

**T.C.  
TRAKYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**Yazılım Bakım Altkarakteristiklerinin  
ISO Tabanlı Modeller Kullanılarak Ölçülmesi**

**Kaan KURTEL  
Doktora Tezi**

**Bilgisayar Mühendisliği Anabilim Dalı  
Tez Yöneticisi: Prof. Dr. Şaban EREN  
2009 - EDİRNE**

**T.C.  
TRAKYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**Yazılım Bakım Altkarakteristiklerinin  
ISO Tabanlı Modeller Kullanılarak Ölçülmesi**

**Kaan KURTEL**

**Doktora Tezi  
Bilgisayar Mühendisliği Anabilim Dalı**

Bu tez 6 Kasım 2009 tarihinde aşağıdaki jüri tarafından kabul edilmiştir.

**Prof. Dr. Şaban EREN  
Tez Yöneticisi ve Başkan**

**Doç. Dr. Yılmaz KILIÇASLAN  
Üye**

**Doç. Dr. Ahmet KOLTUKSUZ  
Üye**

**Yrd.Doç.Dr. Erdem UÇAR  
Üye**

**Yrd.Doç.Dr. Aydın CARUS  
Üye**

**ÖZET****YAZILIM BAKIM ALTKARAKTERİSTİKLERİNİN  
ISO TABANLI MODELLER KULLANILARAK ÖLÇÜLMESİ**

Kaan KURTEL

Doktora Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi : Prof. Dr. Şaban EREN

Kasım, 2009, 145 sayfa

İnternetin yaygın olarak kullanılmasını takiben, bilişim sistemlerine ve onun en önemli bileşeni olan yazılıma yönelik ilginin arttığı ve iş yapma biçiminin önemli ölçüde değiştirdiği görülmektedir. Bu değişikliğin etkilediği önemli alanlardan birisi de, yazılımın yoğun olarak kullanıldığı tedarik zinciri ve lojistik yönetimidir. Bu alandaki ilginin temel nedeniyse, tedarik zinciri ve lojistik yönetimi çözümlerinin küresel pazarda rekabet avantajı sağlamadaki etkileri gibi görülmektedir.

Bütün bu ilgiye, yazılım ve lojistik alanındaki yoğun akademik çalışmalara ve uygulamalara rağmen, lojistik sektörüne hizmet veren yazılımları geliştirmenin ve geliştirilen yazılımları hizmette tutmanın zorluğu artmaktadır. Bu zorluğu aşmak ise ancak daha iyi yazılım ürünleri üretmekle mümkündür. Bu bağlamda, her geçen gün karmaşıklığı artan ve daha heterojen hale gelen iş hayatında, başarılı bir yazılım mühendisliği uygulaması gerçekleştirilebilmek için yazılımın ölçülmesi gereklidir.

Ölçmeye yönelik çalışmalar, gerek yazılım geliştirme ve gerekse de yazılım kalite ve süreç çalışmaları açısından vazgeçilmezdir ve aynı zamanda kaliteli bir yazılım ürünü için de kritik başarı unsurudur. Yazılım mühendisliği açısından yazılımı ölçmeye yönelik uygulamalara bakıldığında, yazılım geliştirmeye yönelik süreçlerin ölçülmesine yönelik yoğun bir ilgi olduğu görülmektedir. Ancak benzer ilginin yazılım ürününün ölçülmesinde de ortaya konması gerekmektedir. Tez bu alandaki boşluğu doldurmayı amaçlamaktadır.

Bu tezde sunulan çalışmada, ISO tabanlı modeller kullanılarak, Lojistik Merkez (LM) yazılım ürünleri ölçülmüş, böylece LM için artırılmış ürün kalite kavramı ortaya konmuş, kullanılan yöntem ve elde edilen sonuçların benzer sistemler için örnek oluşturmasına çalışılmıştır.

Tez, ISO standartlarındaki uygulamaya yönelik belirsizliklerin ve standartların çeşitliliğinden kaynaklanan karmaşıklığın azaltılmasını sağlamaktadır. ISO tabanlı modeller; LM yazılımının taşıdığı özelliklere benzer özelliklere sahip diğer yazılım ürünlerinin ölçülmesindeki süreçleri desteklediğinden, bu çalışmada ISO tabanlı modeller kullanılmıştır. Tez, bunun yanı sıra; yazılım mühendisliği, yazılım kalite güvencesi ve yazılım bakım literatürüne de katkı yapmaktadır.

**Anahtar sözcükler:** Yazılım ölçümü, yazılım bakımı, ISO temelli modeller, lojistik merkez.

## **ABSTRACT**

### **THE MEASURING OF THE SOFTWARE MAINTAINABILITY SUBCHARACTERISTICS BY UTILIZATING ISO-BASED MODELS**

Kaan KURTEL

Ph.D. in Computer Engineering

Supervisor : Prof. Dr. Şaban EREN

November, 2009, 145 pages

The widespread use of internet technologies has encouraged and facilitated the development and distribution of knowledge resources across the global supply chain. One of the essential reasons for this is that as internet technologies expand, there has been an increase in the domain of supply chain and logistics management, which is recognized as a major area of competition for businesses at the present time. Clearly, software will always play an important role in this dynamic business environment.

Despite intensive academic studies on topics of software design, many problems still seem to be unresolved, particularly in the software systems of the logistics sector. In addition, there has been an increase in the number of problems involving the development of complete enterprise logistics software, where the maintenance of the required service levels based on individual customer business needs are becoming more difficult. Therefore, the continual development of improved software products is necessary to overcome these drawbacks which brings forth the necessity that software product values must be measured in this complex, widespread and heterogeneous business environment.

Software measurement studies are complemented by software engineering concepts, both of which aim to encourage the increase in software development efforts and quality improvement. There has been great interest in the measurement of software development processes in the last two decades; however, dealing with the measurement of software products also has equal importance. Hence, this thesis aims to contribute to this area which requires new developments and improvements.

The research study that is presented in this thesis measures the Logistics Center's software products and consequently sets forth an enhanced product quality concept for Logistics Center based on similar systems using ISO-based models. The methods and results in this study are aimed to be a sample model for the generation of similar systems in the future.

This thesis focuses on solutions to overcome the ambiguities and problems that the organizations face during the implementation period of the ISO-based software measurement standards. Also, this study contributes to: software product measurement methodology, software maintenance literature, and practical software applications.

**Keywords:** Software measurements, software maintenance, ISO-based models, logistics center.

*Elif ve Deniz'e*

## TEŞEKKÜR

Hocam Sn. Prof. Dr. Şaban Eren'e (Yaşar Üniversitesi, İstatistik Bölümü) güveni, desteği, deneyim ve bilgisini paylaştığı için çok teşekkür ederim. Hakkını ancak iyi mühendisler yetiştirerek ödeyebilirim.

Çok yoğun iş tempolarına rağmen, bilgi ve verilerini benimle paylaşan, çalışmamı pratik olarak destekleyen Bimar Bilgi İşlem Hizmetleri A.Ş. yazılım geliştirme müdürü Sn. Murat Özemre ve çalışma arkadaşları Sn. Hasan Özkeser, Sn. Hülya Yarkın, Sn. Emel Öndül ve Sn. Güner Mutlu'ya çok teşekkür ederim. Onların desteği çalışmayı anlamlı yaptı.

Doktora çalışmam süresince, verdikleri destek ve olumlu yönlendirmeleri için Sn. Prof. Dr. Tunçdan Baltacıoğlu'na (İzmir Ekonomi Üniversitesi, Lojistik Bölümü), Sn. Sn. Yrd. Doç. Dr. Erdem Uçar'a ve Doç. Dr. Yılmaz Kılıçaslan'a (Trakya Üniversitesi, Bilgisayar Mühendisliği Bölümü) çok teşekkür ederim.

Çalışmamın kritik anlarında çözüm yolunu en basit şekilde gösteren Sn. Dr. Jean-Marc Desharnais'a çok teşekkür ederim.

Akademik çalışmalarında yardımlarını gördüğüm, Sn. Doç. Dr. Ahmet Koltuksuz'a (Yaşar Üniversitesi), Sn. Yrd. Doç. Dr. Serap Atay'a, Sn. Y. Müh. Selma Tekir'e (İzmir Yüksek Teknoloji Enstitüsü, Bilgisayar Mühendisliği Bölümü) ve Sn. Yrd. Doç. Dr. Mehmet S. Ünlütürk'e (İzmir Ekonomi Üniversitesi, Yazılım Mühendisliği Bölümü) çok teşekkür ederim.

Çalışmalarım süresinde beni destekleyen ve zorlukları aşmamda yardımcı olan, Sn. Yrd. Doç. Dr. Yiğit Kazançoğlu'na, Sn. Yrd. Doç. Dr. Coşkun Atay'a, Sn. Y. Müh. Cemalettin Öztürk'e, Sn. Y. Müh. İlker Korkmaz'a, Sn. Y. İstatistikçi Umut Avcı'ya (İzmir Ekonomi Üniversitesi), Sn. Y. Müh. Mete Eminağaoğlu'na (Yaşar Üniversitesi) arkadaşlık, dostluk ve desteklerinden dolayı çok teşekkür ederim.

Kendisine hak ettiği zamanı ayıramama rağmen, çalışmalarım beni anlayışla karşılayıp desteklediği için sevgili eşim Elif Kurtel'e ve varlığı ile bizi her zaman mutlu eden oğlum Deniz Kurtel'e çok teşekkür ederim.



## ŞEKİLLER DİZİNİ

Şekil	Sayfa
1.1 Araştırma alanı ürün uzayı .....	3
1.2 Yazılım ölçüm uzayı ve araştırma alanı .....	4
1.3 Tezin yapısı .....	6
2.1 Yazılım ölçüm süreci ile ilişkili standartlar .....	10
2.2 ISO 15939 ölçme süreci modeli .....	11
2.3 Ölçüm modelindeki temel ilişkiler (ISO 15939) .....	13
2.4 Kalite yaşam döngüsü (ISO 9126-1) .....	16
2.5 Sistemin çeşitli bileşenlerine ait kalite modelleri .....	16
2.6 Karakteristikler, altkarakteristikler ve özellikler arasındaki hiyerarşik yapı ...	17
2.7 Kalite karakteristikleri, altkarakteristikleri ve özellikler .....	20
2.8 Değerlendirme süreci ile değerlendirmeye destek standartlar arasındaki ilişki	21
2.9 Yazılım yaşam döngüsünde kalite (ISO 14598-1) .....	22
2.10 ISO 15939 sürecine ait altsüreçler ve ISO 9126 ile ilişkisi .....	27
3.1 LM'in etki alanı .....	29
4.1 Karar hiyerarşisi ağacı .....	33
5.1 Bakımcının özellikleri .....	41
5.2 Bakım faaliyetlerine ait insan varlıkları .....	41
5.3 Bakım faaliyetleri .....	42
5.4 IEEE Std. 1219 bakım süreci faaliyetleri .....	43
5.5 ISO 12207 yazılım yaşam çevrimi süreçleri ve bakım .....	44
5.6 ISO 14764 yazılım bakım süreci .....	45
5.7 Yazılım bakımı kavramsal modeli .....	45
6.1 Araştırma yöntemi ve standartlarla olan ilişkisi .....	49
7.1 LM'in uygulama altyapısı .....	51
7.2 LM'in yazılım geliştirme ve bakım süreci .....	55
7.3 LM'in geçici çözüme yönelik işlemleri .....	56
7.4 LM'in bakım sistemi ana akışı .....	57

## ŞEKİLLER DİZİNİ

Şekil	Sayfa
7.5 Mainframe ortamındaki bakım sayıları .....	58
7.6 Mainframe ortamındaki bakım sayılarının istatistiksel sonuçları.....	59
7.7 Mainframe ortamındaki bakım sayılarının histogramı ve istatistik dağılımı ...	60
7.8 Web ortamındaki bakım sayıları .....	61
7.9 Web ortamındaki bakım sayılarının istatistiksel sonuçları .....	62
7.10 Web ortamındaki bakım sayılarının histogramı ve istatistik dağılımı .....	63
7.11 LMBS'ne ait ölçme gereksinimlerinin belirlenmesi .....	64
7.12 LM için karar hiyerarşisi ağacı .....	65
7.13 LM'in web ortamına ait ölçüm aralığı .....	76
7.14 Planlama sürecinin panoramik gösterimi .....	83

## ÇİZELGELER DİZİNİ

<b>Çizelge</b>	<b>Sayfa</b>
1.1 ISO 9126 yazılım kalite karakteristikleri ve altkarakteristikleri .....	4
2.1 ISO 15939 standartına göre ölçüm yapısı .....	14
2.2 ISO 9126 standartına ait kalite karakteristikleri .....	18
2.3 ISO 9126-2 temel ölçüler ve ölçüm birimleri .....	25
2.4 ISO 9126-3 temel ölçüler ve ölçüm birimleri .....	26
3.1 Yazılıma ait risklerin problem yaratma olasılığına göre sınıflaması ve lojistik merkez yazılımının konumu .....	31
4.1 Bağıl önem ölçeği .....	33
4.2 Ortalama rassal tutarlılık indeksi .....	36
5.1 ISO 9126-2 ve ISO 9126-3'e göre bakım yapılabirlik iç ve dış ölçüleri.	47
7.1 LM Acentelik Paketi temel taşları .....	52
7.2 Mainframe ortamında yapılan bakımların istatistik değerleri .....	59
7.3 Web ortamında yapılan bakımların istatistik değerleri .....	62
7.4 LM yazılım üstkarakteristikleri önem derecesi .....	68
7.5 LM yazılım karakteristikleri önem derecesi .....	69
7.6 LM yazılımını en iyi temsil eden üstkarakteristikler .....	70
7.7 ISO 9126 standartına göre LM'in bakım ortamından elde edilen ölçüm seti	73
7.8 LM'in bakım ortamından elde edilen temel ölçüler .....	77
7.9 LM'in bakım ortamından elde edilen ölçüm seti .....	78
8.1 Temel ölçülere ait ölçüm sonuçları .....	84
8.2 LM bilgi gereksinimini karşılayan ölçüm sonuçları .....	85
9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu .....	87

## EKLER DİZİNİ

<b>Ek</b>	<b>Sayfa</b>
A McCall'ın yazılım kalite modeli .....	105
B Bohm'in yazılım kalite modeli .....	106
C ISO 9126-2 ve 9126-3 teknik raporlarına ait temel ölçülerin, yazılım kalite altkarakteristikleri ile olan ilişkisi .....	107
Ç Üst kalite faktörlerine göre seçenekleri değerlendirme tabloları .....	112
D ISO 9126 standartına göre değiştirilebilirlik, analiz edilebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler .....	116
E Veri tabloları .....	122
E1 Mainframe Ortamına Ait Genel Hata Üretim ve Bakım Verileri – Yapılan Bakımlar .....	122
E2 Web Ortamına Ait Genel Bakım Verileri .....	123
E3 LM Müşteri Değişiklik İstekleri Veri Alanları - Web Ortamı .....	124
E4 LM Müşteri Hata İstekleri Veri Alanları - Web Ortamı .....	125
E5 Temel ölçülere ait aylık ölçüm sonuçları (01.01.2008-31.07.2009) .....	126
F Temel Ölçüler .....	127

**KISALTMALAR DİZİNİ**

AHP	<i>Analytic Hierarchy Process</i>
AHS	Analitik hiyerarşi süreci
BY	Bakım yapılabilirlik
CMMI	<i>Capability Maturity Model Integration</i>
ELV	Elverişlilik
GK	Gereksinimlerin kalitesi
GSV	Gerçekleştirim ve sınama verimliliği
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISO	<i>International Standards Organization</i>
LBS	Lojistik bilgi sistemi
LM	Lojistik merkez
LMBS	Lojistik merkez bilgi sistemi
MS TFS	<i>Microsoft Team Foundation Server</i>
RI	Rassal indeks
TD	Tutarlılık değeri
TG	Tutarlılık göstergesi
ÜK	Ürün kalitesi
YNA	Yeni nesil acentecilik

# İÇİNDEKİLER

ÖZET	
ABSTRACT	
TEŞEKKÜR	
ŞEKİLLER DİZİNİ	
ÇİZELGELER DİZİNİ	
EKLER DİZİNİ	
KISALTMALAR DİZİNİ	
İÇİNDEKİLER	

1.	GİRİŞ .....	1
1.1.	Amaç .....	1
1.2.	Konunun Önemi .....	1
1.3.	Araştırma Alanı .....	3
1.4.	Tezin Yapısı .....	5
2.	YAZILIM ÜRÜN ÖLÇÜMÜ İÇİN TEMELLER .....	7
2.1.	Yazılım Neden Ölçülmeli? .....	7
2.2.	Yazılım Ölçüm Terimleri .....	8
2.2.1.	Yazılım ölçümü .....	8
2.2.2.	Ölçüm .....	8
2.2.3.	Ölçü .....	9
2.2.4.	Metrik .....	9
2.2.5.	Varlık .....	9
2.2.6.	Özellik .....	9
2.3.	Yazılım Ölçüm Süreci .....	10
2.3.1.	ISO 15939 sistem ve yazılım mühendisliği ölçüm süreci .....	10
2.3.2.	ISO 9126 yazılım ürün değerlendirme kalite modeli .....	15
2.3.3.	ISO 14598 yazılım ürün değerlendirme .....	20
2.3.4.	ISO 25000 standartı .....	22
2.4.	Yazılım Kalite Metrikleri .....	24

3.	LOJİSTİK MERKEZ BİLGİ SİSTEMİ .....	28
3.1.	Lojistik Merkez .....	28
3.2.	Lojistik Merkez Bilgi Sistemi .....	30
4.	KARAR VERME SÜRECİNDE KULLANILAN YÖNTEM .....	32
5.	YAZILIM BAKIMI .....	38
5.1.	Temel Bilgiler .....	38
5.2.	Yazılım Bakımı Terimleri .....	40
5.2.1.	Bakımcı .....	40
5.2.2.	Kullanıcı .....	40
5.2.3.	İstemci .....	41
5.2.4.	Tedarikçi .....	41
5.2.5.	Bakım yöneticisi .....	41
5.2.6.	Bakım talep yöneticisi .....	41
5.2.7.	Bakım kayıt elemanı .....	41
5.2.8.	Bakım faaliyetleri .....	42
5.3.	Yazılım Bakım Süreci .....	43
5.4.	ISO 9126 Standartına Göre Bakım Yapılabilirlik Altkarakteristikleri .....	46
6.	ARAŞTIRMA YÖNTEMİ .....	48
7.	ÖLÇME SÜRECİNİN PLANLANMASI .....	50
7.1.	Ölçme Ortamı .....	50
7.1.1.	LM'in yazılım geliştirme ve bakım ortamına ait genel bilgiler .....	51
7.1.2.	Mainframe ortamı .....	58
7.1.3.	Web ortamı .....	61
7.2.	Gereksinimlerin Belirlenmesi .....	64
7.2.1.	Lojistik merkez için yazılım üst kalite faktörlerinin belirlenmesi .....	65
7.2.2.	Analitik hiyerarşi sürecinin uygulanması .....	66
7.3.	Belirlenen Gereksinimlere Göre Tezin Kapsamına Karar Verme .....	71
7.4.	Ölçümün Planlanması .....	72
7.4.1.	LM'in yazılım ortamlarından elde edilen ölçüm seti ve bilgi gereksinimleri... ..	73
7.4.2.	Temel ölçüler .....	74
7.4.3.	Türetilmiş ölçüler ve yorumlanması .....	78
7.4.4.	Ölçüm setinin oluşturulması sürecinde elde edilen diğer gözlem sonuçları ....	82
7.4.5.	Planlama sürecinin özeti .....	83

8.	ÖLÇMENİN GERÇEKLEŞTİRİLMESİ .....	84
8.1.	Verilerin Toplanması .....	84
8.2.	Verilerin İşlenmesi .....	85
8.3.	Verilerin Doğrulanması .....	86
9.	ÖLÇÜM SONUÇLARININ DEĞERLENDİRİLMESİ .....	87
9.1.	Ölçüm Sonuçlarının Değerlendirilmesi .....	87
9.2.	Ölçüm Sürecinin Değerlendirilmesi .....	94
9.3.	Sonuçların Doğrulanması .....	95
9.4.	Tezin Hipotezi ve İspatı .....	95
10.	SONUÇ VE ÖNERİLER .....	96
	KAYNAKLAR .....	98
	ÖZGEÇMİŞ .....	103
	EKLER .....	104



## 1. GİRİŞ

### 1.1 Amaç

Bu tezin amacı; ISO tabanlı modeller kullanarak, lojistik merkez (LM) yazılım ürününü ölçmek, LM için artırılmış ürün kalite kavramını ortaya koyarak, benzer sistemler için örnek bir çözüm oluşturmaktır.

### 1.2 Konunun Önemi

Sayısal tabanlı bilgisayarların yarım yüzyıllık varlıklarının sonucunda, hemen herkes doğrudan veya dolaylı olarak bilgisayar sistemleri ile ilişki halindedir ve günlük hayatımız bilgisayarlara bağımlı hale gelmiştir. Dünya üzerindeki pek çok faaliyetin başarı ile sürmesi, bilgisayarlara ve onun en önemli parçalarından biri olan yazılımlara bağlıdır. Donanım ve yazılım ürünleri, iş dünyası başta olmak üzere pek çok alanda hayatımıza girmiş ve belirleyici bir konum kazanmıştır. Bu durum bilgi teknolojilerine olan bağımlılığı arttırmakta ve yazılımın iş hayatı içindeki önemini arttırmaktadır.

21. yüzyılın iş ortamında ve sosyal hayatında yazılım ürünlerine olan talep artarken, yazılım ürünleri de gittikçe çeşitlenmekte ve karmaşıklaşmaktadır. Günümüz yazılım geliştirme ortamı, heterojen iş ortamının beklentilerini karşılamak üzere; internet ortamına servis veren dağıtık sistemlerden, farklı ortamlarda yazılmış, farklı platformlarda çalışan ara programlardan ve birbirinden bağımsız çalışan çok sayıdaki yazılımdan oluşmaktadır.

Küreselleşmenin etkisiyle değişen iş dinamikleri, lojistiğin önemini her geçen gün daha da arttırmaktadır. Bu durum lojistik yönetim etkinliğinin en önemli gereklerinden biri olan bilgiye olan gereksinimi de arttırmaktadır. Üreticinin, aracının ve tüketicinin dünyanın herhangi bir yerinde bulunduğu, bilginin elektronik olarak kolaylıkla üretildiği, iletildiği ve tüketildiği günümüz iş koşullarında, bilginin, özellikle de sayısal bilginin önemi daha da artmakta hatta vazgeçilmez olmaktadır. Lojistik alanda üretilen bilgi, internet üzerine taşınarak işletmenin çeşitli seviyelerdeki bilgi sistemlerine girdi olmakta ve karar süreçlerini daha da fazla etkilemektedir. Dolayısıyla işletmeler, ticari etkinliklerini yürütebilmek ve rekabet üstünlüğü sağlayabilmek için lojistik süreçleri ve tedarik zinciri yönetim faaliyetlerini kusursuz hale getirmek zorunda kalmaktadır.

Günümüzde müşteriler, karmaşık taleplerininin karşılanmasında yüksek kalite beklentisi içindedirler. Mevcut dinamikler, yazılım endüstrisini de etkileyerek, yazılım geliştiricilerin müşteri beklentilerini karşılamak konusundaki çabalarını arttırması, daha verimli ve etkin geliştirme süreçleri ile projelerin tamamlamasını gerekli hale getirmektedir. Tüm bu çabalar, rekabet üstünlüğü sağlamada fırsatlar sunarken, süreçlerin izlenmesi, kontrol edilmesi ve iyileştirilmesi ile ilgili mühendislik çabalarını da geliştirmektedir. Yazılım geliştirmede ölçme tekniklerinden yararlanmak, yazılım mühendisliği prensiplerinin gerçekleştirilmesini desteklemekte, geliştirme sürecinin zamanında, doğru ve bütçesinde yapılması anlamına gelmekte ve ürünlerin daha kaliteli bir şekilde üretilmesini sağlamaktadır.

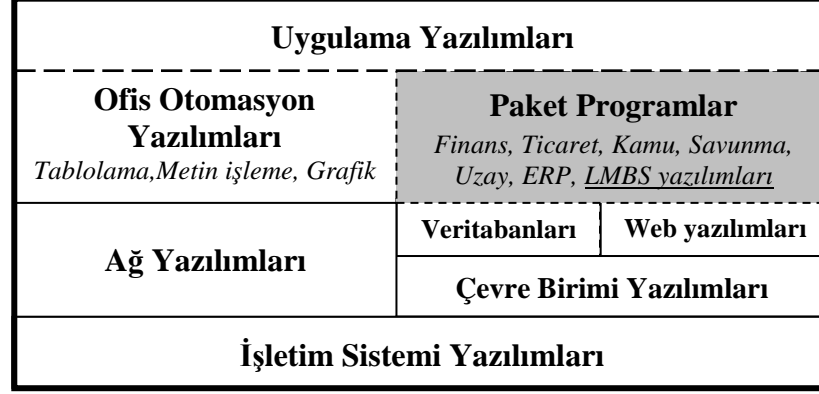
Yazılım mühendisliği, kuramsal olarak yazılım geliştirme metodolojileri, programlama paradigmaları, programlama dilleri ve çeşitli yardımcı araçlar kullanarak mühendislik prensipleri çerçevesinde yazılım geliştirilmesine yönelik çalışmaları kapsamaktadır. Bunu yaparken, müşteri gereksinimlerinin analiz edilmesi, çözümlerin geliştirilmesi, yazılım projelerinin maliyet, zaman ve içerik açısından tasarlanması, gerçekleştirilmesi, test ve bakımı ile ilgilenmektedir. Tüm bu etkinliklerin yürütülmesindeki en temel hareket noktası, müşteri beklentilerini karşılayacak yazılımların geliştirilmesidir.

Yukarıda değinilmiş olan dinamikler nedeniyle, yazılımın kalite faktörüne dikkat edilerek üretilmesi, dünya üzerindeki faaliyetlerin sürekliliği ve ekonomikliği açısından kritik öneme sahiptir. Yazılım endüstrisinin karakteristik özelliği, hızla ve sürekli olarak gelişmesi ve değişmesinin yanı sıra, gerçekte sıkıntılı ve dünya ölçeğinde pek çok başarısız uygulamaya sahne olmasıdır (Dominguez, 2009) (Emam ve Koru, 2008). Yazılımın beklenen kalite düzeyinde olmamasının ve dolayısıyla da müşteri beklentilerini kesin ve net bir şekilde karşılayamamasının neden olduğu maliyetler, ürüne ve ürünü üreten şirketin başarısına yönelik ciddi tehditleri de beraberinde getirmektedir. Bu durum, kaliteli yazılım ürünlerine olan gereksinimi arttırmaktadır.

Tez projesi; ortaya koyacağı yazılım ürün ölçütlerinin, lojistik merkez yazılım geliştiricileri tarafından kullanılarak, daha kaliteli ve ekonomik ürünler üretmeleri, faaliyetlerinde süreklilik sağlayabilmeleri ve rekabet avantajı elde etmeleri açısından önem arz etmektedir. Böylece, benzer bir ortama, süreçlere ve kişilere sahip bir LM'in bu araştırmadan yararlanması amaçlanmaktadır.

### 1.3 Araştırma Alanı

Çalışma kapsamında yazılım ürün ölçütlerinin çalışılacağı uzay, Şekil 1.1’de gri alanda sunulmuştur. Çalışmanın uygulama alanı, paket programlar ürün uzayı içerisinde yer alan lojistik merkez bilgi sistemi uygulama yazılımlarıdır.



**Şekil 1.1 Araştırma alanı ürün uzayı.**

Yazılımın ölçülmesine yönelik yapılan sınıflandırmalarda, üç temel başlık öne çıkmaktadır (Kan, 2003, s:85) (Fenton ve Pfleeger, 1997, s:74) (Lanza ve Marinescu, 2006):

1. Proje ölçümü; yazılım projelerinin geliştirmesine ve yürütülmesine yönelik süreçler ile ilgili olup, verimliliğin ölçülmesi, kaynaklara yönelik gereksinimlerin tahmin edilmesi hakkındadır. Proje ölçümü, tezin çalışma alanına girmediği için incelenmemiştir.
2. Süreç ölçümü; süreçlerle ilgili yazılımın etkinliklerine yönelik olarak, yazılımın geliştirilmesi ve bakımının sağlanmasında kullanılmaktadır. Süreçlerin ölçümü, tezin çalışma alanına girmemektedir.
3. Ürün ölçümü; kod uzunluğu, karmaşıklık, tasarım özellikleri ve kalite seviyesi olarak tanımlanmaktadır. Dolayısı ile yazılımın koduna yöneliktir ve kalite açısından yazılımın incelenmesi ve değerlendirilmesinde kullanılmaktadır. Yazılım ürünün tasarımının ölçülmesi, tezin uygulama alanına girmektir ve bu durum Şekil 1.2’de gri alanda sunulmaktadır.



**Şekil 1.2 Yazılım ölçüm uzayı ve araştırma alanı.**

Yazılım ürün ölçümü ve ürün kalitesinin değerlendirilmesi için gereken bileşenleri içeren uluslararası standartlar, ISO 14598 yazılım ürün değerlendirme ve ISO 9126 yazılım ürün kalite standartıdır. ISO 9126 yazılım kalitesinin tanımlanması için ürün karakteristiklerini ve bunlara bağlı altkarakteristikleri tanımlamaktadır (Jung vd., 2004). Çizelge 1.1’de ISO 9126 kalite standartında bir yazılımın ölçülmesinde kullanılan temel yazılım ürün karakteristikleri gösterilmiştir.

**Çizelge 1.1 ISO 9126 yazılım ürün kalite karakteristikleri ve altkarakteristikleri.**

<b>Karakteristikler</b>	<b>Altkarakteristikler</b>
Fonksiyonellik	Uygunluk, doğruluk, birlikte işlerlik, uyumluluk, güvenlik
Güvenilirlik	Olgunluk, hata toleransı, kurtarılabirlik
Kullanılabilirlik	Anlaşılabilirlik, öğrenebilirlik, işlerlik
Etkinlik	Zaman davranışı, kaynak yararlanımı
Bakım yapılabilirlik	Analiz edilebilirlik, değiştirilebilirlik, durağanlık, sınanabilirlik
Taşınabilirlik	Uyarlanabilirlik, kurulabilirlik, uygunluk, yerdeğiştirilebilirlik

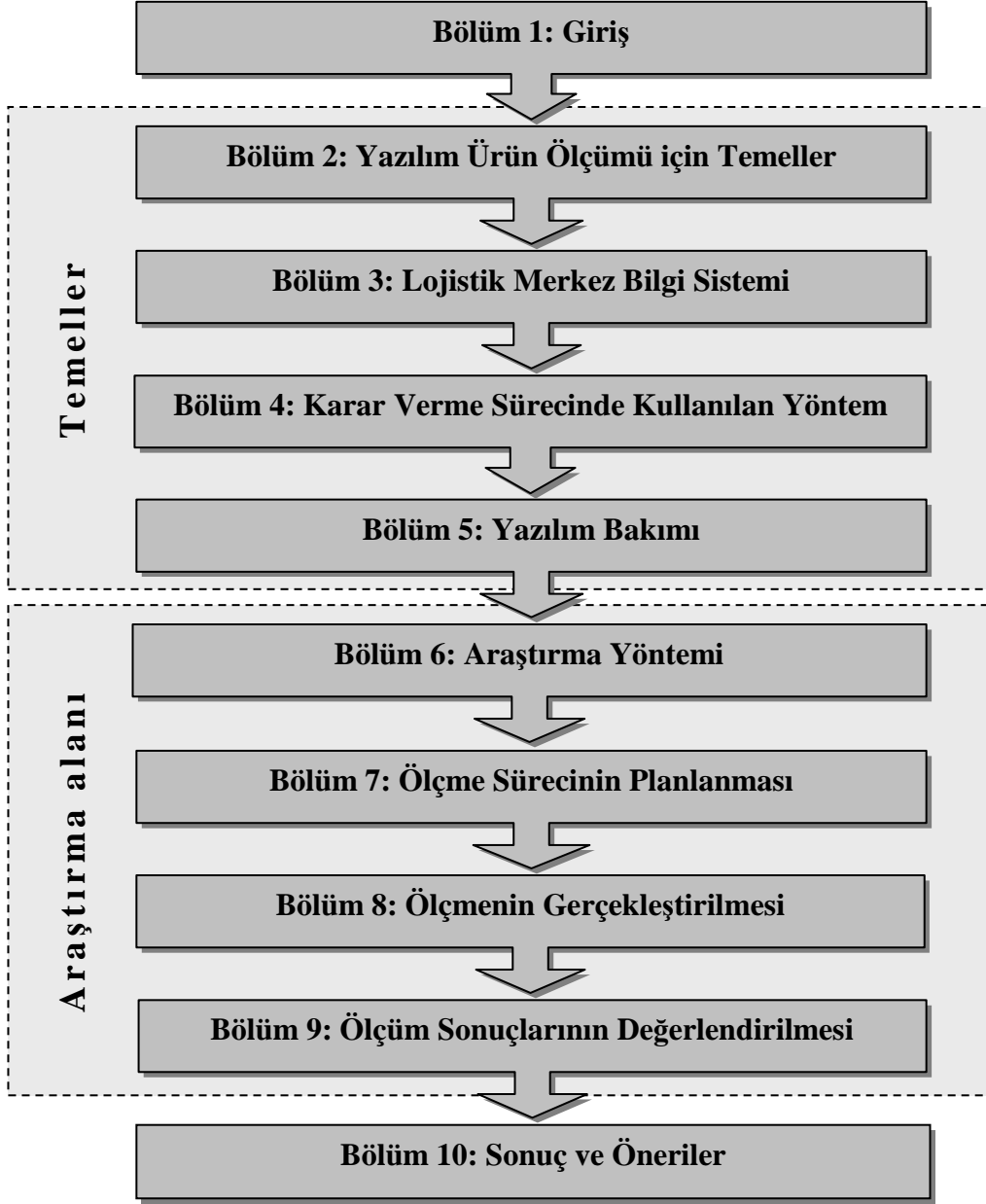
Araştırmanın ilk aşamasını LMBS’nin yazılım ürün karakteristiklerine uygun altkarakteristikleri belirlemek oluşturmaktadır. Burada amaç, gereksinimlerle ve dolayısı ile kalite ile ilişkisi olmayan karakteristikleri ortadan kaldırarak, geliştirilen yazılımda ürün kalite ölçümlenmesinin daha verimli bir şekilde yapılmasını sağlamaktır. İkinci aşamada, elde edilen altkarakteristiklere ait varlıkların ölçülerek, LMBS için arttırılmış ürün kalite kavramı ortaya konması amaçlanmaktadır. Tüm bu araştırmaları yaparken, tezin kendisi, sunulan çözümün benzer sistemler için de uygulanabilmesi için örnek oluşturmaktadır.

#### 1.4 Tezin Yapısı

Şekil 1.3’de sunulduğu gibi, tez çalışması temeller ve araştırma alanı olarak iki ana bölümden oluşmaktadır.

Temeller ana bölümünde, araştırma alanındaki çalışmalara değinilmiştir. Bölüm 2’de yazılım ürün ölçümü ve bununla ilgili ISO standartları anlatılmaktadır. Bölüm 3’te, tezin uygulama alanı olan lojistik merkez tanımı yapılmakta, lojistik merkez bilgi sistemi hakkında bilgi verilmektedir. Bölüm 4’te, lojistik merkez yazılım ürününü ölçmeye yönelik gereksinimlerinin tespit edilmesinde kullanılan yöntem değinilmektedir. Bölüm 5’te ise yazılım bakımı konusu incelenmektedir.

Tezin araştırma alanı başlıklı ikinci ana bölümünde, ölçüm sürecine yönelik faaliyetler, gerçekleştirilme sırası içinde anlatılmaktadır. Bölüm 6’da, araştırmada kullanılan yöntem tezin yapısı ile paralellik gösterecek şekilde anlatılmaktadır. Bölüm 7’de ölçme sürecinin planlanması başlığı altında, benimsenen süreç modeline göre, ölçme ortamının tanınması, gereksinimlerin belirlenmesi ve metriklerin seçilmesi işlemlerine değinilmektedir. Bölüm 8’de, ölçme işlemi anlatılmaktadır. Bölüm 9’da, elde edilen ölçüm sonuçları değerlendirilmektedir. Bölüm 10’da, lojistik merkez bilgi sistemi yazılım ürününün ölçülmesi işlemleri değerlendirilmekte ve öneriler sunulmaktadır.



Şekil 1.3 Tezin yapısı.

## 2. YAZILIM ÜRÜN ÖLÇÜMÜ İÇİN TEMELLER

### 2.1 Yazılım Neden Ölçülmeli?

Yazılım mühendisliğinin temel hedefi, yazılım sistemlerini sezgisel, anlaşılması zor ve kuralsız faaliyetler yerine; kontrol edilebilen, bir yönteme sahip ve tahmin edilebilir faaliyetler haline getirmektir. Yazılımı bir mühendislik ürünü haline getirmek, ancak yazılımı ölçülebilen ve nesnel büyüklükler ile ifade edebilen bir duruma kavuşturmakla mümkündür (Kafura, 1985).

Ölçme sayesinde, müşteri beklentilerinin anlaşılması, yanıtlanması, ürün ve hizmetlere ait verimlilik değerlerinin analiz ve iyileştirilmesine ilişkin faaliyetler, kişilerin duygu ve düşüncelerine bağlı olmaksızın yürütülmekte, böylece kalite konusunda nicel değerlendirmeler yapılabilmektedir. Kalite kavramı büyük ölçüde yazılım kalitesinin ölçülmesi ile ilgilidir (IEEE Std. 1061:1988).

Diğer mühendislik disiplinlerinde olduğu gibi, yazılım mühendisliğinde de ölçmenin ve ölçütlerin önemi büyüktür. Yazılımın ölçülmesi ile yazılım kalite gereksinimlerinin yazılımın yaşam döngüsü boyunca karşılanıp karşılanmadığının anlaşılması amaçlanmaktadır. Yazılım ölçümü; yazılım süreçlerinin anlaşılması, kontrol edilmesi, yönetilmesi ve performansının geliştirilmesi ve izlenmesi yaklaşımıdır. Bu yaklaşım bir yazılım ölçütünün belirli bir yazılım ürününe uygulanması ve yazılım metriklerinin de yazılımın kalitesini artırması ile ilgilidir. Yazılım ölçümünün gerekliliği ve sağladığı faydalar aşağıda özetlenmiştir (IEEE Std. 1061:1988) (Ebert vd., 2005) (Zuse, 1998):

1. Kalite hedeflerine ulaşılmasını sağlar.
2. Yazılımın değiştirilmesi sürecinde kalitenin izlenmesini sağlar.
3. Karmaşıklığı sınırlar.
4. Hataların analiz edilmesini sağlar.
5. Yazılım proje ve ürünlerinin anlaşılmasını sağlar.
6. Yazılımın tahminlenmesini kolaylaştırır.
7. Yazılım projelerinde, süreçlerin izlenmesi, değerlendirilmesi, analizi ve denetimini sağlar.

Uygulamada, ölçme ve değerlendirmeye yönelik sorunlar bulunmaktadır. Bu sorunlar aşağıdaki gibi özetlenebilir:

1. Ölçme zaman alan bir süreçtir.
2. Ölçme iyi anlaşılmamış ve uygulanması güç bir süreçtir.
3. Ölçme sürecinin uzaması, elde edilen bilgilerin hızla değerinin yitirilmesine ve sürekli iyileştirme döngüsünün yavaşlamasına neden olmaktadır.
4. Programcılar yazılım geliştirmeye, ölçme ve veri toplama işlemlerinden daha fazla önem vermektedir.
5. Yazılım endüstrisinin doğasından gelen hızlı yazılım üretme, ürünü pazara çabuk sunma dinamikleri nedeniyle uygulamada yazılımda kalite kavramı dikkate alınmamakta veya göz ardı edilebilmektedir.

## **2.2 Yazılım Ölçüm Terimleri**

### **2.2.1 Yazılım ölçümü**

Yazılım kalite ölçüsünü, belirli bir yazılım ürününe uygulama işlemidir.

### **2.2.2 Ölçüm**

Ölçüm (*measurement*), “*bilinmeyen bir büyüklüğün değerini öğrenmek için, bu büyüklüğü bilinen bir büyüklük (standart) ile karşılaştırma ve bir değer belirleme işlemidir*” (Çınar, 2009). Dolayısı ile ölçüm, bir olgunun belirli bir birim cinsinden hesaplanması ve çeşitli metriklerle ve bu metrikleri içeren modellerle sürekli olarak yinelenen ve geliştirilmesi gereken bir süreç olarak da tanımlanabilir. Yapılan her ölçüm sonucunun, standartlar veya ölçüm sistemleriyle, kesintisiz bir karşılaştırmalar zinciri üzerinden karşılaştırılması gerekmektedir.

ISO 15939:2007 standartına göre ölçüm, ölçülmek istenen varlığa ait ölçüm değerini belirlemek için yapılan işlemler dizisidir. ISO 9126-1:1999 standartına göre ölçüm, bir metrik kullanarak ölçülen bir varlığın özelliğine değer atama (ki bu bir sayı veya seviye değeri de olabilir) işlemidir.



### **2.2.3 Ölçü**

Ölçü, ölçüm yaparken bir varlığın özelliğine atanan sayı veya seviye değeridir (ISO 9126-1:1999).

### **2.2.4 Metrik**

Metrik (ölçüt) tanımlanmış ölçüm yöntemi ve ölçüm cetvelidir. Metrikler niteliğe ait verileri sınıflandıran yöntemleri içermektedir (ISO 9126-1:1999). Her metrik belirli bir cevtele ait belirli ölçüm birimine sahiptir.

### **2.2.5 Varlık**

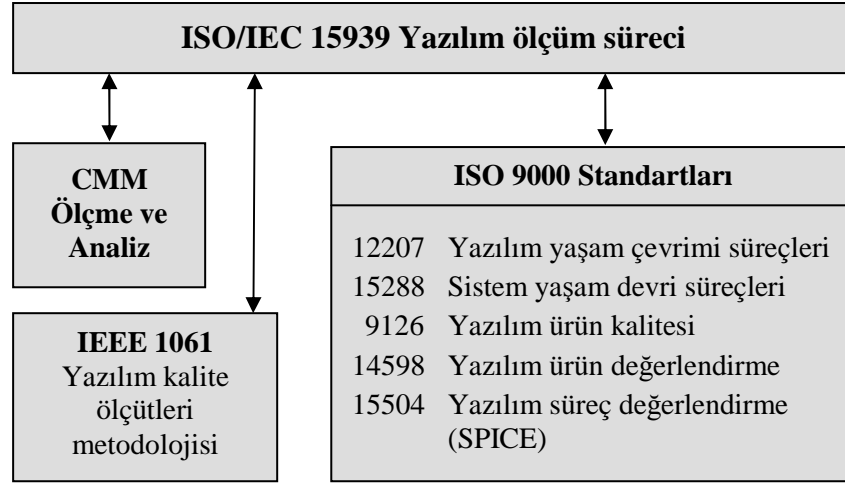
Varlık, süreçleri, ürünleri, projeleri ve çeşitli kaynakları içeren bir nesnedir. Varlıklar, özelliklerinin ölçülmesi ile karakterize edilmektedir. Örneğin, sistem mühendisleri, yazılım mühendisleri, programcılar ve test görevlileri kaynak sınıfına ait mühendislik nesnelere (ISO 15939:2007). Benzer şekilde bir oda, yapılan bir yolculuk veya yazılım projesinin sınama aşaması da ölçülmek istenen varlığa örnek olarak gösterilmektedir (Fenton ve Pfleeger, 1997).

