendirme Formu 4



Yazılım Sürümü Kontrol Listesi

Faaliyetler

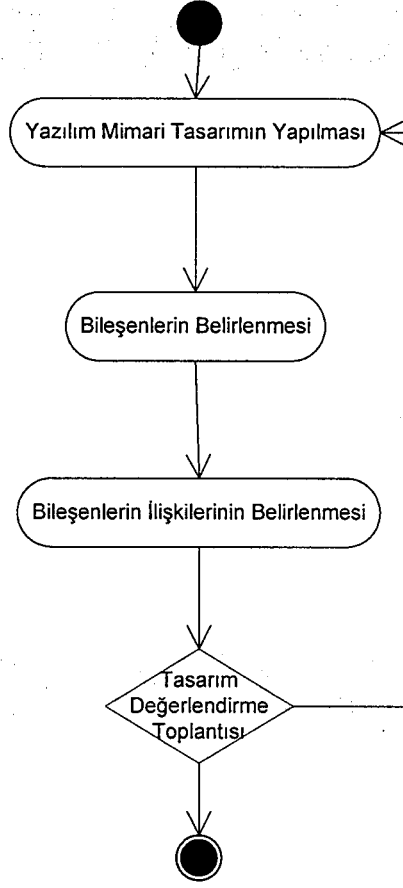
I. Mühendislik İşleri	Sorumlu Birim
<input type="checkbox"/> Versiyon bilgilerini güncelle.	Yazılım Ekibi
<input type="checkbox"/> Debug kodlarını kaldır.	Yazılım Ekibi
<input type="checkbox"/> Bilinen hasarları kaldır.	Yazılım Ekibi
II. Kalite Güvence Faaliyetleri	
<input type="checkbox"/> Bilinen bütün problemlerin giderildiğini kontrol et.	Yazılım Ekibi/Kalite
<input type="checkbox"/> Son bir defa test et.	Yazılım Ekibi/Kalite
<input type="checkbox"/> Programı CD'den temiz bir bilgisayara kur ve dene.	Yazılım Ekibi
<input type="checkbox"/> Programı eski versiyonun üzerine kur ve dene.	Yazılım Ekibi
<input type="checkbox"/> Programı kaldırmayı dene.	Yazılım Ekibi
III. Sürüm Faaliyetleri	
<input type="checkbox"/> Dağıtılacak Dosya Listesini oluştur.	Yazılım Ekibi
<input type="checkbox"/> Dosyaların tarih/zaman bilgilerini değiştir.	Teknik Destek
<input type="checkbox"/> Sarı ve Kırmızı disket ve cd'leri oluştur.	Teknik Destek
<input type="checkbox"/> Sarı ve Kırmızı disket gerekli bütün dosyaların olduğunu kontrol et.	Teknik Destek
<input type="checkbox"/> Sarı ve Kırmızı diskette virüs taraması yap.	Teknik Destek
<input type="checkbox"/> Ana kopyanın olduğu ortamda scandisk yap.	Teknik Destek
<input type="checkbox"/> Geliştirme bilgisayarının bir kopyasını CD'ye kopyala.	Teknik Destek
IV. Dokümantasyon Faaliyetleri	
<input type="checkbox"/> Okubeni.txt'teki versiyon bilgilerini kontrol et.	Teknik Destek
<input type="checkbox"/> Yazılım Paketinin Dokümantasyon Merkezi'ne kaydını yaptır.	Yazılım Ekibi
<input type="checkbox"/> Yardım dosyalarındaki versiyon bilgilerini kontrol et.	Yazılım Ekibi
V. Diğer Faaliyetler	
<input type="checkbox"/> Lisans anlaşmasını gözden geçir.	Proje Müdürü

Ali Tüzünkan
Proje Mühendisi
HAZIRLAYANMüveddet Başda IŞIK
Kalite Güvence Müdürü
ONAYLAYAN