### **2.2.6 Özellik**

Özellik, bir varlığın ölçülebilen fiziksel veya soyut özelliğidir (ISO 9126-1:1999). Örnek olarak, bir odanın rengi veya alanı, bir yolculuğun maliyeti veya sınama için geçen süre verilebilir. Bir varlık pek çok özelliğe sahip olabileceği gibi bir özellik farklı bilgi gereksinimleri için birden fazla ölçümde de kullanılabilir. Örneğin bir odanın ölçülmesinden söz edildiğinde, odaya ait uzunluk, alan veya ısı derecesi anlaşılmaktadır. Benzer şekilde sıcaklığın ölçülmesi istendiği zaman, özel bir alanın ve/veya zamanın belirtilmesi gerekmektedir (Fenton ve Pfleeger, 1997).

### 2.3 Yazılım Ölçüm Süreci

Yazılım ölçüm süreci, başta yazılımın projelendirilmesi olmak üzere pek çok endüstriyel standart ile ilişkilidir. Yazılım kalitesini sağlamak üzere temel ve destekleme faaliyetlerini ve görevlerini düzenleyen standartların birlikte uyum içinde kullanılması, süreçlere ilişkin düzeltici faaliyetlerde bulunulması gereklidir. Şekil 2.1’de yazılım ölçüm süreci ile ilişkili standartlar sunulmaktadır.



Şekil 2.1 Yazılım ölçüm süreci ile ilişkili standartlar.

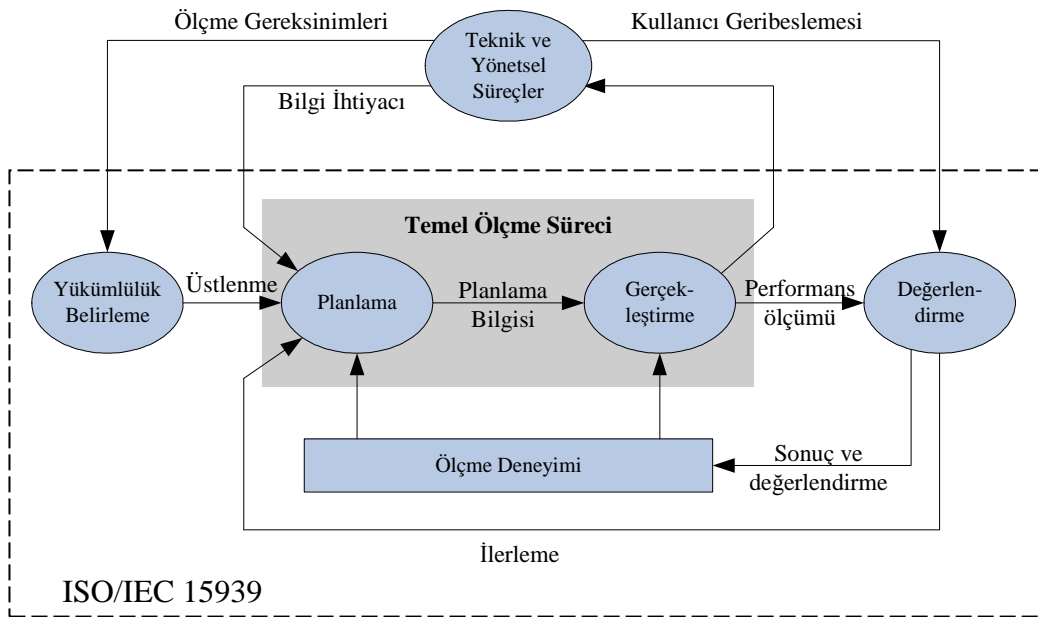
Ölçüm süreci, sıra ile yapılan bir dizi etkinliği ve bununla ilişkili işleri içermektedir. Yazılım ölçüm süreci için endüstriyel standartların başında, ISO 15939 Sistem ve Yazılım Mühendisliği Ölçüm Süreci standardı, ISO 14598 Yazılım Ürün Değerlendirme standardı ve ISO 9126 Yazılım Ürün Kalitesi standardı gelmektedir. Her üç standart da, bu çalışmanın temelini oluşturmaktadır.

#### 2.3.1 ISO 15939 Sistem ve Yazılım Mühendisliği Ölçüm Süreci

ISO 15939, yazılım endüstrisinde, ölçme ile ilişkili olarak tüm projeye veya ilişkili kısımlara yönelik etkinlikleri ve işleri tanımlayan, belirleyen, uygulayan ve süreci geliştiren bir Uluslararası Standartlar Kuruluşu (*ISO, International Standards Organization*) standardıdır. Standart, yazılım biriminin ürün geliştirmeye ve bununla ilgili süreçlerin verimli bir şekilde yönetilmesine ve de ürünün kalitesine yönelik olarak gerçekçi sonuçların elde edilmesini sağlamak için, ilgili verilerin toplanması, analiz edilmesi ve raporlanmasını sağlayan faaliyetleri ve ilişkin görevleri tanımlamaktadır.

Ölçüm süreci, işletmenin bilgi gereksinimi ile ilgilidir ve bu gereksinim sonucunda elde edilen bilginin işletmenin ürüne yönelik çeşitli karar süreçlerini, kalite beklentilerini ve gereksinimlerini ne kadar karşıladığının anlaşılması istenmektedir ve süreç bilgiye gereksinimi olan kişiler tarafından yürütülmektedir.

Standart temel kalite geliştirme modeli olan Planla-Uygula-Denetle-Harekete geç (*Plan-Do-Check-Act*) döngüsüne uygun olarak; yinelemeli, sürekli geribildirim ve geliştirmeye dayanan dört ana süreçten oluşmaktadır (Şekil 2.2):



Şekil 2.2 ISO 15939 ölçme süreci modeli.

1. **Yükümlülük belirleme.** Bu süreçte, ölçmenin amacının tanımlanması ve yönetim isteklerine göre sorumlulukların tespit edilmesi daha sonra da ölçmeye ilişkin görevlerin ve kaynakların atanması yer almaktadır.
2. **Ölçme sürecinin planlanması.** Bu süreçte hangi bilgilere gereksinim duyulacağını tespit edilmesi, bunların önceliklendirilmesi, ölçümün tanımlanması, ölçütlerin seçimi, değerlendirme kriterlerinin saptanması, ölçmenin planlanması ve ölçme ile ilgili araçların elde edilmesi ve kurulması bulunmaktadır.

3. **Ölçme sürecinin gerçekleştirilmesi.** Bu süreç, verilerin toplanması, sürecin ve verilerin doğrulanması, verilerin analizi ve anlamlandırılması, analiz sonuçlarının belgelendirilmesi ve süreçten yararlanan kişiler arasındaki iletişimi içermektedir.
4. **Sonuçların değerlendirilmesi.** Ölçme sürecinin verimliliğinin ve ölçütlerin değerlendirilmesi işlemlerini içermektedir.

Teknik ve yönetsel süreçler, ölçme faaliyetinin dış işlevine yönelik genel bir kalite tanımlaması olmasından dolayı, ISO 15939 standartına ait değildir. Ölçme sürecinin çekirdek faaliyetleri olan Planlama ve Gerçekleştirme, temelde ölçme işini yapacak olan kişiler ile ilgilidir. Bu iki çekirdek faaliyeti destekleyen Ölçmenin Kurulması ve Sürdürülmesi, Ölçmenin Değerlendirilmesi faaliyetleri ise ölçme sürecinden sorumlu olan kişiler ile ilgilidir. Standart, değerlendirme sonucunda elde edilen ve daha önceki uygulamalara ait deneyim ve verilerin, iş hedefleriyle uyumlu olması koşuluyla sürece dahil edilmesine izin vermektedir (ISO 15939:2007; s:8).

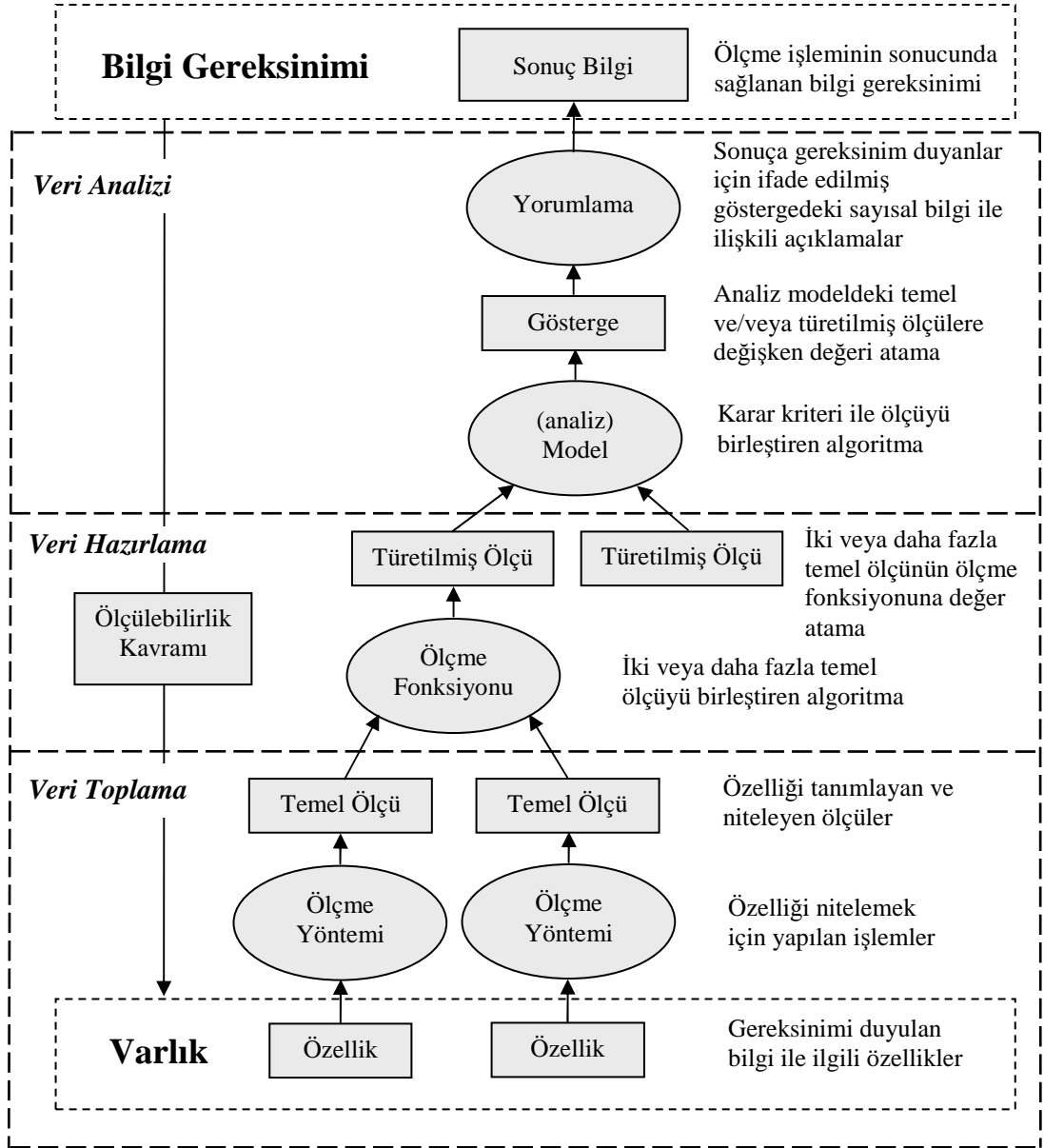
ISO 15939 standardı ölçüyü,<sup>1</sup> ölçme işleminin sonucunda elde edilen değer olarak tanımlamaktadır. Buna göre:

- **Temel ölçü.** Diğer metriklerden bağımsız olarak ölçülebilen metriktir.
- **Türetilmiş ölçü.** İki ya da daha fazla temel metriktan yararlanılarak bir matematiksel fonksiyon kullanılarak türetilmiş metriktir.

ISO 15939 Ölçme Süreci bilgi modelindeki temel ilişkiler Şekil 2.3'te gösterilmektedir. Modelde; gereksinimi duyulan bilgi ile bu bilgiyi sağlayacak olan varlıklar arasındaki ilişkinin nasıl sağlanacağı, karar vermeye yönelik olarak varlığın özneliklerinin göstergelere nasıl çevrileceği açıklanmaktadır.

Gereksinimi duyulan bilgi ile ilişkili seçim veya tanımlamalar, ölçülebilirlik kavramı ile ilişkilidir. Ölçüm modeli, ölçümü planlayan kişiler tarafından tanımlanmalıdır. Şekil 2.3'te gösterilen ölçüm modelinde, gereksinimi duyulan bilgiyi sağlayan varlığa ait özelliklerden başlayarak, bir dizi etkinliğin nasıl modellendiği anlatılmaktadır. Varlığın özelliğini nitelemek için temel ölçülere değer atanmasını takiben, temel ölçülerden ölçme fonksiyonu sayesinde türetilen ölçüler elde edilmekte ve ölçüm gerçekleştirilmektedir. Sonrasında ise elde edilen sayısal ölçüm bilgilerine ilişkin yorumlama yapılarak, bilgi gereksinimi karşılanmaktadır.

<sup>1</sup> ISO 9126 standartında metrik (*metric*) terimi, ISO 15939 standartındaki ölçü (*measure*) terimi yerine de kullanılmaktadır. Bu tezde de benzer kullanım benimsenmiştir.



Şekil 2.3 Ölçüm modelindeki temel ilişkiler (ISO 15939).

Çizelge 2.1’de ISO 15939 standartına göre ölçüm yapısı gösterilmektedir. Yapı aynı standartdaki ilişki modeline (Şekil 2.3) ve ISO 9126 standartında belirtilen metriklerin açıklandığı tablo ile uyumludur. Ölçüm yapısı ile ölçme ile ilgili kavramlar ve bu kavramların nasıl kullanılacağı anlatılmaktadır.

**Çizelge 2.1 ISO 15939 standartına göre ölçüm yapısı.**

<b>Ölçüm Yapısı</b>	
<b>Bilgi Gereksinimi</b>	Amaç ve hedeflere ulaşabilmek, risk ve sorunlarla başedebilmek için gereken bilgidir.
<b>Ölçülebilirlik Kavramı</b>	Bilgi gereksinimi ile bu gereksinimi karşılayan varlık arasındaki ilişkiyi tanımlamaktadır.
<b>Varlık</b>	Varlık bir yazılım nesnesidir ve özelliklerinin ölçülmesi ile karakterize edilmektedir. Yazılım nesnelere; 1) Ürünle ilgili: tasarım dokümanı, ağ, kaynak kodu, sına 2) Süreçlerle ilgili: tasarım, sına, gereksinim analizi süreçleri 3) Proje ve kaynaklarla ilgili: sistem ve yazılım mühendisleri, programcılar ve sına yaparı kişiler olabilir. Her varlık bilgi gereksinimi ile ilgili bir veya daha fazla özelliğe sahip olduğu gibi, varlığa ait özellikler birden fazla bilgi gereksinimi ile ilişkili de olabilir.
<b>Özellikler</b>	Varlığın, gereksinim duyulan bilgi ile ilgili sayısal veya niteleyici özellikleridir.
<b>Temel Ölçüler</b>	Temel ölçü, diğer ölçülerden işlevsel olarak bağımsız ve tek bir özellik hakkında bilgi sağlamaktadır.
<b>Ölçüm Yöntemleri</b>	Ölçüm yöntemi, bir özelliği niteleyebilmek için kullanılan sıralı ve belirlenmiş işlemleri tanımlanmaktadır.
<b>Ölçek</b>	Özellik ile eşleşen bir dizi ile sürekli veya kesik sıralı değerler kümesidir.
<b>Ölçüm Birimi</b>	Ölçüm birimi, kararlaştırılmış, tanımlanmış, benimsenmiş ve kabul edilmiş özel bir değerdir.
<b>Türetilmiş Ölçüler</b>	İki ya da daha fazla temel ölçüden yararlanılarak bir fonksiyon yardımıyla türetilmiş ölçülerdir.
<b>Ölçüm Fonksiyonu</b>	Ölçüm fonksiyonu, iki ya da daha fazla temel ölçüye uygulanacak algoritma veya hesaplama.
<b>Gösterge</b>	Gösterge, tanımlanan bilgi gereksinimine bağlı olarak bir modelden, tahminlemenin yapılması veya belirlenmiş özelliklerin değerlendirilmesini sağlayan bir ölçüdür.
<b>Model</b>	İlgili karar kriterleri için bir ya da daha fazla temel ve/veya türetilmiş ölçümü birleştiren hesaplama veya algoritmadır. Model, ölçülen bileşenler arasındaki beklenen ilişkilerin ve/veya zaman ile ilişkili davranışların anlaşılması veya tüm bunların varsayımları ile ilişkilidir. Model, bilgi gereksinimiyle ilişkili tahminleri ve değerlendirmeleri göstermektedir. Ölçek ve ölçüm yöntemi, analiz tekniklerinin seçimini veya modelde kullanılan göstergeleri etkilemektedir.
<b>Karar Kriteri</b>	Karar kriteri, ölçmenin sonucunun anlaşılmasına yardımcı olmakta ve sayısal eşik değerlerini, hedefleri, elde edilen sonuçların daha ayrıntılı incelemeye olan gerekliliğine karar vermek güvenilirlik düzeyini tanımlamaktadır. Karar kriteri sayısal veya kavramsal olarak ifade edilebilir. Karar kriteri geçmiş verilerden, planlardan ve sezgilerden veya istatistiksel yöntemler kullanılarak türetilir.

### 2.3.2 ISO 9126 yazılım ürün değerlendirme kalite modeli

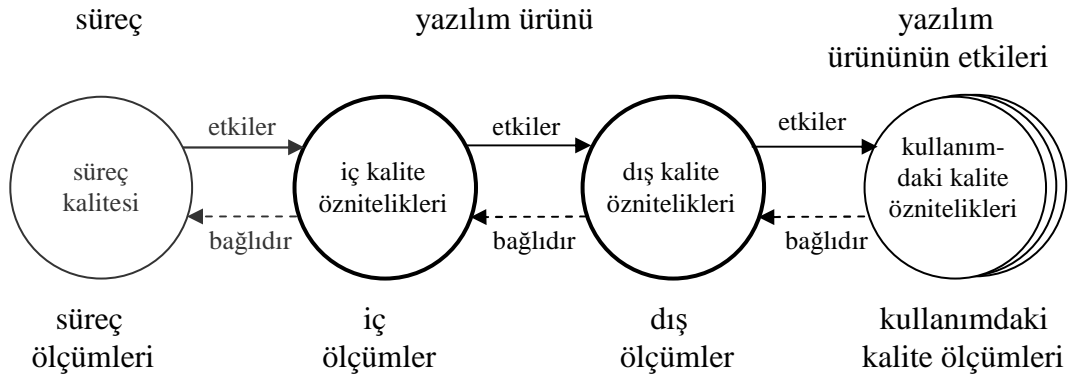
Yazılım ürün kalite modelleri 1970'lerden beri yazılım geliştiriciler tarafından çalışılan bir konudur. McCall (1977) ve Boehm (1978) tarafından tanımlanan erken dönem modellerde (Ek-A ve Ek-B), modeli geliştirenler yazılım kalitesine ait nitelikleri, bitmiş bir ürünün özelliklerine ve kullanıcının bakış açısına göre nitelemiştir. Her iki modelde de anahtar özellikler, dış ve iç nitelikler olarak belirtilmiş ve kalite faktörlerine ait niteliklerin soyutlama derecesine göre altniteliklere veya altkarakteristiklere göre sınıflanmış, kalite kriterleri olarak adlandırılmıştır (Fenton ve Pfleeger, 1997).

Yazılım kalitesinde faktör, kriter ve metriklerin, McCall veya Boehm'in modelindeki şekliyle değiştirilmeden kullanılması durumunda bu yaklaşım, değişmeyen model yaklaşımı (*fixed model approach*) olarak adlandırılmaktadır. Kalite modellenmesinde kullanılan pek çok faktör ve kriterden bazılarının ürünün özelliklerine göre seçilerek kullanılması ise "kendi modelini tanımla" yaklaşımı olarak adlandırılmaktadır (Fenton ve Pfleeger, 1997, s.340).

ISO/IEC 9126 Yazılım Mühendisliği – Yazılım ürünü kalite modeli standardı 1991 yılında yazılım ürünlerinin kalite karakteristiklerini tanımlayan bir çalışmanın sonucu olarak yayımlanmıştır. Standart, gözden geçirilerek sınırlı sayıdaki değişiklikler ile 2001 ve 2004 yıllarında tekrar yayımlanmıştır.

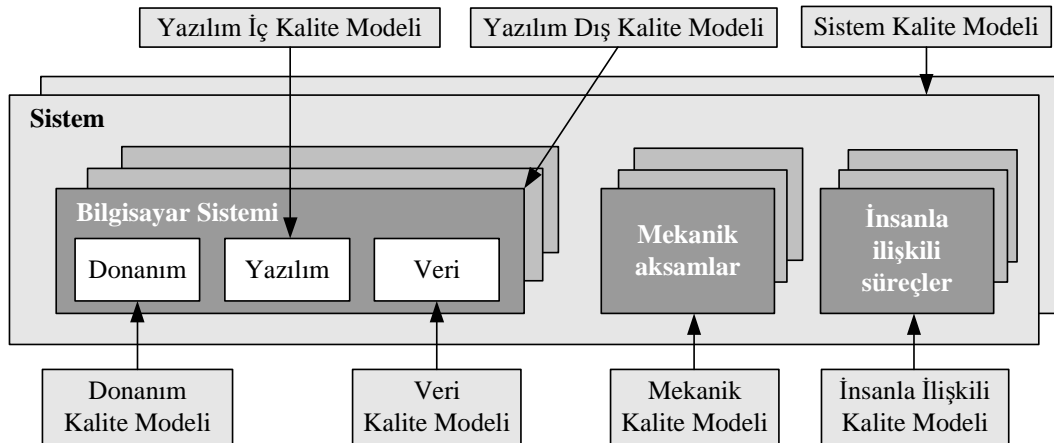
ISO 9126 kalite modelinde, yazılım ürünün karakteristiklerine ait üzerinde anlaşılmiş olan metrikler teknik rapor olarak sunulmaktadır (ISO 9126-1:1999). Standart dört temel bölüme ayrılmıştır ve iç kalite, dış kaliteyi ve her ikisi de kullanım sırasında oluşan kalite metriklerini etkilemektedir (Şekil 2.4):

- **IS 9126-1 Kalite Modeli**
- **TR 9126-2 Dış Metrikler:** Yazılım çalıştırıldığında gösterdiği davranış özellikleri ile ilgilidir ve sadece sınaama aşamasında kullanılmaktadır.
- **TR 9126-3 İç Metrikler:** Yazılımın çalıştırılma haricindeki, statik özelliklerinin ölçülmesi ile ilgilidir. İç metrikler gereksinimlerin tanımlanması, tasarım özellikleri ve kaynak kod gibi yazılımın geliştirilme sürecinde uygulanmaktadır.
- **TR 9126-4 Kullanımdaki Kalite Metrikleri:** İç ve dış kaliteye yönelik gereksinimleri karşılamış ve yazımı tamamlanmış ürünün, özel çevre ve durumda kullanımı sırasındaki kalite beklentilerini karşılaması ile ilgilidir.



**Şekil 2.4 Kalite yaşam döngüsü (ISO 9126-1).**

ISO 9126 yazılım kalite modeliyle, yazılımın üretilme sürecine ait kalite modellemesinden çok, yazılıma ve yazılımın kullanımdaki etkilerine yönelik karakteristikler tanımlanmaktadır. Bu konuda ISO 12207 Yazılım Yaşam Döngüsü Süreçleri standardı da ayrıca yol gösterici olmaktadır. ISO 9126 kalite modeli öncelikle bilgisayar sistemi ile ilgilidir. Şekil 2.5 sistemin kendisine ve parçalarına ait kalite modellerinin, birbirleri ile ilişkilerini ISO açısından göstermektedir (Boegh, 2008).



**Şekil 2.5 Sistemin çeşitli bileşenlerine ait kalite modelleri.**

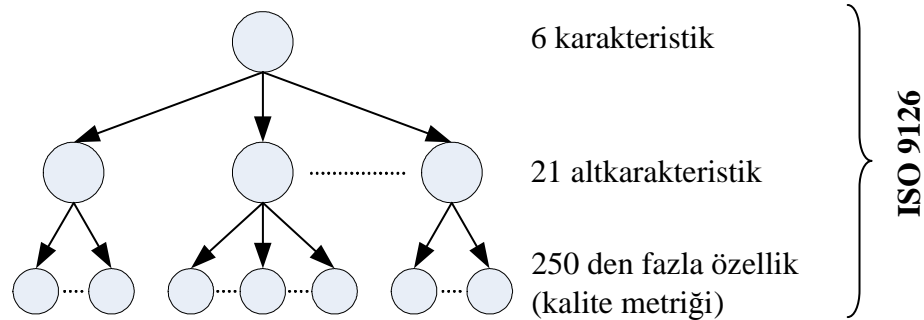


ISO 9126 standardı, yazılım kalitesine ait fonksiyonellik, güvenilirlik, kullanılabilirlik, etkinlik, bakım yapılabilirlik ve taşınabilirlik temel karakteristiklerinden oluşmakta ve temel özellikleri Koscianski ve Costa (1999) tarafından aşağıdaki şekilde belirtilmektedir:

- hiyerarşik özellik göstermeleri,
- birbirleri ile ortak özellikleri bulunmaması,
- gereksinimlere göre düzenlenmeye uygun olmalarıdır.

ISO 9126 kalite modeli, iç ve dış kalite gereksinimleri için 6 karakteristik<sup>2</sup> ve 21 altkarakteristik<sup>3</sup> tanımlamaktadır. ISO 9126 standartında belirtilen karakteristikleri ölçmek için 250'den fazla özelliğin kullanılması gerekmektedir<sup>4</sup> (Desharnais vd., 2009).

ISO 9126 standartında belirtilen karakteristikler ve altkarakteristikler, ISO 15939 standartında belirtilen ölçme bilgi modeliyle birlikte yorumlandığında yazılım kalite karakteristikleri, altkarakteristikleri ve özellikler arasında oluşan hiyerarşi Şekil 2.6'da sunulmaktadır.



**Şekil 2.6 Karakteristikler, altkarakteristikler ve özellikler arasındaki hiyerarşik yapı.**

<sup>2</sup> ISO 9126 standardı kullanımdaki kalite metrikleri için de 4 adet karakteristik tanımlamaktadır. Bu karakteristikler yazılım ürününün üretim aşamasındaki kalite hedefleri yerine bitmiş üründe kullanıcının kalite açısından görüşünü dikkate almaktadır. Şekil 2.4'te sunulduğu gibi dış kalite hedeflerine ulaşılması durumunda gerçek anlamda bu karakteristikler anlamlı olmaktadır. Bu nedenden dolayı ISO 9126-4: Kullanımdaki kalite metrikleri tezin kapsamı dışında tutulmuştur.

<sup>3</sup> ISO 9126 standardı 27 altkarakteristik tanımlamaktadır. Her karakteristiğe ait uygunluk altkarakteristiği bulunmaktadır. Örneğin; fonksiyonellik karakteristiği için fonksiyonelliğe uygunluk altkarakteristiği vardır. Çalışmada, bu kapsamdaki 6 altkarakteristik dikkate alınmamıştır.

<sup>4</sup> Desharnais ve diğerleri 2009 yılındaki çalışmalarında, 80 temel ölçüm kullanılarak türetilen 250'den fazla ölçümden söz etmektedirler. Söz konusu çalışmada, ISO 9126 standartının gelecek nesil ISO 25000 serisinde, özellikle de ISO 25021 standartında güncellenerek temel ölçüme ait öncelik ve sıralamalara ait öneriler sunulmaktadır (Desharnais ve diğerlerinin dahil olduğu ISO/IEC JTC1/SC7 WG6 kodlu ISO çalışma grubu).

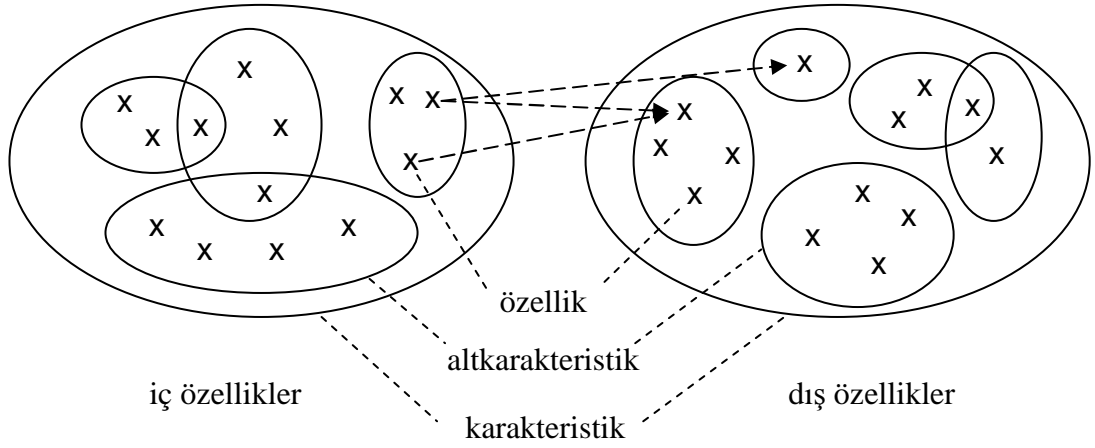
Her alt-karakteristik ve bu karakteristiklere ait açıklamalar Çizelge 2.2’de verilmiş (TSE, 1998) (Aydın, 2003) ve ayrıca her altkarakteristik, ilgili karakteristiğin ilk harfi ve altkarakteristiğin sırasını gösteren rakam ile sembolize edilmiştir. Örneğin fonksiyonellik karakteristiğinin uygunluk altkarakteristiği “F1” ile gösterilmiştir.

**Çizelge 2.2 ISO 9126 standartına ait kalite karakteristikleri.**

Kod	Karakteristikler	Tanım
	<b>Fonksiyonellik</b>	<b>Yazılım belirlenen şartlar altında kullanıldığında, belirtilen ve ihtiyacı duyulan fonksiyonları sağlaması</b>
F1	Uygunluk	Yazılımın belirlenmiş fonksiyonlar setine sahip olması ve bu fonksiyonlar setine uygunluk özelliği
F2	Doğruluk	Yazılımın üzerinde uzlaşmaya varılmış, doğru etkiyi veya sonucu doğurması özelliği
F3	Birlikte işlerlik	Yazılımın belirlenmiş sistemlerle etkileşim kurabilme becerisi
F4	Uyumluluk	Yazılımın uygulama ile ilgili standartlara, anlaşmalara, yasal düzenlemelere ve talimatlara uygun olması özelliği
F5	Güvenlik	Yazılımın, programlara ve verilere, hatayla veya kasten yetkilendirilmemiş erişimi önleyebilme özelliği
	<b>Güvenilirlik</b>	<b>Yazılımın belirlenmiş şartlar altında performans seviyesini koruyabilme becerisi</b>
G1	Olgunluk	Yazılımdaki hatalar nedeniyle yazılımın çökme sıklığının düşük olması özelliği
G2	Hata toleransı	Yazılım hatalarında veya tanımlanmış ara yüzün ihlal edilmesi durumlarında, yazılımın belirlenmiş performans seviyesini sürdürebilme becerisi
G3	Kurtarılabirlik	Sistemin göçmesi durumunda doğrudan etkilenen verinin geri yüklenebilme ve performans seviyesinin yeniden oluşturulabilme becerisi
	<b>Kullanılabilirlik</b>	<b>Belirlenmiş şartlar altında kullanıldığında, anlaşılabilen, öğrenilebilen, kullanılan ve kullanıcıya çekici gelen yazılım ürün yeteneği</b>
K1	Anlaşılabilirlik	Yazılımın mantıksal yapısının ve uygulanabilirliğinin kullanıcı tarafından anlaşılabilmesi için harcanması gereken kullanıcı çabası ile ilgili yazılım özelliği
K2	Öğrenebilirlik	Yazılım uygulamalarının kullanıcı tarafından öğrenilebilmesi için harcanacak kullanıcı çabası ile ilgili yazılım özelliği
K3	İşlerlik	Operasyon ve operasyon kontrolü için gereken kullanıcı çabası ile ilgilenen yazılım özelliği

Kod	Karakteristikler	Tanım
	<b>Etkinlik</b>	<b>Kaynağa bağlı olmak kaydıyla ifade edilmiş olan şartlarda yazılımın gereken performansı sağlaması</b>
E1	Zaman davranışı	Üretilen iş oranı ve işlem zamanları fonksiyonunu yerine getirirken talebin yanıtlanması ile ilgili yazılım özellikleri
E2	Kaynak yararlanımı	Kaynak kullanımı ve kaynak kullanım süresinin fonksiyonları yerine getirirken kullanılan miktarı ile ilgilenen yazılım özellikleri
	<b>Bakım yapılabilirlik</b>	<b>Yazılımın çalışma çevresinde, gereksinimlerde ve fonksiyonel özelliklerde değişiklik söz konusu olduğunda yazılımın düzeltilmesi, geliştirilmesi ve adaptasyon becerisi</b>
B1	Çözümenebilirlik	Tanı eksikliği ve çökme sebepleri veya düzenlenecek parçaların belirlenmesinde tanımlama için gereken beceriler
B2	Değiştirilebilirlik	Hata giderme, uyarılama veya çevresel değişiklik için gereken çaba ile ilgilenen yazılım özellikleri
B3	Durağanlık	Yapılan değişiklikler sonucunda yazılımın beklenmeyen etki doğurması riski
B4	Sınanabilirlik	Değiştirilen yazılımı geçerli hale getirmek için gereken çabaya dayanan yazılım niteliği
	<b>Taşınabilirlik</b>	<b>Yazılımın bir çalışma ortamından diğerine aktarılabilmesi</b>
T1	Uyarlanabilirlik	Göz önüne alınan yazılım için belirlenen amacın sağlanabilmesinde değişiklik gerçekleştirilmeden yazılımın belirli bir farklı çevrede adapte edilebilme özellikleri
T2	Kurulabilirlik	Tanımlanmış çevrede yazılımın kurulumu için gereken çaba ile ilgili beceriler
T3	Uygunluk	Taşınabilirlik ile ilgili anlaşmalar veya standartlara bağlı yazılım geliştirme becerisi
T4	Yer değiştirilebilirlik	Yazılımın başka bir yazılım ortamında kullanılabilmesi için fırsat sağlama ve harcanması gereken çaba ile ilgili yazılım özellikleri

Şekil 2.7’de sunulan hiyerarşide en üst seviyede kalite karakteristikleri, en alt seviyede de özellikler bulunmaktadır. Hiyerarşide bazı özelliklerin birden fazla altkarakteristikle ilgisinin bulunması, hiyerarşinin mükemmel olmadığını göstermektedir. İç veya dış metrikler ile ölçülen altkarakteristiklerler, ortak özellikler tarafından ölçülmektedir. Bu durum yazılımda karmaşıklığa neden olabilmektedir. Bunun nedeni yazılımın üretilmesi sırasında iç veya dış özelliklerin değişmesi durumunda, bunun altkarakteristikleri de etkilemesi ve iç ve dış özelliklerin altkarakteristik setlerinin elemanları arasında oluşan eşleşme fonksiyonlarının, giderek daha üst seviyeli polinomlarla ifade edilmesi gereği ve zorluğudur.



**Şekil 2.7 Kalite karakteristikleri, altkarakteristikleri ve özellikler.**

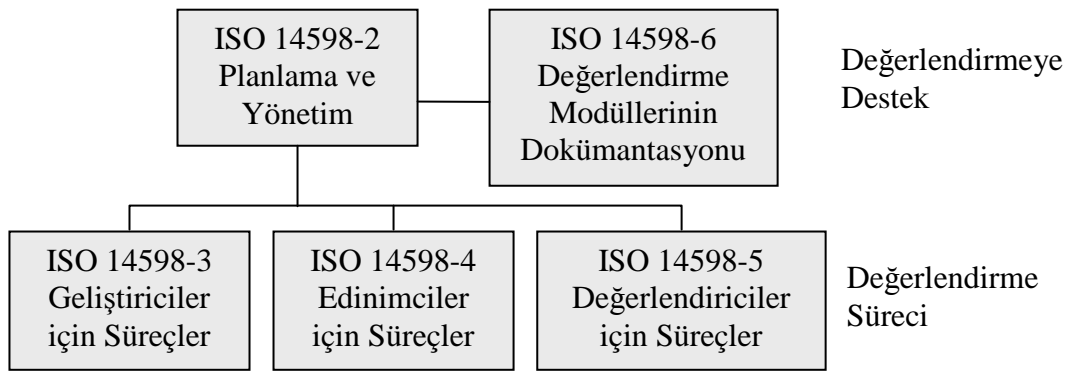
### 2.3.3 ISO 14598 Yazılım Ürün Değerlendirme

ISO 9126 serisi kalite modelini ve karakteristiklerini tanımlamakta, bu karakteristikleri ölçen metrikler hakkında bilgi vermektedir. ISO 14598 serisi ise yazılım ürün değerlendirme sürecine genel bir bakış açısı sunmaktadır ve ürün değerlendirmeye ait gereksinimlere beş farklı durum için kılavuzluk etmektedir (Şekil 2.8). Bu durumların üçü, değerlendirme sürecine aittir:

1. Geliştirme (ISO 14598-3). Yeni ürünün geliştirilmesi veya mevcut ürünün büyütülmesi aşamasına ya da ürünü değerlendiren teknik kişileri planlayarak performanslarına kılavuzluk edilmesinde kullanılmaktadır.
2. Elde etme (ISO 14598-4): Yazılımın elde edilmesi veya daha önceden geliştirilmiş yazılımın tekrar kullanılması sürecinin planlanmasına kılavuzluk etmektedir.
3. Değerlendirme (ISO 14598-5): Yazılımın bağımsız değerlendirilmesinde, değerlendiriciler tarafından uygulamada kullanılmaktadır. Standartın bu kısmı genellikle üçüncü parti firmalar tarafından tercih edilmektedir.

ISO 14598 standardı yazılım ürününe ait değerlendirme sürecinin yanı sıra, bu süreci destekleyen ISO 14598-2 (Planlama ve Yönetim) ile ISO 14598-6 (Değerlendirme Modüllerinin Dokümantasyonu) standartlarına da sahiptir. Bunlar:

4. Planlama ve Yönetim (ISO 14598-2). Standart Planlama ve Yönetim gereksinimlerini içermekte ve değerlendirme sürecine kılavuzluk etmektedir.
5. Değerlendirme modülleri (ISO 14598-6): Değerlendirme modüllerinin dokümantasyonuna kılavuzluk etmektedir. Standart; karakteristikleri, altkarakteristikleri ve bunlarla ilgili iç ve dış metrikleri içermektedir.



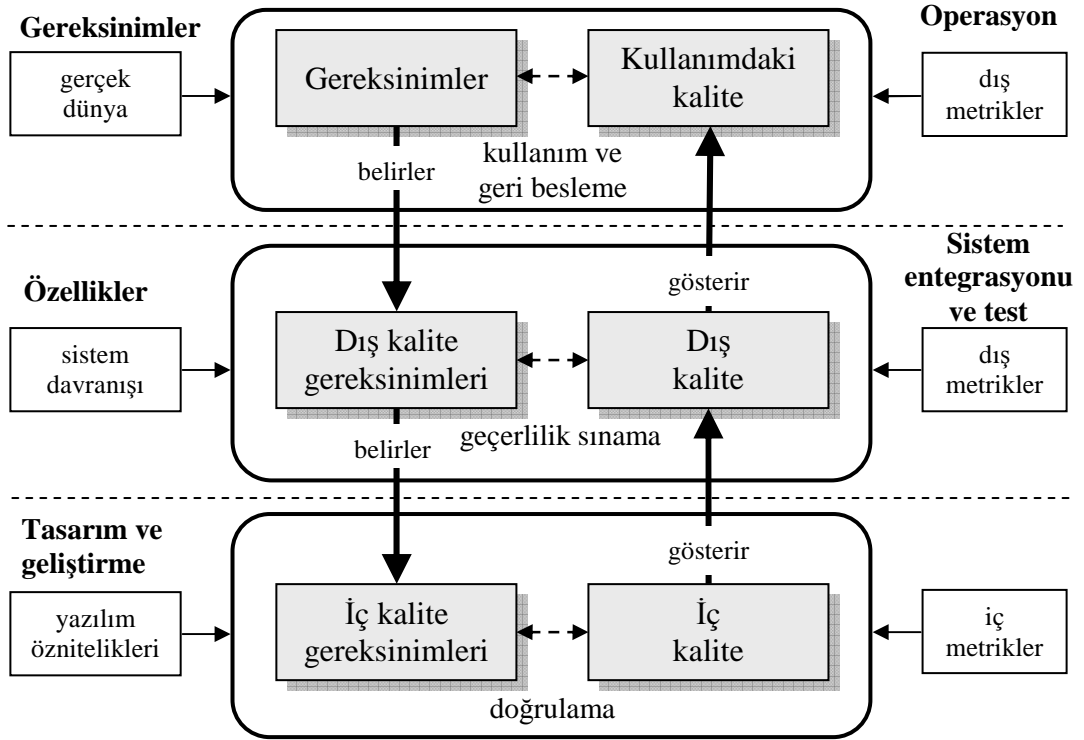
**Şekil 2.8 Değerlendirme süreci ile değerlendirmeye destek standartlar arasındaki ilişki.**

ISO 14598 standardının iç, dış ve kullanımdaki kalite metriklerinin birbirleri ile olan ilişkisi Şekil 2.9’te gösterilmektedir.

Kaynak kodun geliştirilmesine yönelik olarak iç metrikler, yazılım geliştirme sürecinde etkili olan metriklerdir ve aynı zamanda dış metrikleri de etkilemektedir.

Dış metrikler yazılımın çalışma esnasında uygulanabildiği için, sadece yazılımın bir parçası olarak değil, dış çevresini oluşturan donanım ve yazılım sistemleri ile de ilgilidir ve çalışan sistemi değerlendirmektedir.

Kullanımdaki kalite ise, özellikle son kullanıcılar, operatörler veya bakım yapanlar tarafından ilişkili karakteristiklerin bir araya getirilmesi ile kullanılmaktadır. Burada önemli nokta: yazılımın bir ortamdan diğerine taşınması halinde, kullanımdaki kaliteye etki eden dış metrikler açısından kalite beklentilerinin karşılanmasıdır.



Şekil 2.9 Yazılım yaşam döngüsünde kalite (ISO 14598-1).

### 2.3.4 ISO 25000 Standartı

ISO 9126 yazılım ürün değerlendirme kalite karakteristikleri 1991 yılında ilk kez yayımlandığından beri genişletilmiş, eklentilerle tekrar yayımlanmıştır. Bu süreçte, ISO 9126 standardı bilgi teknolojileri alanında temel bir çalışma olmasına rağmen, “standartın ölçütler konusunda yeterli olmayışı, ölçütlerin ağırlıkları ve birbirleri ile nasıl kullanılacağı konusundaki zayıflıkları” (Al-Kilidar vd., 2005) nedeniyle eleştirilmiştir. Ayrıca ISO’nun yazılıma yönelik standartlarının fazla sayıda ve birbirlerini tamamlayıcı nitelikte olması, yani yazılımın geliştirilmesinde birden fazla standartın birlikte kullanılmasındaki zorluklar nedeniyle de eleştiri toplamaktadır (Al-Kilidar vd., 2005).

ISO, 2005 yılında ISO 25000 standartını, ISO 9126 (Yazılım ürün kalitesi) ve ISO 14598 (Yazılım ürün değerlendirme) standartlarını tekrar düzenleyerek, SQuaRE serisi uluslararası standartı duyurmuş ve standartının temel özellikleri aşağıdaki gibi tanımlanmıştır (ISO 25000, 2005):

- ISO 9126 ve ISO 14598 ile aynı kurallara, referanslara ve fonksiyonel temellere sahiptir.
- ISO 9126 ve ISO 14598’i tamamlayıcı özelliğe sahiptir.
- Her iki standarta ait bağımsız yaşam döngülerinin neden olduğu tutarsızlıklar giderilmiştir.

SQuaRE serisi beş bölümden oluşacak şekilde yapılandırılmıştır (ISO 25000, 2005):

1. Kalite yönetim bölümü (ISO 2500n)
2. Kalite model bölümü (ISO 2501n)
3. Kalite ölçme bölümü (ISO 2502n)
4. Kalite gereksinimleri bölümü (ISO 2503n)
5. Kalite değerlendirme bölümü (ISO 2504n)

SQuaRE serisinin ve mevcut ISO 9126 serisinin temel hedefi, ISO 15939 yazılım ölçüm sürecinde belirtilen konuların uyum ve koordinasyonunu sağlamaktır. Özellikle kalite ölçme bölümüne ait (ISO 2502n) alttaki beş doküman ile mevcut ISO 9126 serisinin değiştirilerek yerine yeni standartın gelmesi planlanmaktadır:

1. ISO 25020: Ölçüm Kaynak Modeli ve Yol Göstericisi
2. ISO 25021: Kalite Ölçü Unsurları
3. ISO 25022: Kalite İç Ölçümü
4. ISO 25023: Kalite Dış Ölçümü
5. ISO 25024: Kalitenin Kullanımdaki Ölçümü

## 2.4 Yazılım Kalite Metrikleri

Yazılım kalite metrikleri, ISO 9126-1 standartında belirtilen altkarakteristikler ve karakteristikler açısından sınıflandırılmakta ve tablolar halinde gösterilmektedir. ISO 9126-2 ve ISO 9126-3 teknik raporlarında her bir metrik için detaylı tablolar bulunmaktadır. Metrik tablosu, yukarıda Çizelge 2.1’de verilmiş olan ISO 15939:2007 standardı ölçüm yapısı ile tam bir benzerlik göstermekte ve aşağıdaki başlıklarda bir metriği incelemektedir.

- a. **Metrik adı:** İç ve dış metrik tablolarındaki benzer metrikler aynı ada sahiptir.
- b. **Metriğin amacı:** Tablodaki sorular, metriğin uygulama alanına ait cevapları anlatmaktadır.
- c. **Uygulama metodu:** Uygulamanın genel hatlarını anlatmaktadır.
- d. **Ölçüm, formül ve veri elemanının hesaplanması:** Kullanılan veri elemanının anlamını da içerecek şekilde ölçmeye ilişkin formülü vermektedir.
- e. **Ölçüm değerinin yorumu:** Ölçüm değerinin alabileceği değer kümesini ve tercih edilmesi gereken değeri açıklamaktadır.
- f. **Metrik ölçüm cetveli tipi:** Metriğin ölçülmesinde kullanılan ölçüm cetvelinin tipini ismen, sırasına göre, aralık, oransal veya mutlak değer olarak belirtmektedir.
- g. **Ölçüm tipi:** Ölçüm tipleri; boyut veya büyüklük (örn. İşlev büyüklüğü, Kod büyüklüğü), zaman (örn. Olaydan itibaren geçen süre, Kullanım zamanı), sayma (örn. Değişiklik sayısı, Hata sayısı) olarak belirtilmektedir.
- h. **Ölçüm girdisi:** Ölçümde kullanılacak olan verinin kaynağı.
- i. **ISO/IEC 12207 SLCP referansı:** Metriğin yazılım yaşam çevrimi sürecinin neresinde kullanılacağına ait referans bilgi.
- j. **Hedef alıcı:** Ölçüm sonuçları ile ilgili kullanıcılar belirtilir.

ISO 9126-2 ve ISO 9126-3 teknik raporlarındaki tablolarda listelenen metrikler, işlevsel kod uzunluğunun ölçülmesi veya zaman verimlilik ölçümünün yapılması gibi özel amaçlar için, ek metriklerin de kullanılmasına izin vermektedir.

Yazılım kalite metriklerine ait detaylar Ek-D’de verilmektedir.



ISO 9126-2 teknik raporunda tanımlanan temel ölçüler ve bu ölçülere ilişkin ölçüm birimleri Çizelge 2.3’de gösterilmektedir (Abran vd., 2008) (ISO 9126-2).

**Çizelge 2.3 ISO 9126-2 temel ölçüler ve ölçüm birimleri.**

Temel Dış Ölçüler		
	Ölçü İsmi	Ölçüm Birimi
1	Fonksiyonların sayısı	Fonksiyon (sayı)
2	Operasyon süresi	Dakika
3	Kullanıcının karşılaştığı hatalı hesaplamaların sayısı	Durum (sayı)
4	Veri biçimlerinin toplam sayısı	Biçim (sayı)
5	Geçersiz işlemlerin sayısı	İşlem (sayı)
6	Aksaklıkların sayısı	Aksaklık (sayı)
7	Hataların sayısı	Hata (sayı)
8	Ürün uzunluğu	Byte
9	Sınama durumlarının sayısı	Durum (sayı)
10	Bozulmaların sayısı	Bozulma (sayı)
11	Onarım zamanı	Dakika
12	Sistemin çalışmadığı süre	Dakika
13	Yeniden başlamların sayısı	Yeniden başlama (sayı)
14	Gerekli onarım sayısı	Onarım (sayı)
15	Bilgilendirici belgelerin sayısı	Belge (sayı)
16	I/O veri öğeleri sayısı	Öge (sayı)
17	Fonksiyon öğrenmenin kolaylığı	Dakika
18	Görevlerin sayısı	Görev (sayı)
19	Yardım sıklığı	Erişim (sayı)
20	Hata ayıklama	Dakika
21	Ekranların veya formların sayısı	Erkanlar (sayı)
22	Kullanıcı yanlışlarının veya değişikliklerinin sayısı	Yanlış (sayı)
23	Uyarılma çabalarının sayısı	Çaba (sayı)
24	Uyumlu bir kullanılabilirlik için belirlenmiş öğelerin sayısı	Öge (sayı)
25	Yanıt zamanı	Saniye veya Milisaniye
26	Değerlendirmelerin sayısı	Değerlendirme (sayı)
27	Dönüşlerin gecikme süresi	Saniye veya Milisaniye
28	Görev süresi	Dakika
29	I/O ilişkili yanlış sayısı	Yanlış (sayı)
30	Kullanıcının I/O birimlerinden yararlanmayı bekleme süresi	Saniye veya Milisaniye
31	Bellek ilişkili yanlışların sayısı	Yanlış (sayı)
32	İletimle ilgili yanlışların sayısı	Yanlış (sayı)
33	İletim kapasitesi	Byte
34	Gözden geçirilmiş sürümlerin sayısı	Sürüm (sayı)
35	Çözümlenen hataların sayısı	Hata (sayı)
36	Veri giriş çıkışlarında kolay kullanım	Dakika

ISO 9126-3 teknik raporunda tanımlanan temel ölçüler ve bu ölçülere ilişkin ölçüm birimleri Çizelge 2.4’de gösterilmektedir (Abran vd., 2008) (ISO 9126-3).

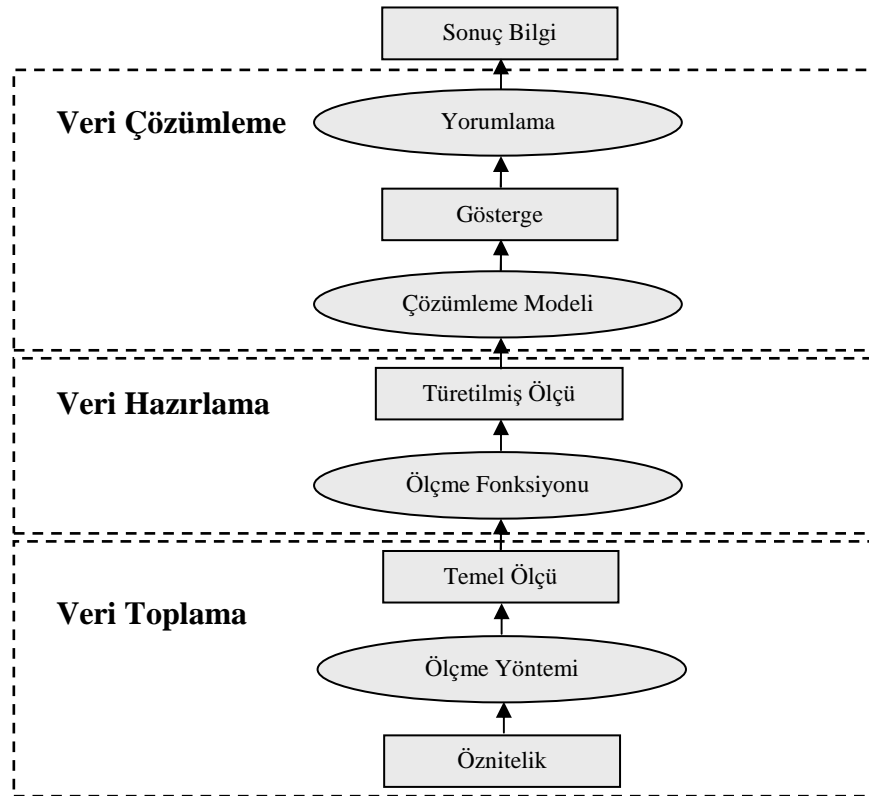
**Çizelge 2.4 ISO 9126-3 temel ölçüler ve ölçüm birimleri.**

Temel İç Ölçüler		
	Ölçü İsmi	Ölçüm Birimi
1	Fonksiyonların sayısı	Fonksiyon (sayı)
2	Veri öğelerinin sayısı	Öğe (sayı)
3	Veri biçimlerinin sayısı	Format (sayı)
4	Arabirim protokollarının sayısı	Protokol (sayı)
5	Erişim türlerinin sayısı	Erişim türü (sayı)
6	Erişim denetlenebilirlik ihtiyaçlarının sayısı	İhtiyaç (sayı)
7	Bozulmuş veri örneklerinin sayısı	Örnek (sayı)
8	Düzeltilen aksaklıkların sayısı	Aksaklık (sayı)
9	Sınama durumlarının sayısı	Durum (sayı)
10	Onarım sayısı	Gereksinim (sayı)
11	Verilerin geçerliliği için denetlenen girdi öğe sayısı	Öğe (sayı)
12	İşlemlerin sayısı	İşlem (sayı)
13	Gerçekleştirilmiş iletilerin sayısı	İleti (sayı)
14	Arabirim elemanlarının sayısı	Eleman (sayı)
15	Yanıt zamanı	Saniye veya Milisaniye
16	Dönüşlerin gecikme süresi	Saniye veya Milisaniye
17	I/O kullanımı (arabellek sayısı)	Arabellek (sayı)
18	Bellek kullanımı	Byte
19	Doğrudan sistemin çağırması ile ilişkili kod uzunluğu sayısı	Satır (sayı)
20	I/O ilişkili yanlış sayısı	Yanlış (sayı)
21	Bellek ilişkili yanlış sayısı	Yanlış (sayı)
22	Log edilmiş öğelere duyulan gereksinim sayısı	Öğe (sayı)
23	Yapılan değişikliklerin sayısı	Değişiklik (sayı)
24	Değişkenlerin sayısı	Değişken (sayı)
25	Tanılamaya yönelik gereken fonksiyon sayısı	Fonksiyon (sayı)
26	Varlıkların sayısı	Varlık (sayı)
27	Önceden tanımlanmış gerekli sınama fonksiyon sayısı	Fonksiyon (sayı)
28	Sınamanın diğer sistemlere olan bağımlılık sayısı	Bağımlılık (sayı)
29	Tanılamaya yönelik denetleyicilerin sayısı	Denetleme noktası (sayı)
30	Veri yapılarının sayısı	Veri yapısı (sayı)
31	Toplam kurulum işlemlerinin sayısı	İşlem (sayı)
32	Kurulum adımlarının sayısı	Adım (sayı)

ISO 9126 kalite modelinde ölçülmesi gereken yazılım özelliklerini nitelerek üzere gerekli temel ölçüler bir veya birden fazla defa kullanılmak üzere toplanır, sonra amaca uygun diğer metrikler türetilir. ISO 9126-2 ve 9126-3 teknik raporlarına ait temel metriklerin, yazılım kalite karakteristikleri ile olan ilişkisi Ek-C’de ki örnek çapraz referans tablosunda gösterilmektedir (Abran vd., 2008). Tablo özellikle yazılım geliştiricilere aşağıdaki konularda yol göstermektedir:

- Temel metrikleri bir kez tanımlayarak, seçerek ve toplayarak, türetilmiş metriklerde değerlendirmeye imkan sağlamak.
- Gerekli özel bir yazılım kalite niteliği için, hangi ölçülerin kullanılması gerektiğine ilişkin bilgi sağlamak.

Dolayısı ile ISO 9126 kalite modeli ile Şekil 2.3’te belirtilen ISO 15939 ölçme süreci modelindeki temel ilişkileri “Veri Toplama”, “Veri Hazırlama” ve “Veri Çözümleme” açısından tekrar yorumlanarak Şekil 2.10’da gösterilmiştir (Abran vd., 2008). Böylelikle her iki temel standart arasındaki ilişkilendirme de yapılmıştır.



Şekil 2.10 ISO 15939 sürecine ait alt süreçler ve ISO 9126 ile ilişkisi.

### 3. LOJİSTİK MERKEZ BİLGİ SİSTEMİ

Tedarik zinciri yönetimi ve lojistik çözümleri, üretim ve hizmet sektörleri için küresel pazarda rekabet avantajı elde etmede önemli bir etkidir. İşletmenin ve tedarik zincirinin katma değer yaratan faaliyetlerini bütünleşik süreçler haline getirerek, müşteri memnuniyetinin sağlanması temel hedeflerden biridir. İşletmeler, bu faaliyetleri kendi bünyelerinde yürütebilecekleri gibi işletme dışında lojistik hizmet veren kurumlar üzerinden de gerçekleştirebilir.

#### 3.1 Lojistik Merkez

Lojistik Merkez (LM); işletmelere rekabet avantajı sağlamak üzere bütünleşik faaliyetlerin merkezinde yer alacak şekilde lojistik hizmetler sunar. LM, tedarikçi, aracı ve ortaklarıyla birlikte değer yaratırken, mal, hizmet ve bunlara ait bilgilerin taraflar arasında akışını düzenler. Değer yaratma sürecinde, işletme içindeki süreçlerin entegrasyonu ile birlikte, tedarik zinciri içindeki taraflar arasında da entegrasyonu sağlamak zorundadır.

LM; tüm lojistik etkinlik, iş süreçleri ve bilgi sistemi bileşenlerinin, tek bir çatı altında ve iş ortakları arasında değer zinciri yaratacak şekilde organizasyonundan ve işletiminden sorumludur. Bir LM, üretim şirketleri için üretilecek olan ürüne ait hammadde ve yarı mamullerin girişinden, depolanmasına, üretime tam zamanında sokulmasından, üretim araçlarının yüklenmesine, fire ve atıkların yönetilmesinden, mamul maddenin saklanmasına, depo alanlarının daha verimli kullanılması ve ürünlerin müşteriye sevk edilmesine kadar geniş bir alanı kapsayabileceği gibi, bir perakende şirketi için raf düzeninin sağlanması, satınalma fiyat araştırması, fiyatlama gibi etkinliklerden de sorumlu olabilir. LM, hizmet verdiği işletme için müşteri anlaşmaları yapabilir, iş süreçlerini iyileştirebilir, değer zincirini oluşturan taraflar arasında eşgüdümü sağlayabilir. LM, tedarikçiler, depolar, taşıyıcılar, gümrükçüler, üreticiler, bayi ve perakendeciler arasında merkezi bir konuma sahiptir. Tedarik zincirini oluşturan şirketlerin her birine ve bunların da tedarikçilerine hizmet verirken, bir orkestra şefi gibi tüm etkinlikleri yönetir. LM, müşterisinin rekabet stratejisi hakkında bilgi sahibi olmalı ve hatta stratejinin oluşturulmasında müşterisi ile birlikte çalışabilmelidir. Şekil 3.1 LM'in etkileşimde olduğu çevresi gösterilmektedir. (Kurtel, Atay, Tunçay, 2006)



**Şekil 3.1 LM'in etki alanı.**

LM'in müşterisine sağlayabileceği faydalar aşağıda listelenmiştir (Murphy, 2004):

1. Sipariş sürecini iyileştirmek,
2. Esneklik ve kapasite kullanımında artışlar sağlamak,
3. Zamanında ve beklentileri karşılayan dağıtım süreçleri ile müşteri memnuniyetini sağlamak,
4. İşletmelere mali avantajlar sağlamak,
5. Lojistik operasyonları ölçeklemek, ölçülebilir ve denetlenebilir hale getirmek,
6. İşletmelerin ana işlerine odaklanmalarını sağlamak.

LM'e ait örnek hedef, vizyon ve misyon tanımlaması aşağıda yapılmış olup (Kurtel, Atay, Tunçay, 2006), böylelikle LM'in yazılımına ait kalite beklentilerinin de biçimlendirilmesi gerçekleştirilmiş olmaktadır.

- **Hedef:** "Hızlı ve bütünleşik lojistik hizmetleri, yeni teknolojik çözümleri de içerecek şekilde rekabet avantajı yaratmak üzere müşterisinin hizmetine sunmak."
- **Vizyon:** "LM X, tüm paydaşlarının dengeli mutluluğunu sağlamayı hedefleyerek, en geç 2015 yılında "Y" sektöründe Entegre Lojistik Hizmetler PAZAR LİDERİ olacak, En geç 2020 yılına kadar aynı sektörde AB içinde PAZAR LİDERİ olacak ve son adım olarak da hizmet yelpazesini genişleterek V, W, Y, Z sektörlerinde (tüm sektörlerle hizmet verecek şekilde) dünya pazarında BİRİNCİ ya da en fazla İKİNCİ sırada yer alacak ve pozisyonunu koruyacaktır."

- **Misyon:** *“Tüm paydaşlarının dengeli mutluluğunu sağlayacak şekilde, Y, V, W, Z sektörlerine lojistik hizmetler vermek üzere, yaratıcı, yenilikçi sistem tasarım ve uygulamaları sonucunda hedef sektörlerindeki tüm kuruluşlara en yeni çözümler sunarak katma değer yaratmak.”*

### 3.2 Lojistik Merkez Bilgi Sistemi (LMBS)

LM'in sahip olması gereken bilgi sistemi, yukarıda hedef, vizyon ve misyon cümleleriyle ifade edilen hedeflere ulaşmayı sağlamalı, stratejiyi desteklemelidir.

LMBS, günümüzde verimli ve etkili bir lojistik ve tedarik zinciri yönetimi için gerekli olan ürün ve hizmetleri sağlamalı, bu hizmetleri sürdürülebilir ve süreçlerin iyileştirilmesine yardımcı olmalıdır. Ayrıca maliyetler açısından rekabetçi konumun ve müşteri memnuniyetinin sağlanmasına yönelik operasyonların yürütülebilmesini ve de tüm paydaşlar ile doğru ve zamanında bilgi akışının yapılmasını sağlamalıdır.

LMBS, yazılım ve donanım ürünlerinin bir araya gelmesiyle oluşmuş, lojistik etkinlikleri yöneten, denetleyen ve ölçen bir sistemdir. Bu sistem bir işletmeye ait olabileceği gibi, tedarik zinciri boyunca da olabilir (Lambert vd., 1998).

Donanım ürünleri bilgisayarlar, sunucular, girdi, çıktı ve iletişim, barcode ve RF cihazları, depolama üniteleri olabilmektedir. Yazılım ürünleri lojistik ve tedarik zinciri yönetimi etkinliklerine ait çeşitli sistem, uygulama programları ve bunların birbirleriyle haberleşmesini sağlayan arabirim yazılımları, bu programların çalışabileceği ortamı sağlayan işletim sistemi, veri tabanı, internet programları ve diğer yazılım ortamlarıdır.

LMBS, teknik olarak organizasyonda ve tedarik zincirinde birbirleriyle ilişkili ve karar vermeye yarayan bilgilerin elde edilmesi, işlenmesi, saklanması, dağıtılması ve gösterimi ile ilgilidir. Karar verme sürecini tamamlayacak şekilde, süreçlerin denetlenmesi ve koordinasyonu açısından yöneticilere ve sorunları çözen kişilere yardımcı olmayı amaçlar. LMBS, tedarik zinciri yönetimini çeşitli yollardan etkilemektedir. Başlıca etkileri (Handfield ve Nichols, 2002) (Dornier vd., 1998):

1. Planlama, tahminleme, denetleme ve izleme süreçlerini düzenlemek ve süreçlerin verimliliğini arttırmak.
2. Tedarik zincirinin ilişkilerindeki karmaşıklığı azaltmak.
3. Tedarik zincirinin tarafları arasındaki veri bütünlüğünü sağlamak.
4. Doğru bilgiye zamanında ulaşılmasını sağlamak.
5. Kullanılan dış kaynakların maliyetlerini düşürmek.

Bilgi sistemleri açısından, işletme içindeki, dışındaki ve iş ortakları arasındaki, çeşitli büyüklüğe ve öneme sahip tüm etkinliklerde belirsizliğin ortadan kaldırılması temel hedeftir. Belirsizliğin en önemli nedeni, talepteki belirsizlik ve tahminlenmesindeki zorluktur. Gerçekte de talep, pek çok farklı nedene bağlıdır ve rekabet, fiyat, hava koşulları, teknolojik yenilikler bunlardan başlıcalarıdır. Belirsizliğin diğer önemli nedeni, ürünlerin dağıtımı ile ilgili zamanlamadır. Bu da üretim araç ve gereçlerinden, depolama ve yol koşullarına kadar uzanan geniş bir yelpazedeki belirsizliklerin toplamıdır. LMBS'nin temel hedefi, işletmenin genel bilgi sistemi ve kendi alt sistemleri ile beraber bu belirsizlikleri ortadan kaldırmaktır (Chopra, Meindl, 2004).

Yazılıma ait risklerin, problem yaratma olasılıklarına göre yapılan sınıflandırmadan (IEEE Std. 1012:2004) yararlanılarak, LM yazılımına ait konumlama Çizelge 3.1'de gösterilmiştir. Buna göre LM yazılımı kritik ve/veya sınırlı risk içeren gruba girmektedir ve gri alanda gösterilmektedir.

**Çizelge 3.1 Yazılıma ait risklerin problem yaratma olasılığına göre sınıflaması ve lojistik merkez yazılımının konumu.**

<b>Sınıflar</b> <b>Etki</b>	<b>Felaket</b>	<b>Kritik</b>	<b>Sınırlı</b>	<b>Önemsiz</b>
<b>Çevresel</b>	Giderilemez çevresel zararlar	Giderilebilir çevresel zararlar	Sınırlı çevresel zararlar	Önemsiz çevresel etki
<b>Yaşamsal</b>	Can yitimi, sosyal kayıplar	Büyük ve kalıcı yaralanma	Küçük yaşamsal etkiler	Zararsız
<b>Ekonomik</b>	Geniş ekonomik zarar	Önemli ekonomik kayıp	Küçük ekonomik kayıp	Yok sayılabilir
<b>Görev</b>	Tamamen yok olması	Kısmi görev kayıpları	İkincil görev kayıpları	Küçük performans kayıpları
<b>Çalışma süresi</b>	Sürekli çalışma	Sürüm değişikliklerinde durabilir	Belirli aralıklarla durabilir	İstek ya da gereksinimlerde

#### 4. KARAR VERME SÜRECİNDE KULLANILAN YÖNTEM

Bu bölümde ISO 9126 kalite standartına ait hangi karakteristiklerin, LM için ölçülmesinin gerekli olduğuna ilişkin kararın verilmesinde kullanılan yöntem açıklanmaktadır. Bu kararların verilmesinde gerekli olan nedenler aynı zamanda bu tezin yapılmasındaki nedenlerle de koşutluk göstermektedir. Bunlar:

1. LMBS yazılımına hangi karakteristiklerin ölçülmesinde yarar olduğunu tespit etmektir.
2. ISO 9126 kalite standartına ait 21 adet altkarakteristiğin, 250'den fazla metrik ile ölçülmesinden doğan ölçme zorluğunu azaltmaktır.

Tezde, karar verme sürecinde kullanılan temel yöntemlerden biri olan Analitik Hiyerarşi Süreci (AHP<sup>5</sup>, *Analytic Hierarchy Process*) kullanılmıştır. Bunun nedeni AHP yönteminin sınanmış, güvenilir ve bilimsel ölçütler dikkate alınarak uygulanabilirliği, grup ve bireyin önceliklerini dikkate alırken, aynı zamanda farklı görüşlerdeki kişilerin de görüşlerinin sayısal değerlere ifade edilmesine olanak vermesi, grup çalışmasına uygun olması, kolay uygulanabilmesi, karar verme süreçlerinde özellikle benzer büyüklüklerdeki problemlerde başarı ile kullanılıyor olmasıdır (Kuruüzüm ve Atsan, 2001; Kazançoğlu, 2008).

Karar vericiler için seçeneklerden birinin seçilmesi genellikle zor bir süreçtir ve karar sürecinin sayısal büyüklüklerle desteklenmesi özellikle istenmektedir. AHP çok kriterli karar verme problemlerinin çözümünü sağlamak üzere 1971 yılında Thomas Saaty tarafından geliştirilmiştir (Saaty, 1980). AHP'de amaç karar verici yargısını kullanarak, seçeneklerden hangisinin amacına uygun olduğuna karar vermektedir (Winston, 2004). ISO 9126 kalite karakteristiklerinden hangilerinin LM yazılımı için daha uygun olduğuna karar vermek yazılım geliştiricileri için önemlidir ve bu alanda AHP yönteminin kullanılması uygun görülmüştür.

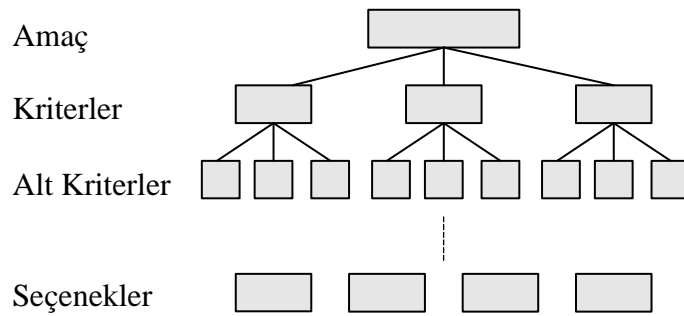
---

<sup>5</sup> Analitik Hiyerarşi Sürecinin kısaltması olarak İngilizce kısaltma karşılığı olan AHP kullanılmıştır. Bunun nedeni, AHP kısaltmasının Türkiye'de kullanılan literatürdeki bilinirliğinin fazla olması nedeniyle, kavramlar arasında koşutluk sağlanmasıdır.



AHP hiyerarşik bir yapıya sahiptir ve Partovi (1994) tarafından AHP'nin uygulanması üç adımda tanımlanmaktadır: 1) hiyerarşinin tasarımı; 2) ağırlıkların belirlenmesi ve 3) sonuçların hesabı.

AHP'nin başarısı, problem ile ilgili kriterlerin ve karar hiyerarşinin belirlenmesi ile doğrudan ilgilidir. Problemin karmaşık bir yapıdan anlaşılabilir ve analiz edilebilir bir yapıya kavuşturulmasında hiyerarşinin tasarımı büyük öneme sahiptir. Karar hiyerarşisinin tasarımında; ana amaç, kararı etkileyen faktörler açısından kriterler, alt kriterler ve seçenekler arasındaki ilişkilerin belirlenmesidir (Şekil 4.1). Amaç en iyi çözüme ulaşmaktır (Kazançoğlu, 2008).



**Şekil 4.1 Karar hiyerarşisi ağacı.**

Hiyerarşinin belirlenmesinden sonra, her seviyedeki öğelerin göreceli üstünlüklerin bulunması için ikili karşılaştırma işlemi yapılmaktadır. Böylelikle her seviyedeki karşılaştırmalar ile kare matrisler ortaya çıkmaktadır. AHP'de ikili karşılaştırma yargılarının oluşturulmasında, diğer bir deyişle A kriterinin B kriterine göre ne kadar önemli olduğu kararının verilmesinde 1-9 bağıl önem ölçeği kullanılmaktadır (Çizelge 4.1) (Saaty, 1999, s.73).

**Çizelge 4.1 Bağıl önem ölçeği.**

Önem Değerleri (1-9)	Değer Tanımları
1 Eşit	Her iki karakteristikte eşit öneme sahiptir.
3 Orta	Karakteristiklerden biri, diğerinden hafifçe daha önemlidir.
5 Güçlü	Karakteristiklerden biri, diğerinden daha önemlidir.
7 Çok güçlü	Karakteristiklerden biri, diğerine göre çok önemlidir.
9 Mutlak öneme sahip	Karakteristiklerden biri, diğerine göre mutlak öneme sahiptir.
2, 4, 6, 8	Gerektiğinde kullanılmak üzere uzlaşma değerleri.

1, 3, 5, 7 ve 9 rakamlarına karşılık gelmek üzere ikili karşılaştırmayı yapan kişi veya kişiler iki öge arasında tercihini belirtirken sözel olarak "Eşit Önemde", "Orta Önemde", "Güçlü Derecede Önemli", "Çok Güçlü Derecede Önemli" ve "Mutlak Öneme Sahip" kelimelerini kullanırlar. Burada dikkat edilmesi gereken konu, terslik ilkesine paralel olarak A ögesi B'ye karşı 3 kat önemliyse B'de A'ya karşı 1/3 önemli olarak ifade edilip matrise yazılmasıdır. Bu ölçeğin kullanılmasında ikili karşılaştırmayı yapan kişinin sezgi ve deneyimi önemlidir ve karşılaştırma tablosunun oluşturulmasında kişiye bu yeteneğini yansıtabileceği bir yapı sunulmaktadır. Farklı özelliklerdeki kişilerin bir araya gelmesi, bir tartışma ortamı içinde ortak bir görüşe varılarak tabloların doldurulması yararlı sonuçlar vermektedir.

AHP'nin uygulanmasına yönelik adımlar aşağıda sunulmaktadır (Winston, 2004):

### **Adım 1: Amaçlara ait ağırlıkların bulunması**

Bu adımda, (1) nolu eşitlikte sunulan ikili karşılaştırma matrisi oluşturulmaktadır ve  $n$ , karşılaştırılan elemanların sayısını,  $a_{ij}$  karşılaştırılan elemanların birbirlerine göre önemini belirtmektedir. A matrisinde  $a_{12}$ ; özellik 1'in özellik 2'ye göre olan önemini ifade etmektedir. Matrisin köşegeni boyunca yer alan  $a_{11}$ ,  $a_{22}$ , ...,  $a_{nn}$  elemanları kendileri ile karşılaştırıldıkları için 1'e eşittir.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad (1)$$

### **Adım 2: Normalizasyon işlemi**

Bu adımda, normalizasyon işlemi yapılmaktadır. Normalizasyon işleminde; A matrisinin sütunları toplanmaktadır.

$$A_{i1} = \sum_{k=1}^n a_{k1}, A_{i2} = \sum_{k=1}^n a_{k2}, \dots, A_{in} = \sum_{k=1}^n a_{kn} \quad (2)$$

Takip eden adımda, (3) nolu eşitlikte verildiği gibi  $A_{\text{norm}}$  matrisi oluşturulmaktadır.

$$A_{\text{norm}} = \begin{bmatrix} \frac{a_{11}}{A_{i1}} & \frac{a_{12}}{A_{i2}} & \dots & \frac{a_{1n}}{A_{in}} \\ \frac{a_{21}}{A_{i1}} & \frac{a_{22}}{A_{i2}} & \dots & \frac{a_{2n}}{A_{in}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n1}}{A_{i1}} & \frac{a_{n2}}{A_{i2}} & \dots & \frac{a_{nn}}{A_{in}} \end{bmatrix} \quad (3)$$

### Adım 3: Ağırlıkların hesaplanması

Bu adımda, karara etki eden faktörlerin ağırlığı ( $w$ ), (4) nolu ifadeye gösterildiği gibi hesaplanmaktadır ve ağırlıklar toplamı 1'e eşit olmalıdır.

$$w_1 = \frac{\sum_{i=1}^n A_{\text{norm}1i}}{n}, w_2 = \frac{\sum_{i=1}^n A_{\text{norm}2i}}{n}, \dots, w_n = \frac{\sum_{i=1}^n A_{\text{norm}ni}}{n} \quad (4)$$

Elde edilen değerler, ağırlık veya öncelik vektörü olarak adlandırılmakta ve en büyük değerde olanı, karar vermede ağırlığı en fazla olan faktör olarak işleme girmektedir.

### Adım 4: Tutarlılık Denetimi

Bu adımda, elde edilen değerlerin tutarlılık denetimi yapılmaktadır. Tutarlılık denetimi için (5) nolu eşitlikte sunulduğu gibi  $Aw^T$  vektörü hesaplanmaktadır.

$$Aw^T = A \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} Aw^T_1 \\ Aw^T_2 \\ \vdots \\ Aw^T_n \end{bmatrix} \quad (5)$$

Daha sonra T değeri hesaplanır;

$$T = \frac{1}{n} \sum_{i=1}^n \frac{Aw^T_i \text{'nin } i. \text{ değeri}}{w^T_i \text{'nin } i. \text{ değeri}} \quad (6)$$

Takip eden adımda, T değeri kullanılarak tutarlılık göstergesi (*Consistency index*) hesaplanmaktadır.

$$TG = \text{Tutarlılık Göstergesi} = \frac{T - n}{n - 1} \quad (7)$$

Bu aşamada; Adım 1'de oluşturulan ve (1) nolu eşitlikte sunulan önem matrisinin tutarlılık değeri (*Consistency ratio*); (7) nolu eşitlikte sunulan TG değerinin, Çizelge 4.2'de sunulan rassal indeks (*Random Index, RI*) tablosunda<sup>6</sup> n, seçenek sayısına karşılık gelen değere bölünmesiyle elde edilmektedir.

$$TD = \text{Tutarlılık değeri} = \frac{TG}{RI} \quad (8)$$

**Çizelge 4.2 Ortalama rassal tutarlılık indeksi.**

n	2	3	4	5	6	7	8	9	10
R.I.	0	0,52	0,89	1,11	1,25	1,35	1,40	1,45	1,49

Eğer  $TD=0$  ise karar verici için mükemmel bir tutarlılık sözkonusudur. Saaty  $\frac{TG}{RI}$  oranının 0,10'den küçük olmasını önermektedir<sup>7</sup> (Saaty ve Vargas, 2001) (Bhushan ve Rai, 2004). Eğer  $\frac{TG}{RI} < 0,10$  ise tutarlılık sonucu tatmin edicidir, fakat  $\frac{TG}{RI} > 0,10$  olması durumunda ise değerlendirmelerde tutarsızlığın bulunduğu sonucuna ulaşılmaktadır. Bu durumda karar vericilerin yargılarını tekrar gözden geçirerek, tutarlılık oranını istenilen seviyeye düşürmeleri gerekmektedir.

#### **Adım 5: Amaca göre seçeneklerin değerlendirilmesi**

Bu adımda, seçeneklerden hangisinin karar vericinin amacına ulaşmasında daha önemli olduğunun irdelenmesi işlemleri yapılmaktadır. AHP yöntemi, problemi Şekil 4.1'de de sunulduğu gibi birden fazla seviyeden oluşan bir hiyerarşik yapı ile ele almakta olduğundan, her bir amaç için seçenekleri birbirleri ile ikili olarak karşılaştırarak sonuca ulaşılması gerekmektedir.

<sup>6</sup> Çizelge 4.2'deki ortalama rassal tutarlılık indeks tablosu, 1/9, 1/8, ..., 1, ..., 8, 9 önem ölçeği kullanılarak 500 adet rastgele üretilen matrislerin sonucunda Saaty tarafından elde edilmiş ve daha sonra da her bir matris için tutarlılık indeksinin ortalaması alınarak hesaplanmıştır (Saaty, 1980; Saaty ve Vargas, 2001, s.9) (Saaty, 1991) (Saaty, 1999, s.83).

<sup>7</sup> Saaty (1999, s.81) çalışmasında tutarlılık oranı için %10 değerinin dikkate alınması sonucuna varmaktadır (özellikle 3x3 matrislerde %5, 4x4 matrislerde de %9, daha büyük matrislerde ise %10). Benzer şekilde Vargas (1982) tarafından da tutarlılığın %10'dan küçük olması önerilmektedir.

Bu doğrultuda, (9) nolu ifadede de sunulduğu gibi, her bir kritere (amaç) ilişkin seçeneklerin ikili karşılaştırma işlemi yukarıda verilmiş olan Çizelge 4.1'deki bağıl önem ölçeği kullanılarak yapılmakta ve  $A_{\text{amaç}_1}$  matrisi oluşturulmaktadır. Böylelikle ortak kriterlere göre ikili karşılaştırılmaları esas alan AHP yöntemi ile, karar verme problemi çözülmektedir (Kazançoğlu, 2008).

$$A_{\text{amaç}_1} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \dots\dots\dots A_{\text{amaç}_n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad (9)$$

Bu aşamadan sonra Adım 2 ve 3'ekilere benzer şekilde normalizasyon ve ağırlıkların belirlenmesi işlemleri yapılmakta, sonuçta da seçenekler, amaçlara uygun şekilde önem sırasına göre listelenmektedir. Bu doğrultuda elde edilen sonuç, karar vericinin amacına ulaşmasındaki en önemli etkiye sahip seçeneği göstermektedir.

## 5. YAZILIM BAKIMI

### 5.1 Temel Bilgiler

Yazılım geliştirme çabasının amacı, yazılımın, kullanıcı gereksinimlerini karşılamaıdır. Ancak geliřtilmesi tamamlanan ve müşteriye teslim edilen yazılım ürünü, kullanıcının yeni gereksinimlerinden, yazılım ortamının veya uygulama alanının deęişmesinden, yazılımdaki mevcut kusur ve eksikliklerin giderilmesine yönelik deęişim isteklerinden dolayı, sürekli deęişim baskısı altındadır. Bu noktada yazılımın bakım süreci başlamıř olmaktadır.

Yazılım bakımı, yazılım mühendislięinin temel konulardan birisi ve yazılım yařam döngüsünün ayrılmaz bir parçasıdır (Abran ve Moore, 2004) (ISO 12207). Yazılım bakımı için harcanan kaynaklar ve giderler, yazılımın yařam döngüsüne ait tüm kaynak ve giderlerin yaklaşık %40 ila %70'ini oluřturmaktadır (Grupp ve Takang, 2003). Boehm'in çalıřmasına göre ise (1987) yazılım yařam döngüsü maliyetlerinin %50 ila %90'ı yazılım bakımına aittir ve geliştirme için yapılan harcamanın iki misli bakım için harcanmaktadır. Jones (2008, s:217) içinde bulunduęumuz yüzyılın sonuna doęru yazılım bakımı alanında çalıřacak olan yazılım mühendislerinin sayısını 4,000,000 olarak tahmin etmekte ve bu sayının da yazılım üretimi alanında çalıřacak olan toplam iř gücünün %12 ila %15'ine denk geldięini belirtmektedir.

Yazılım bakımı, yazılım faaliyetlerinin bütünsellik içinde, maliyet etkin olarak gereksinimleri karşılamaı ve desteęin saęlanması olarak tanımlanmaktadır. Yazılım bakımına olan gereksinim, yazılım kullanıldıęı sürece devam etmekte ve bakıma gerek duyulmasının temel nedenleri Grupp ve Takang (2003) tarafından ařaęıdaki řekilde belirtilmektedir:

- **Hizmet süreklilięini saęlamak:** Sistemi çalıřır ve erişilebilir durumda tutmak için gerekli çabaların tümü.
- **Zorunlu güncelleřtirme desteęini saęlama:** Yasalarda ve yazılım platformunda meydana gelen deęişiklikler nedeniyle yazılımdaki deęişiklik isteklerinin karşılanması.
- **Kullanıcı gereksinimlerindeki deęişim isteklerini saęlamak:** Kullanıcıların fonksiyonellięi arttırma yönündeki istekleri ve gereksinimlerindeki deęişimler nedeniyle yazılımdaki deęişiklik istekleri.

- **Gelecekte bakıma gereksinim duyma:** Ticari ve finansal nedenlerden dolayı yazılım geliştirmenin kısayoldan yapılması nedeniyle kodda veya veritabanında yeniden yapılandırma ve dokümantasyonda güncelleme gereksinimleri.

Summerville (2007) yazılım bakım tiplerini, yazılım hatalarının giderilmesi, yazılımın farklı bir çalışma ortamına uyarlanması, yazılımın fonksiyonelliğinin değiştirilmesi veya yeni fonksiyonların eklenmesi olarak belirtmektedir.

Yazılım bakım türleri aşağıdaki gibi tanımlanmaktadır (IEEE Std. 1219:1998) (ISO 14764:2006) (Saridoğan, 2008):

1. **Düzeltilici Bakım:** Yazılımın teslim edilmesinden sonra kullanıcı tarafından bildirilen yazılım kusurlarının düzeltilmesi, yazılımda değişiklikler yapılması amacını taşımaktadır. Arıza meydana geldikten sonra arızanın giderilmesi için yapılan bakım türüdür.
2. **Uyarlayıcı Bakım:** Yazılımın teslim edilmesinden sonra, çalışma ortamının, işletim sisteminin, üzerinde veya birlikte çalıştığı donanım ürünlerinin değişmesi durumunda, yeni çalışma ortamına ait gerekli değişikliklerin ve uyarlamaların yapılması amaçlanmaktadır.
3. **İyileştirici Bakım:** Yazılımın teslim edilmesinden sonra, yazılımın performansının artırılması veya ek işlevler kazandırılması amacıyla yapılan değişikliklerdir.
4. **Önleyici (Koruyucu) Bakım:** Yazılımın teslim edilmesinden sonra, olası yanlışları önlemek amaçlı olarak yapılan bakım türüdür. Sistemin performansını ve güvenilirliğini belirli bir seviyede tutmak için yapılan bakım türüdür.

Lientz ve Swanson'un çalışmasına göre, bakım türlerine göre harcanan çaba oranları; iyileştirici bakım için %50, uyarlayıcı bakım için %25, düzeltilici bakım için %20 ve önleyici bakım için %5 olarak verilmektedir (Grupp ve Takang, 2003). Yazılım bakımına duyulan gereksinim %65'i yeni gereksinimlerden dolayı, %18'i sistemin çalışma ortamının değişmesinden dolayı meydana gelirken, yazılımdaki hatalardan dolayı meydana gelen bakım oranı ise %17 olarak gerçekleşmektedir (Lientz ve Swanson, 1980; Nosek ve Palvia, 1990; Summerville'den 2007).

Yazılım geliştirme modellerinden şelale modelinde ve yazılım mühendisliğinin çekirdek bilgisinin tanımlandığı *Software Engineering Body of Knowledge*'da (Abran ve Moore, 2004) yazılım mühendisliğinin ilk beş temel bilgi alanı sıra ile; gereksinim analizi, tasarım, kodlama, sına ve bakım olarak tanımlanmaktadır. Artırımsal ve spiral yazılım geliştirme modellerinde de benzer şekilde yazılım geliştirmenin temel süreçleri; yazılım belirtileri, yazılım tasarımı ve gerçekleştirme, yazılım geçerlilik sınaması ve yazılımın evrimleştirilmesi olarak verilmektedir. Modellerin tümünde bakım faaliyeti ve bu faaliyeti destekleyen diğer faaliyetler, tamamlanmış ve kullanıcıya teslim edilmiş ürünlerde uygulanmaktadır (Summerville, 2007).

Yazılım yaşam döngüsü içinde yazılım bakımı evresi, garanti süresinin başlamasıyla veya yazılımın hayata geçirilmesinin hemen öncesinde başlamaktadır. Ancak yazılım bakımı faaliyetleri, yazılımın teslim edilmeden öncesine ait faaliyetleri de içerebilmektedir. Bu süreçte, bakıma ilişkin işlemlerin planlanması, geçiş sürecine ait lojistik kararların alınması bulunmaktadır. Teslim sonrasında ait faaliyetler ise yazılımın değiştirilmesi, eğitim, çağrı merkezinin ve yardım masasının çalışma ve işlemlerini kapsamaktadır. Ayrıca, değişim isteklerinin izlenmesi, başta kod olmak üzere yazılımcılar tarafından yapılan diğer çalışmaların değiştirilmesi, sınamanın yönlendirilmesi, yeni sürümlerin duyurulması, kullanıcıların eğitimi ve günlük desteğinin sağlanması ile ilgili konular yer almaktadır (Abran ve Moore, 2004).