KAL.F.075

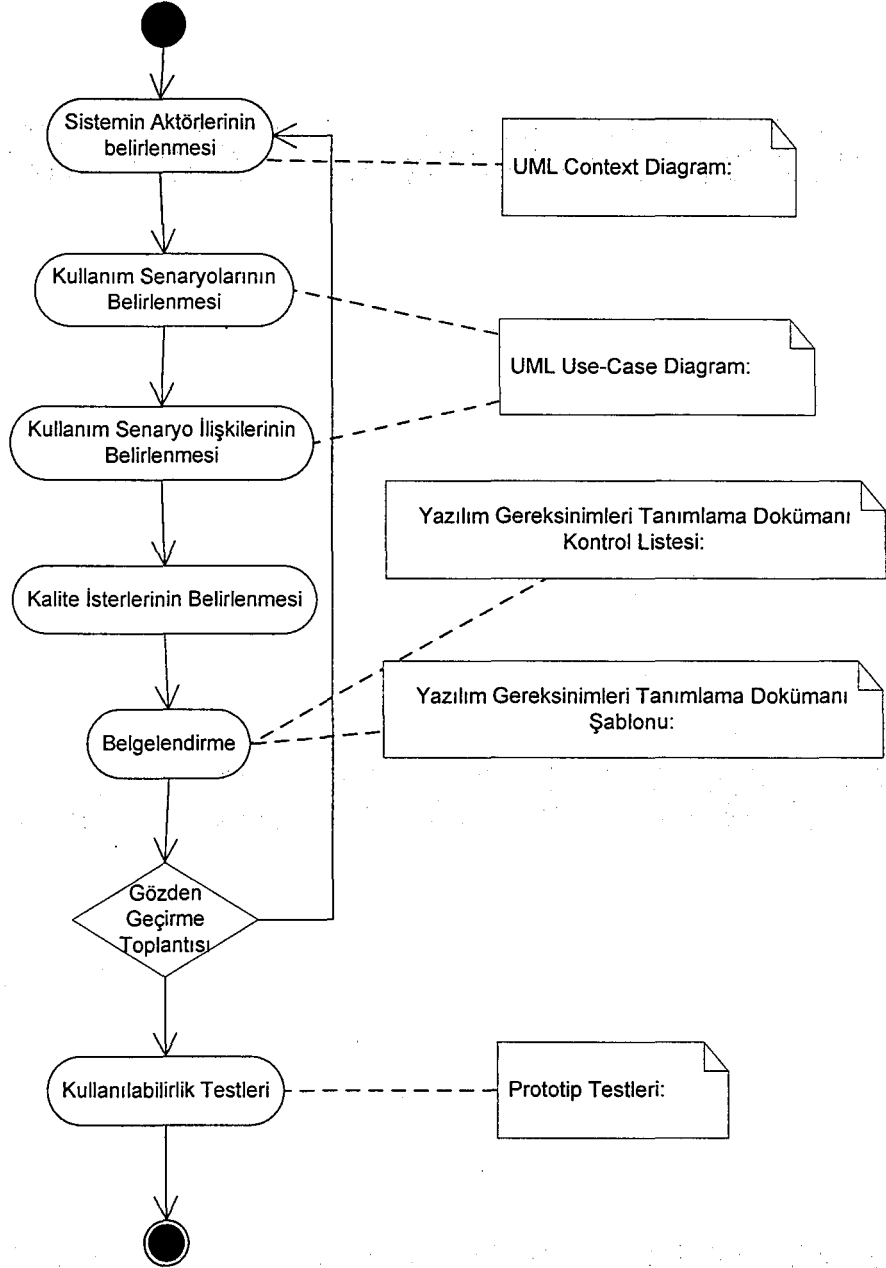
Şekil B-10. Sürüm Kontrol Listesi

Yazılım Tasarım Süreci



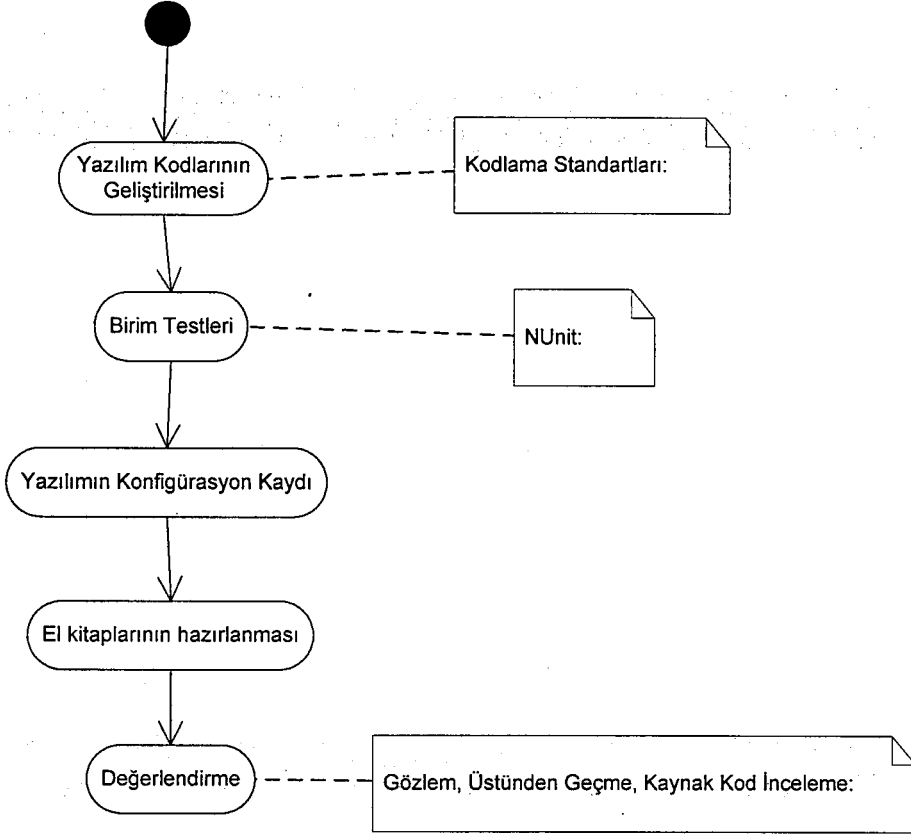