## **5.2 Yazılım Bakımı Terimleri**

Yazılım bakımına ait temel varlıklar ve açıklamaları aşağıda verilmektedir.

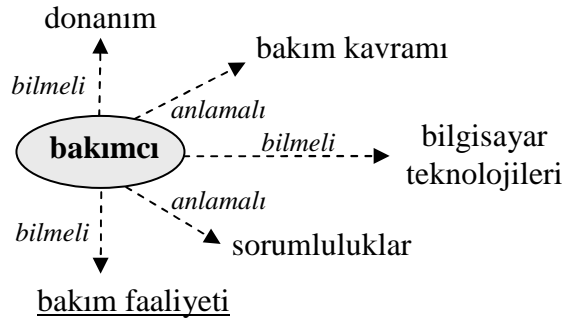
### **5.2.1 Bakımcı**

Bakımcı, bakım faaliyetlerini gerçekleştiren kişi veya kuruluştur. Bakım yapan bir kişide bulunması gereken özellikler Şekil 5.1'de gösterilmektedir (Anquetil vd., 2006).

### **5.2.2 Kullanıcı**

Yazılım ürününü kullanarak belirli bir görevi gerçekleştirmek isteyen operatör, geliştirici, yazılımın sonucu ile ilgilenen kişi veya kuruluştur.





**Şekil 5.1 Bakımcının özellikleri.**

### 5.2.3 İstemci

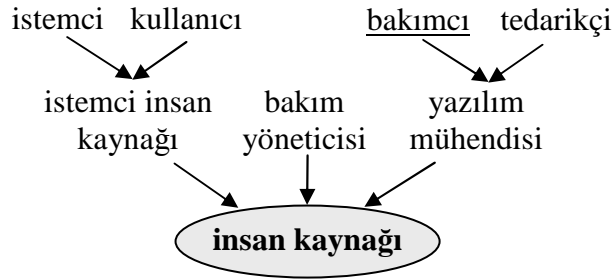
Yazılımda değişim gereksinimini ileri süren kişi veya kuruluştur.

### 5.2.4 Tedarikçi

Sözleşme hükümleri ile işletmedeki sistemin gereksinimlerinin karşılaması, yazılım ürününü veya hizmetini elde edilmesinden sorumlu kişi veya kuruluştur.

### 5.2.5 Bakım Yöneticisi

Bakım sürecini planlayan ve yöneten, kullanıcı, bakımcı ve tedarikçi ile ilgili ilişkileri düzenleyen, ürünün yapılmasını denetleyen kişidir. Şekil 5.2’de bakım faaliyetlerine ilişkin insan varlıkları gösterilmektedir (Anquetil vd., 2006).



**Şekil 5.2 Bakım faaliyetlerine ait insan varlıkları.**

### 5.2.6 Bakım Talep Yöneticisi

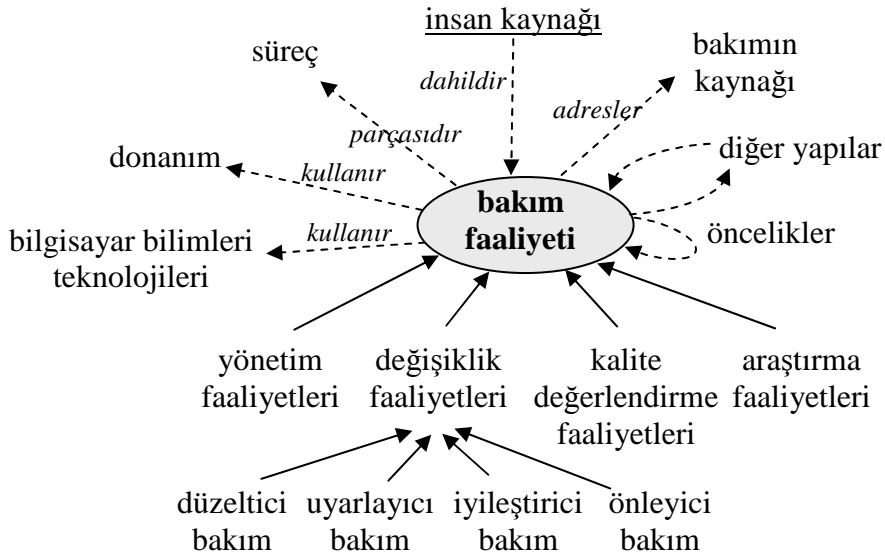
Bakım sürecinde, bakım talebinin kabul edilmesi sonrasındaki işlemleri takip eden ve özellikle de bakım isteğine ait faturalama işlemlerini yürüten kişidir.

### 5.2.7 Bakım Kayıt Elemanı

Bakım sürecinde, bakım talebinin gelmesini takiben, isteklerin kayıtlanması, güncelleştirilmesi işlemlerini yürüten ve bu işlemlere ait programları kullanan kişidir.

### 5.2.8 Bakım Faaliyetleri

Bakım faaliyetleri Anquetil ve diğerlerine göre (2006); (a) Bakım yönetimi, ürünün yapılanmasının denetlenmesi ile ilişkili yönetim faaliyetlerini; (b) Değişiklik faaliyetleri, yazılım bakımı türlerine ait faaliyetler; (c) Kalite değerlendirme faaliyetleri, değişimlerin ürünün bütününe zarar vermemesinden emin olunması ile ilgili olan faaliyetleri; (d) Araştırma faaliyetleri, değişimi gerektiren etkilerin belirlenmesi alt faaliyetlerini içermektedir. Bakım faaliyetleri sınama ve konfigürasyon planı ve diğer yazılım bileşenleri gibi yapılan diğer yapıları da içermektedir. Bu yapılar, bakım ile ilişkili süreçlerde; yazılım geliştirme, kalite değerlendirme, sınama ve konfigürasyon yönetimi planları, bakım sisteminin yürütülmesine yardımcı olan çeşitli kullanıcı, operatör, bakım notlarını içermektedir. Bakımın kaynağı, bakım ile ilgili gereksinimleri, çevrimiçi yardım dokümantasyonları, dinamik kütüphane kullanımını, güvenlik, diğer sistemlerle olan iletişim, veri yapıları ve veri tabanlarına ait yapılar ilişkilidir ve diğer bir taraftan da yapılan işler olarak adlandırılmaktadır. Bakım ile ilişkili faaliyetler Şekil 5.3'de gösterilmektedir.

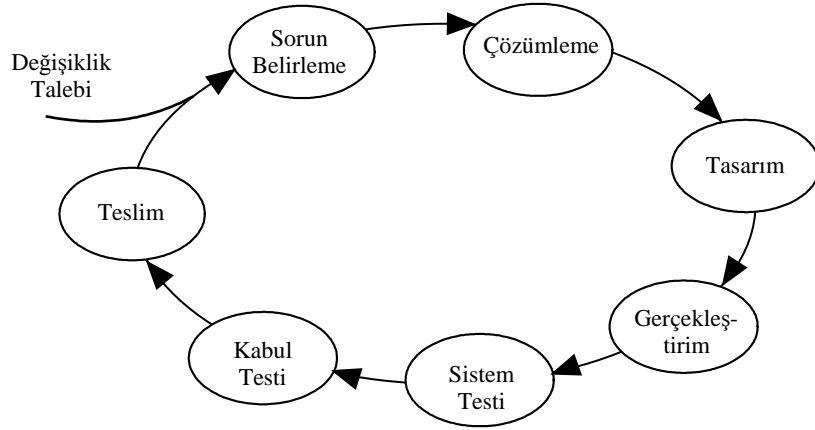


Şekil 5.3 Bakım faaliyetleri.

### 5.3 Yazılım Bakım Süreci

Yazılım bakım süreci, bakım ile ilgili faaliyetleri ve bu faaliyetler ile ilgili girdi ve çıktıları sağlamaktadır (Abran ve Moore, 2004). Yazılım bakım süreci, IEEE Std. 1219 ve ISO 14764 standartlarında detaylı olarak ele alınmakta ve tanımlanmaktadır.

IEEE Std. 1219 (1998) yazılım bakım faaliyetlerinin yürütülmesi ve yönetilmesi için gereken süreç gereksinimlerini, bu süreçlerle ilgili denetim, planlama, dokümantasyon işlemlerini tanımlamaktadır. IEEE 1219 standartında değişiklik talebinin gelmesini takip eden süreçlerde, bakım talebinin belirlenmesi ve tanımlanması daha sonra da ürünün teslim öncesindeki bakım faaliyetlerini içermektedir (Şekil 5.4).

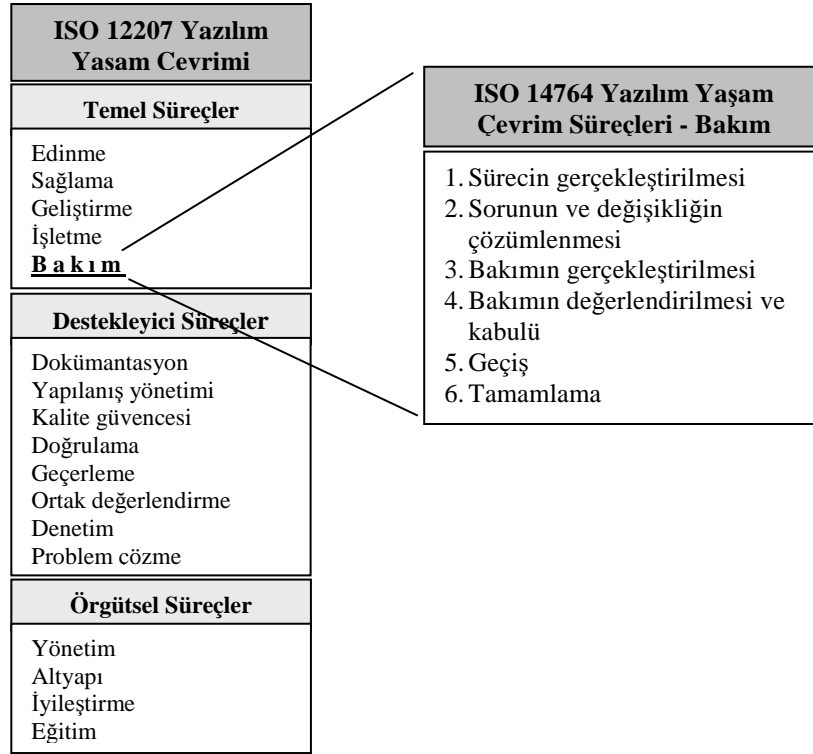


**Şekil 5.4 IEEE Std. 1219 bakım süreci faaliyetleri.**

IEEE Std. 1219 bakım sürecine ait faaliyetlerin yürütülmesi, IEEE tarafından oluşturulan bir dizi endüstriyel standart ile ilişkilidir:

1. Çözümleme aşamasında; IEEE 830 Yazılım gereksinim belirtimi ve IEEE 1074 Yazılım proje yaşam çevrimi;
2. Tasarım aşamasında; IEEE 830 ve IEEE 1016 Yazılım tasarım betimlemesi ve IEEE 1074;
3. Gerçekleştirim aşamasında; IEEE 1008 Yazılım birim sına ve IEEE 1074;
4. Sistem testi için; IEEE 829 Yazılım sına dokümantasyonu, IEEE 1012 Yazılım doğrulama ve geçerlilik sına, IEEE 1028 Yazılım inceleme ve denetimi ve IEEE 1074;
5. Kabul testi için; IEEE 1012 ve IEEE 1074 için standartları ile ilişkilidir.

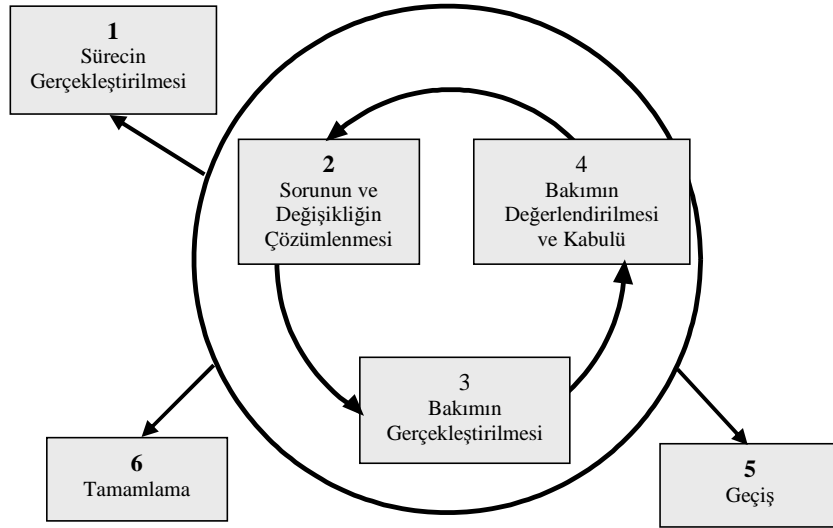
ISO 14764 Yazılım yaşam çevrim süreçleri - bakım standardı, ISO 12207 (veya IEEE 12207) Yazılım yaşam çevrimi standartında açıklanan bakım süreçlerini genişleterek tanımlanmaktadır. ISO 12207:1995 standartına göre yazılım mühendisliği süreçleri temel, destekleyici ve örgütsel süreçler olarak üç ana gruba ayrılmakta ve yazılım bakımı, beş temel süreçten biri olarak ele alınmaktadır (Şekil 5.5).



**Şekil 5.5 ISO 12207 yazılım yaşam çevrimi süreçleri ve bakım.**

ISO 14764 yazılım bakım süreci, yazılımın bütünlüğünü korumak üzere gerekli olan bir dizi faaliyet ve görevleri içermektedir (Şekil 5.6). Bu faaliyet ve görevlerin yerine getirilmesi ile ilgili sorumluluk bakımcıya aittir. Bakımcı, değişiklik isteği belirdiğinde bakım sürecine ilişkin adımları gerçekleştirmeli ve yazılım bakım sürecinin planlandığı şekilde yürütüldüğünden emin olmalıdır.

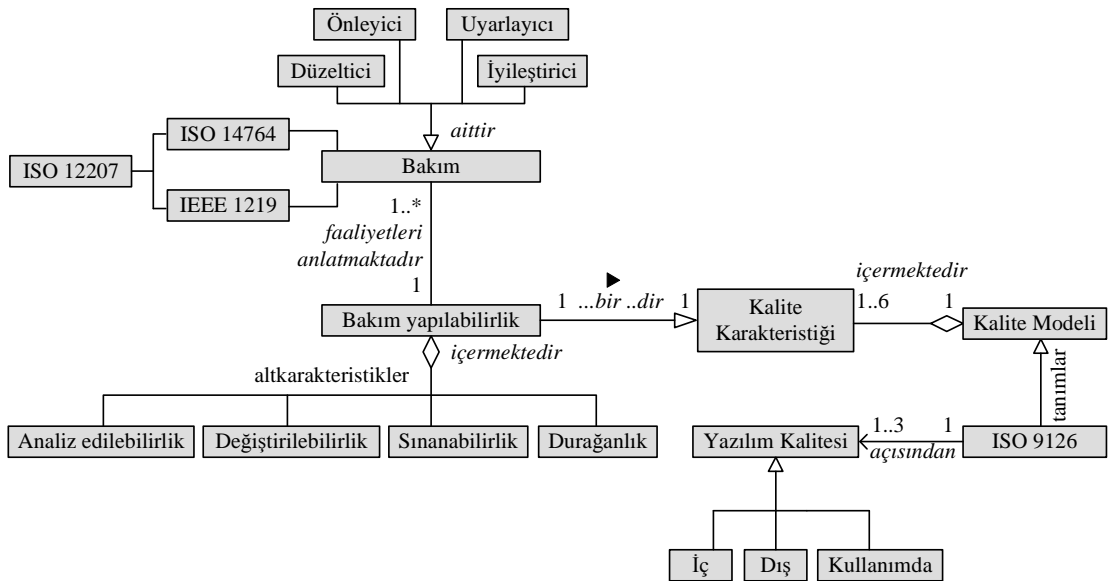
ISO 14764 standartına göre, bakım süreci başladığında, bakım planları ve çözüm adımları belirlenmeli, bakım için gerekli kaynak tahsisi yapılmalıdır. Bakımı yapan kişiler, yazılımın teslim edilmesi sonrasında, değişiklik isteği veya bir sorunun rapor edilmesini takiben koda meydana gelecek değişiklikler ve bununla ilişkili dokümantasyon işlemlerini yapmalıdır. Bu destek sürecinin yazılımın başlangıç safhasından yeni ortamına taşınmasına kadar olan sürede ve daha sonra da yazılımın devre dışı kalmasına kadar geçecek olan sürede sürdürülmesi gerekmektedir.



**Şekil 5.6 ISO 14764 yazılım bakım süreci.**

Çeşitli araştırmacılar tarafından bakım faaliyetlerini düzenleyen yeni bir süreç modeli, Yazılım Bakımı Olgunluk Modeli (*Software Maintenance Maturity Model - SM<sup>mm</sup>*) adıyla sunulmuş (April vd., 2005) (April ve Desharnais, 2005) ve çalışmalar April ve Abran tarafından kitap haline getirilerek yayımlanmıştır (2008).

Şekil 5.7'de yazılım bakımı kavramlarına ait model, ISO 9126 temeline dayandırılarak sunulmaktadır (Grimán vd., 2006).



**Şekil 5.7 Yazılım bakımı kavramsal modeli.**

#### 5.4 ISO 9126 Standartına Göre Bakım Yapılabilirlik Altkarakteristikleri

Bakım yapılabilirliğin ölçülmesi, yazılımın anlaşılması, geliştirilmesi ve düzeltilmesi açısından değerlidir. Böylelikle yazılımın geliştirilme amaçları ile elde edilen sonuçların karşılaştırılması mümkün olabilmektedir. Altta yazılımın bakım yapılabilirliğine ait altkarakteristikler (ISO 9126-2:2001) (ISO 9126-3:2001) açıklanmaktadır.

**Değiştirilebilirlik** dış metrikleri ile, yazılımda değişikliğin gerçekleştirilmesi sırasında, bakımıcının veya kullanıcının yazılımı değiştirmek için harcadığı çabayı ölçebilmek için bakımıcı, kullanıcı veya sistem davranışlarına ait özellikleri ölçmektedir. İç değiştirilebilirlik metrikleri ise bakımıcı veya kullanıcının yazılım ürünündeki değişikliği gerçekleştirilmesi için harcayacağı çabanın tahmin edilmesinde kullanılmaktadır.

**Çözümlenebilirlik** dış metrikleri ile, eksikliklerin veya hataya neden olan durumların teşhis edilmesi sırasında, bakımıcının veya kullanıcının harcadığı çabaya veya harcanan kaynaklara ait nitelikler ölçülmektedir. İç çözümlenebilirlik metrikleri, yazılımdaki eksikliklerin veya hataya neden olan durumların değişiklik yapılan yazılımda saptanarak, teşhisine yönelik bakımıcının veya kullanıcının harcadığı çabanın veya harcanan kaynakların tahmin edilmesine yönelik özellikleri belirtmektedir.

**Sınanabilirlik** dış metrikleri ile, değiştirilmiş veya değiştirilmemiş yazılımın sınanmasına yönelik işlemler sırasında, bakımıcının veya kullanıcının harcadığı çabayı ölçebilmek için, bakımıcı, kullanıcı veya sistem davranışlarına ait nitelikler ölçülmektedir. Sınanabilirlik iç metrikleri, yazılımda tasarlanmış ve gerçekleştirilmiş sınamaya yardımcı özerk fonksiyonların sayısının tahmin edilmesine yönelik özellikleri belirtmektedir.

**Durağanlık** dış metrikleri ile, değişikliklerin yapılmasından sonra yazılım çalıştırıldığında, yazılımı da içeren sistemin beklenmeyen etkileri ile ilgili özellikler ölçülmektedir. Durağanlık iç metrikleri, değişiklik sonrasında yazılımın durağanlığının tahmin edilmesine yönelik özellikleri göstermektedir.

Çizelge 5.1’de bakım yapılabilirlik karakteristiklerine ait iç ve dış ölçüleri sunulmaktadır (ISO 9126-2:2001) (ISO 9126-3:2001) (Abran vd., 2008). Çizelgede her bir ölçü için “M” harfi ile başlayan ve sıra takip eden numaralandırma yapılmıştır.

**Çizelge 5.1 ISO 9126-2 ve ISO 9126-3’e göre bakım yapılabilirlik iç ve dış ölçüleri.**

Kod	Altkarakteristik	No	Ölçü Adı	ISO 9126-2	ISO 9126-3
B1	Çözümenebilirlik	M1	Teşhisin desteklenmesi	✓	
		M2	Denetlenebilirlik	✓	
		M3	Hata çözümleme etkinliği	✓	
		M4	Hatayı çözümleyebilme	✓	
		M5	Durumun izlenebilmesi	✓	
		M6	Faaliyet kaydetme		✓
		M7	Tanılama fonksiyonu hazırlama		✓
B2	Değiştirilebilirlik	M8	Değişikliği denetleme yeteneği	✓	
		M9	Parametreleri belirlenmiş değiştirilebilirlik	✓	
		M10	Değiştirme zorluğu	✓	
		M11	Değişim döngüsünün verimliliği	✓	
		M12	Değişikliğin gerçekleştirilme süresi	✓	
		M13	Değişikliklerin kaydedilebilmesi		✓
B3	Durağanlık	M14	Değişikliğin başarı oranı	✓	
		M15	Düzeltilme sonrasında değişim etkisinin belirlenmesi	✓	
		M16	Değişim etkisi		✓
		M17	Değişim etkisinin belirlenmesi		✓
B4	Sınanabilirlik	M18	Tekrar sınanabilirliğin verimliliği	✓	
		M19	Yerleşik sınaama fonksiyonlarının kullanılması	✓	
		M20	Sınamayı yeniden başlatabilme	✓	
		M21	Önceden tanımlanmış sınaama fonksiyonlarını tamlığı		✓
		M22	Sınanabilirliğin özerkliği		✓
		M23	Sınanabilirlik gelişiminin izlenmesi		✓

Çizelge 5.1’deki bakım yapılabilirlik karakteristiğine ait değiştirilebilirlik, analiz edilebilirlik ve sınanabilirlik altkarakteristiklerinin ölçümü ile ilgili açıklamalar (ISO 9126-2:2001) (ISO 9126-3:2001) Ek-D’de verilmektedir.

## 6. ARAŞTIRMA YÖNTEMİ

Bu bölümde, Bölüm 1’de belirtilen “*ISO tabanlı modeller kullanarak, lojistik merkez (LM) yazılım ürününi ölçmek, LM için artırılmış ürün kalite kavramını ortaya koyarak, benzer sistemler için örnek bir çözüm oluşturmak*” amacına uygun olarak yapılan sistematik veri toplama ve analiz etme süreci anlatılmaktadır.

Araştırmada konular, araştırmanın güvenilirliği, geçerliliği ve genellenebilirliği açısından ele alınmıştır. Elde edilecek olan verilerin “iyi” veya “kötü” olarak belirlenmemesine özen gösterilmiş, olayların oluşuyla verilerin uyumluluğuna dikkat edilmiştir. Araştırmanın geçerliliği ise, araştırmadaki yönetsel varlıkların ve süreçlerin, araştırmanın amacında belirtilenleri gerçekten araştırılmasına yönelik olarak düzenlenmesi ile sağlanmıştır. Kullanılan yöntemlerin ve elde edilen bulguların gelecekte benzer ortamlarda da kullanılabilmesi veya kaynak olarak gösterilmesi amaçlanarak araştırmanın genellenebilirliği sağlanmıştır.

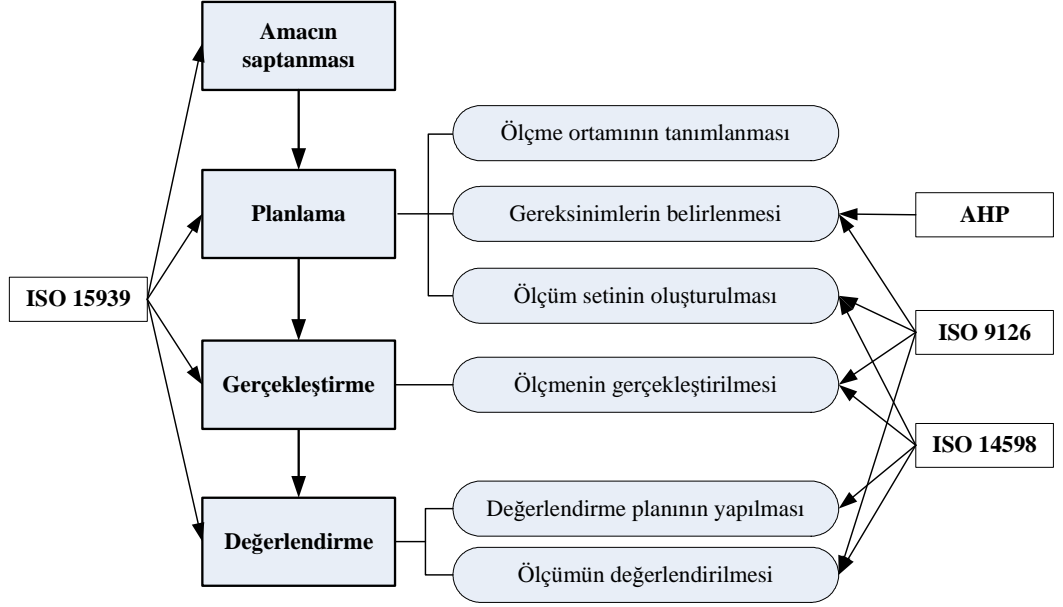
Yukarıda tanımlanan temel nitelikler, konunun ele alınış şekli ve ölçme sürecinin planlanması; yoğun bir iş yükü altında sınırlı sayıda çalışanı olan, oldukça karmaşık bir duruma sahip ve yüksek beklentili müşterilerinden, zaman baskısını yoğun olarak hisseden LM’den bilgilenilerek yapılmıştır. Bu işlemler yapılırken mümkün olduğu kadar LM’in var olan olaylarını ve süreçlerini etkilemeden, LM’in bir üyesi gibi, LM’in yaptığı işlerin boyutu kavranmaya çalışılmıştır.

Araştırmada yöntem olarak Bölüm 2.3.1’de açıklanan ISO 15939 süreç modeli benimsenmiş ve modelin temeli değiştirilmeden çalışmanın gereksinimlerine uygun olarak uyarlanmış, araştırmanın yapıldığı LM’in yararına ve çalışmanın amacına uygun bir süreç modeli ortaya konmuştur. Bu model aşağıdaki adımları içermektedir:

1. **Amacın saptanması**
2. **Planlama;** araştırmanın ikinci temel adımını oluşturmakta ve ölçme ortamının anlaşılması, ölçme kriterlerinin belirlenmesi ve ölçütlerin seçilmesi adımlarını içermektedir.
3. **Gerçekleştirme;** ölçme sürecinin üçüncü temel adımıdır ve bu adımda belirlenen metriklere göre ölçme işlemi yapılmıştır.
4. **Değerlendirme;** elde edilen verilere göre ölçme sonuçları değerlendirilmiştir.



Araştırma yöntemi, ölçme sürecine ait standartlarla ilişkili olarak Şekil 6.1’de sunulmaktadır.



**Şekil 6.1 Araştırma yöntemi ve standartlarla olan ilişkisi.**

## 7. ÖLÇME SÜRECİNİN PLANLANMASI

Bu bölümde, Bölüm 1’de verilen amaçlara uygun olarak ve Bölüm 2 ve Bölüm 3’de sunulan kuramsal temeller kullanılarak, ISO 15939 standartına uygun olarak tasarlanan ölçme sürecine ait planlama çalışmaları ele alınmıştır. Bu bölümün temel konu başlıkları; LM ölçme ortamının tanımlanması, LMBS gereksinimlerinin belirlenmesi ve bu gereksinimleri karşılayan ölçülerin tespit edilmesidir. Bu süreçte, ölçme ortamının tanımlanmasını takiben, LM bilgi gereksinimleri tespit edilmiş, ölçme yöntemindeki kriterler saptanmış, gereksinimleri karşılayan yazılım ürün karakteristikleri ve bu karakteristiklere ait ölçüler tespit edilmiştir. Daha sonra da tespit edilen karakteristiklere ilişkin ölçüler ve bu ölçüleri sağlayan özellikler incelenmiştir.

Bu süreçte ölçme işlemlerinin yürütülmesi için benimsenen iş yapma biçimi; işletmenin mevcut iş yüküne ilave yük getirmeden, ölçme işlemi için gerekli bilgilerin elde edilmesidir. Dolayısı ile ISO 9126-1 standardında detaylı olarak belirtilen yazılım ürün ölçütlerinin hangilerinin LMBS için daha önemli olduğunun tespit edilmesi gerekmiştir ve bu durum Bölüm 1.3’te tanımlanan araştırmanın alanı ve hedefleri ile doğrudan ilişkilidir. Araştırma sürecinin planlanmasındaki temel ilkeler:

1. Ölçme ile ilgili somut verilerin zamanında ve verimli bir şekilde elde edilmesi,
2. Bu verilerin hızla sonuca dönüştürerek, faydaya yönelik bir uygulamayı gerçekleştirmesi,
3. Bu işlemler sırasında işletmenin mevcut iş yükünü artırmadan ve iş süreçlerini kesintiye uğratmadan verilerin elde edilmesi

olarak belirlenmiştir.

### 7.1 Ölçme Ortamı

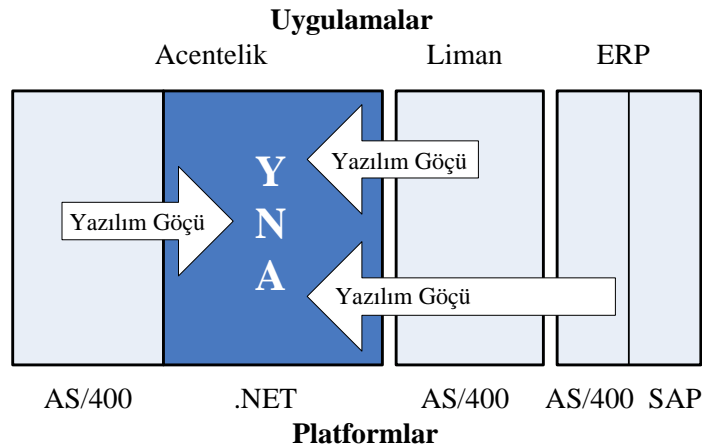
Ölçme ortamının anlaşılması, ölçme gereksinimlerinin belirlenmesi ve ölçmenin gerçekleştirilmesi açısından önemlidir. LM’in mevcut yazılım altyapısı ve süreçleri izleyen kısımda anlatılmaktadır.

### 7.1.1 LM'in yazılım geliştirme ve bakım ortamına ait genel bilgiler

Bu araştırmada, birlikte çalışılan LM'in<sup>8</sup> bilgi teknolojisi alanındaki temel amacı, "iş hedeflerini gerçekleştirebilmek için etkin, ekonomik ve yenilikçi bilgi teknolojileri çözümlerini sunmaktır". LM, bu amaca ulaşmak için sunduğu çözümün odağında olan mevcut yazılımlarını yenileyerek, acente, şube ve müşterilerine internet platformunda çalışan bir yazılım sunmayı hedeflemekte, yazılımını tekrar gözden geçirerek iyileştirmeyi ve yeni gereksinimlere cevap verebilecek değişiklikleri yapmayı ön planda tutmaktadır.

LM, 50'den fazla yerleşimde, 2000'e yakın kullanıcıya hizmet verirken, Intel tabanlı sunucular, AS/400 bilgisayar sistemleri ile taşımacılık ve lojistik alanında yazılım geliştirmektedir. Geliştirilen belli başlı yazılım uygulamaları; Acentelik, Armatörlük, Limancılık, Lojistik, Taşımacılık, Elektronik Veri Alışverişi ve Internet/Intranet'tir. Şekil 7.1'de görüldüğü gibi; araştırmanın yapıldığı LM'deki temel yazılım varlıkları üç gruba ayrılmaktadır:

1. **Web ortamı:** .NET ortamındaki YNA (Yeni Nesil Acentelik) uygulamaları.
2. **Mainframe ortamı:** IBM AS/400 platformunda geliştirilen Acentelik ve Liman uygulamaları.
3. SAP, NAVIS, Catlogic ve AS/400 üzerinde çeşitli ERP uygulamaları.



**Şekil 7.1 LM'in uygulama altyapısı.**

<sup>8</sup> LM, 40'dan fazla şirketin bir araya gelmesinden oluşan, deniz, kara, hava ve demiryolu taşımacılığı, bunları tamamlayan lojistik hizmetler, liman işletmeciliği, armatörlük ve gemi işletmeciliği gibi bir çok alanda çalışan bir gruptur. LM, bilgi teknolojisi hizmetlerini de içerecek şekilde, lojistik faaliyetlere katma değer yaratan tüm hizmetleri üreten bir üst yapıdır. LM'in, gruba bilgi teknolojisi hizmetlerini sunan ayrı bir şirketi bulunmaktadır ve araştırmaya konu olan çalışmalar LM'in bu şirketi ile yürütülmüştür. Araştırmada gurubun bu şirketi LM olarak anılmaktadır.

LM yukarıdaki amaçlarına ulaşabilmek için AS/400 platformundaki uygulama yazılımlarını .NET ortamına taşıma projesini 2005 tarihinden beri yürütmektedir. Çizelge 7.1’de LM’in Acentelik Paketinin temel taşları sunulmaktadır.

**Çizelge 7.1 LM Acentelik Paketi temel taşları.**

<b>2009</b>	<b>Eylül</b>	• Acentelik Paketi ikinci aşamasına ait tamamlanan modüllerin canlı kullanıma alınma çalışmalarına başlandı.
	<b>Nisan</b>	• Devrede olan Acente Paketinde yaşanan performans sorunları nedeniyle performans iyileştirme çalışmaları yapıldı.
	<b>Ocak</b>	• Acente Paketinin ikinci aşaması projelendirildi ve proje başlatıldı. Yeni projeye başlandığı için değişiklik isteklerinin en az seviyede karşılanmasına karar verildi. Bu aşamaya ait modüllerin analiz ve kodlama çalışmalarına başlandı.
<b>2008</b>		• Acentelik Paketi ilk aşamasına ait tüm modüllerin tamamlanması ile birlikte yaygınlaştırma çalışmaları yapıldı.
	<b>Temmuz</b>	• Yaygınlaştırma çalışmalarını hızlandırmak amacıyla mevcut hataların düzeltilmesi önceliklendirildi.
<b>2007</b>		• Acentelik Paketinin ilk aşamasına ait diğer modüllerin analiz ve kodlama çalışmalarına devam edildi.
	<b>Ekim</b>	• Acentelik Paketi ilk aşamasına ait tamamlanan modülleri canlı kullanıma alındı.
<b>2006</b>		• Acentelik Paketinin ilk aşamasına ait diğer modüllerin analiz ve kodlama çalışmalarına devam edildi.
	<b>Kasım</b>	• Mainframe ortamındaki son büyük devreye alım yapıldı.
	<b>Mart</b>	• Acentelik Paketi ilk aşamasına ait tamamlanan modülleri canlı kullanıma alındı.
<b>2005</b>	<b>Temmuz</b>	• Acente Paketinin ilk aşaması, en kritik modüllerden oluşacak şekilde projelendirildi ve proje başlatıldı.
	<b>Ocak</b>	• Tüm acentelik paketinin MS .NET ortamında yeniden yazılmasına karar verildi.
<b>2004</b>		• Holding bünyesinde alınan çeşitli iş alanlarına ait SAP, NAVIS, CATLOGIC gibi paketler ile kurumsal entegrasyon projelerine başlandı.
<b>1989</b>		• IBM AS/400 ortamında RPG programlama dili ile yazılım geliştirilmeye başlandı.
<b>1984</b>		• Bilgi İşlem Servisi IBM'in System/23 bilgisayarlarını kullanarak faaliyete geçti.

LM, müşteri memnuniyeti sağlamaya yönelik yapılanma sürecinde aşağıdaki teknolojileri kullanmaktadır:

1. Yazılım geliştirmede gereksinimlerin yönetimi için IBM Telelogic DOORS,<sup>9</sup>
2. Yazılım kaynak kodunun denetimi, veri toplama, raporlama, proje izleme konularında Microsoft Team Foundation Server<sup>10</sup> (MS TFS),
3. Verilerin modellenmesi ve analizi için Sybase PowerDesigner,<sup>11</sup>
4. UML tabanlı modelleme ve tasarım için Sparke Systems,<sup>12</sup>
5. Bilgi teknolojileri hizmet yönetimini düzenleyen CA Service Desk.<sup>13</sup>

Yukarıdaki teknolojilerin kullanılması sayesinde gereksinimlerin tespiti, tasarım ve kodlama süreçlerinin her aşamasında, sınama, süreç yönetimi, kalite, yapılanış yönetimi ve ölçmenin yapılması mümkün olmaktadır. LM güncel yazılım mühendisliği ilkelerini kullanarak, yazılım kalitesi konusundaki çalışmalarını yürütmekte, geliştirme platformuna ve süreçlerine ilişkin düzenlemeleri yaparak belirlediği yazılım proje, süreç ve ürüne ait ölçme ve değerlendirme işlemleri ile yazılımı ölçmektedir.

LM .NET ortamında, birden fazla programcı tarafından aynı kod üzerinde yapılan değişiklikleri izlemek ve tek bir noktadan yönetmek için Microsoft Team Foundation Server (MS TFS) yazılımını kullanmaktadır.

LM AS/400 ortamını kullanan müşterinin bakım isteklerini karşılamak için bir bakım ekibine sahiptir ve bu alandaki müşteri hizmetlerini yönetmek için CA Service Desk yazılımını kullanmaktadır. AS/400 ortamındaki bakım süreçlerine ait veriler, stratejik olarak LM'in bu ortamı daha fazla desteklememe kararı nedeniyle detaylı olarak toplanmamakta, toplanan veriler de süreçlerin iyileştirilmesi yerine, bakım için harcanan eforun, LM'in müşterilerine fatura edilmesine yönelik olarak kullanılmaktadır.

LM, CMMI 2. uygunluk seviyesini hedeflemektedir. Böylece gereksinim yönetimi, proje planlama, proje izleme ve takip, tedarikçi sözleşme yönetimi, yapılandırma yönetimi, ölçme ve çözümleme, süreç ve ürün kalite güven süreç alanlarında sürekli gelişmeyi amaçlamaktadır.

<sup>9</sup> <http://www.telelogic.com/Products/doors/doors/index.cfm>

<sup>10</sup> <http://msdn.microsoft.com/>

<sup>11</sup> <http://www.sybase.com/products/modelingdevelopment/powerdesigner>

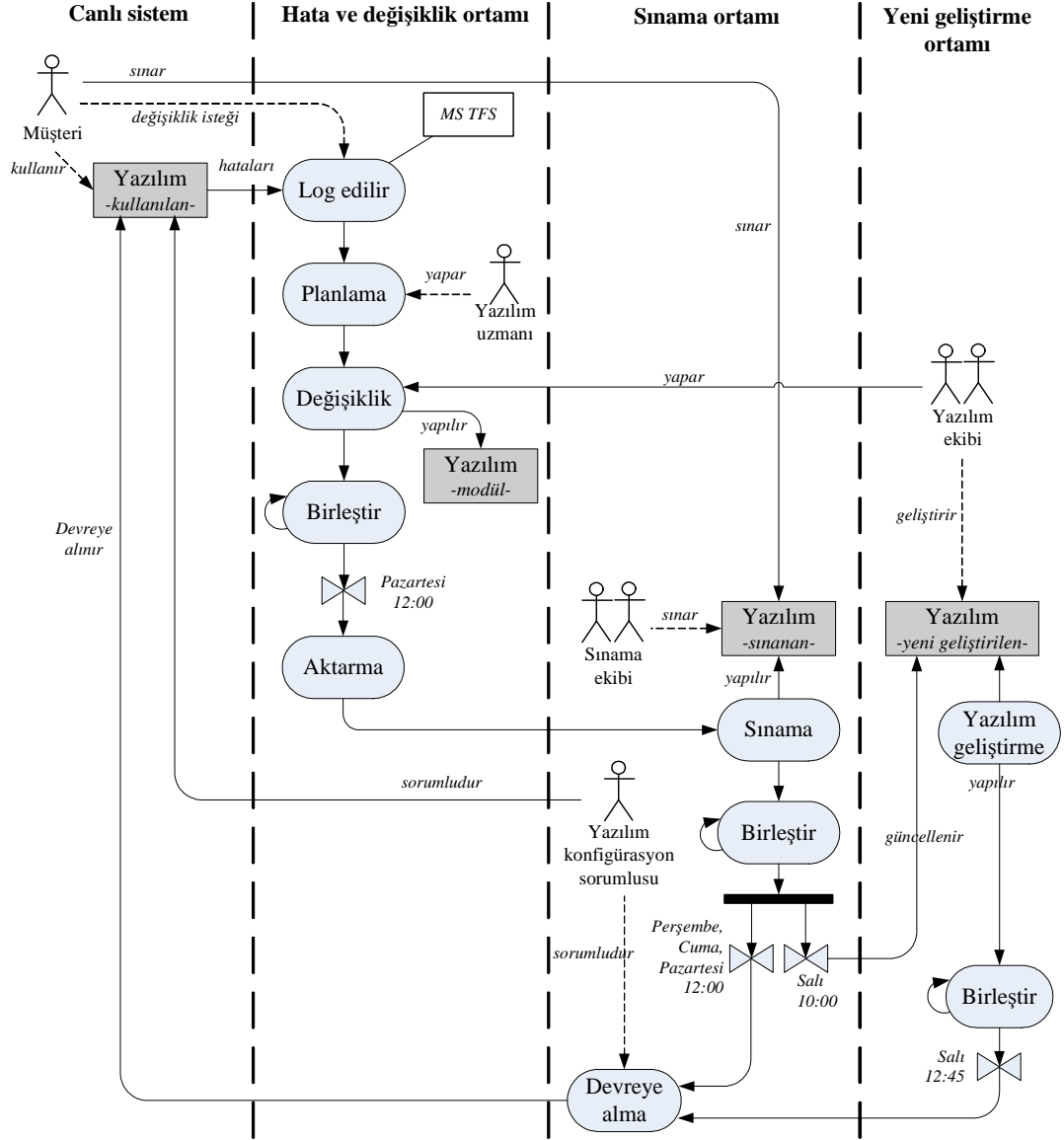
<sup>12</sup> <http://www.sparxsystems.com.au/>

<sup>13</sup> <http://www.ca.com/us/service-desk.aspx>

LM, sadece YNA (Yeni Nesil Acentecilik) projesiyle .NET ortamında yazılım geliştirmektedir. AS/400 ortamında yazılım geliştirmeye ilişkin herhangi bir işlem yapmamakta, sadece ürünün bakımını ve bazı acil durumlarda gerekli değişiklikleri yapmaktadır. LM'in YNA projesine ait yazılım geliştirme ortamı aşağıdaki temel faaliyetleri içermektedir.