Şekil B-11. Yazılım Tasarım Süreci İş Akışı

Yazılım İsterleri Süreci



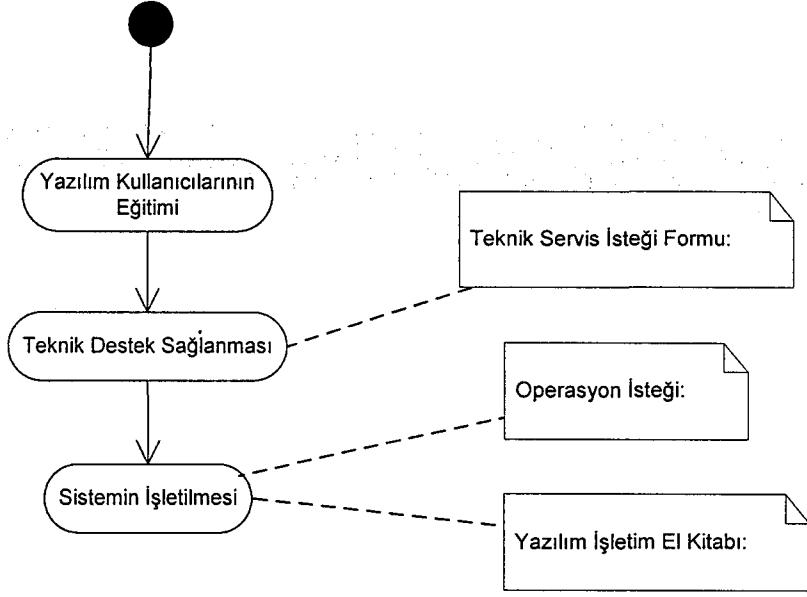
Şekil B-12. Yazılım İsterleri Süreci İş Akışı