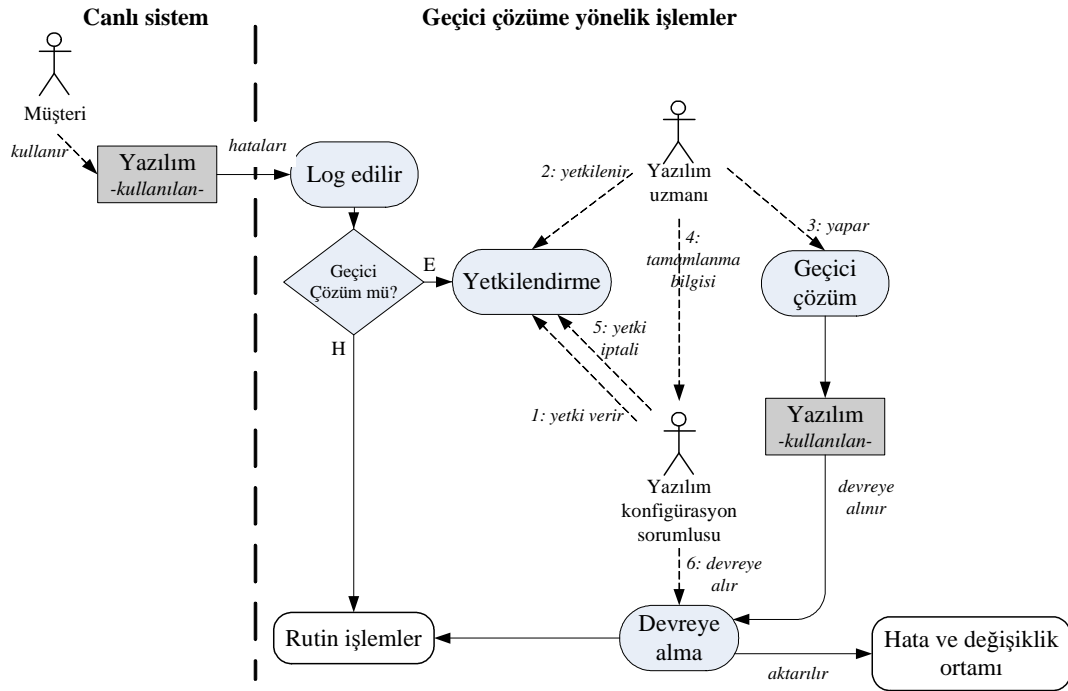
- 1. Yeni geliştirme ortamı.** Yazılım geliştirmenin yapıldığı ortamdır. İçerisinde yeni geliştirilen ve henüz canlı sistemde hiç yer almamış modülleri barındırmaktadır. Bu ortamdaki işlerin yürütülmesinden tüm yazılım geliştirme ekibi sorumludur.
- 2. Hata ve değişiklik ortamı.** İçerisinde canlı sistemde yer alan modüller yer almaktadır. Bu ortamdaki talepler, müşteri tarafından kullanılan canlı sistemden gelen hata ve değişiklik istekleridir ve bu istekler MS TFS tarafından düzenlenir. Bu ortamda yazılım uzmanı tarafından planlama ve düzeltme işlemleri yapılır. Yazılım uzmanı, düzeltmelerin hangilerinin yapılacağına ve canlı sistemde nasıl devreye alınacağına karar vermektedir ve her Pazartesi Saat 12:00'den, bir sonraki Pazartesi 12:00'ye kadar gelen değişiklikleri yaparak gruplandırır ve bir birleştirme işlemi ile sınama ortamına aktarır.
- 3. Sınama ortamı.** Bu ortamda, geliştirilen yazılım canlı sisteme aktarılmadan önceki sınama işlemleri yapılır. Yapılan değişikliklerin ve giderilen hataların canlı sistemde bir soruna neden olup olmadığı araştırılır. Müşteri tarafından bildirilen hata ve değişiklik isteklerinin karşılanıp karşılanmadığı her hafta Perşembe, Cuma ve takip eden haftanın Pazartesi saat 12:00'de birleştirilerek, LM'in sınama mühendisleri ve müşteri tarafından sınanır. Bu ortamdaki işleri yapmaktan sadece Proje Koordinasyon Sorumlusu yetkilidir.
- 4. Canlı sistem.** Müşteriye teslim edilmiş ve çalışan yazılım ürünlerinin bulunduğu ortamdır. Sınama ortamında sınıanan değişiklik istekleri ve hata düzeltmeleri, her hafta Salı günü saat 10:00 itibari ile sınama ortamından, canlı sisteme aktarılır. Ardından her Salı saat 12:45'de canlı sistem kesilir ve aktarılan kodlar canlı sistemde devreye alınır. Bu ortamdaki işleri yapmaktan sadece Proje Koordinasyon Sorumlusu yetkilidir.

LM'in YNA projesi için yazılım geliştirme ortamı Şekil 7.2'de sunulmaktadır.



Şekil 7.2 LM'in yazılım geliştirme ve bakım süreci.

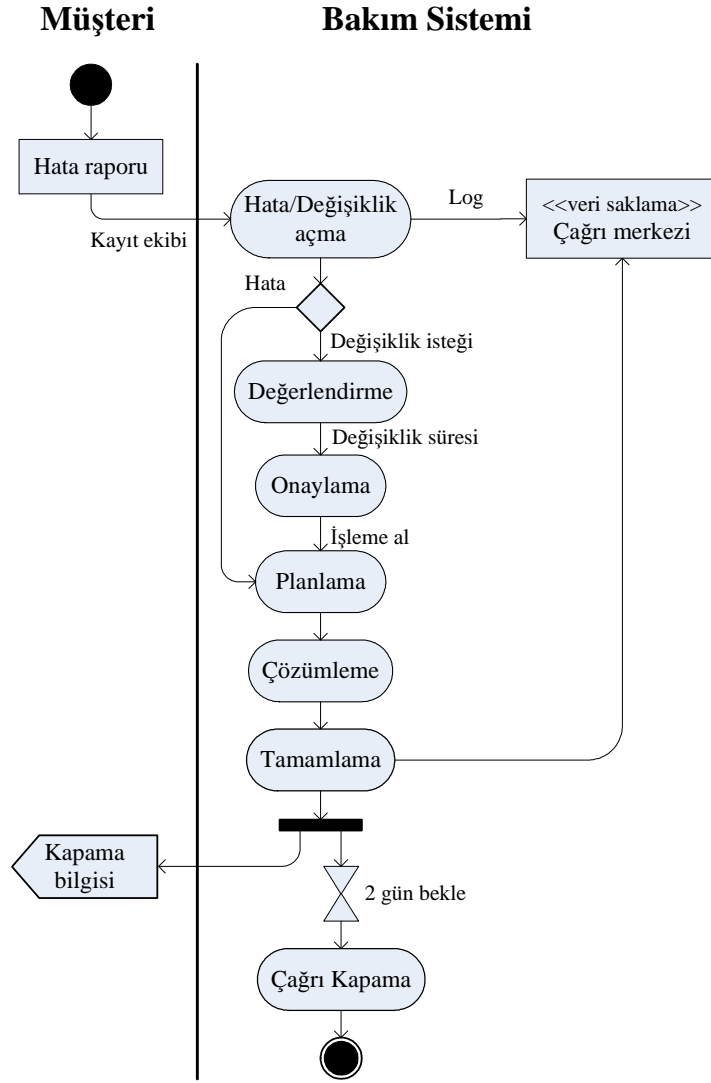
Sistemde, hata veya deęişiklik isteklerinin hızla yapılmasının istendięi durumda (geçici çözüm veya *quick fix*) yazılım konfigürasyon sorumlusu, yazılım uzmanını yetkilendirerek soruna hızlı bir şekilde müdahale edilmesini sağlamaktadır (Şekil 7.3). Bu durumda yazılımın sınanması işlemi yazılımcılar tarafından yapılmamakta, bunun yerine müşteri sınanma işlemini bizzat canlı sistem üzerinde gerçekleştirmektedir.



**Şekil 7.3 LM'in geçici çözüme yönelik işlemleri.**

LM bakım için gelen müşteri isteklerini karşılamak üzere bakım sistemine sahiptir ve sistemin ana akışı Şekil 7.4'de gösterilmektedir. Bakım sistemi müşteriden gelen istekle başlamakta ve gelen istek çağrı merkezinde saklanmaktadır. Gelen bakım isteğinin kapsamına göre yönlendirilmesi işlemi proje talep yöneticisi tarafından yapılmaktadır. Deęişiklik istekleri öncelikle analiz edilerek onaylanmakta, daha sonra da işleme alınarak planlanmaktadır. Bakım işlemi, bakım talebinin çeşidine göre, bakımcı veya yazılımcı tarafından gerçekleştirilmekte ve bakım dosyaları bakım kayıt elemanı tarafından güncelleştirilmektedir. Bu noktada, bakıma ait deęişikliğin sınanması işlemi Yazılım Destek Uzmanı tarafından yapılmakta, fakat sistematik bir test prosedürü uygulanmamaktadır. Tamamlanma bilgisi müşteriye gönderilmekte ve iki gün beklendikten sonra çağrı kapatılmaktadır.





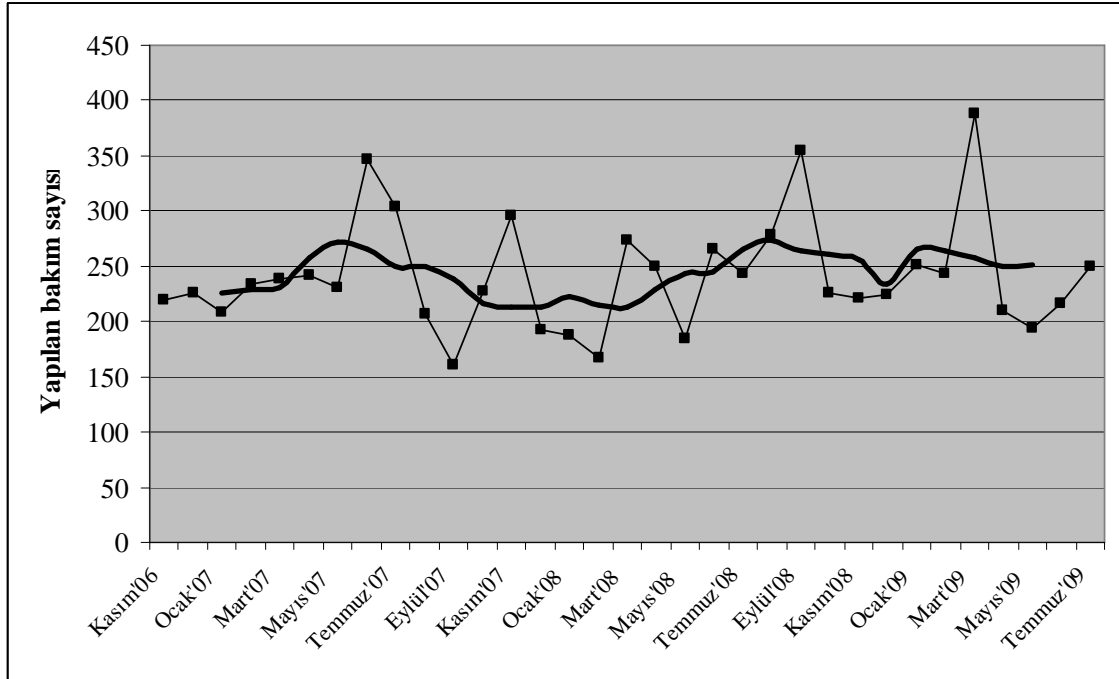
**Şekil 7.4 LM'in bakım sistemi ana akışı.**

LM'in bu bölümde bahsedilen uygulamaları ve kullanılan araçları, yazılım mühendisliğinin ilkeleri ile bağdaşmaktadır. LM'in kalite sürecini takip etmesi, ölçme ve değerlendirmeye yönelik süreç yapılanması, lojistik alandaki deneyimi ve destek verdiği grup şirketlerinin ticari başarısı dikkate alındığında, LM tezin uygulama alanı için ideal bir ortam sunmaktadır.

### 7.1.2 Mainframe ortamı

Mainframe ortamı IBM AS/400 sunucuların bulunduğu ve gereksinime dayalı acil durumlar dışında yazılımın üretilmediği bir ortamdır. LM tarafından mainframe ortamındaki son büyük devreye alma Kasım 2006'da gerçekleşmiş olup, bu tarihten itibaren büyük çaplı devreye alma işlemi gerçekleştirilmemiştir.

Mainframe ortamındaki yazılımda gerçekleştirilen, yazılım destek ve geliştirme faaliyetlerine yönelik veriler Ek-E1'de sunulmaktadır. Ek-E1'deki veriler kullanılarak, Kasım 2006 ile Temmuz 2009 ayları arasında, LM tarafından mainframe ortamında yapılan bakım sayıları ve bakım sayılarının genel eğilimini elde etmek amacıyla hareketli ortalama ile hesaplanan ve koyu çizgi ile gösterilen bakım yoğunluğu fonksiyonu Şekil 7.5'de sunulmaktadır.



**Şekil 7.5 Mainframe ortamındaki bakım sayıları.**

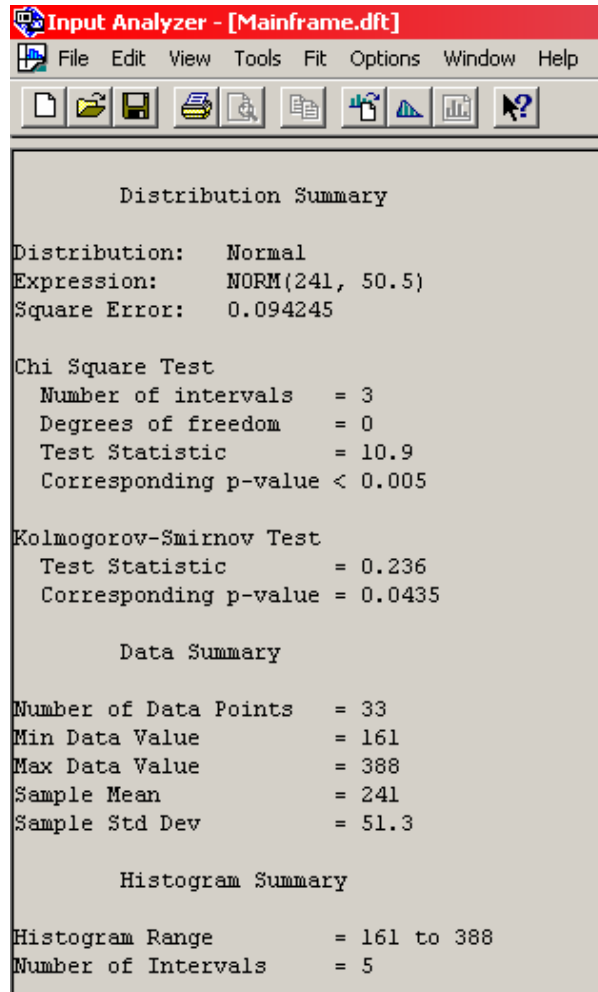
Şekil 7.5'teki grafikten, mainframe ortamındaki bakım sayılarının kararlılık gösterdiği gözlem sonuçlarına göre anlaşılmaktadır. Kasım 2006'dan sonra büyük devreye alımların yapılmadığı mainframe ortamındaki bu durum, Musa (1987, S:11-13) tarafından da belirtildiği gibi uzun süredir kullanımda olan bir yazılım için beklenen bir sonuçtur.

Mainframe ortamındaki bakım sayılarına ait aritmetik ortalama ve standart sapma değerleri hesaplanmış ve Çizelge 7.2’de sunulmuştur.

**Çizelge 7.2 Mainframe ortamında yapılan bakımların istatistik değerleri.**

Bakım ortamı	N	$X_{ort}$	Ss
Mainframe	33	241	51,3

Şekil 7.6’da Arena<sup>14</sup> yazılımının kullanılması sonucunda, mainframe ortamındaki bakım sayılarının istatistiksel sonuçları sunulmaktadır.



**Şekil 7.6 Mainframe ortamındaki bakım sayılarının istatistiksel sonuçları.**

<sup>14</sup> Arena, Rockwell Automation Inc. (<http://www.rockwellautomation.com>) tarafından üretilen bir simülasyon yazılımıdır. Tezde, Arena yazılımının standart bileşenlerinden olan *Input Analyzer*, bakım verilerin, olasılık dağılım fonksiyonlarına uygunluğunun araştırılmasında kullanılmıştır.

Yapılan analiz sonucunda, mainframe ortamındaki aylık bakım sayısını en iyi temsil eden dağılımın normal dağılım olduğu görülmüştür. Ross (2006) tarafından verilen normal dağılımın olasılık yoğunluk fonksiyonuna, Şekil 7.6'daki değerler konularak (10) nolu eşitlikte gösterilen fonksiyonu değeri elde edilmiştir.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} = \frac{1}{50,5\sqrt{2\pi}} e^{-(x-241)^2/(2.(50,5)^2)} \quad (10)$$

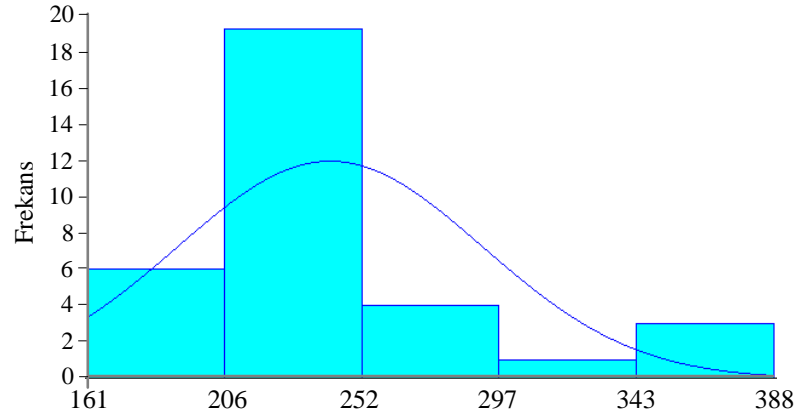
Mainframe ortamındaki bakım sayılarının, Arena yazılımının önerdiği normal dağılıma uyum gösterip göstermediğini sınamak için aşağıdaki hipotez testi kurulmuştur.

Ho: Mainframe ortamındaki bakım sayısı, normal dağılıma uyum göstermektedir.

H1: Mainframe ortamındaki bakım sayısı, normal dağılım göstermemektedir.

Buna göre, 0,05 güven düzeyi ve 32 serbestlik derecesiyle (gözlem yapılan ay sayısı-1), ki-kare ( $\chi^2$ ) tablo değeri 46.1942 olarak bulunmuştur. Arena yazılımı tarafından test istatistiği 10.9 olarak hesaplanmıştır. Bu değer, tablo değerinden küçük olduğundan dolayı, Ho hipotezi red edilememiştir. Bir başka ifadeyle, normal dağılım, mainframe ortamındaki bakım sayılarını temsil etmektedir.

Şekil 7.7'de mainframe ortamındaki bakım verilerinin histogram grafiği ve Arena yazılımı ile elde edilen en uygun istatistik dağılımı sunulmaktadır. Grafiğin incelenmesi sonucunda; aylık bakım verilerinin, 206-252 aralığında yoğunlaştığı görülmektedir.

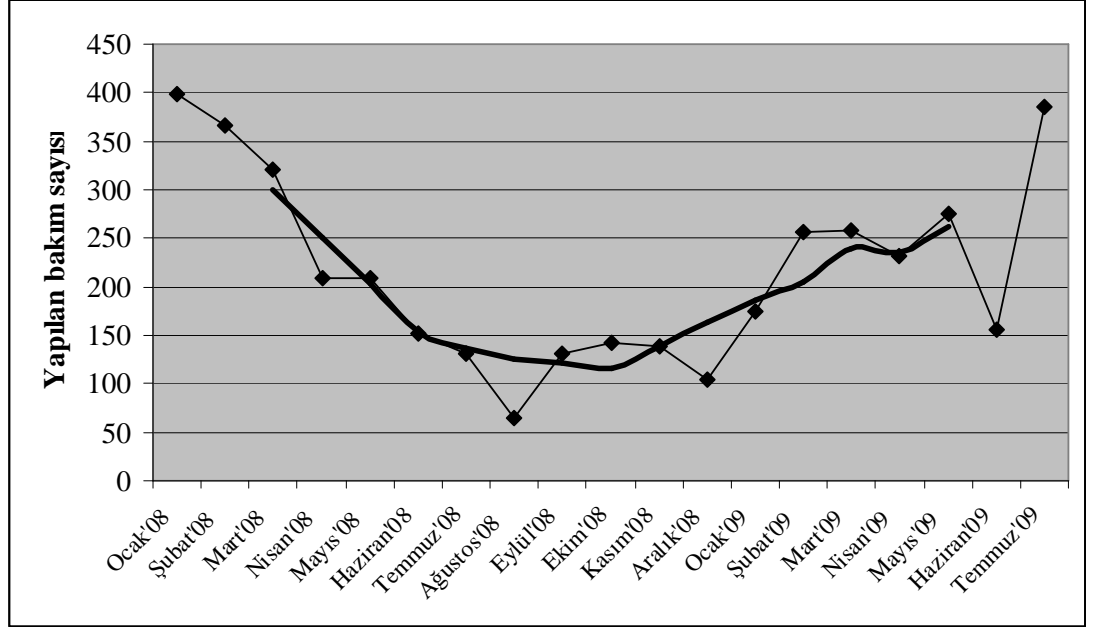


**Şekil 7.7 Mainframe ortamındaki bakım sayılarının histogramı ve istatistik dağılımı.**

Elde edilen bulgular, LM yazılım yöneticileri ve bakımçılarına, mainframe ortamındaki yazılım destek ve geliştirme süreçlerinin tahminlenmesinde yardımcı olacak ve ileriye yönelik bakım kaynaklarının daha iyi planlanmasını sağlayacaktır.

### 7.1.3 Web ortamı

Web ortamı, LM'in acentelik paketinin MS .NET ortamında yeniden yazıldığı, halen de yazılım geliştirme ve iyileştirme çabalarının devam ettiği bir ortamdır. LM tarafından web ortamında, geliştirilmesi tamamlanan modüller, Ocak 2008 tarihinden itibaren devreye alınmıştır. Bu tarihten önceki çalışmalar, tamamen LM'in yazılımın geliştirme çabalarına yönelik olduğundan, Ek-E2'de sunulan web ortamına ait genel bakım verilerinden, Ocak 2008 ile Temmuz 2009 ayları arasında gerçekleşen ve müşterilerinden gelen hata ve değişiklik isteklerini içeren veriler tezde dikkate alınmıştır. Bu doğrultuda, LM tarafından web ortamında yapılan bakım sayıları ve bakım sayılarının genel eğilimini elde etmek amacıyla hareketli ortalama ile hesaplanan ve koyu çizgi ile gösterilen bakım yoğunluğu fonksiyonu Şekil 7.8'de sunulmaktadır.



**Şekil 7.8 Web ortamındaki bakım sayıları.**

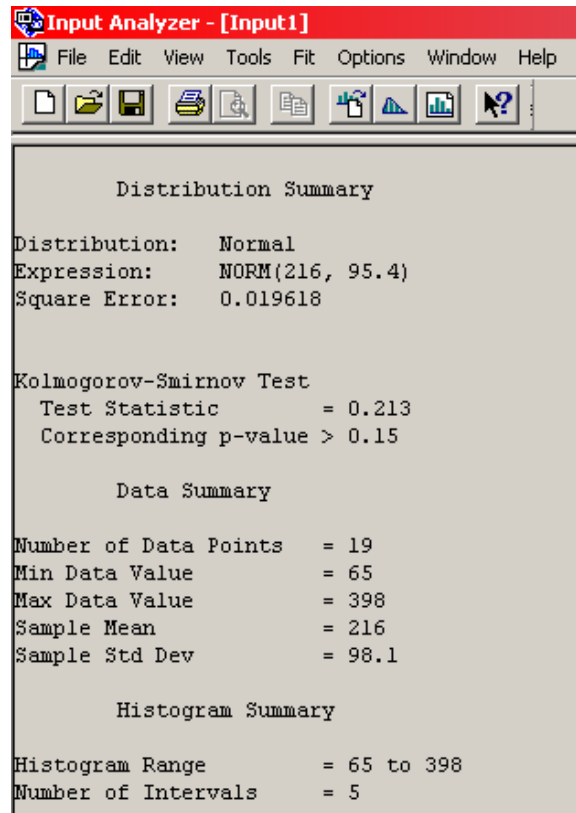
Şekil 7.8'deki grafikten, web ortamındaki bakım sayılarının Çizelge 7.1'de sunulan Acentelik Paketinin geliştirilmesindeki yönetim kararları ile uyumluluk gösterdiği gözlem sonuçlarından anlaşılmaktadır. Örneğin; 2008 yaz aylarında hataların düzeltilmesine hız verilmesi nedeniyle, aynı dönemdeki bakım sayılarında bir azalma gözlemlenmiş ve Nisan 2009'da mainframe ortamında yaşanan performans sorunlarının, web ortamında entegrasyon sorunlarına yol açması sonucunda, hem mainframe (Şekil 7.5) hem de web ortamındaki bakım çalışmalarının arttığı gözlemlenmiştir.

Web ortamındaki bakım sayılarına ait aritmetik ortalama ve standart sapma değerleri hesaplanmış ve Çizelge 7.3’de sunulmuştur.

**Çizelge 7.3 Web ortamında yapılan bakımların istatistik değerleri.**

Bakım ortamı	N	X <sub>ort</sub>	Ss
Web	19	216	98,1

Şekil 7.9’da Arena yazılımının kullanılması sonucunda, web ortamındaki bakım verilerinin istatistiksel sonuçları sunulmaktadır.



**Şekil 7.9 Web ortamındaki bakım sayılarının istatistiksel sonuçları.**

Yapılan analiz sonucunda, web ortamındaki aylık bakım sayısını en iyi temsil eden dağılımın normal dağılım olduğu görülmüştür. Ross (2006) tarafından verilen normal dağılımın olasılık yoğunluk fonksiyonuna, Şekil 7.9’daki değerler konularak (11) nolu eşitlikte gösterilen olasılık yoğunluk fonksiyonu değeri elde edilmiştir.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} = \frac{1}{95,4\sqrt{2\pi}} e^{-(x-216)^2/(2.(95,4)^2)} \quad (11)$$

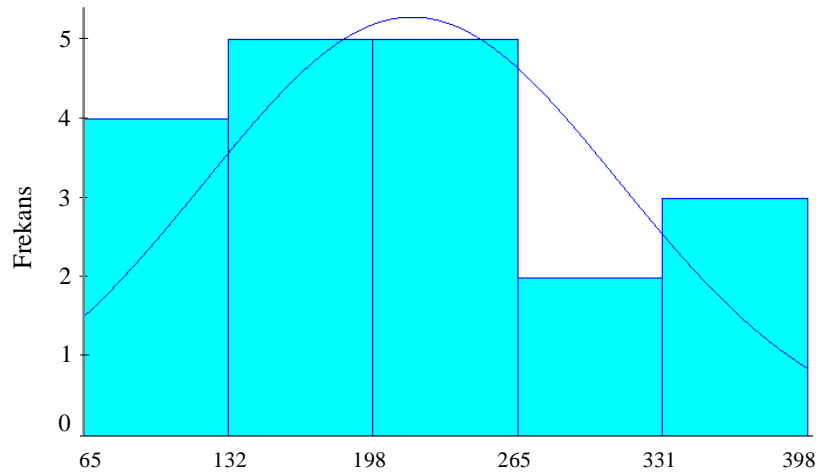
Web ortamındaki bakım sayılarının, Arena yazılımının önerdiği normal dağılıma (Şekil 7.9) uyum gösterip göstermediğini test etmek için aşağıdaki hipotez testi kurulmuştur.

Ho: Web ortamındaki bakım sayısı, normal dağılıma uyum göstermektedir.

H1: Web ortamındaki bakım sayısı, normal dağılım göstermemektedir.

Buna göre, 0,05 güven düzeyi ve 19 serbestlik derecesiyle (gözlem yapılan ay sayısı), Kolmogorov-Smirnov (K-S) tablo değeri 0,301 olarak bulunmuştur. Arena yazılımı tarafından test istatistiği 0,213 olarak hesaplanmıştır. Bu değer, tablo değerinden küçük olduğundan dolayı, Ho hipotezi red edilememiştir. Bir başka ifadeyle, normal dağılım, web ortamındaki bakım sayılarını temsil etmektedir.

Şekil 7.10'da web ortamındaki bakım verilerinin histogram grafiği ve Arena yazılımı ile elde edilen en uygun istatistik dağılımı sunulmaktadır. Bu grafiğin incelenmesi sonucunda; aylık bakım verilerinin, 132-265 aralığında yoğunlaştığı görülmektedir.

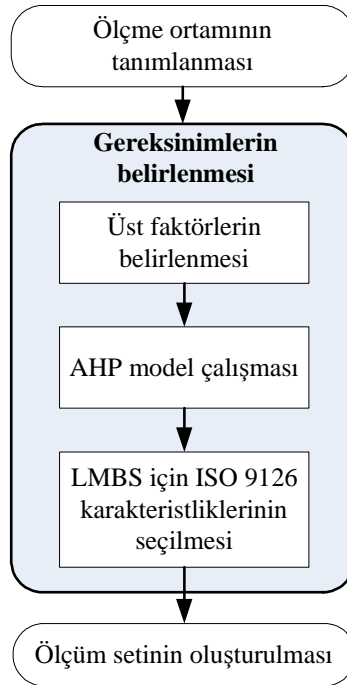


**Şekil 7.10 Web ortamındaki bakım sayılarının histogramı ve istatistik dağılımı.**

Elde edilen bulgular, LM yazılım yöneticileri ve bakımcılarına, web ortamındaki yazılım destek ve geliştirme süreçlerinin tahminlenmesinde yardımcı olacak ve ileriye yönelik bakım kaynaklarının daha iyi planlanmasını sağlayacaktır.

## 7.2 Gereksinimlerin Belirlenmesi

LM yazılım ürününü ölçmeye yönelik gereksinimlerin belirlenmesi, bu gereksinimleri karşılayan kalite kriterlerinin belirlenmesi anlamına gelmektedir. Gereksinimlerin belirlenmesi üç aşamadan oluşmaktadır. Bunlar LM'in yazılım kalite hedefine ulaşmasını sağlayacak üst faktörlerin belirlenmesi, belirlenen üst faktörlerle ISO 9126 standartının ilişkilendirilerek, LMBS için hangi kalite karakteristiklerinin daha önemli olduğuna karar verilmesi ve son olarak da bir sonraki adım olan hangi karakteristikleri ölçmekte kullanılacak olan metriklerle ilişkin verilerin toplanmasıdır. Şekil 7.11'de ölçme sürecinde gereksinimlerin belirlenmesine ilişkin alt süreçler gösterilmektedir.

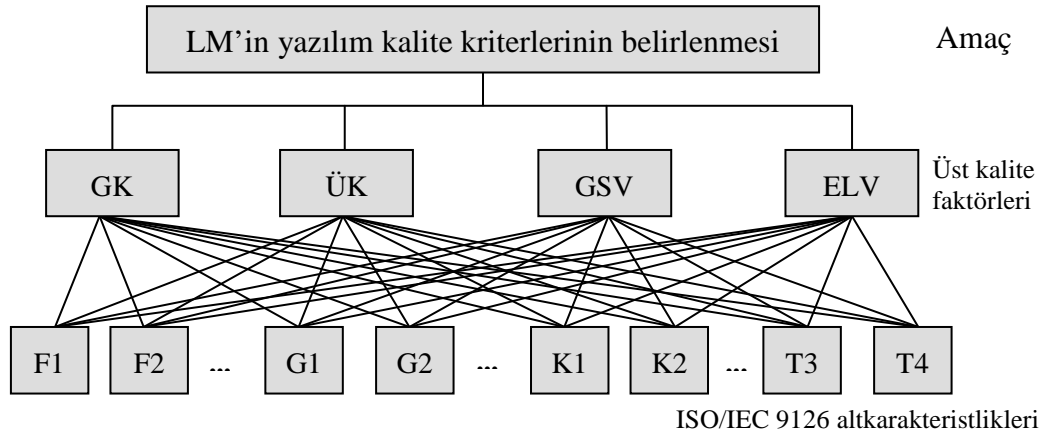


**Şekil 7.11 LMBS'ne ait ölçme gereksinimlerinin belirlenmesi.**

Üst faktörlerin belirlenmesindeki temel amaç, ISO 9126 standartına ait altkarakteristiklerden hangilerinin LM için gerekli olduğuna yönelik kararın verilebilmesi için AHP yönteminin uygulamada getirdiği gerekliliktir. Üst kalite faktörleri, LM'in yazılım kalite hedeflerine ulaşılmasında gerekli olan faktörleri temsil etmektedir. Böylelikle yazılımın genel kalite standartlarından ödün vermeden, kullanılacak metrik sayılarında sınırlamalar yapılabilmesi ve ölçme işleminin daha verimli, uygulanabilir hale gelebilmesi mümkün olabilecektir.



Üst faktörlerin belirlenmesinden sonra, ISO 9126 standartında belirtilen karakteristiklerden hangilerinin kullanılacağını saptanabilmesi için AHP yöntemi seçilmiştir. AHP yönteminin karar verme sürecinde uygulanabilmesi için, LM yazılım üst kalite faktörleri ana kriterler olarak alınmış, ISO 9126 altkarakteristikleri de alt kriter olarak kullanılmıştır (Şekil 7.12).



**Şekil 7.12 LM için karar hiyerarşisi ağacı.**

### 7.2.1 Lojistik merkez için yazılım üst kalite faktörlerin belirlenmesi

LM için yazılım üst kalite faktörlerin belirlenmesi işlemi, LMBS yazılımını üreten yazılım geliştirme müdürü, proje yöneticisi, takım yöneticisi ve yazılım uzmanları ve bu alandaki akademisyenlerin görüşleri alınarak yapılmış ve aşağıda sunulmuştur:

1. **Gereksinimlerin kalitesi (GK)**, müşteri gereksinimlerinin ne kadar iyi anlaşıldığı ile ilgilidir ve kalite hedeflerine ulaşılması, gereksinimlerin tam ve doğru olarak anlaşılması ile mümkündür. Bu da yazılım kalitesi açısından karakteristikler ile başta müşteri olmak üzere, diğer paydaşların gereksinimlerini anlamak, bu gereksinimleri sistemin parçası haline getirmek ve alt sistemlere de bu gereksinimlerin hayata geçirilmesi ile ilişkilidir (Hull vd., 2005).
2. **Ürün Kalitesi (ÜK)**, yazılımın konulan koşullara uygunluğu (örneğin şartname ile belirtilen koşullar) ve sahip olması gereken nihai özellikleridir. Koşullara uygun olarak üretilmeyen yazılım, hataya neden olmakta ve ürün kalitesini tehdit etmektedir. Yazılımın gerekli fonksiyonlara sahip olması, güvenilirlik beklentileri, kullanım kolaylığı örnek olarak verilebilir.

3. **Gerçekleştirim ve sınaama verimliliği (GSV)**, yazılımın geliştirmesinden başlayıp, hataların ayıklanmasına kadar olan süreçlerdeki operasyon verimliliğidir. Geliştirme sürecindeki sorunların farkedilmesini sağlamak, yazılım kalitesindeki değişimleri izleyebilmek ve bu iyileştirmelere olan katkı GSV olarak tanımlanabilir.
4. **Elverişlilik (ELV)**, yazılım kalitesine yönelik veri setlerinin elde edilmesindeki kolaylık, ekonomiklik, elde edilen verinin kullanılabilirliği, etkinliği ve şirket kültürüyle/alışkanlıklarıyla uyumluluğudur.

### 7.2.2 Analitik Hiyerarşi Sürecinin uygulanması

Bu bölümde, Bölüm 4’te sunulan ve karar verme sürecinde kullanılacak olan yöntemin, teorik yapısına uygun olarak yapılan uygulama anlatılmaktadır.

#### Adım 1: Üst kalite faktörlerine ait ağırlıkların bulunması

Bu adımda, LMBS yazılım kalite kriterlerinin en önemli üst karakteristiklerinin ağırlıklarının bulunması işlemi anlatılmaktadır. Bunun için, üst karakteristiklere ait önem değerleri, LM yazılım geliştirme ekibindeki uzman kişiler tarafından yapılan ikili karşılaştırmalar sonucunda elde edilmiştir ve üst kalite faktörlerine ait karşılaştırmalı değerler matrisi (12) nolu eşitlikte sunulmuştur.

$$A = \begin{matrix} & \begin{matrix} GK & ÜK & GSV & ELV \end{matrix} \\ \begin{matrix} GK \\ ÜK \\ GSV \\ ELV \end{matrix} & \begin{bmatrix} 1 & 0,33 & 1 & 0,20 \\ 3 & 1 & 3 & 0,33 \\ 1 & 0,33 & 1 & 0,33 \\ 5 & 3 & 3 & 1 \end{bmatrix} \end{matrix} \quad (12)$$

#### Adım 2: Normalizasyon işlemi

Normalizasyon işlemi, (12) nolu eşitlikte elde edilen ikili karşılaştırma matrisine, (2) ve (3) nolu ifadelerde sunulduğu şekilde yapılmıştır. Örneğin, (13) nolu eşitlikte görüldüğü gibi,  $A_{norm_{11}}=0,10$  değeri, (12) nolu eşitlikteki  $a_{11}=1$  değerinin, bu elemanın bulunduğu sütundaki elemanların toplamına bölünmesi ile elde edilmiştir.

$$\sum_{i=1}^4 A_{i1} = 1+3+1+5 = 10, \quad A_{norm_{11}} = 1/10 = 0,10.$$

$$A_{\text{norm}} = \begin{bmatrix} 0,10 & 0,07 & 0,12 & 0,11 \\ 0,30 & 0,21 & 0,37 & 0,18 \\ 0,10 & 0,07 & 0,12 & 0,18 \\ 0,50 & 0,64 & 0,37 & 0,54 \end{bmatrix} \quad (13)$$

### Adım 3: Ağırlıkların hesaplanması

Üst kalite faktörlerine ait ağırlıklar (13) nolu eşitlikten, (4) nolu formül kullanılarak hesaplanmıştır.

$$w_{\text{GK}} = \frac{0,10 + 0,07 + 0,12 + 0,11}{4} = 0,10 \quad w_{\text{ÜK}} = \frac{0,30 + 0,21 + 0,37 + 0,18}{4} = 0,27$$

$$w_{\text{GSV}} = \frac{0,10 + 0,07 + 0,12 + 0,18}{4} = 0,12 \quad w_{\text{ELV}} = \frac{0,50 + 0,64 + 0,37 + 0,54}{4} = 0,51$$

Yukarıda hesap edilen ağırlıklar, vektör olarak (14) nolu eşitlikte gösterilmiştir.

$$w = [0,10 \quad 0,27 \quad 0,12 \quad 0,51] \quad (14)$$

### Adım 4: Tutarlılık Denetimi

Bu adımda, LM yazılım geliştirme ekibi tarafından yapılan ve (12) nolu eşitlikte sunulan ikili karşılaştırmaların tutarlılık denetimi yapılmıştır. Buna göre  $Aw^T$  vektörü, (5) nolu eşitlikte sunulan formüle göre, (15) nolu eşitlikteki gibi hesaplanmıştır.

$$Aw^T = \begin{bmatrix} 1 & 0,33 & 1 & 0,20 \\ 3 & 1 & 3 & 0,33 \\ 1 & 0,33 & 1 & 0,33 \\ 5 & 3 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0,10 \\ 0,27 \\ 0,12 \\ 0,51 \end{bmatrix} = \begin{bmatrix} 0,41 \\ 1,10 \\ 0,48 \\ 2,18 \end{bmatrix} \quad (15)$$

(15) nolu eşitliğin sonucu kullanılarak, T değeri (6) nolu eşitlikten

$$T = \left( \frac{1}{4} \right) \left\{ \frac{0,41}{0,10} + \frac{1,10}{0,27} + \frac{0,48}{0,12} + \frac{2,18}{0,51} \right\} = \frac{16,44}{4} = 4,11 \quad (16)$$

olarak hesaplanmıştır. (16) nolu eşitlikten elde edilen T değeri ve seçenek sayısı kullanılarak, (7) nolu eşitlikten tutarlılık göstergesi

$$TG = \text{Tutarlılık göstergesi} = \frac{4,11 - 4}{4 - 1} = 0,036 \quad (17)$$

olarak elde edilmiştir.

(17) nolu eşitlikten elde edilen tutarlılık göstergesi ve Çizelge 4.2'deki n=4 için rassal indeks değeri olan 0,89 değeri yardımıyla, (8) nolu eşitlik kullanılarak tutarlılık değeri;

$$TD = \text{Tutarlılık değeri} = \frac{0,036}{0,89} = 0,041 \quad (18)$$

olarak hesaplanmıştır.

(18) nolu ifadede elde edilen tutarlılık değeri, Saaty tarafından belirtilen (Bhushan & Rai, 2004, s.17) ve genel olarak kabul edilen sınır değer olan 0,10'den daha küçüktür (0,041 < 0,10). Dolayısı ile LM için yapılan üst kalite faktörlerine ait ikili karşılaştırmaların yeterli tutarlılığa sahip olduğu sonucuna ulaşılmıştır. Ancak AHP kendi içinde ne kadar tutarlı bir yöntem olsa da, elde edilen sonuçlar, karar vericilerin (12) nolu eşitlikte sunulan değerlendirme matrisindeki yargılarına bağlıdır.

Böylece, LMBS yazılım kalite kriterlerinden en önemli üst karakteristiklerinin ağırlıkları (14) nolu eşitlikte sunulduğu gibi bulunmuş, (18) nolu eşitlikte sunulduğu gibi bulunan ağırlıkların yeterli tutarlılığa sahip olduğu anlaşılmıştır.

Tezin bu aşamasında, (14) nolu ifadede elde edilen sonuçlar, önem sırasına göre sıralanmış (Çizelge 7.4) ve LM yazılım ekibine göre, en önemli yazılım kalite üst karakteristiği “**Elverişlilik**” olarak belirlenmiş ve Ara Sonuç 1'e ulaşılmıştır.

**Çizelge 7.4 Yazılım kalite üst karakteristikleri önem derecesi.**

	<b>AĞIRLIK</b>
<b>Elverişlilik</b>	<b>0,51</b>
<b>Ürün kalitesi</b>	0,27
<b>Gerçekleştirim ve sınama verimliliği</b>	0,12
<b>Gereksinimlerin kalitesi</b>	0,10

**Ara Sonuç 1 : Elverişlilik, LM yazılımı için en ağırlıklı üst kalite faktörüdür.**

Elde edilen sonuca göre, LM tarafından, yazılım kalite kriteri olarak, yazılımı ölçmeye yönelik veri setlerinin elde edilmesindeki kolaylık, ekonomiklik, elde edilen verinin kullanılabilirliği, etkinliği ve şirket kültürüyle/alışkanlıklarıyla uyumluluğu en ağırlıklı üst kalite faktörü olarak algılanmaktadır.

### Adım 5: Üst kalite faktörlerine göre seçeneklerin değerlendirilmesi

Bu aşamada GK, ÜK, GSV ve ELV üst kalite faktörleri esas alınarak, her bir faktör için ISO 9126 standartındaki altkarakteristiklere ait ikili karşılaştırma yargılarının oluşturulması işlemi, Adım 1, 2 ve 3'te anlatılan işlemlere benzer biçimde MS Excel yazılımı kullanılarak gerçekleştirilmiştir. Bu işlemlerin sonucunda elde edilen matrisler Ek-Ç'de sunulmaktadır. Bu aşamada, tutarlılık analizi; 21 adet parametrenin fazla sayıda olması nedeniyle anlamlı sonuçların üretilmesine engel olduğu için yapılmamıştır. İlk dört adımdaki işlemlere benzer biçimde yapılan hesaplamaların sonucunda, LM yazılımı için elde edilen kalite karakteristikleri, önem derecesine göre sıralı olarak Çizelge 7.5'de sunulmaktadır.

**Çizelge 7.5 LM yazılım karakteristikleri önem derecesi.**

Sıra	Kod	Karakteristik	Sonuç	Yığımsal toplam
1	B2	Değiştirilebilirlik	8	8
2	K1	Anlaşılabilirlik	7	15
3	B1	Çözümenebilirlik	7	22
4	F2	Doğruluk	7	29
5	B4	Sınanabilirlik	7	36
6	F4	Uyumluluk	7	43
7	F1	Uygunluk	6	49
8	K3	İşlerlik	6	55
9	F3	Birlikte işlerlik	5	60
10	F5	Güvenlik	5	65
11	T1	Uyarlanabilirlik	4	69
12	G1	Olgunluk	4	73
13	B3	Durağanlık	4	77
14	G3	Kurtarılabilirlik	4	81
15	K2	Öğrenebilirlik	4	85
16	G2	Hata toleransı	3	88
17	T3	Uygunluk	3	91
18	E1	Zaman davranışı	3	94
19	E2	Kaynak yararlanımı	2	96
20	T2	Kurulabilirlik	2	98
21	T4	Yer değiştirilebilirlik	2	100

Çizelge 7.5'ten elde edilen sonuçlardan, ISO 9126 standartını oluşturan 21 yazılım kalite alt karakteristiğın, LM tarafından hangi önem sırasında algılandığı anlaşılmaktadır. Bu önem sırası sayesinde, Ara Sonuç 2 elde edilmiştir.