Kod Geliştirme Süreci



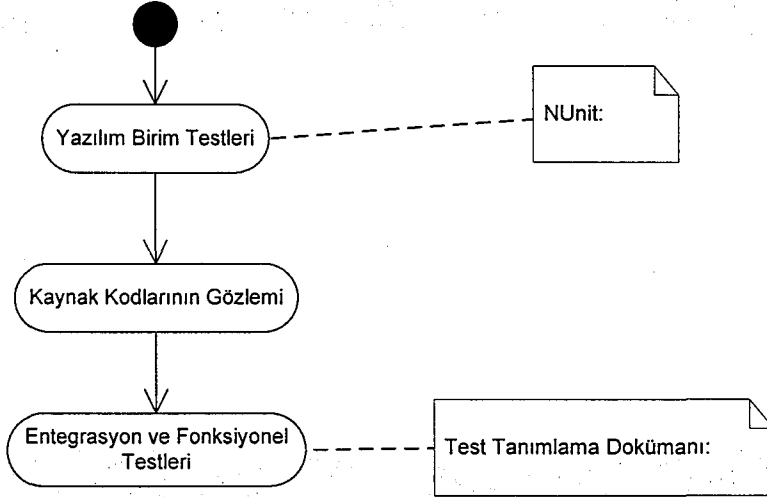
Şekil B-13. Kod Geliştirme Süreci İş Akışı

Yazılım İşletim Süreci



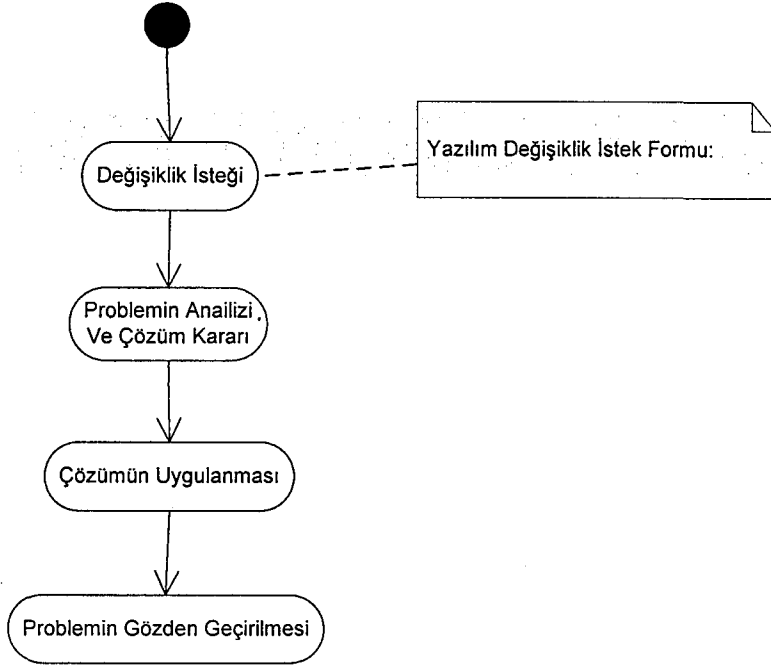
Şekil B-14. Yazılım İşletim Süreci İş Akışı