**Ara Sonuç 2 :** Değıştirilebilirlik, LM'in yazılım kalite gereksinimlerini temsil eden en önemli ISO 9126 altkarakteristiğidir.

Tezin bu aşamasında, en önemli üstkarakteristiğın hangisi olduğunun belirlenmesi amacıyla, Çizelge 7.5'te önem değeri sunulan LM yazılım altkarakteristikleri, ait oldukları üstkarakteristikte gruplanarak toplanmış ve üstkarakteristik için bir ortalama değeri hesaplanmıştır. Örneğın; bakım yapılabilirliğe ait dört altkarakteristiğın önem dereceleri toplanmış ve üstkarakteristiğın sahip olduğu altkarakteristik sayısına bölünerek ortalama bir değeri elde edilmiştir. Bu değeri, bakım yapılabilirlik için  $BY_{ort} = ( 8 + 7 + 7 + 4 ) / 4 = 6,50$ 'dir. Benzer hesaplamaların sonucunda, ISO 9126 standartına ait üstkarakteristiklerin önem değeri Çizelge 7.6'da gösterilmektedir.

**Çizelge 7.6 LM yazılımını en iyi temsil eden üstkarakteristikler.**

Sıra	ISO 9126 üstkarakteristikleri	Aritmetik Ortalama	Altkarakteristikler
1	<b>Bakım yapılabilirlik</b>	<b>6,50</b>	<b>B1, B2, B3, B4</b>
2	Fonksiyonellik	6,00	F1, F2, F3, F4, F5
3	Kullanılabilirlik	5,67	K1, K2, K3
4	Güvenilirlik	3,67	G1, G2, G3
5	Taşınabilirlik	2,75	T1, T2, T3, T4
6	Etkinlik	2,50	E1, E2

Çizelge 7.6, LM'in ISO 9126 yazılım ürün kalite standartına ait üstkarakteristiklerin hangi önem sırasında algılandığını göstermektedir. Bu algı, Ara Sonuç 3'ün elde edilmesini sağlamıştır.

**Ara Sonuç 3 :** Bakım yapılabilirlik, LM için en önemli ISO 9126 karakteristiğidir.

### 7.3 Belirlenen Gereksinimlere Göre Tezin Kapsamına Karar Verme

Çalışmanın bu aşamadan sonraki kapsamı, 1, 2 ve 3 nolu ara sonuçlarda sunulan yargılar kullanılarak belirlenmiştir. Bir nolu ara sonuçta elde edilen “**Elverişlilik**” üst kalite faktörünün etkisiyle, tezin amacına ve çalışma şekline uygun olarak ve Derharnais<sup>15</sup>,’ın yorumları doğrultusunda, LM için karakteristiklerin mümkün olabildiği kadar sınırlandırılmasına karar verilmiştir. Bunun temel nedeni, tez kapsamında LM’in gereksinimleriyle ilişkisi olmayan karakteristiklerin incelenmesini engellemek ve yazılım ürünü kalite ölçme işleminin LM için yapılabilirliğini göstermektedir. Gereksinimler ile ilgili olmayan kalite karakteristikleri yazılım kalitesini düşüren, kalite ile ilişkisiz karakteristikler olarak tanımlanmaktadır (Xenos, 2001) ve üretken olmayan karakteristiklerin ortadan kaldırılması da yazılım kalitesi açısından önemlidir (Aydın, 2003).

Bu doğrultuda, bakım yapılabilirlik karakteristiğine ait olan durağanlık altkarakteristiği Çizelge 7.5’de de sunulduğu gibi düşük öneme sahip olmasından dolayı elenerek alttaki sonuca ulaşılmıştır:

#### **Sonuç 1**

**Bakım Yapılabilirlik** karakteristiği ve bu karakteristiğe ait **Değiştirilebilirlik**, **Çözümlenebilirlik**, **Sınanabilirlik** altkarakteristikleri, LM için en önemli ISO 9126 karakteristikleridir.

<sup>15</sup> Dr. Jean-Marc Desharnais ile 11 Eylül 2008 tarihinde yapılan görüşmenin sonucunda sadece bakım yapılabilirlik karakteristiğinin seçilmesi önerilmiştir. Dr. Desharnais, Kanada, Quebec Üniversitesi, Yüksek Teknoloji Okulu (*ÉTS, École de technologie supérieure*) Uygulamalı Metrikler alanında Yazılım Mühendisliği Labotaruvarında (*Software Engineering Laboratory in Applied Metrics*) ve Boğaziçi Üniversitesi TAM ve Softlab projelerinde araştırmacı olarak akademik çalışmalarını sürdürmektedir. Dr. Desharnais, ayrıca ISO’nun yazılım kalite geliştirme komitelerinde çalışmaktadır. Kendisine [desharnaisjm@gmail.com](mailto:desharnaisjm@gmail.com) adresinden veya <http://profs.logti.etsmtl.ca/jmdeshar/Accueil/index.html> adlı web sitesinden ulaşılabilmektedir.

#### 7.4 Ölçümün Planlanması

Bu bölümde, Sonuç 1’de elde edilen bakım yapılabilirlik altkarakteristiklerine göre ölçüm setinin oluşturulması ve hangi ölçümlerin ne şekilde yapılacağı planlanmıştır. Yöntem olarak, genel ölçüm kümesinden özel ölçüm kümesine doğru gidilerek ölçüm setinin oluşturulması benimsenmiş ve elde edilen sonuçlar bu bölümün alt bölümlerinde adım adım açıklanmıştır. Bu doğrultuda 1 nolu sonuçtan elde edilen altkarakteristiklerden başlayarak, LM’e özel ölçüm kümesinin elde edilmesine, daha sonra da bu kümenin ölçülmesinde kullanılacak temel ve türetilmiş ölçülerin belirlenmesine çalışılmıştır.

Ölçüm setinin oluşturulması için LM’in yazılım müdürü, web ve mainframe proje konfigürasyon sorumluları, bakım yöneticileri ve bakımcılar ile görüşmeler yapılmıştır. Yapılan görüşmelerin sonucunda, LM tarafından seçilen bakım yapılabilirlik altkarakteristiklerine yönelik ölçüm setinin oluşturulması için altta sunulan çalışmaların yapılmasına karar verilmiştir:

1. LM’in sahip olduğu web ve mainframe yazılım ortamları açısından, bakım yapılabilirlik altkarakteristiklerine ait hangi verilerin sağlanacağını araştırılması.
2. Tezin 2. bölümünde açıklandığı gibi ISO 9126-1 standartına, ISO 9126-2 ve 3 teknik raporlarına uygun olacak şekilde hangi temel ölçülerin elde edilebileceği ve bu temel ölçülerden hangi ölçülerin türetileceğinin araştırılması.



#### 7.4.1 LM'in yazılım ortamlarından elde edilen ölçüm seti ve bilgi gereksinimleri

LM'in sahip olduğu yazılım ortamları açısından, elverişlilik üst kalite faktörü de göz önünde bulundurularak (Ara Sonuç 1), bakım yapılabilirlik altkarakteristiklerinden hangilerinin LM tarafından ölçülebileceği araştırılmış ve LM yetkilileri ile yapılan çalışmalar sonucunda elde edilen sonuçlar Çizelge 7.7'de sunulmuştur. Bu araştırma için, özellikleri Ek-D'de sunulan ISO 9126 standartına göre değiştirilebilirlik, çözümlenebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler ve bu metriklerin elde edilmesinde kullanılan hesaplamalar dikkate alınmıştır. Araştırma sonucuna göre, LM'in bakım ortamından; yazılımdaki değişikliklerin denetlenmesi, değişikliklerin gerçekleştirilme süresi ve hatayı çözümleyebilme yeteneğine ilişkin veriler elde edilebilmektedir.

**Çizelge 7.7 ISO 9126 standartına göre  
LM'in bakım ortamından elde edilen ölçüm seti.**

Kod	Altkarakteristik	No	Bilgi Gereksinimi	Web ortamı	Mainframe
<b>B2</b>	<b>Değiştirilebilirlik</b>	M8	Yazılımdaki değişiklikleri denetleme yeteneği	✓	✓
		M9	Parametreleri belirlenmiş değiştirilebilirlik	-	-
		M10	Değiştirme zorluğu	-	-
		M11	Değişim döngüsünün verimliliği	-	-
		M12	Değişikliğin gerçekleştirilme süresi	✓	-
		M13	Değişikliklerin kaydedilebilmesi	-	-
<b>B1</b>	<b>Çözümlenebilirlik</b>	M1	Teşhisin desteklenmesi	-	-
		M2	Denetlenebilirlik	-	-
		M3	Hata çözümleme etkinliği	-	-
		M4	Hataıı çözümleyebilme yeteneđi	✓	-
		M5	Durumun izlenebilmesi	-	-
		M6	Faaliyet kaydetme	-	-
		M7	Tanıılama fonksiyonu hazırlama	-	-
<b>B4</b>	<b>Sınanabilirlik</b>	M18	Tekrar sınanabilirliđin verimliliđi	-	-
		M19	Yerleşik sınaıa fonksiyonlarının kullanılması	-	-
		M20	Sınamayı yeniden başlatabilme	-	-
		M21	Önceden tanımlanmış sınaıa fonksiyonlarınıı tamlığı	-	-
		M22	Sınanabilirliđin özerkliđi	-	-
		M23	Sınanabilirlik gelişiminin izlenmesi	-	-

Çizelge 7.7'nin yorumlanması sonucunda alttaki sonuçlara ulaşılmıştır.

### **Sonuç 2**

Mainframe ortamında yazılım bakımına yönelik ölçüm yapılmamakta ve yapılmasının da LM'e bir faydası bulunmamaktadır.

### **Sonuç 3**

Yazılım bakım sürecinde, bakımın sınanmasına yönelik veriler elde edilememektedir.

#### **7.4.2 Temel ölçüler**

Bu bölümde, 1, 2 ve 3 nolu sonuçların yorumlanması, Bölüm 2'de temelleri verilen, Şekil 2.3 ve Çizelge 2.1'de detayları anlatılan ISO 15939 standartına göre, Ek-E3 ve Ek-E4'te sunulan veri alanları kullanılarak, LM yazılım ürününe ait temel ölçüler belirlenmiştir. Temel ölçülerin belirlenmesinde, Bölüm 7.1'de sunulan LM'in yazılım ortamı belirleyici olmuştur.

Temel ölçülere ilişkin ölçümler, yazılımın devreye alınmasını takiben, Ocak 2008 ile Temmuz 2009 tarihleri arasındaki müşteriler tarafından talep edilen bakım isteklerini içermektedir. Bunun nedeni, Ocak 2008 tarihinden önceki verilerin, yazılımın geliştirilmesi safhasına ait olması ve web ortamında geliştirilmesi tamamlanan modüllerin bu tarihten itibaren devreye alınmasıdır. Benzer mantıkla, Ocak 2008 tarihinden sonraki geliştirme çabalarına yönelik bakım verileri de dikkate alınmamıştır.

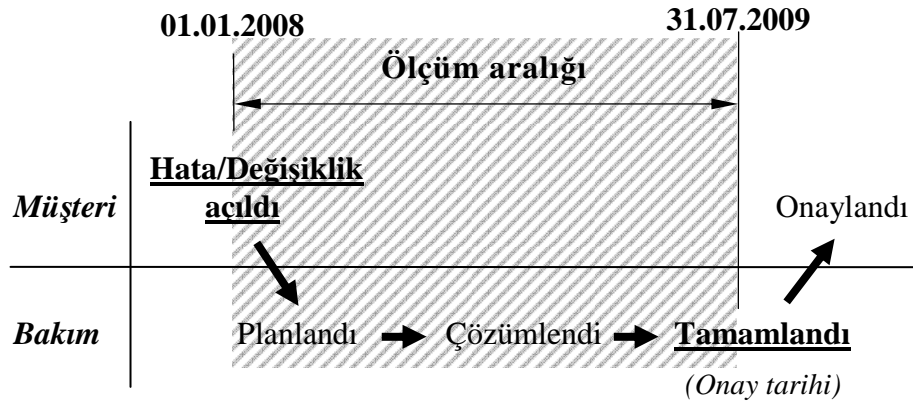
Bakıma konu olan olaylar, LM tarafından iki temel alanda gerçekleştirilmektedir. Bunlardan ilki; müşteri tarafından bildirilen hatalardır, ikincisi ise yine müşteri tarafından yazılımda yapılması istenilen değişikliklerdir. Bakıma konu olan olaylar aşağıda ayrıntılı olarak anlatılmaktadır:

1. **Hatalar:** Hata (*failure*); bir sistemin kendisinden beklenen işlevi yerine getirememesi durumudur (IEEE Std. 610.12:1990) (IEEE Std 982.1:1988). Elde edilen hata verileri, Ek-E2'deki verilerde ve Ek-E4'teki veri alanlarında sunulduğu gibi, müşteri tarafından hatanın bildirilmesi esasına göre Ocak 2008 ile Temmuz 2009 tarihleri arasındaki verilerden elde edilmektedir. LM tarafından hatalara yönelik yapılan bakım çalışmaları, Bölüm 5'te detayları verilen düzeltici bakım faaliyetleri sınıfına girmektedir.
2. **Değişiklik istekleri:** Değişiklik istekleri, müşteri tarafından yazılıma ek işlevler kazandırılmasına yönelik çabaları içermektedir. Bu çabalar, yasalar ve mevzuatta meydana gelen değişiklikler ve kullanıcı isteklerini yerine getirmek için gereken ilave yazılım uyarlamaları olarak tanımlanabilir. LM tarafından değişiklik isteklerine yönelik yapılan bakım çalışmaları, Bölüm 5'te detayları verilen iyileştirici bakım faaliyetleri sınıfına girmektedir.

LM bakım süreci incelendiğinde, web ortamında iki temel büyüklüğü ölçmenin olası olduğu anlaşılmıştır. Bunlar;

- Bakım sürecini oluşturan olaylar arasındaki süre,
- Bakıma konu olan olayların sayısıdır.

Bakıma yönelik sürelerin hesaplanmasında, Şekil 7.4'ten basitleştirilerek türetilen ve Şekil 7.13'de sunulan, bakım operasyonlarına ait süreler dikkate alınmıştır. Bir bakım kaydı, MS TFS üzerinde müşteriden gelen hata veya değişiklik isteği ile açılmakta ve yine müşteri tarafından kapatılmaktadır. Ancak tamamlanmış bakımların, müşteri tarafından kesin onayının geciktirilmesi veya önemsenmeyerek kapanma onayının yapılmaması nedeniyle, sürece ait doğru değerlerin elde edilemediği tespit edilmiştir. Bundan dolayı, müşteri tarafından bakımın onaylanma zamanı yerine, bakımın tamamlanma zamanı (onay tarihi) dikkate alınmış ve dolayısı ile bakım sürecine yönelik hesaplamalarda, bakımın açılması ile tamamlanma zamanı arasındaki süre kullanılmıştır.



**Şekil 7.13 LM'in web ortamına ait ölçüm aralığı.**

Bakım verilerinden sürenin hesaplanmasına yönelik işlemlerde, LM'in resmi çalışma saatleri temel alınarak hesaplamalar yapılmıştır. Bu durum, tatil ve çalışma saatleri dışındaki sürelerin bakım verilerinden ayrıştırılarak normalleştirilmesi anlamına gelmektedir. Ayrıştırma işlemi ve sürelerin hesaplanması, MS Office Project 2007 yazılımı kullanılarak gerçekleştirilmiştir.

Sürelerin hesaplanmasında, bakımıcının, bir bakım olayını sonuçlandırmak için, o olay üzerinde aralıksız çalıştığı ve bakım süresince de başka bir işle uğraşmadığı LM tarafından varsayılmaktadır. Bunun nedeni, bakıma yönelik sıcak çalışmanın herhangi bir nedenle kesintiye uğraması durumunda, duraklama süresine yönelik ölçümler yapılamamasıdır. Bu olumsuzluktan dolayı, bakım süreçleri arasındaki zaman aralıkları doğru olarak ölçülememektedir. Ancak bu durum, yine de, LM'in bakım süreleri hakkında genel bir fikir edinmesine engel teşkil etmemektedir. Bu doğrultuda, bakım sürelerinin ölçülmesine ilişkin alttaki sonuca ulaşılmıştır.

#### **Sonuç 4**

LM tarafından bakım sürelerinin ölçülmesinde,  
ölçme sorunu bulunmaktadır.  
Her şeye rağmen bu durum, LM'e bilgi sağlamaktadır.

Yukarıda sunulan sonuca ve Bölüm 2.1'de verilen ölçmenin sağladığı fayda ve sorunlar dikkate alındığında, süreye yönelik eldeki bilgilerin bir temel ölçü olarak kullanılmamasına karar verilmiştir.

ISO 9126-1 standartına ve standartın teknik raporlarına, Çizelge 2.3, 2.4, 7.7 ve Ek-C de sunulan değiştirilebilirlik ve çözümlenebilirlik altkarakteristiklerine, LM'in bilgi gereksimine uygun olarak, LM tarafından sağlanan temel ölçüler, Çizelge 7.8'de sunulmuştur. Ek-F'de, Çizelge 7.8'de sunulan temel ölçüler detaylı olarak tanımlanmaktadır.

**Çizelge 7.8 LM'in bakım ortamından elde edilen temel ölçüler.**

Ölçü		Birim	Dış/İç Ölçü
<b>T1</b>	Hata düzeltme süresi	Gün	Dış
<b>T2</b>	Değişiklik süresi	Gün	Dış
<b>T3</b>	Hata sayısı	Hata	Dış
<b>T4</b>	Değişiklik sayısı	Değişiklik	Dış
<b>T5</b>	Düzeltilen hata sayısı	Hata	Dış
<b>T6</b>	Yapılan değişiklik sayısı	Değişiklik	Dış
	Fonksiyonların sayısı	Fonksiyon	İç
	Log edilmesi gereken öge sayısı	Öge	İç
	Tanımlama için gerekli fonksiyon sayısı	Fonksiyon	İç

Çizelge 7.8'deki temel ölçülerin elde edilebilirlikleri açısından yorumlanması ile alttaki sonuca ulaşılmıştır.

### **Sonuç 5**

LM tarafından, yazılımın iç kalite özellikleri ölçülmemektedir.

### 7.4.3 Türetilmiş ölçüler ve yorumlanması

Bu bölümde, 7.4.2. bölümde tanımlanan temel ölçüler kullanılarak ve detayları Ek-D'de verilen LM'in bilgi gereksinimini karşılayan türetilmiş ölçülerin tespiti ve bu verilerin nasıl yorumlanacağı sunulmaktadır.

Bu noktada LM'in tespit edilen bilgi gereksinimlerini sağlayan temel ölçüler, elde edilebilen ve fakat ISO 9126-1 standartının teknik raporlarına yer almayan ilavelerle güncelleştirilerek, LM'in bilgi gereksinimini daha iyi karşılamasının mümkün olup olmadığı araştırılmıştır. Bunun sonucunda;

- M8 bilgi gereksinimi, bakımı denetleme yeteneği olarak incelenmiştir.
- M12 nolu bilgi gereksiniminin daha detaylı olarak üç alt kısımda değerlendirilmesi halinde daha fazla bilginin sağlanacağı anlaşılmıştır.
- M4 hatayı çözümlenebilirlik yeteneği ise M4a ve M4b olarak bakım ve değişikliklerin çözümlenebilirlik yeteneği olarak iki alt kısımda incelenmiştir.

Dolayısı ile 1 - 5 nolu sonuçlara göre, gereksinimlerin yeniden düzenlenmesi ile elde edilen ölçüm seti Çizelge 7.9'da sunulmaktadır.

**Çizelge 7.9 LM'in bakım ortamından elde edilen ölçüm seti.**

Kod	Alt karakteristik	No	Bilgi Gereksinimi	Web ortamı
<b>B2</b>	<b>Değiştirilebilirlik</b>	M8a	Bakımı denetleme yeteneği	✓
		M12a	Hatanın giderilme süresi	✓
		M12b	Değişikliğin gerçekleştirilme süresi	✓
		M12c	Bakımın tamamlanma süresi	✓
<b>B1</b>	<b>Çözümlenebilirlik</b>	M4a	Hatayı çözümlenebilirlik yeteneği	✓
		M4b	Değişikliği çözümlenebilirlik yeteneği	✓

Çizelge 7.9'da sunulan türetilmiş ölçüler, yukarıdaki esaslara göre altta detaylı olarak tanımlanmıştır.

### **Türetilmiş ölçü 1: Bakımı denetleme yeteneği (M8a)**

Veri toplama:

- Temel ölçü 3 : Hata sayısı (T3)
- Temel ölçü 4 : Değişiklik sayısı (T4)
- Temel ölçü 5 : Düzeltilen hata sayısı (T5)
- Temel ölçü 6 : Yapılan değişiklik sayısı (T6)

Veri hazırlama:

- Türetilmiş ölçü :  $(T5+T6)/(T3+T4)$

Veri analizi:

- Karakteristik : Bakım yapılabilirlik
- Alt karakteristik : Değiştirilebilirlik (B2)
- Kalite grubu : Dış kalite ölçümü
- Elde edilen değer, yapılan bakım (hata+değişiklik) ile müşteriden gelen bakım istekleri arasındaki ilişkiyi gösterir.
- Yorumlama : Değerin 1'e yakınlığı daha iyidir

### **Türetilmiş ölçü 2: Hatanın giderilme süresi (M12a)**

Veri toplama:

- Temel ölçü 1 : Hata düzeltme süresi (T1)
- Temel ölçü 5 : Düzeltilen hata sayısı (T5)

Veri hazırlama:

- Türetilmiş ölçü :  $T1/T5$

Veri analizi:

- Karakteristik : Bakım yapılabilirlik
- Alt karakteristik : Değiştirilebilirlik (B2)
- Kalite grubu : Dış kalite ölçümü
- Elde edilen değer, ortalama hata giderme süresini vermektedir.
- Yorumlama : Değerin küçük olması iyidir

### **Türetilmiş ölçü 3: Değişikliğin gerçekleştirilme süresi (M12b)**

Veri toplama:

- Temel ölçü 2 : Değişiklik süresi (T2)
- Temel ölçü 6 : Yapılan değişiklik sayısı (T6)

Veri hazırlama:

- Türetilmiş ölçü :  $T2/T6$

Veri analizi:

- Karakteristik : Bakım yapılabilirlik
- Alt karakteristik : Değiştirilebilirlik (B2)
- Kalite grubu : Dış kalite ölçümü
- Elde edilen değer, ortalama değişiklik gerçekleştirme süresini vermektedir.
- Yorumlama : Değerin küçük olması iyidir

### **Türetilmiş ölçü 4: Bakımın tamamlanma süresi (M12c)**

Veri toplama:

- Temel ölçü 1 : Hata düzeltme süresi (T1)
- Temel ölçü 2 : Değişiklik süresi (T2)
- Temel ölçü 5 : Düzeltilen hata sayısı (T5)
- Temel ölçü 6 : Yapılan değişiklik sayısı (T6)

Veri hazırlama:

- Türetilmiş ölçü :  $(T1+T2)/(T5+T6)$

Veri analizi:

- Karakteristik : Bakım yapılabilirlik
- Alt karakteristik : Değiştirilebilirlik (B2)
- Kalite grubu : Dış kalite ölçümü
- Elde edilen değer, ortalama bakım tamamlanma süresini vermektedir.
- Yorumlama : Değerin küçük olması iyidir



### **Türetilmiş ölçü 5: Hatayı çözümleyebilme yeteneği (M4a)**

Veri toplama:

- Temel ölçü 3 : Hata sayısı (T3)
- Temel ölçü 5 : Düzeltilen hata sayısı (T5)

Veri hazırlama:

- Türetilmiş ölçü :  $1-(T3-T5)/T3$

Veri analizi:

- Karakteristik : Bakım yapılabilirlik
- Alt karakteristik : Çözümenebilirlik (B1)
- Kalite grubu : Dış kalite ölçümü
- Elde edilen değer, hataların çözümenebilme yeteneğini vermektedir.
- Yorumlama : Değerin 1'e yakınlığı iyidir

### **Türetilmiş ölçü 6: Değişikliği çözümleyebilme yeteneği (M4b)**

Veri toplama:

- Temel ölçü 4 : Değişiklik sayısı (T4)
- Temel ölçü 6 : Yapılan değişiklik sayısı (T6)

Veri hazırlama:

- Türetilmiş ölçü :  $1-(T4-T6)/T4$

Veri analizi:

- Karakteristik : Bakım yapılabilirlik
- Alt karakteristik : Çözümenebilirlik (B1)
- Kalite grubu : Dış kalite ölçümü
- Elde edilen değer, değişikliklerin çözümenebilme yeteneğini vermektedir.
- Yorumlama : Değerin 1'e yakınlığı iyidir

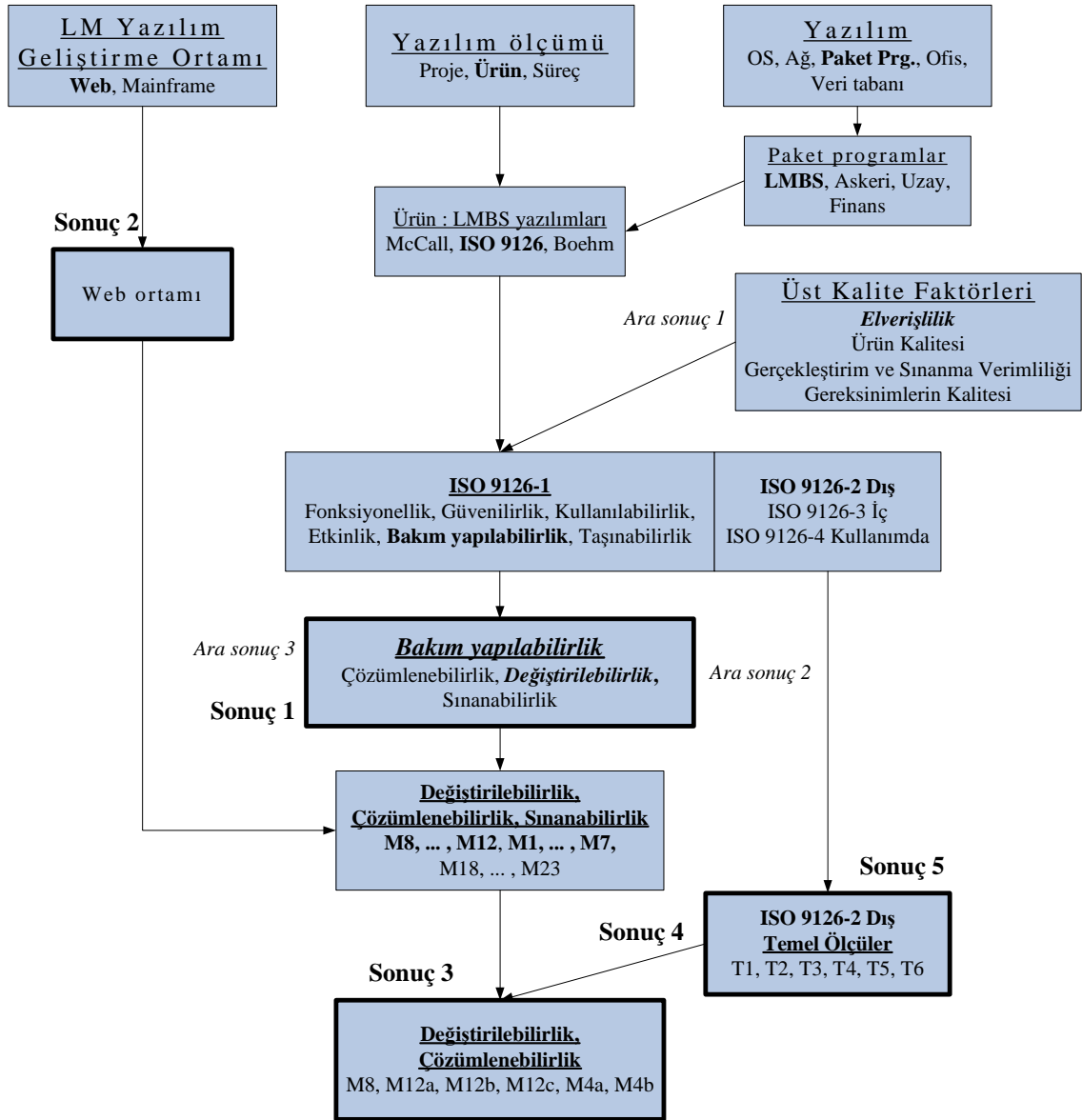
#### 7.4.4 Ölçüm setinin oluşturulması sürecinde elde edilen diğer gözlem sonuçları

Ölçüm setinin oluşturulması sürecinde, yukarıdaki sonuçları destekleyen gözlemler altta sunulmaktadır.

1. LM, yazılım geliştirme sürecinde sınama işlemlerini yazılım mühendisliği ilkelerine göre yapmasına rağmen, bakımın sınanması ağırlıklı olarak müşteri tarafından yapılmakta ve LM bakımın sınanmasına yönelik veri tutmamaktadır.
2. LM tarafından, yazılımın karmaşıklığına ve satır sayısına ilişkin özelliklerine ait veriler mevcut süreçlerden elde edilememektedir. LM, web ortamındaki yazılımının geliştirilmesinin erken safhalarında, karmaşıklık ve kod satır sayısının elde edilmesine yönelik çalışmaları yapmasına rağmen, bunun çok fazla çaba gerektirmesinden dolayı bu konudaki çalışmayı sonlandırmıştır. Bu durum yazılım bakımı hakkındaki bazı kritik ölçülerin elde edilememesinin de başlıca nedenidir.
3. Bazı özelliklerin ölçülmesi mevcut süreçler ile olası değildir ve süreçler de değiştirilmek istenmemektedir (örneğin sınamaya yönelik süreçler).
4. Bazı ölçüler LM'in bilgi ihtiyacını karşılamamakta, bu nedenle ölçülmek istenmemektedir (örneğin yazılım parametrelerinin değiştirilebilirliği).
5. LM için en önemli bakım yapılabilirlik karakteristiği olan değiştirilebilirlik için diğer karakteristiklere göre daha fazla veri elde edilmektedir.
6. Web ortamına ait daha fazla verinin elde edilmesinin nedeni, LM'in Web ortamını desteklemesi ve kullanılan Microsoft TFS yazılımı sayesinde verilerin elde edilmesindeki kolaylıktır.
7. Mainframe ortamı veri toplamaya uygun operasyonel ve yazılımsal özellikler göstermemektedir. Mainframe üzerindeki yazılımlarda hataların nedenlerini ortaya çıkarmak için kayıt tutulmamakta ve LM mainframe ortamına artık daha fazla yatırım yapmamaktadır.

### 7.4.5 Planlama sürecinin özeti

Planlama sürecinde, genelden özele doğru sonuç çıkarma yöntemi kullanılarak ölçüler bulunmuş ve sonuca ulaşılmıştır. Şekil 7.14’de, türetilmiş ölçülerin elde edilmesine kadar olan tercihler, sınırlamalar ve gereksinimle ilgili olmayan karakteristiklerin/altkarakteristiklerin ortadan kaldırılmasına yönelik çalışmalar bir bütün olarak gösterilmektedir.



Şekil 7.14 Planlama sürecinin panoramik gösterimi.

## 8. ÖLÇMENİN GERÇEKLEŞTİRİLMESİ

Bu bölümde, 6. Bölümde belirtilen araştırma yöntemine uygun şekilde, ölçmenin gerçekleştirilmesi sürecine ilişkin uygulamalar ele alınmıştır.

### 8.1 Verilerin Toplanması

Bu bölümde; bilgi gereksinimi tespit edilmiş, temel ölçüleri belirlenmiş ve planlanması tamamlanmış olan ölçüm yapısına ilişkin temel ölçülerin elde edilmesi işlemleri anlatılmaktadır.

LM müşterilerinin değişiklik istekleri ve hata kayıtları; web ortamı için MS TFS yazılımı kullanılarak elde edilmiş ve değişiklik isteklerine ait veri alanları Ek-E3'de ve hata kayıtları ise Ek-E4'de sunulmuştur. Elde edilen verilerin raporlanmasında MS Office Excel ve MS Office Project 2007 yazılımı kullanılmış olup, rapor sonucu Ek-E5'da sunulmaktadır.

LM ile ölçüm sonuçlarının paylaşılması ve sonrasında yapılan değerlendirme sonucunda (ISO 15939 standardı geri besleme süreci), Ek-E5'de sunulan temel ölçülerin aylık bilgileri içermesi sayesinde, Ocak 2009 ile Temmuz 2009 arasındaki ölçüm sonuçlarının da ayrı olarak yorumlanmasına karar verilmiştir. Temel ölçülere ilişkin Ek-E5'de detayları verilen ölçüm sonuçları, **01.01.2008-31.07.2009** ve **01.01.2009-31.07.2009** tarihleri arasında olmak üzere Çizelge 8.1'de sunulmaktadır.

**Çizelge 8.1 Temel ölçülere ait ölçüm sonuçları.**

Ölçü	Birim	Ölçüm Sonucu		
		01.01.08-31.07.09	01.01.09-31.07.09	
T1	Hata düzeltme süresi	Gün	8.409	3.322
T2	Değişiklik süresi	Gün	7.109	949
T3	Hata sayısı	Hata	3.160	1.575
T4	Değişiklik sayısı	Değişiklik	943	162
T5	Düzeltilen hata sayısı	Hata	2.965	1.503
T6	Yapılan değişiklik sayısı	Değişiklik	742	131

## 8.2 Verilerin İşlenmesi

LM'in bilgi gereksinimini karşılamak üzere Çizelge 8.1'de sunulan temel ölçülere ait ölçüm sonuçlarından, türetilmiş ölçüler hesaplanmış ve elde edilen değerler Çizelge 8.2'de sunulmuştur. Elde edilen sonuçların sunulması sırasında, LM'in Ocak 2008 ile Temmuz 2009 ayları arasındaki türetilmiş ölçümler ile Ocak 2009 ile Temmuz 2009 arasındaki ölçümleri, son altı aydaki bakım performansın izlenebilmesi için ayrı ayrı sunulmuştur. Böylece, Çizelge 7.1'de sunulan LM'in son altı aylık yönetim kararlarının, bakıma olan etkileri de incelenmiştir.

**Çizelge 8.2 LM bilgi gereksinimini karşılayan ölçüm sonuçları.**

No	Bilgi Gereksinimi	Türetilmiş ölçü	Sonuç		Yorum <sup>16</sup>
			Ocak'08 Temmuz'09	Ocak'09 Temmuz'09	
M8a	Bakımı denetleme yeteneği	$(T5+T6)/(T3+T4)$	<b>0,90</b>	<b>0,94</b>	Değerin 1'e yakınlığı iyidir
M12a	Hatanın giderilme süresi	T1/T5	<b>2,84 gün/hata</b>	<b>2,21 gün/hata</b>	Değerin küçük olması iyidir
M12b	Değişikliğin gerçekleştirilme süresi	T2/T6	<b>9,58 gün/değ.</b>	<b>7,24 gün/değ.</b>	Değerin küçük olması iyidir
M12c	Bakımın tamamlanma süresi	$(T1+T2)/(T5+T6)$	<b>4,19 gün</b>	<b>2,61 gün</b>	Değerin küçük olması iyidir
M4a	Hatayı çözümlenebilme yeteneği	$1-(T3-T5)/T3$	<b>0,94</b>	<b>0,95</b>	Değerin 1'e yakınlığı iyidir
M4b	Değişikliği çözümlenebilme yeteneği	$1-(T4-T6)/T4$	<b>0,79</b>	<b>0,81</b>	Değerin 1'e yakınlığı iyidir

<sup>16</sup> Ölçüm sonuçlarının yorumlanmasında, ISO 9126 standartının teknik raporlarında belirtilen yorumlama şekli benimsenmiştir.

### 8.3 Verilerin Doğrulanması

LM tarafından toplanan bakım verilerinin MS TFS yazılımı kullanılarak toplanması ve uygun ortamda saklanmasından dolayı, verilerin doğrulanmasına gerek olmadığına karar verilmiştir. Yapılan veri toplama çalışmasında aynı hata/değişiklik istekleri, müşteri tarafından birden fazla açılan hata kayıtları, iptal edilen bakım istekleri değerlendirme dışı bırakılarak ölçüme dahil edilmemiş, temel ölçülerin hesaplanmasında bakım sürecinde belirtilen veri alanlarına ait kısıtlar LM ile birlikte denetlenmiş ve doğrulanmıştır.

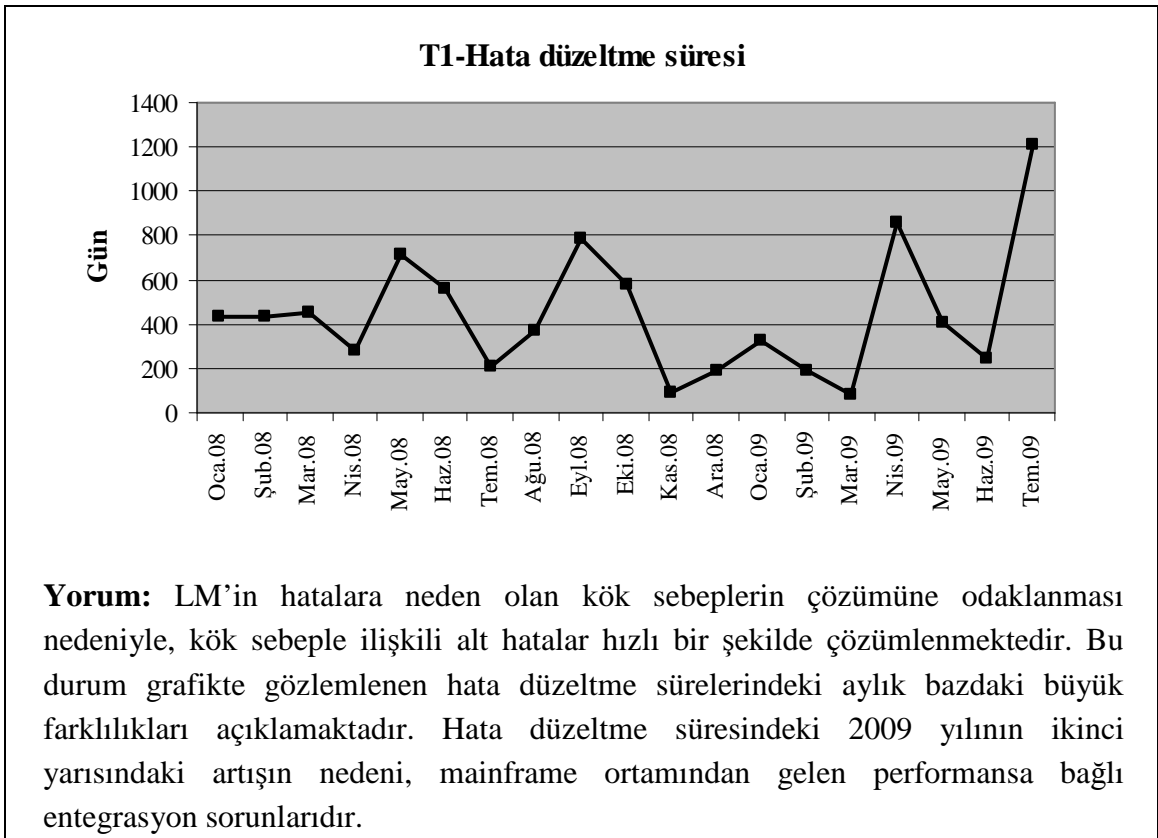
## 9. ÖLÇÜM SONUÇLARININ DEĞERLENDİRİLMESİ

Bu bölümde, temel ve türetilmiş ölçülerin değerlendirilmesinden elde edilen sonuçlar yer almaktadır. Değerlendirme ve doğrulama işlemleri, elde edilen ölçüm sonuçları, LM'in proje ve konfigürasyon yönetimi konusundaki kararları ile (Çizelge 7.1) karşılaştırılarak yapılmıştır. Ölçüm sonuçları ve ölçme sürecinin değerlendirilmesi aşağıda sunulmaktadır.

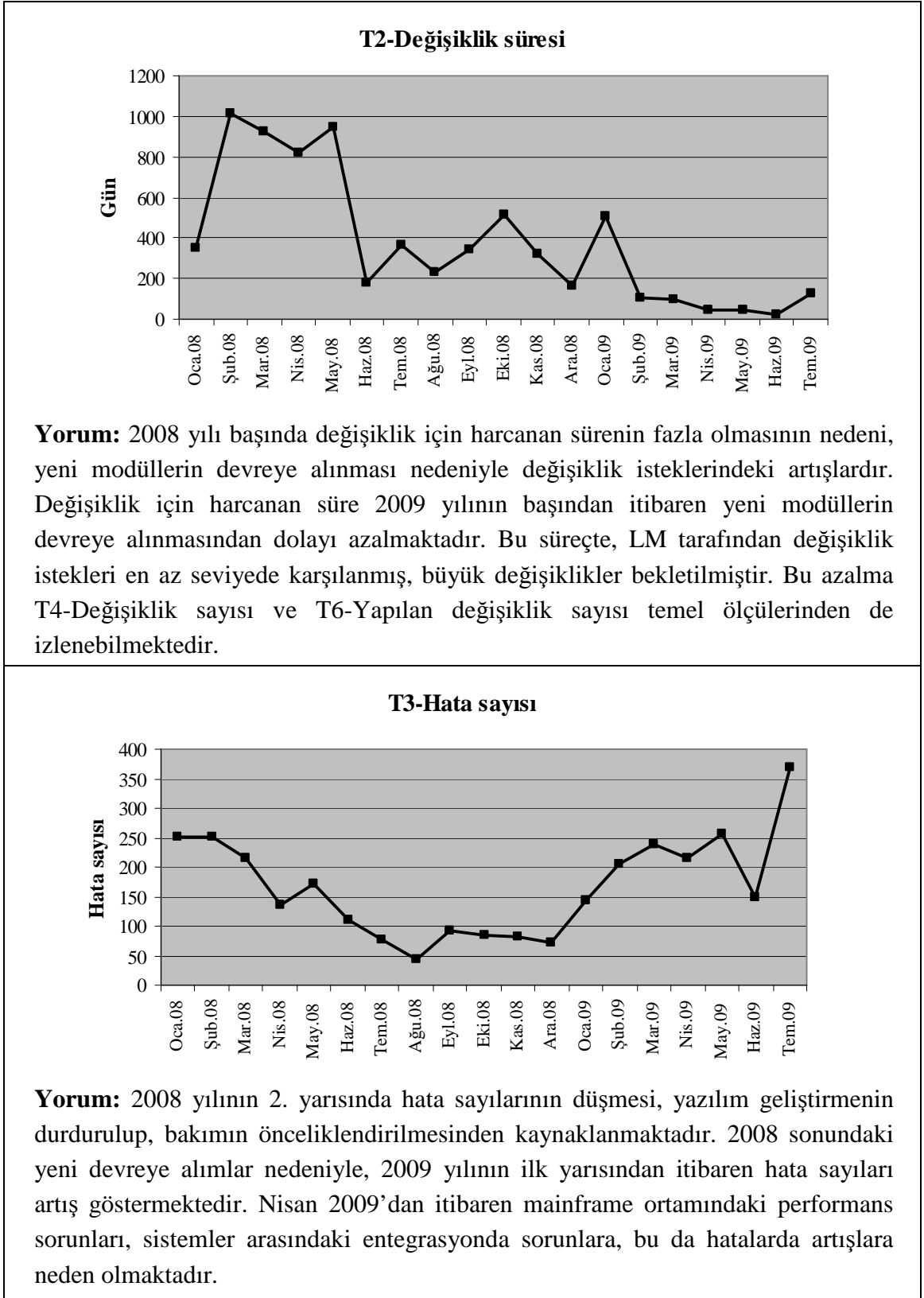
### 9.1 Ölçüm Sonuçlarının Değerlendirilmesi

Çizelge 8.1 ve 8.2'de sunulan temel ve türetilmiş ölçüm değerlerinin, ölçüm süresince değişimlerini izlemek ve yorumlamak için Ek-E5'daki veriler grafiksel olarak ifade edilmiştir. Söz konusu grafikler ve yorumları Çizelge 9.1'de sunulmaktadır.

**Çizelge 9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu.**

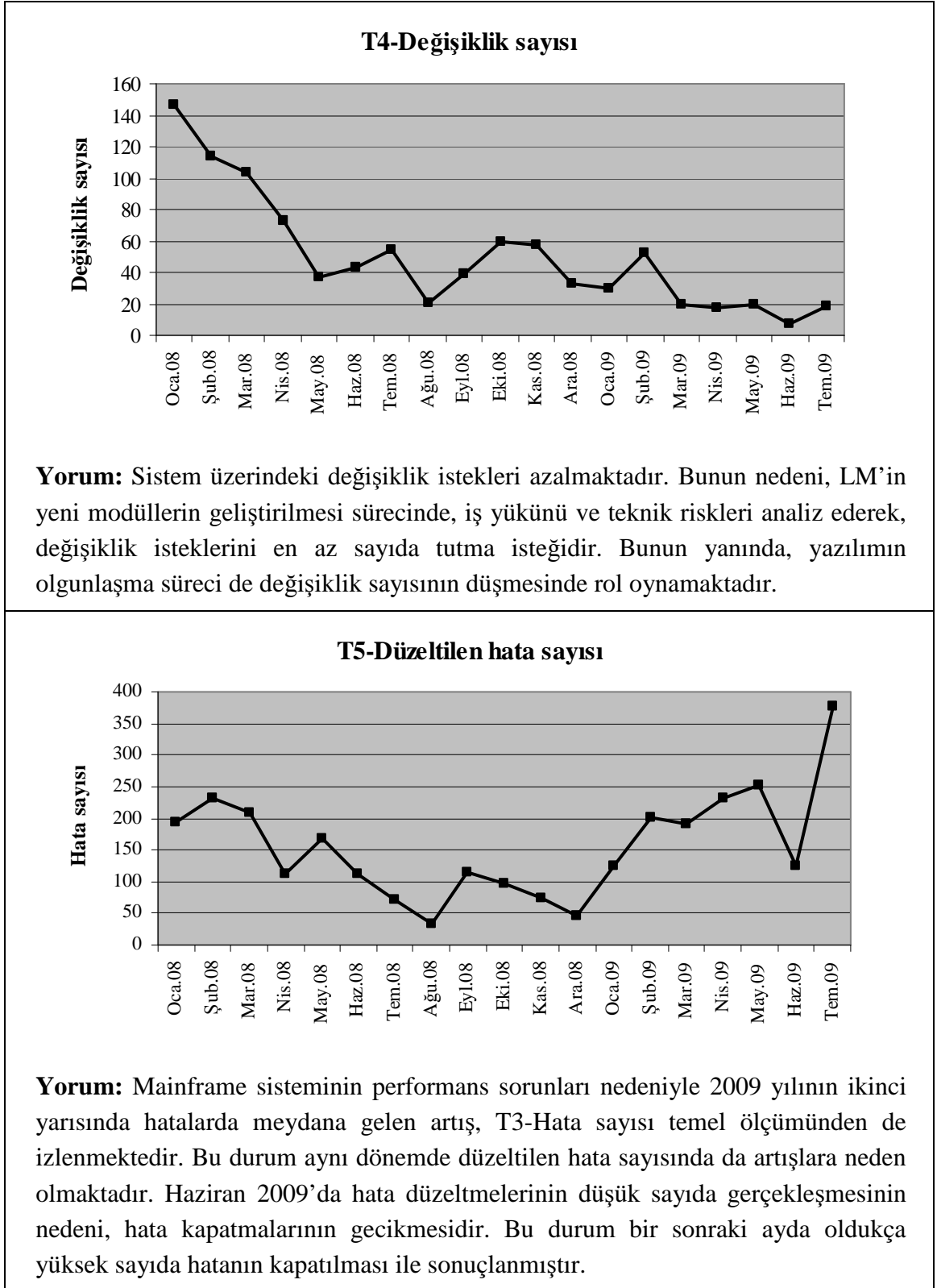


Çizelge 9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu. (Devam)

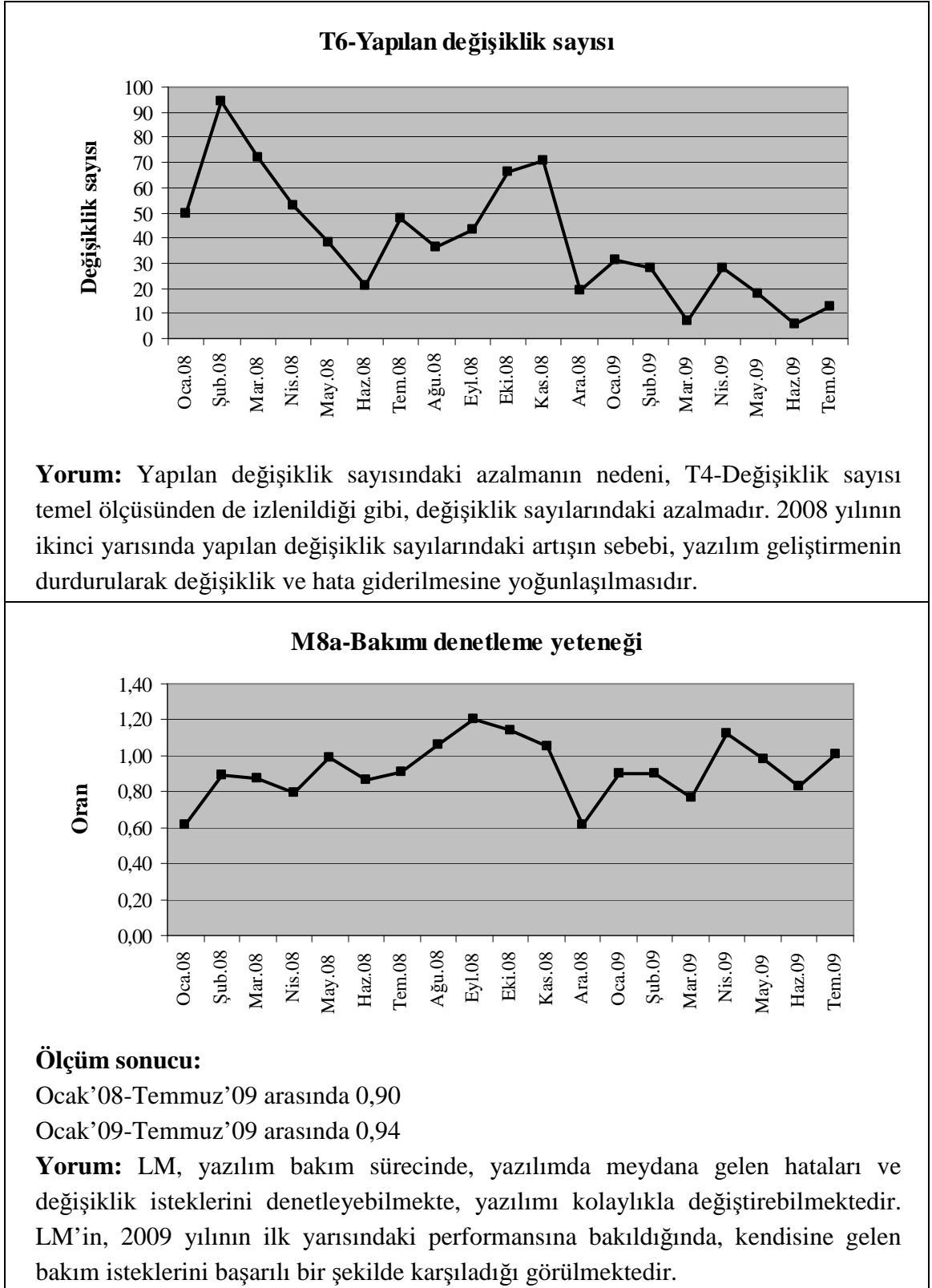




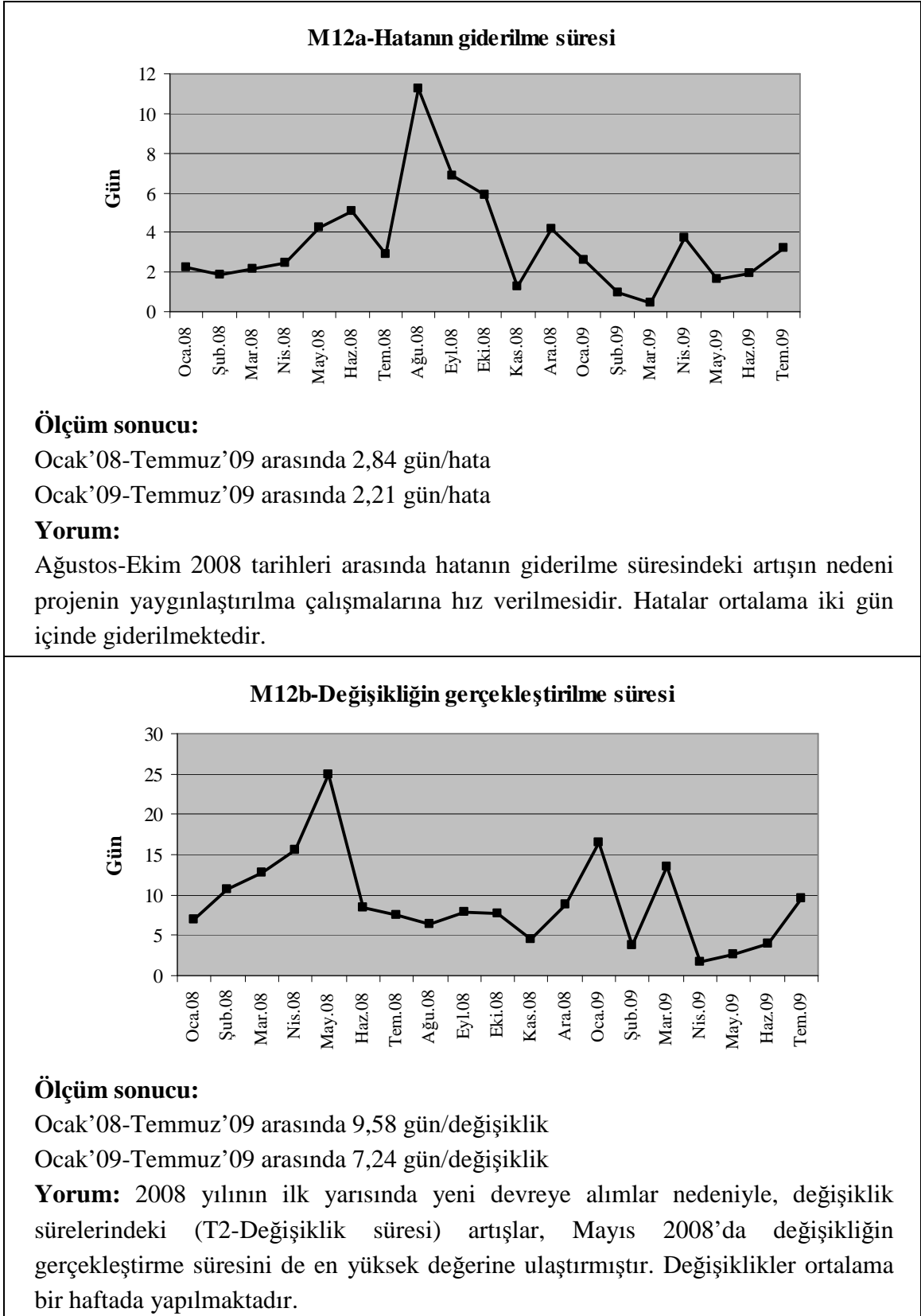
Çizelge 9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu. (Devam)



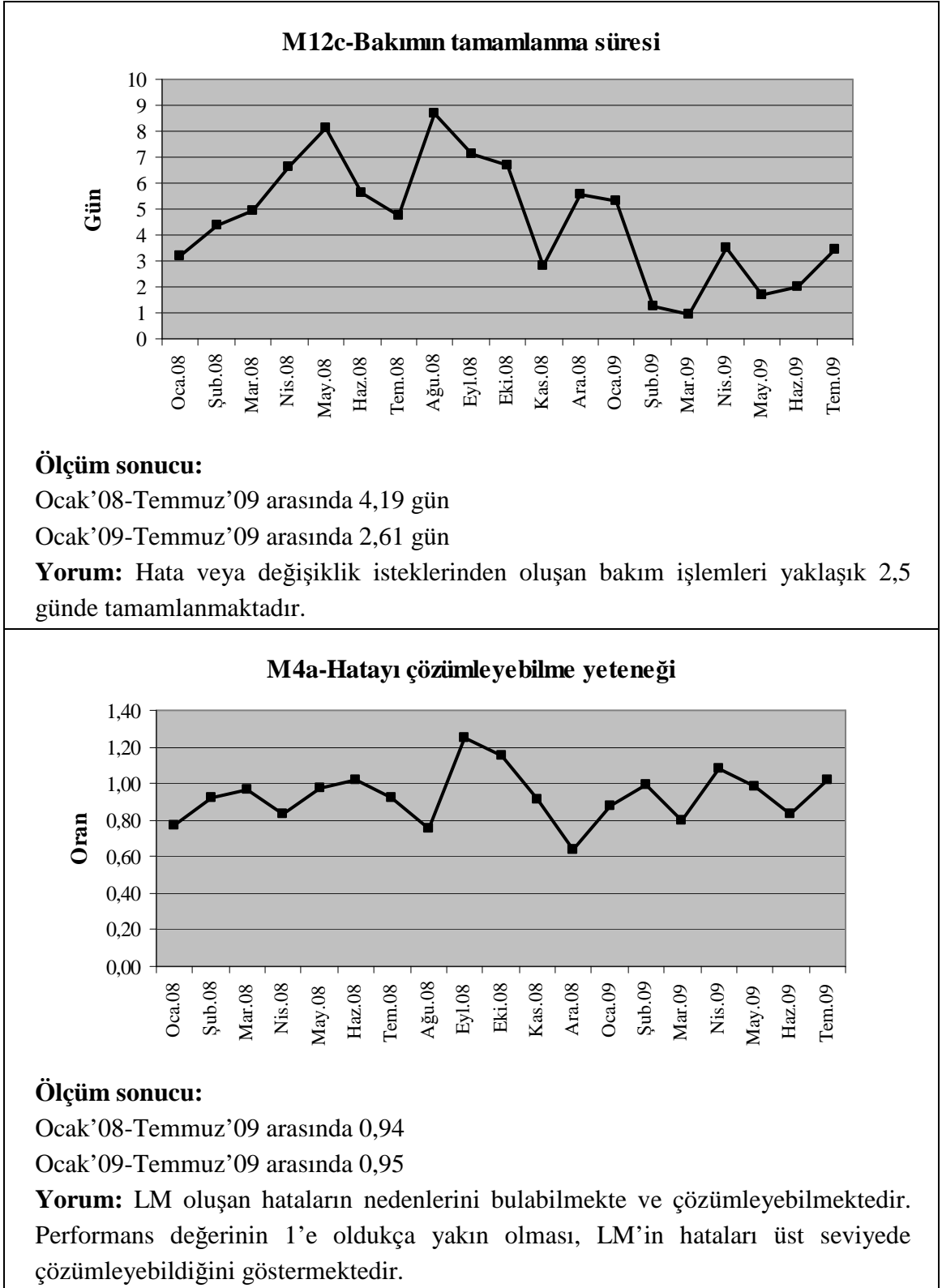
Çizelge 9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu. (Devam)



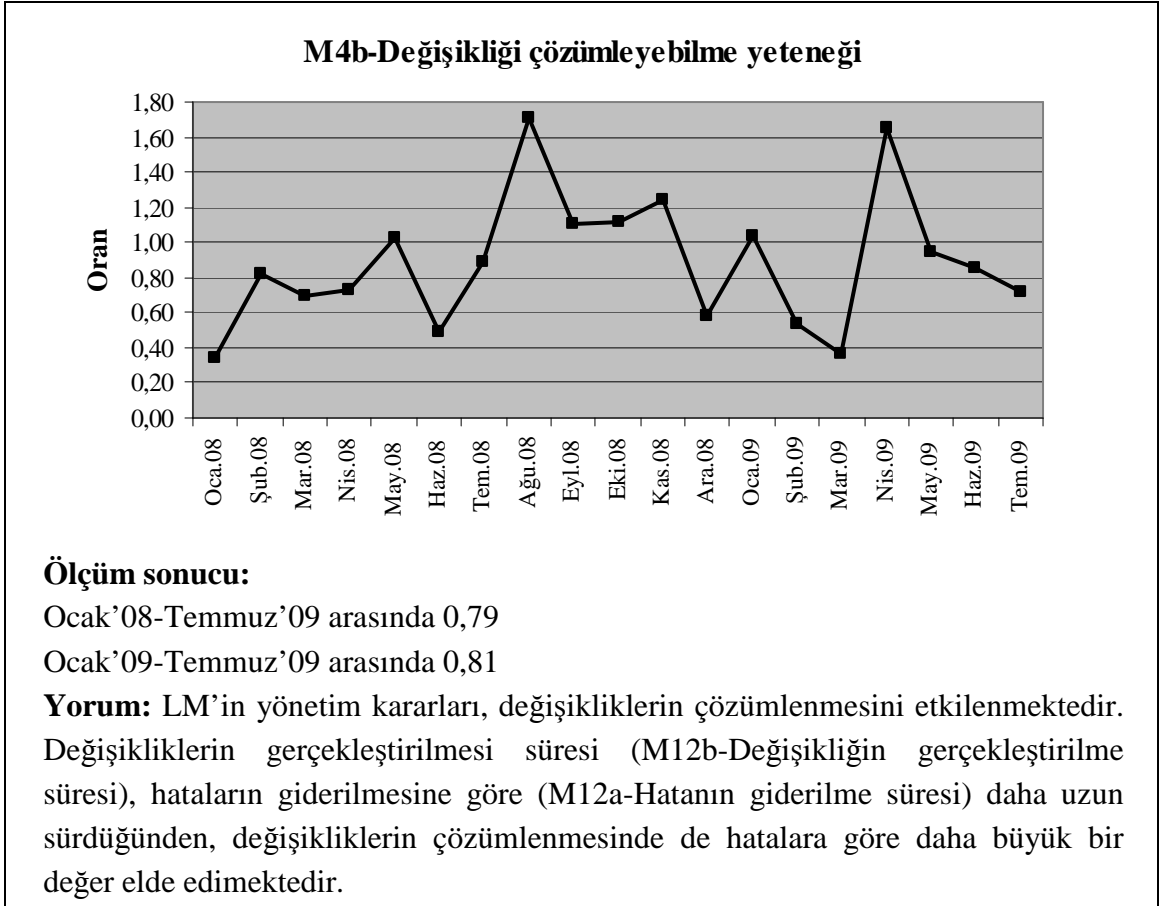
Çizelge 9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu. (Devam)



Çizelge 9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu. (Devam)



**Çizelge 9.1 Temel ve türetilmiş ölçülere ait grafikler ve ölçülerin yorumu. (Devam)**



Sonuç olarak;

- LM yazılım hatalarını yaklaşık 2 gün içinde giderebilmektedir.
- LM bakım gereksinimlerini yaklaşık 7 gün içinde sonuçlandırabilmektedir.
- LM hataları çözümleyebilmekte ve bakım isteklerini başarılı bir şekilde yanıtlamaktadır.
- LM hataların giderilmesine (düzeltici bakım), değişiklik isteklerinin karşılanmasından (iyileştirici bakım) daha fazla önem vermektedir. Bu doğrultuda, LM'in yaptığı bakımın, Bölüm 5.1'de sunulan bakım türleri için harcanan çaba oranlarına ters düştüğü anlaşılmaktadır. Bu durum geliştirilmesi süren bir yazılım için beklenen bir sonuçtur.

## 9.2 Ölçüm Sürecinin Değerlendirilmesi

Ölçüm süreci değerlendirildiğinde;

1. LM, yazılım bakımının ölçülmesi ve değerlendirmesi için yeterli ve nitelikli insan ve makina kaynağı ayırmaktadır.
2. Ölçüm süreci bakım mühendisleri tarafından benimsenmiş ve LM bilgi sistemleri yöneticileri tarafından desteklenmektedir.
3. LM'in yazılım bakım süreçlerini tanımlamış olması, LM bilgi gereksinimlerinin tanımlanmasını ve süreçlerin ölçümde kullanılmasını kolaylaştırmıştır. Elde edilen sonuçların anlamlı olmasına katkı sağlamıştır.
4. LM'in yazılım geliştirme sürecini ve/veya bakım ve değişiklik isteklerini gerçekleştirme koşullarını durdurması, yavaşlatması veya önceliklendirmesi, ölçüm sürecinin ve sonuçlarının doğrulanmasında yarar sağlamıştır.
5. Ölçüm sürecinde, MS TFS yazılımının veri toplama sağladığı imkan ve kolaylıklar belirleyici olmuştur. Otomatik bir veri toplama yazılımının olmaması durumunda, verilerin toplanmasının mümkün olmayacağı LM yetkilileri tarafından belirtilmiştir. Dolayısı ile ölçüm sürecinin kalitesinin, veri toplama sürecinin kalitesine çok yakından bağlı olduğu anlaşılmıştır.
6. Bakım sürecinin kaynak planlaması ve maliyet tahmini yapılmasında sayısal büyüklüklerin kullanılması sağlanmıştır. Ölçüm değerleri, müşteri ile yapılacak bakım anlaşmalarının daha iyi ve gerçekçi koşullarda yapılmasının önünü açmıştır.
7. Ölçümlerin sürdürülmesi durumunda, yazılım bakımındaki iyileştirme ve gelişmelerin izlenebileceği LM tarafından bilinmektedir.
8. Değişiklik yönetimi ile ilgili sonuçların sayısal olarak izlenmesi mümkün olmuştur.
9. Özelde bakım süreçlerinin, genelde de projenin tamamını etkileyen belirsizlik ve risklerin LM tarafından algılanmasını kolaylaştırmıştır.
10. Yeterli sayıda verinin olması sonuçların elde edilmesinde yararlı olmuştur.

Sonuç olarak;

- LM, yazılım mühendisliği standartlarını ve modellerini, geliştirme ve bakım süreçlerinde rehber olarak kullanmaktadır.
- LM bakım süreçlerini tanımlamakta ve ölçebilmektedir.

### **9.3 Sonuçların Doğrulanması**

Bölüm 8.2’de sunulan grafiklerde yer alan temel ve türetilmiş ölçülerin, Çizelge 7.1’de sunulan LM’in yazılımın entegre değişiklik yönetimine ait kararlarıyla uyumlu olduğu görülmüştür. Yönetim kararlarının bakım süreç karakteristiklerine, bunun da elde edilen verilere ve dolayısı ile ölçülere yansımış olması beklenen bir sonuçtur. Bu durum ölçüm sonuçlarını doğrulamakta, ölçümün, ölçülen varlığa ait bilgi gereksinimini karşıladığını göstermektedir.

### **9.4 Tezin Hipotezi ve İspatı**

Bu bölümde ortaya konmuş olan sonuçlar; sayfa (ii)’de ve sayfa 1’de tanımlanmış olan bu doktora tezinin hipotezlerini, açıkça ve hiçbir tartışmaya meydan vermeyecek biçimde ispatlar durumdadır.

LM yazılım ürününü ölçmek ve LM yazılım ürünü için arttırılmış kalite kavramını ortaya koymak olarak tekrarlanabilecek olan söz konusu bu hipotezler; sonuçlar bölümünde de görüldüğü gibi bütünüyle ispatlanmış ve bu haliyle tez hedeflerine ulaşmıştır.

## 10. SONUÇ VE ÖNERİLER

Bu tezde; ISO tabanlı modeller kullanarak, lojistik merkez yazılım ürününde yazılım kalite kavramını arttırmak amacıyla yapılan araştırma yöntemi sunulmuş, endüstriyel standartlar lojistik merkez yazılım ürününde uygulanmış, lojistik merkezin bilgi gereksinimlerini karşılayan ölçüler saptanmış ve ölçülmüştür. Elde edilen ölçüm sonuçları değerlendirilmiş ve kullanılan yöntem ve elde edilen sonuçların benzer sistemler için örnek oluşturmasına çalışılmıştır.

Tezde yapılan çalışmalar, günümüz iş ve sosyal hayatında yazılımın artan önemi düşünüldüğünde önem arz etmektedir. Daha iyi ve kaliteli yazılımlara olan gereksinim açıktır ve bu gereksinim artmaktadır. Dolayısı ile bir mühendislik ürünü olan yazılım geliştirme ve bakımını sayısal olarak ölçmek ve değerlendirmek yazılım ürünlerinin daha iyi üretilmesi fikrini desteklemektedir.

ISO standartları bu alandaki çalışmalarını düzenlemekte ve geliştirmektedir. ISO tabanlı standartlar kullanılarak; Çizelge 9.1’de sonuçları sunulan yazılım ürün ölçümlerini, genelden özele doğru sonuç çıkararak, gerçekleştirmenin mümkün olduğu bu çalışmada gösterilmiştir. Bunu başarmanın önündeki tehdit ve engeller aşağıdaki sunulmaktadır:

1. Yazılım kalitesi konusuna yeterli önemin verilmemesi.
2. Yazılım geliştirme ve bakım süreçlerinin olmaması, belirsizliği veya verimsizliği.
3. Belirsizlik ve risk algılamasındaki sorunlar.
4. Rekabet, müşteri, zaman ve maliyet baskıları.
5. İnsan ve makina kaynaklarındaki eksiklikler.
6. Uygulama yöntemine ilişkin belirsizlikler.
7. Veri toplama sürecindeki zorluk ve sorunlar.

Yazılım ürününün ölçülmesi sürecinde yukarıda sıralanan sorunların çözümlenmesi, sonuç ve etkileri itibarıyla yüksek risk içeren sistemlerden sınırlı risk içeren sistemlere doğru uygulamanın yaygınlaşmasını sağlayacaktır. Bu noktada tezin, bunu gerçekleştirmek isteyen işletme ve kişilere yardımcı olacağı umulmaktadır.



Bu tezde uygulanan yöntemler ve elde edilen sonuçlar; nesnel, neden ve sonuç ilişkisini desteklemektedir, güvenilir, tekrarlanabilir, belirsizlikleri ortadan kaldırmaktadır, modellenebilme özelliği göstermektedir ve ölçeklenebilir. Bu açıdan büyük, küçük projelere, gruplara ve firmalara uygulanabilme özelliğine sahiptir. Yazılım proje yönetimini, yazılım kalite çalışmalarını desteklemekte ve bu süreçlerle uyum sağlayabilmektedir. Şekil 1.1’de sunulan paket programlar ürün uzayındaki herhangi bir uygulama yazılımında başarı ile kullanılabilme özelliğine sahiptir.

Tezin amacına uygun olarak ve yukarıda elde edilen sonuçları destekleyecek ve geliştirecek şekilde, gelecekte yapılacak araştırma çalışmalarına yol göstermesi yönündeki öneriler, iki başlık altında aşağıda sunulmuştur:

**Kısa dönemli öneriler:**

- ISO 25021 standartında ve Munson’nun (2002) belirttiği gibi hatalarla birlikte aksaklıkların (*fault*) da sayılabilmesi önem arz etmektedir.
- Yazılım bakım türlerinin ISO 14764 standartına ve yazılım anomalilerinin de ISO 1044 standartına göre sınıflandırılması özellikle yazılım ürününe yönelik ölçme çalışmalarını olumlu etkileyecektir.
- Yazılım bakımına yönelik olarak iyileştirici, düzeltici ve uyarlayıcı bakım türlerine göre istatistik modellerin oluşturulması, bakım kaynaklarının daha verimli kullanılmasını ve yönetilmesini sağlayacaktır.
- Yazılımın ölçülmesinde en iyi uygulamaları sunan çalışmaların arttırılması, bu alanda çalışacak olan araştırmacı ve yararlanıcıları için önem arz etmektedir.

**Uzun dönemli öneriler:**

- Yazılımın diğer fiziksel büyüklüklerden farklı olmasını sağlayan ve de yazılımı ölçmenin önündeki en temel sorun olan; yazılımın sınırlarının belirsizliği, kullanılan dillere göre farklılıklar ve yazılımcının verimliliğini belirlemek gibi çeşitli zorlukları içeren bu alandaki tüm kuramsal çalışmalar önem arz etmektedir.

Yazılım ürün ölçümü ve yazılım bakımı alanında ülkemizde yapılmış olan sınırlı sayıdaki çalışmadan biri olan bu tez, yukarıda tanımlanmış olan alanlarda gerçekleştirilecek yeni araştırmalara ve uygulamalara bir temel oluşturması açısından önem taşımaktadır. Yazılım kalitesi, ölçümü ve bakımı üzerinde yapılacak olan her yeni araştırma ve uygulama bu satırların yazarını mutlu edecektir.

## KAYNAKLAR

- Abran A., Al-Qutaish R.E., Desharnais J.M. and Habra N.**, 2008, “ISO-Based Models to Measure Software Product Quality”, in ‘*Software Quality Measurement – Concepts and Approaches*’, Ed. R. Kumar Jain B., Institute of Chartered Financial Analysts of India, ICFAI University Press, Hyderabad, India, pp. 61-96.
- Abran A., Moore J.W.** (Executive Editors), **Bourque P., Dupuis R.** (Editors), 2004, “Guide to the Software Engineering Body of Knowledge: 2004 Edition”. IEEE Computer Society Press, April 2005.
- Al-Kilidar H., Cox K., Kitchenham B.**, 2005, “The Use and Usefulness of the ISO/IEC 9126 Quality Standard.” in *2005 International Symposium on Empirical Software Engineering (ISESE 2005), 17-18 November 2005, Noosa Heads, Australia*. IEEE, pp. 126–132.
- Anquetil N., Olivera K.M., Dias M.G.B.**, 2006, “Software Maintenance Ontology.” in *Ontologies for Software Engineering and Software Technology*, Ed. Calero C., Ruiz F., Piattini M., Springer Berlin Heidelberg, pp:153-173.
- April A., Abran A.**, 2008, “Software Maintenance Management”, IEEE Computer Society, Wiley & Sons Publication.
- April A., Desharnais, J.M.**, 2005, “Software Maintenance Maturity Model (SM<sup>mm</sup>): A Software Maintenance Process Model”, in *Statistic Canada Conference 2005*.
- April A., Hayes J.H., Abran A., Dumke R.**, 2005, “Software Maintenance Maturity Model (SM<sup>mm</sup>): the software maintenance process model.” in *Journal of Software Maintenance and Evolution: Research and Practice*, Volume 17, Issue 3 (May 2005), Pages: 197 – 223.
- Aydın, A.O.**, 2003, “Yazılım Kalitesi ve Karakteristikleri.” Gazi Üniversitesi Fen Bilimleri Enstitüsü: Ankara.
- Bhushan N., & Rai K.**, 2004, “Strategic Decision Making, Applying the Analytic Hierarchy Process”, Springer-Verlag.
- Boegh J.**, 2008, “A New Standard for Quality Requirements,”*IEEE Software*, vol.25, no.2, pp.57-62, May/April 2008.
- Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M.**, 1978, “Characteristics of Software Quality”, North Holland.
- Boehm, B.W.**, 1987, “Industrial Software Metrics Top 10 List”, *IEEE Software*, 4, 5, September 1987, pp. 84-85.
- Chopra S., Meindl P.**, 2004, “Supply Chain Mangement”, Pearson Prentice Hall.

## KAYNAKLAR (devam)

- Çınar A.**, Temel Ölçümbilim ve Kalibrasyon, <http://www.etik.com.tr/>, bağlantı tarihi: 23.09.2009.
- Desharnais J.M., Abran A., Suryn W.**, 2009, "Attributes within ISO 9126: a Pareto Analysis", Software quality Management 2009, British Computer Society.
- Dominguez J.**, 2009, "The Curious Case of the CHAOS Report 2009", <http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html>, bağlantı tarihi: 24.09.2009.
- Dornier P.P., Ernst R., Fender M., Kouvelis P.**, 1998, "Global Operations and Logistics Text and Cases", John Wiley&Sons.
- Ebert C., Dumke R., Bundschuh M., Schmietendorf A.**, 2005, "Best Practices in Software Measurement", Berlin: Springer-Verlag.
- Emam K., Koru A.G.**, 2008, "A Replicated Survey of IT Software Project Failures," *IEEE Software*. September/October 2008 (vol. 25 no. 5), pp. 84-90.
- Fenton N.E., Pfleeger S.L.**, 1997, "Software Metrics, A Rigorous & Practical Approach", 2 ed., PWS Publishing.
- Grimán A., Chávez L., Pérez M.A., Mendoza L.E., Domínguez K.**, 2006, "Towards a Maintainability Evaluation in Software Architectures", ICEIS (3)2006: 555-558.
- Grupp P., Takang A.A.**, 2003, "Software Maintenance, Concepts and Practice", 2 ed., World Scientific Publishing.
- Handfield R.B, Nichols Jr. E.L.**, 2002, "Supply Chain Redesign: Transforming Supply Chains into Integrated Value Systems", FT Press.
- Hull E., Jackson K., Dick J.**, 2005, "Requirements Engineering", 2 ed., Springer.
- IEEE Std. 610.12-1990. Standard Glossary of Software Engineering Terminology. Institute of Electrical and Electronics Engineers, 1990.
- IEEE Std 982.1-1988. Standard Dictionary of Measures to Produce Reliable Software. Institute of Electrical and Electronics Engineers, 1988.
- IEEE Std. 1012-2004. Standard for Software Verification and Validation. Institute of Electrical and Electronics Engineers, 2004.
- IEEE Std. 1044-1993. Standard Classification for Software Anomalies. Institute of Electrical and Electronics Engineers, 1993.
- IEEE Std. 1061-1988. Standard for a Software Quality Metrics Methodology. Institute of Electrical and Electronics Engineers, 1988.

**KAYNAKLAR (devam)**

- IEEE Std. 1219-1998. Standard for Software Maintenance. Institute of Electrical and Electronics Engineers, 1998.
- ISO/IEC FDIS 9126-1. Software Engineering – Product Quality – Part 1: Quality model. International Organization for Standardization, 1999.
- ISO/IEC DTR 9126-2. Software Engineering – Product Quality - Part 2: External Metrics. International Organization for Standardization, 2001.
- ISO/IEC DTR 9126-3. Software Engineering – Product Quality - Part 3: Internal Metrics. International Organization for Standardization, 2001.
- ISO/IEC TR 9126-4. Software Engineering – Product Quality – Part 4: Quality in Use Metrics. International Organization for Standardization, 2001.
- ISO/IEC FDIS 12207. Systems and software engineering - Software life cycle processes. International Organization for Standardization, 2007.
- ISO/IEC IS 14598-1. Information Technology - Software Product Evaluation – Part 1: General Overview. International Organization for Standardization, 1999.
- ISO/IEC FDIS 14598-2, Software Engineering - Product Evaluation – Part 2: Planning and management. International Organization for Standardization, 1999.
- ISO/IEC FDIS 14598-3. Information Technology - Software product evaluation – Part 3: Process for developers. International Organization for Standardization, 1998.
- ISO/IEC IS 14598-4. Software Engineering- Product Evaluation – Part 4: Process for acquirers. International Organization for Standardization, 1999.
- ISO/IEC IS 14598-5. Information Technology - Software Product Evaluation – Part 5: Process for evaluators. International Organization for Standardization, 1998.
- ISO/IEC FDIS 14598-6. Software Engineering - Product evaluation – Part 6: Documentation of evaluation modules. International Organization for Standardization, 1999.
- ISO/IEC 14764. Software Engineering-Software Life Cycle Processes-Maintenance. International Organization for Standardization, 2006.
- ISO/IEC 15939. Systems and Software engineering – Measurement process. International Organization for Standardization, 2007.
- ISO/IEC 25000. Software engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Guide to SquaRE. International Organization for Standardization, 2005.

## KAYNAKLAR (devam)

- ISO/IEC 25021. (Draft). Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality measure elements. International Organization for Standardization, July 2009.
- Jones C.**, 2008, "Applied Software Measurement, Global Analysis of Productivity and Quality", 3. ed., McGraw-Hill.
- Kafura D.**, 1985, "A Survey of Software Metrics," *Proceedings of the 1985 ACM annual conference on The range of computing : mid-80's perspective: mid-80's perspective*, s: 502 – 506.
- Kan S.H.**, 2003, "Metrics and Models in Software Quality Engineering", 2 ed., Addison Wesley.
- Kazançođlu Y.**, 2008, "Lojistik Yönetimi Sürecinde Tedarikçi Seçimi ve Deđerlendirilmesinin Yöneylem Arařtırması Teknikleri ile Gerçekeřtirilmesi", Ege Üniversitesi Sosyal Bilimler Enstitüsü iřletme doktora tezi.
- Koscianski A., Costa, J.C.B.**, 1999, "Combining analytical hierarchical analysis with ISO/IEC 9126 for acomplete quality evaluation framework," *Software Engineering Standards, 1999. Proceedings. Fourth IEEE International Symposium and Forum on*, pp.218-226.
- Kurtel K., Atay S., Tunçay A.**, 2006, "Information Technologies for Contemporary Logistics Centers," *4<sup>th</sup> International Logistics and Supply Chain Congress*, s: 616-622, 29-30 November, 1 December 2006.
- Kuruüzüm A., Atsan N.**, 2001, "Analitik Hiyerarři Yöntemi ve İřletmecilik Alanındaki Uygulamaları," *Akdeniz İ.İ.B.F Dergisi* (1), s.83-105.
- Lambert D.M., Stock J.R., Ellram L.M.**, 1998, "Fundamentals of Logistics Management", McGraw-Hill.
- Lanza M., Marinescu R.**, 2006, "Object-Oriented Metrics in Practice", Berlin Heidelberg: Springer-Verlag.
- Lientz B.P., Swanson E.B.**, 1980, "Software Maintenance Management", Reading, MA; Addison-Wesley. (Ch.21).
- McCall, J. A., Richards, P. K., Walters, G. F.**, 1977, "Factors in Software Quality", *Nat'l Tech.Information Service*, no. Vol. 1, 2 and 3.
- Munson J.C., Nikora A.P.**, 2002, "Toward A Quantifiable Definition of Software Faults", *Proceedings of the 13 th International Symposium on Software Reliability Engineering (ISSRE'02)*, IEEE.
- Murphy P.R., Wood D.F.**, 2004, "Contemporary Logistics", 8th. ed., Prentice Hall.

**KAYNAKLAR (devam)**

- Musa J.D., Iannino A., Okumoto K.**, 1987, “Software Reliability Measurement, Prediction, Application”, McGraw-Hill.
- Nosek, J.T., Palvia, P.**, 1990, “Software Maintenance Management: Changes in the Last Decade,” *Journal of Software Maintenance*, Vol. 2, pp. 157-174.
- Partovi, F.**, 1994, “Determining What to Benchmark: An Analytic Hierarchy Process Approach” *International Journal of Operations & Production Management*. Vol 14 Issue:6, s.25–39.
- Ross S.**, 2006, “A First Course in Probability”, 7. ed., Pearson Prentice Hall.
- Saaty T.L.**, 1980. “The Analytic Hierarchy Process”, New York: McGraw-Hill.
- Saaty T.L.**, 1999. “Decision Making for Leaders”, 3 ed., RWS Publication.
- Saaty T.L., Vargas L.G.**, 2001, “Models, Methods, Concepts & Application of the Analytic Hierarchy Process”, Kluwer Academic Publication.
- Sarıdoğan M.E.**, 2008, “Yazılım Mühendisliği”, İstanbul: Papatya Yayıncılık.
- Sommerville I.**, 2007, “Software Engineering”, 8 ed., Addison Wesley.
- TSE, 1998, “TS ISO/IEC 9126 Bilgi teknolojisi - Yazılım ürün değerlendirmesi - Kalite özellikleri ve kullanım kılavuzu.” Türk Standardları Enstitüsü, Ankara.
- Winston W.L.**, 2004, “Operations Research”, 4 ed., Thomson International.
- Xenos M.**, 2001, “Usability Perspective in Software Quality,” *Usability Engineering Workshop, Proceedings of the 8th Panhellenic Conference on Informatics with International Participation*, Vol. 2, pp. 523-529.
- Zuse H.**, 1998, “A Framework of Software Measurement”, Walter de Gruyter.

## ÖZGEÇMİŞ

Kaan Kurtel, 1961 yılında İzmir’de doğdu. İzmir Atatürk Lisesi’nden 1978 yılında mezun olduktan sonra, 1982 yılında Dokuz Eylül Üniversitesi’nde mühendislik eğitimi görerek lisans eğitimini, 2005 yılında Ege Üniversitesi Bilgisayar Mühendisliği Bölümünde yüksek lisans eğitimini tamamladı.

1981-1990 yılları arasında çeşitli bilgi işlem firmalarında programcı ve sistem analist olarak çalıştı. 1990-2002 döneminde bilgi sistemleri endüstrisinin çeşitli alanlarında teknik destek, eğitim, pazarlama ve satış müdürlüğü görevlerinde bulundu.

2002-2008 yılları arasında İzmir Ekonomi Üniversitesi’nde öğretim görevlisi olarak Meslek Yüksekokulu Müdür Yardımcılığı görevinde bulundu. 2008 yılından beri İzmir Ekonomi Üniversitesi, Bilgisayar Bilimleri Fakültesi, Yazılım Mühendisliği Bölümü’nde öğretim görevlisi olarak akademik çalışmalarını sürdürmektedir.

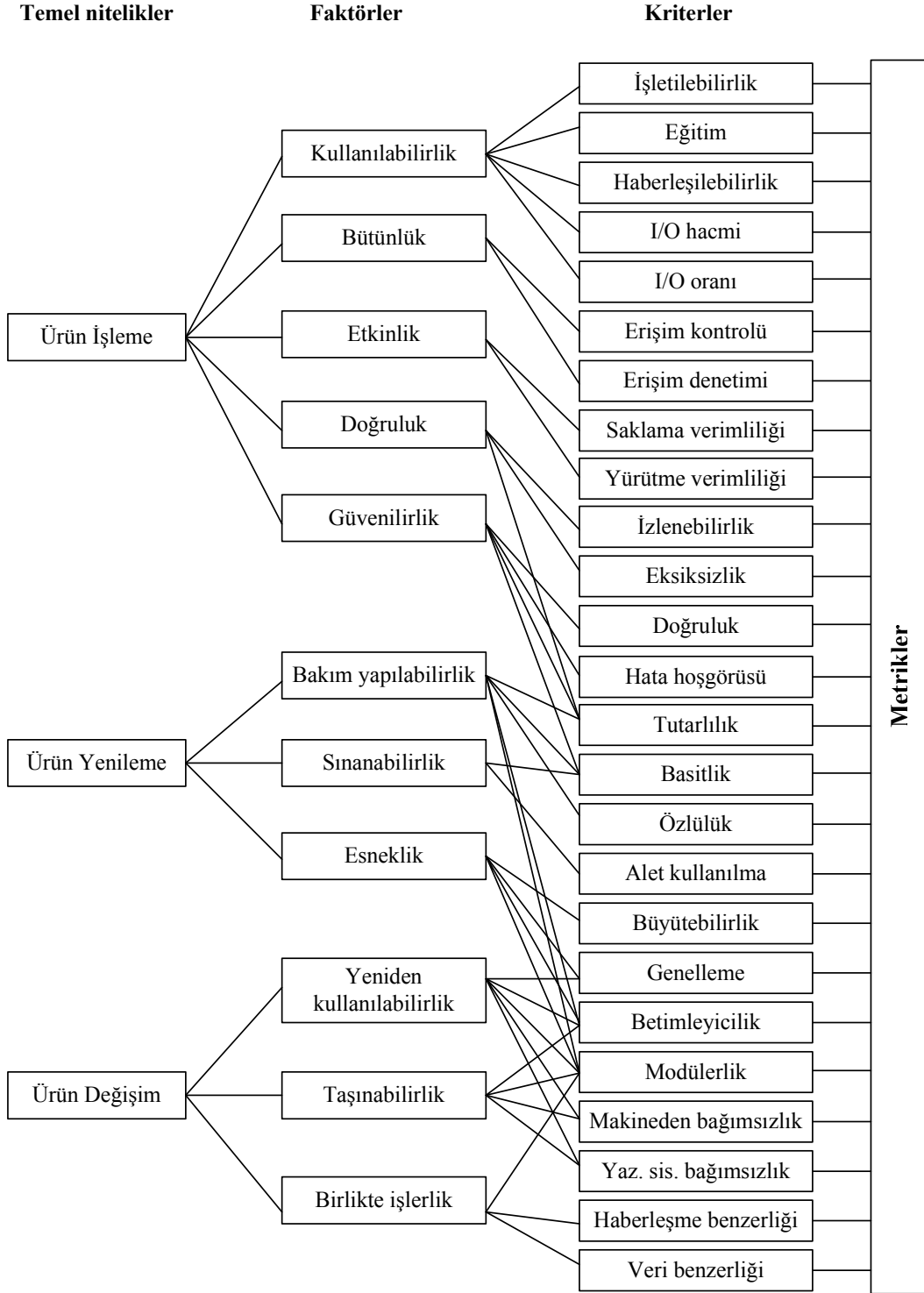
## **EKLER**

Çalışma 7 adet ek içermektedir. Ek-A ve Ek'B'de erken dönem ürün kalite modelleri sunulmaktadır. Ek-C'de ISO 9126 teknik raporlarına ait temel ölçülerin yazılım altkarakteristikleri ile olan ilişkisi sunulmaktadır. Ek-Ç, AHP yönteminin uygulaması sırasında kullanılan değerlendirme tablolarını içermektedir. Ek-D'de ISO 9126 yazılım bakımı altkarakteristiklerine ait metrikler ve açıklamaları sunulmaktadır. Ek-E'de veri tablolarına ait detaylar bulunmaktadır. Ek-F'de temel ölçülerin detayları sunulmaktadır.