Yazılım Test Süreci



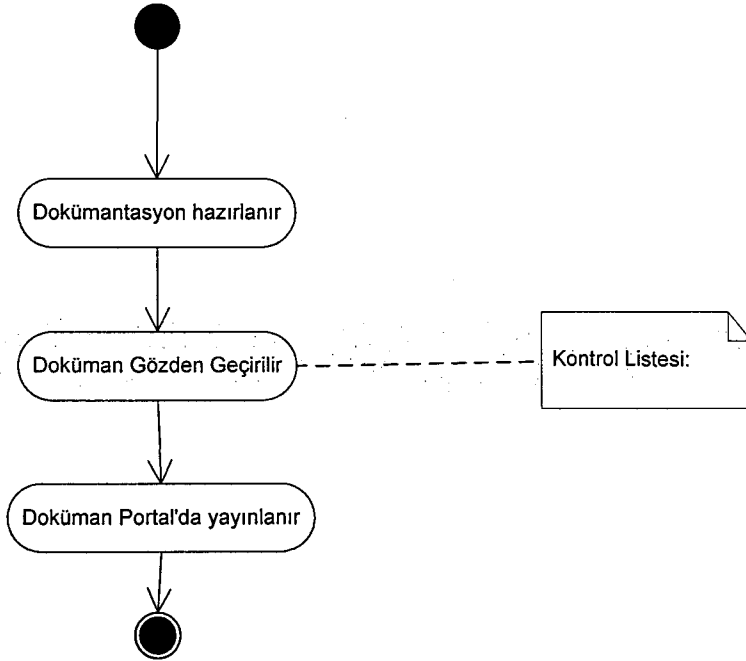
Şekil B-15. Yazılım Test Süreci

Yazılım Bakım Süreci



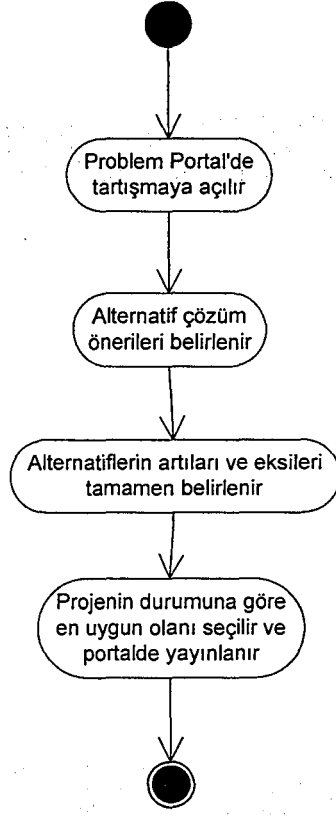
Şekil B-16. Yazılım Bakım Süreci İş Akışı

Belgelendirme Süreci



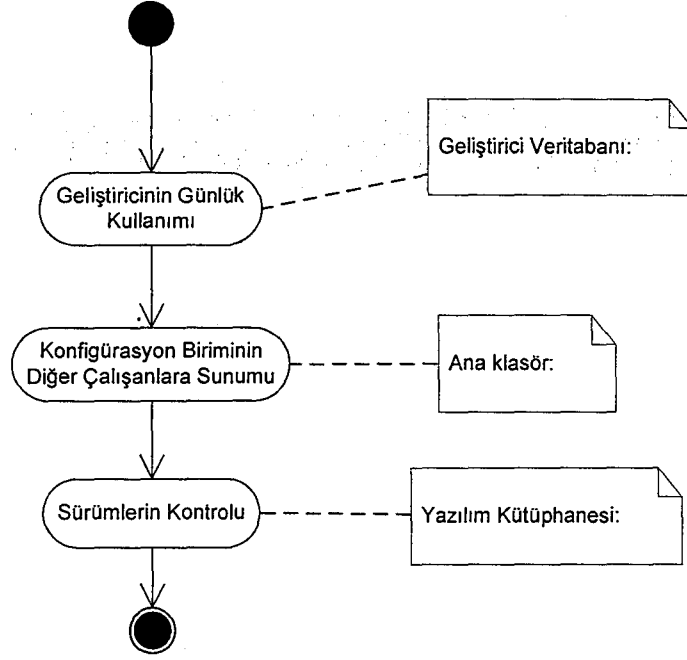
Şekil B-17. Belgelendirme Süreci İş Akışı

Problem Çözme Süreci



Şekil B-18. Problem Çözme Süreci

Yazılım Konfigürasyon Yönetimi Süreci



Şekil B-19. Konfigürasyon Yönetimi Süreci İş Akışı