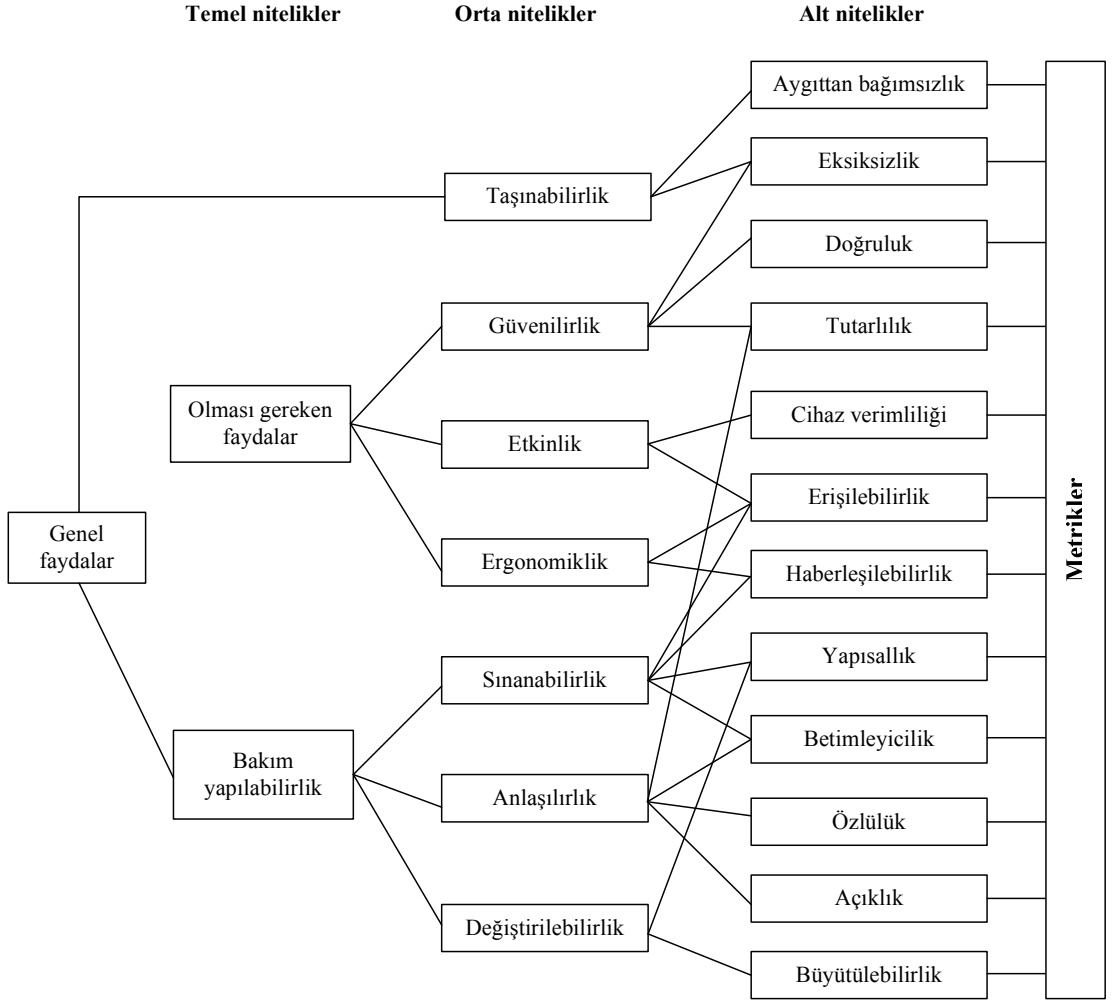
## EK-A

## McCall'ın yazılım kalite modeli.



## EK-B

## Boehm'in yazılım kalite modeli.



**EK-C**

**ISO 9126-2 ve 9126-3 teknik raporlarına ait temel ölçülerin, yazılım kalite altkarakteristikleri ile olan ilişkisi.**

Ölçü	Birim	Fonksiyonellik					Güvenilirlik			Kullanılabilirlik			Etkinlik		Bakım yapılabilirlik				Taşınabilirlik			
		F1	F2	F3	F4	F5	G1	G2	G3	K1	K2	K3	E1	E2	B1	B2	B3	B4	T1	T2	T3	T4
1	Fonksiyonların sayısı	Fonksiyon	Dİ	İ				İ		Dİ	Dİ	Dİ				İ			Dİ		İ	D
2	Operasyon süresi	Dakika		D	D		D		D		D	D	D	D		D	D	D	D		D	
3	Kullanıcının karşılaştığı hatalı hesaplamaların sayısı	Durum		D																		
4	Veri biçimlerinin toplam sayısı	Biçim			Dİ																	
5	Geçersiz işlemlerin sayısı	İşlem					D		İ													
6	Aksaklık sayısı	Aksaklık					D												D	D		
7	Hata sayısı	Hata					D	D					D		D		D				D	
8	Ürün uzunluğu	Byte					D															
9	Sınama durum sayısı	Durum					D	D	D		İ							D				
10	Bozulmaların sayısı	Bozulma						D	D													
11	Onarım zamanı	Dakika							D													
12	Sistemin çalışmadığı süre	Dakika							D													
13	Yeniden başlamaların sayısı	Yeniden başlatma							D													

**D** – Temel dış ölçüler    **İ** – Temel iç ölçüler

**EK-C (devam)**

**ISO 9126-2 ve 9126-3 teknik raporlarına ait temel ölçülerin, yazılım kalite altkarakteristikleri ile olan ilişkisi.**

Ölçü	Birim	Fonksiyonellik					Güvenilirlik			Kullanılabilirlik			Etkinlik		Bakım yapılabilirlik				Taşınabilirlik				
		F1	F2	F3	F4	F5	G1	G2	G3	K1	K2	K3	E1	E2	B1	B2	B3	B4	T1	T2	T3	T4	
14	Gerekli onarım sayısı	Onarım							Dİ														
15	Bilgilendirici belgelerin sayısı	Belge								D													
16	I/O veri öğeleri sayısı	Öge								D													
17	Fonksiyon öğrenmenin kolaylığı	Dakika									D												
18	Görevlerin sayısı	Görev									D		Dİ										
19	Yardım sıklığı	Erişim									D												
20	Hata ayıklama	Dakika										D											
21	Ekranların veya formların sayısı	Erkanlar										D											
22	Kullanıcı yanlışlarının veya değişikliklerinin sayısı	Yanlış										D											
23	Uyarılma çabalarının sayısı	Çaba										D											
24	Düzeltilen aksaklıkların sayısı	Aksaklık						İ															
25	Yanıt zamanı	Saniye											Dİ										
26	Değerlendirmelerin sayısı	Değerlendirme											D	D									

**D** – Temel dış ölçüler    **İ** – Temel iç ölçüler

**EK-C (devam)**

ISO 9126-2 ve 9126-3 teknik raporlarına ait temel ölçülerin, yazılım kalite altkarakteristikleri ile olan ilişkisi.

Ölçü	Birim	Fonksiyonellik					Güvenilirlik			Kullanılabilirlik			Etkinlik		Bakım yapılabilirlik				Taşınabilirlik					
		F1	F2	F3	F4	F5	G1	G2	G3	K1	K2	K3	E1	E2	B1	B2	B3	B4	T1	T2	T3	T4		
27	Dönüşlerin gecikme süresi	Saniye												Dİ										
28	Görev süresi	Dakika												D										
29	I/O ilişkili yanlış sayısı	Yanlış											D		Dİ									
30	Kullanıcının I/O birimlerinden yararlanmayı bekleme süresi	Saniye													D									
31	Bellek ilişkili yanlışların sayısı	Yanlış													Dİ									
32	İletimle ilgili yanlışların sayısı	Yanlış													D									
33	İletim kapasitesi	Byte													D									
34	Gözden geçirilmiş sürümlerin sayısı	Sürüm															D							
35	Çözümlenen hata sayısı	Hata						D									D	D	D					
36	Veri giriş çıkışlarında kolay kullanım	Dakika																			D			
37	Veri öğelerinin sayısı	Öğe		İ													D				D		İ	D
38	Arayüz protokollerinin sayısı	Protokol			İ																			

**D** – Temel dış ölçüler    **İ** – Temel iç ölçüler

**EK-C (devam)**

**ISO 9126-2 ve 9126-3 teknik raporlarına ait temel ölçülerin, yazılım kalite altkarakteristikleri ile olan ilişkisi.**

Ölçü	Birim	Fonksiyonellik					Güvenilirlik			Kullanılabilirlik			Etkinlik		Bakım yapılabilirlik				Taşınabilirlik				
		F1	F2	F3	F4	F5	G1	G2	G3	K1	K2	K3	E1	E2	B1	B2	B3	B4	T1	T2	T3	T4	
39	Erişim türlerinin sayısı	Erişim türü				İ																	
40	Erişim denetlenebilirlik ihtiyaçlarının sayısı	İhtiyaç				İ																	
41	Bozulmuş veri örneklerinin sayısı	Örnek				Dİ																	
42	Verilerin geçerliliği için denetlenen girdi öge sayısı	Öge										İ											
43	Operasyon Sayısı	Operasyon										Dİ											
44	Gerçekleştirilen Mesajların Sayısı	Mesaj										İ											
45	Arayüz Elemanlarının Sayısı	Eleman										İ											
46	I/O kullanımı (arabellek sayısı)	Arabellek											İ										
47	Doğrudan sistemin çağırması ile ilişkili kod uzunluğu sayısı	Satır											İ										
48	Log edilmesi gereken öğelerin sayısı	Öge													İ								
49	Yapılan değişiklik sayısı	Değişim															İ						
50	Değişkenlerin sayısı	Değişken															İ						
51	Tanılamaya gerekli fonksiyon sayısı	Fonksiyon													İ								

**D** – Temel dış ölçüler      **İ** – Temel iç ölçüler

**EK-C (devam)**

**ISO 9126-2 ve 9126-3 teknik raporlarına ait temel ölçülerin, yazılım kalite altkarakteristikleri ile olan ilişkisi.**

Ölçü	Birim	Fonksiyonellik					Güvenilirlik			Kullanılabilirlik			Etkinlik		Bakım yapılabilirlik				Taşınabilirlik				
		F1	F2	F3	F4	F5	G1	G2	G3	K1	K2	K3	E1	E2	B1	B2	B3	B4	T1	T2	T3	T4	
52	Varlıkların sayısı	Varlık																					İ
53	Önceden tanımlanmış gerekli sınaama fonksiyon sayısı	Fonksiyon																					İ
54	Sınamanın diğer sistemlere olan bağımlılık sayısı	Bağımlılık																					İ
55	Denetim noktalarının sayısı	Denetim Noktası																					İ
56	Veri yapılarının sayısı	Veri yapısı																					İ
57	Kurulum işlemlerinin sayısı	İşlem																					Di
58	Kurulum adımlarının sayısı	Adım																					İ
59	Hata Zamanı	Dakika															D	D					

**D** – Temel dış ölçüler     **İ** – Temel iç ölçüler

Alt karakteristiklere ait semboller:

F1	Uygunluk	G2	Hata toleransı	E2	Kaynak yararlanımı	T2	Kurulabilirlik
F2	Doğruluk	G3	Kurtarılabirlik	B1	Çözümenebilirlik	T3	Uygunluk
F3	Birlikte işlerlik	K1	Anlaşılabilirlik	B2	Değiştirilebilirlik	T4	Yer değiştirilebilirlik
F4	Uyumluluk	K2	Öğrenebilirlik	B3	Durağanlık		
F5	Güvenlik	K3	İşlerlik	B4	Sınanabilirlik		
G1	Olgunluk	E1	Zaman davranışı	T1	Uyarlanabilirlik		

**EK-Ç Üst kalite faktörlerine göre seçenekleri değerlendirme tabloları:**

**Gereksinimlerin kalitesi (GK) matrisi**

	Uygunluk	Doğruluk	Birlikte işlerlik	Uyumluluk	Güvenlik	Olgunluk	Hata toleransı	Kurtarılabilirlik	Anlaşılabilirlik	Öğrenilebilirlik	İşlerlik	Zaman davranışı	Kaynak yararlanımı	Çözümenebilirlik	Değiştirilebilirlik	Durağanlık	Sınanabilirlik	Uyarlanabilirlik	Kurulabilirlik	Uygunluk	Yer değiştirilebilirlik
Uygunluk	1,00	0,20	0,33	1,00	1,00	1,00	3,03	3,03	0,33	1,00	1,00	3,03	3,03	1,00	3,03	3,03	1,00	1,00	3,03	3,03	3,03
Doğruluk	5,00	1,00	1,00	1,00	1,00	1,00	3,03	3,03	1,00	1,00	1,00	3,03	3,03	1,00	1,00	3,03	1,00	1,00	3,03	3,03	3,03
Birlikte işlerlik	3,00	1,00	1,00	1,00	3,03	1,00	3,03	3,03	0,33	0,33	1,00	1,00	1,00	1,00	3,03	3,03	1,00	1,00	3,03	3,03	1,00
Uyumluluk	1,00	1,00	1,00	1,00	1,00	3,03	1,00	3,03	0,33	1,00	1,00	1,00	3,03	1,00	1,00	1,00	1,00	1,00	3,03	3,03	1,00
Güvenlik	1,00	1,00	0,33	1,00	1,00	3,03	3,03	1,00	1,00	1,00	3,03	1,00	3,03	1,00	1,00	1,00	1,00	3,03	3,03	3,03	1,00
Olgunluk	1,00	1,00	1,00	0,33	0,33	1,00	1,00	3,03	0,33	1,00	1,00	1,00	1,00	0,33	1,00	1,00	1,00	1,00	3,03	3,03	1,00
Hata toleransı	0,33	0,33	0,33	1,00	0,33	1,00	1,00	3,03	0,33	0,33	1,00	1,00	1,00	0,33	1,00	1,00	1,00	0,33	3,03	3,03	3,03
Kurtarılabilirlik	0,33	0,33	0,33	0,33	1,00	0,33	0,33	1,00	0,33	0,33	0,33	0,33	1,00	1,00	0,33	0,33	1,00	1,00	3,03	1,00	3,03
Anlaşılabilirlik	3,00	1,00	3,00	3,00	1,00	3,00	3,00	3,00	1,00	1,00	1,00	3,03	1,00	1,00	1,00	1,00	1,00	3,03	3,03	1,00	3,03
Öğrenilebilirlik	1,00	1,00	3,00	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,03	1,00	1,00	1,00	1,00	3,03	3,03	3,03	3,03
İşlerlik	1,00	1,00	1,00	1,00	0,33	1,00	1,00	3,00	1,00	1,00	1,00	3,03	3,03	1,00	1,00	3,03	3,03	0,33	1,00	1,00	1,00
Zaman davranışı	0,33	0,33	1,00	1,00	1,00	1,00	1,00	3,00	0,33	1,00	0,33	1,00	3,03	0,33	0,33	0,33	0,33	1,00	1,00	1,00	1,00
Kaynak yararlanımı	0,33	0,33	1,00	0,33	0,33	1,00	1,00	1,00	1,00	0,33	0,33	0,33	1,00	0,33	0,33	0,33	0,33	1,00	1,00	1,00	1,00
Çözümenebilirlik	1,00	1,00	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,03	3,03	3,03	3,03
Değiştirilebilirlik	0,33	1,00	0,33	1,00	1,00	1,00	1,00	3,00	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,03	3,03	3,03	3,03
Durağanlık	0,33	0,33	0,33	1,00	1,00	1,00	1,00	3,00	1,00	1,00	0,33	3,00	3,00	1,00	1,00	1,00	3,03	1,00	3,03	3,03	3,03
Sınanabilirlik	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,33	3,00	3,00	1,00	1,00	0,33	1,00	1,00	3,03	3,03	3,03
Uyarlanabilirlik	1,00	1,00	1,00	1,00	0,33	1,00	3,00	1,00	0,33	0,33	3,00	1,00	1,00	0,33	0,33	1,00	1,00	1,00	1,00	3,03	3,03
Kurulabilirlik	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33	1,00	1,00	1,00	0,33	0,33	0,33	0,33	1,00	1,00	0,33	1,00
Uygunluk	0,33	0,33	0,33	0,33	0,33	0,33	0,33	1,00	1,00	0,33	1,00	1,00	1,00	0,33	0,33	0,33	0,33	0,33	3,00	1,00	3,03
Yer değiştirilebilirlik	0,33	0,33	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,33	1,00	1,00	1,00	0,33	0,33	0,33	0,33	0,33	1,00	0,33	1,00
	22,97	14,84	19,64	19,65	18,34	27,05	34,44	43,84	14,32	15,65	21,68	35,78	43,21	15,65	20,38	24,44	21,72	28,48	51,42	46,06	45,36



**EK-Ç Üst kalite faktörlerine göre seçenekleri değerlendirme tabloları (Devam)**

**Ürün kalitesi (ÜK) matrisi**

	Uygunluk	Doğruluk	Birlikte işlerlik	Uyumluluk	Güvenlik	Olgunluk	Hata toleransı	Kurtarılabirlik	Anlaşılabilirlik	Öğrenebilirlik	İşlerlik	Zaman davranışı	Kaynak yararlanımı	Çözümenebilirlik	Değiştirilebilirlik	Durağanlık	Sınanabilirlik	Uyarlanabilirlik	Kurulabilirlik	Uygunluk	Yer değiştirilebilirlik
Uygunluk	1,00	0,33	0,33	1,00	0,33	0,33	1,00	1,00	0,33	1,00	0,33	1,00	1,00	3,03	1,00	1,00	1,00	3,03	3,03	3,03	3,03
Doğruluk	3,00	1,00	3,03	1,00	1,00	0,33	0,33	1,00	1,00	3,03	1,00	1,00	1,00	3,03	3,03	1,00	3,03	3,03	3,03	3,03	3,03
Birlikte işlerlik	3,00	0,33	1,00	1,00	3,03	1,00	3,03	3,03	0,33	0,33	1,00	1,00	1,00	1,00	3,03	3,03	1,00	1,00	3,03	3,03	1,00
Uyumluluk	1,00	1,00	1,00	1,00	1,00	3,03	1,00	3,03	0,33	1,00	1,00	1,00	3,03	1,00	1,00	1,00	1,00	1,00	3,03	3,03	1,00
Güvenlik	3,00	1,00	0,33	1,00	1,00	3,03	3,03	1,00	1,00	1,00	3,03	1,00	3,03	1,00	1,00	1,00	1,00	3,03	3,03	3,03	1,00
Olgunluk	3,00	3,00	1,00	0,33	0,33	1,00	1,00	3,03	0,33	1,00	1,00	1,00	1,00	0,33	1,00	1,00	1,00	1,00	3,03	3,03	1,00
Hata toleransı	1,00	3,00	0,33	1,00	0,33	1,00	1,00	3,03	0,33	0,33	1,00	1,00	1,00	0,33	1,00	1,00	1,00	0,33	3,03	3,03	3,03
Kurtarılabirlik	1,00	1,00	0,33	0,33	1,00	0,33	0,33	1,00	0,33	0,33	0,33	0,33	1,00	1,00	0,33	0,33	1,00	1,00	3,03	1,00	3,03
Anlaşılabilirlik	3,00	1,00	3,00	3,00	1,00	3,00	3,00	3,00	1,00	1,00	1,00	3,03	1,00	1,00	1,00	1,00	1,00	3,03	3,03	1,00	3,03
Öğrenebilirlik	1,00	0,33	3,00	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,03	1,00	1,00	1,00	1,00	3,03	3,03	3,03	3,03
İşlerlik	3,00	1,00	1,00	1,00	0,33	1,00	1,00	3,00	1,00	1,00	1,00	3,03	3,03	1,00	1,00	3,03	3,03	0,33	1,00	1,00	1,00
Zaman davranışı	1,00	1,00	1,00	1,00	1,00	1,00	1,00	3,00	0,33	1,00	0,33	1,00	3,03	0,33	0,33	0,33	0,33	1,00	1,00	1,00	1,00
Kaynak yararlanımı	1,00	1,00	1,00	0,33	0,33	1,00	1,00	1,00	1,00	0,33	0,33	0,33	1,00	0,33	0,33	0,33	0,33	1,00	1,00	1,00	1,00
Çözümenebilirlik	0,33	0,33	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,03	3,03	3,03	3,03
Değiştirilebilirlik	1,00	0,33	0,33	1,00	1,00	1,00	1,00	3,00	1,00	1,00	1,00	3,00	3,00	1,00	1,00	1,00	1,00	3,03	3,03	3,03	3,03
Durağanlık	1,00	1,00	0,33	1,00	1,00	1,00	1,00	3,00	1,00	1,00	0,33	3,00	3,00	1,00	1,00	1,00	3,03	1,00	3,03	3,03	3,03
Sınanabilirlik	1,00	0,33	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,33	3,00	3,00	1,00	1,00	0,33	1,00	1,00	3,03	3,03	3,03
Uyarlanabilirlik	0,33	0,33	1,00	1,00	0,33	1,00	3,00	1,00	0,33	0,33	3,00	1,00	1,00	0,33	0,33	1,00	1,00	1,00	1,00	3,03	3,03
Kurulabilirlik	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33	1,00	1,00	1,00	0,33	0,33	0,33	0,33	1,00	1,00	0,33	1,00
Uygunluk	0,33	0,33	0,33	0,33	0,33	0,33	0,33	1,00	1,00	0,33	1,00	1,00	1,00	0,33	0,33	0,33	0,33	0,33	3,00	1,00	3,03
Yer değiştirilebilirlik	0,33	0,33	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,33	1,00	1,00	1,00	0,33	0,33	0,33	0,33	0,33	1,00	0,33	1,00
	29,65	18,30	21,67	19,65	17,67	25,72	29,71	39,78	14,32	17,68	21,02	31,72	39,15	19,71	20,38	20,38	23,75	32,54	51,42	46,06	45,36

### EK-Ç Üst kalite faktörlerine göre seçenekleri değerlendirme tabloları (Devam)

#### Gerçekleştirim ve sınamaya verimliliği (GSV) matrisi

	Uygunluk	Doğruluk	Birlikte işlerlik	Uyumluluk	Güvenlik	Olgunluk	Hata toleransı	Kurtarılabilirlik	Anlaşılabilirlik	Öğrenebilirlik	İşlerlik	Zaman davranışı	Kaynak yararlanımı	Çözümlenebilirlik	Değiştirilebilirlik	Durağanlık	Sımanabilirlik	Uyarlanabilirlik	Kurulabilirlik	Uygunluk	Yer değiştirilebilirlik
Uygunluk	1,00	1,00	3,00	0,33	3,00	1,00	3,00	3,00	1,00	5,00	1,00	3,00	1,00	1,00	0,33	3,00	0,33	0,33	1,00	3,00	5,00
Doğruluk	1,00	1,00	0,33	0,33	1,00	1,00	3,00	3,00	1,00	3,00	1,00	3,00	1,00	1,00	1,00	3,00	0,33	1,00	1,00	1,00	3,00
Birlikte işlerlik	0,33	3,00	1,00	1,00	3,00	0,33	1,00	1,00	0,33	1,00	1,00	1,00	1,00	3,00	1,00	3,00	0,33	3,00	3,00	3,00	3,00
Uyumluluk	3,00	3,00	1,00	1,00	1,00	3,00	3,00	3,00	1,00	1,00	1,00	3,00	3,00	1,00	0,33	1,00	0,33	1,00	3,00	3,00	3,00
Güvenlik	0,33	1,00	0,33	1,00	1,00	3,00	1,00	1,00	1,00	3,00	1,00	1,00	1,00	0,33	0,33	0,33	1,00	1,00	1,00	1,00	1,00
Olgunluk	1,00	1,00	3,00	0,33	0,33	1,00	0,33	0,33	1,00	1,00	0,33	1,00	1,00	1,00	1,00	1,00	0,33	1,00	3,00	3,00	3,00
Hata toleransı	0,33	0,33	1,00	0,33	1,00	3,00	1,00	3,00	1,00	1,00	1,00	3,00	3,00	1,00	0,33	1,00	0,33	1,00	3,00	3,00	1,00
Kurtarılabilirlik	0,33	0,33	1,00	0,33	1,00	3,00	0,33	1,00	3,00	3,00	3,00	3,00	1,00	1,00	0,33	1,00	0,33	3,00	3,00	3,00	1,00
Anlaşılabilirlik	1,00	1,00	3,00	1,00	1,00	1,00	1,00	0,33	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,33	0,33	0,33	1,00	1,00	3,00
Öğrenebilirlik	0,20	0,33	1,00	1,00	0,33	1,00	1,00	0,33	1,00	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,33	1,00	1,00	1,00	1,00
İşlerlik	1,00	1,00	1,00	1,00	1,00	3,00	1,00	0,33	1,00	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,33	0,33	1,00	1,00	3,00
Zaman davranışı	0,33	0,33	1,00	0,33	1,00	1,00	0,33	0,33	1,00	1,00	1,00	1,00	1,00	1,00	0,33	1,00	0,33	1,00	1,00	1,00	1,00
Kaynak yararlanımı	1,00	1,00	1,00	0,33	1,00	1,00	0,33	1,00	3,00	1,00	1,00	1,00	1,00	1,00	0,33	1,00	1,00	1,00	1,00	1,00	1,00
Çözümlenebilirlik	1,00	1,00	0,33	1,00	3,00	1,00	1,00	1,00	3,00	3,00	3,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	3,00	3,00	3,00
Değiştirilebilirlik	3,00	1,00	1,00	3,00	3,00	1,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	1,00	1,00	3,00	1,00	3,00	3,00	3,00	3,00
Durağanlık	0,33	0,33	0,33	1,00	3,00	1,00	1,00	1,00	3,00	3,00	3,00	1,00	1,00	1,00	0,33	1,00	0,33	1,00	3,00	3,00	3,00
Sımanabilirlik	3,00	3,00	3,00	3,00	1,00	3,00	3,00	3,00	3,00	3,00	3,00	3,00	1,00	1,00	1,00	3,00	1,00	3,00	3,00	3,00	3,00
Uyarlanabilirlik	3,00	1,00	0,33	1,00	1,00	1,00	1,00	0,33	3,00	1,00	3,00	1,00	1,00	1,00	0,33	1,00	0,33	1,00	1,00	1,00	3,00
Kurulabilirlik	1,00	1,00	0,33	0,33	1,00	0,33	0,33	0,33	1,00	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,33	1,00	1,00	3,00	1,00
Uygunluk	0,33	1,00	0,33	0,33	1,00	0,33	0,33	0,33	1,00	1,00	1,00	1,00	1,00	0,33	0,33	0,33	0,33	1,00	0,33	1,00	1,00
Yer değiştirilebilirlik	0,20	0,33	0,33	0,33	1,00	0,33	1,00	1,00	0,33	1,00	0,33	1,00	1,00	0,33	0,33	0,33	0,33	0,33	1,00	1,00	1,00
	22,73	23,00	23,67	18,33	29,67	30,33	27,00	27,67	33,67	39,00	31,67	35,00	26,33	18,33	11,00	26,33	10,33	26,33	38,33	43,00	47,00

### EK-Ç Üst kalite faktörlerine göre seçenekleri değerlendirme tabloları (Devam)

#### Elverişlilik (ELV) matrisi

	Uygunluk	Doğruluk	Birlikte işlerlik	Uyumluluk	Güvenlik	Olgunluk	Hata toleransı	Kurtarılabilirlik	Anlaşılabilirlik	Öğrenebilirlik	İşlerlik	Zaman davranışı	Kaynak yararlanımı	Çözümlenebilirlik	Değiştirilebilirlik	Durağanlık	Sımanabilirlik	Uyarlanabilirlik	Kurulabilirlik	Uygunluk	Yer değiştirilebilirlik
Uygunluk	1,00	0,33	1,00	0,33	1,00	3,00	3,00	0,33	1,00	1,00	3,00	5,00	5,00	5,00	1,00	1,00	0,33	0,33	3,00	1,00	5,00
Doğruluk	3,00	1,00	5,00	1,00	1,00	0,33	3,00	3,00	1,00	5,00	3,00	5,00	3,00	1,00	0,33	3,00	0,33	1,00	1,00	1,00	3,00
Birlikte işlerlik	1,00	0,20	1,00	0,33	3,00	1,00	3,00	1,00	0,33	3,00	0,33	3,00	3,00	0,33	0,33	3,00	1,00	1,00	1,00	1,00	3,00
Uyumluluk	3,00	1,00	3,00	1,00	3,00	3,00	5,00	5,00	1,00	3,00	3,00	3,00	3,00	1,00	0,33	1,00	1,00	1,00	3,00	3,00	3,00
Güvenlik	1,00	1,00	0,33	0,33	1,00	1,00	3,00	0,33	0,33	1,00	0,33	5,00	3,00	0,33	0,33	3,00	0,33	0,33	3,00	1,00	3,00
Olgunluk	0,33	3,00	1,00	0,33	1,00	1,00	1,00	0,33	0,33	3,00	0,33	3,00	3,00	0,33	0,33	1,00	0,33	1,00	3,00	1,00	3,00
Hata toleransı	0,33	0,33	0,33	0,20	0,33	1,00	1,00	1,00	0,33	1,00	0,33	1,00	1,00	0,20	0,20	0,33	0,33	1,00	3,00	1,00	3,00
Kurtarılabilirlik	3,00	0,33	1,00	0,20	3,00	3,00	1,00	1,00	0,33	3,00	1,00	3,00	3,00	0,33	0,33	1,00	0,33	1,00	1,00	1,00	1,00
Anlaşılabilirlik	1,00	1,00	3,00	1,00	3,00	3,00	3,00	3,00	1,00	5,00	1,00	3,00	3,00	1,00	1,00	3,00	1,00	3,00	3,00	3,00	3,00
Öğrenebilirlik	1,00	0,20	0,33	0,33	1,00	0,33	1,00	0,33	0,20	1,00	0,33	1,00	3,00	0,33	0,33	1,00	0,33	1,00	1,00	1,00	1,00
İşlerlik	0,33	0,33	3,00	0,33	3,00	3,00	3,00	1,00	1,00	3,00	1,00	3,00	3,00	0,33	1,00	3,00	1,00	1,00	3,00	3,00	3,00
Zaman davranışı	0,20	0,20	0,33	0,33	0,20	0,33	1,00	0,33	0,33	1,00	0,33	1,00	1,00	0,20	0,20	1,00	0,33	1,00	1,00	1,00	1,00
Kaynak yararlanımı	0,20	0,33	0,33	0,33	0,33	0,33	1,00	0,33	0,33	0,33	0,33	1,00	1,00	0,33	0,33	1,00	0,33	0,33	1,00	1,00	3,00
Çözümlenebilirlik	0,20	1,00	3,00	1,00	3,00	3,00	5,00	3,00	1,00	3,00	3,00	5,00	3,00	1,00	1,00	3,00	1,00	1,00	3,00	3,00	3,00
Değiştirilebilirlik	1,00	3,00	3,00	3,00	3,00	3,00	5,00	3,00	1,00	3,00	1,00	5,00	3,00	1,00	1,00	3,00	1,00	1,00	3,00	3,00	3,00
Durağanlık	1,00	0,33	0,33	1,00	0,33	1,00	3,00	1,00	0,33	1,00	0,33	1,00	1,00	0,33	0,33	1,00	0,33	1,00	1,00	1,00	3,00
Sımanabilirlik	3,00	3,00	1,00	1,00	3,00	3,00	3,00	3,00	1,00	3,00	1,00	3,00	3,00	1,00	1,00	3,00	1,00	3,00	1,00	1,00	3,00
Uyarlanabilirlik	3,00	1,00	1,00	1,00	3,00	1,00	1,00	1,00	0,33	1,00	1,00	1,00	3,00	1,00	1,00	1,00	0,33	1,00	3,00	1,00	1,00
Kurulabilirlik	0,33	1,00	1,00	0,33	0,33	0,33	0,33	1,00	0,33	1,00	0,33	1,00	1,00	0,33	0,33	1,00	1,00	0,33	1,00	1,00	3,00
Uygunluk	1,00	1,00	1,00	0,33	1,00	1,00	1,00	1,00	0,33	1,00	0,33	1,00	1,00	0,33	0,33	1,00	1,00	1,00	1,00	1,00	1,00
Yer değiştirilebilirlik	0,20	0,33	0,33	0,33	0,33	0,33	0,33	1,00	0,33	1,00	0,33	1,00	0,33	0,33	0,33	0,33	0,33	1,00	0,33	1,00	1,00
	25,13	19,93	30,33	14,07	34,87	33,00	47,67	31,00	12,20	44,33	21,67	55,00	50,33	16,07	11,40	35,67	13,00	22,33	40,33	31,00	53,00

**EK-D ISO 9126 standartına göre değiştirilebilirlik, analiz edilebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler.**

Metrik adı	Metriğin amacı	Uygulama metodu	Ölçüm, formül ve veri elemanın hesaplanması	Ölçüm değerinin yorumu
<b>B1 Çözümenebilirlik Dış Metrikleri</b>				
M1	<i>Teşhisin desteklenmesi</i>	Teşhis fonksiyonları, nedensel analiz desteğini verebilmede ne kadar yetenekli? Kullanıcı hata olduğunda özel bir süreç tanımlayabiliyor mu?	Hataları çözmek için teşhise yönelik fonksiyonların kullanılması sırasında, kullanıcı veya bakımıcının davranışları gözlemlenir.	$X=A/B$ A=Bakımcının neden-sonuç ilişkisini anlaması için teşhis edebileceği hata sayısı B=Kayıtlı hata sayısı $0 \leq X \leq 1$ Değerin 1'e yakınlığı iyidir.
M2	<i>Denetlenebilirlik</i>	Kullanıcı hataya neden olan belirli operasyonları tanımlayabiliyor mu? Bakımcı hataya neden olan belirli operasyonları kolaylıkla bulabiliyor mu?	Hataları çözmek için uğraşan kullanıcı veya bakımıcının davranışları gözlemlenir.	$X=A/B$ A=Operasyon sürecinde fiilen kayıtlanan veri sayısı B=Yazılımın operasyon sırasında durumunu izlemeye yeterli olan kayıtlanması planlanmış veri sayısı $0 \leq X$ Değerin 1'e yakınlığı iyidir.
M3	<i>Hata çözümleme etkinliği</i>	Kullanıcı hata nedenini etkin olarak analiz edebiliyor mu? Bakımcı hata nedenlerini kolaylıkla bulabiliyor mu? Hata nedenleri kolaylıkla nasıl hallediliyor?	Hataları çözmek için uğraşan kullanıcı veya bakımıcının davranışları gözlemlenir.	$X=Toplam(T)/N$ T=Tout-Tin Tout=Hata nedenlerinin ortaya çıkarılma zamanı (veya durum raporunun kullanıcıya gönderilmesi) Tin=Hata raporlarının alınma zamanı N=Kayıtlı hata sayısı $0 \leq X$ Oranın küçük olması iyidir.
<i>Not: En kötü duruma ait en fazla süreyi ve sapmayı gösteren zaman aralığının ölçülmesi önerilmektedir.</i>				

**EK-D ISO 9126 standartına göre değiştirilebilirlik, analiz edilebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler. (Devam)**

Metrik adı	Metriğin amacı	Uygulama metodu	Ölçüm, formül ve veri elemanın hesaplanması	Ölçüm değerinin yorumu
M4 <i>Hatayı çözümlenebilme yeteneği</i>	Kullanıcı hataya neden olan operasyonları tanımlayabiliyor mu? Bakımcı hata nedenlerini kolaylıkla bulabiliyor mu?	Hataları çözmek için uğraşan kullanıcı veya bakımcının davranışları gözlemlenir.	$X=1-A/B$ A=Nedeni henüz bulunamamış hataların sayısı B=Kayıtlı hata sayısı	$0 \leq X \leq 1$ Değerin 1'e yakınlığı iyidir.
M5 <i>Durumun izlenebilmesi</i>	Kullanıcı hataya neden olan belirli operasyonları, operasyon sırasında verileri izleyerek tanımlayabiliyor mu? Bakımcı hata nedenlerini operasyon sırasında verileri izleyerek kolaylıkla bulabiliyor mu?	Yazılımındaki operasyon süresinde veriler izlenerek, durumu kayıt eden kullanıcı veya bakımcının davranışları gözlemlenir.	$X=1-A/B$ A=Bakımcının (veya kullanıcının) verileri izlerken hata olarak tespit ettiği durumların sayısı B=Bakımcının (veya kullanıcının) operasyon süresinde yazılıma ait verileri izleyerek, kaydettiği durumların sayısı	$0 \leq X \leq 1$ Değerin 1'e yakınlığı iyidir.
<b>B1 Çözümlenebilirlik İç Metrikleri</b>				
M6 <i>Faaliyet kaydetme</i>	Sistemin durumu eksiksiz olarak nasıl kaydedilir?	Faaliyet kayıtlarından, belirlenmiş öğeler sayılır ve kayıtlanması gereken öğelerin sayısı ile kıyaslanır.	$X=A/B$ A= Değerlendirme sonucunda belirlenerek onaylanmış ve girişi yapılmış veri sayısı B=Kayıtlanmak üzere özellikleri tanımlanmış veri sayısı	$0 \leq X \leq 1$ Değerin 1'e yakınlığı, kayıt sisteminin daha fazla veri sağladığını gösterir.
M7 <i>Tanımlama fonksiyonu hazırlama</i>	Tanımlama fonksiyonları eksiksiz olarak nasıl hazırlanır?	Gerçekleştirilen tanımlama fonksiyonların sayısı bulunur ve gerekli tanımlama fonksiyonlarının sayısı ile kıyaslanır.	$X=A/B$ A= Değerlendirme sonucunda belirlenerek onaylanmış ve tanımlanmış veri sayısı B=Gerekli tanımlama fonksiyonu sayısı	$0 \leq X$ Değerin 1'e yakınlığı, tanımlama fonksiyonunun daha iyi gerçekleştirildiğini gösterir.

**EK-D ISO 9126 standartına göre değiştirilebilirlik, analiz edilebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler. (Devam)**

Metrik adı	Metriğin amacı	Uygulama metodu	Ölçüm, formül ve veri elemanın hesaplanması	Ölçüm değerinin yorumu	
<b>B2 Değiştirilebilirlik Dış Metrikleri</b>					
M8	<i>Yazılımdaki değişiklikleri denetleme yeteneği</i>	Kullanıcı gözden geçirilmiş sürümleri kolaylıkla tanımlayabiliyor mu? Bakımcı problemleri çözmek için yazılımı kolaylıkla değiştirebiliyor mu?	Yazılımı değiştirmeye uğraşırken kullanıcı veya bakımcının davranışları gözlemlenir. Ayrıca, sorunu çözüm önerileri veya bakım raporları ile araştırılır.	$X=A/B$ A= Tutulan verilerde fiilen kaydedilmiş değişiklik sayısı B= Tutulan verilerde yazılım değişikliklerini izlemek için yeterli olan kaydedilmesi planlanan değişiklik sayısı	$0 \leq X \leq 1$ Değerin 1'e yakınlığı daha iyidir veya 0'a yakın olma daha az değişiklik anlamına gelmektedir.
M9	<i>Parametreleri belirlenmiş değiştirilebilirlik</i>	Kullanıcı veya bakımcı yazılımı değiştirirken ve sorunları çözerken kolaylıkla parametreleri değiştirebilir mi?	Yazılımı değiştirmeye uğraşırken kullanıcı veya bakımcının davranışları gözlemlenir. Ayrıca, sorunu çözüm önerileri veya bakım raporları ile araştırılır.	$X=1-A/B$ A=Bakımcının yazılımı değiştirmeden parametre kullanarak gerçekleştirdiği durum sayısı B=Bakımcının parametre kullanarak yazılımı değiştirmek için girişimde bulunduğu durum sayısı	$0 \leq X \leq 1$ Değerin 1'e yakınlığı iyidir.
M10	<i>Değiştirme zorluğu</i>	Bakımcı problemi çözebilmek için yazılımı kolaylıkla değiştirebiliyor mu?	Yazılımı değiştirmeye uğraşan kullanıcı veya bakımcının davranışları gözlemlenir. Ayrıca, ürünün tanımı ile beraber soruna ait çözüm önerileri ve bakım raporları araştırılır.	$T=Toplam(A/B)/N$ A=Değişim için harcanan çalışma zamanı B=Yazılım değişikliğinin büyüklüğü (size) N=Değişiklik sayısı	$0 < T$ Oranın küçük olması iyidir veya aşırı oranda değişiklik gereksinimini temsil eder.
<i>Not: Yazılım değişikliğinin büyüklüğü, program kodunun yürütülür deyimlerine, isterlerin gereksinimine göre değişebilir.</i>					

**EK-D ISO 9126 standartına göre değiştirilebilirlik, analiz edilebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler. (Devam)**

Metrik adı	Metriğin amacı	Uygulama metodu	Ölçüm, formül ve veri elemanın hesaplanması	Ölçüm değerinin yorumu
M11 <i>Değişim döngüsünün verimliliği</i>	Kullanıcıların sorunları, uygun bir sürede ve tatmin edici bir şekilde çözümlenebiliyor mu?	Kullanıcı ile tedarikçi arasındaki etkileşim izlenir. Kullanıcıların problemin çözülmesi konusundaki istekleri için geçen süre kayıt edilir.	Ortalama Süre: $T_{ort} = \frac{\text{Toplam}(T_u)}{N}$ $T_u = T_{rc} - T_{sn}$ $T_{sn} = \text{Kullanıcının, problem raporuyla tedarikçiye bakım gereksinimini gönderme zamanı}$ $T_{rc} = \text{Kullanıcının, düzenlenmiş sürüm yayını (veya durum raporunu) alma zamanı}$ $N = \text{Yayınlanmış sürüm sayısı}$	$0 < T_{ort}$ Oranın küçük olması iyidir. Ancak, gözden geçirilmiş büyük sürümler buna dahil edilmemelidir.
M12 <i>Değişikliğin gerçekleştirilme süresi</i>	Bakımcı, hata ile ilgili sorunları çözebilmek için yazılımı kolaylıkla değiştirebiliyor mu?	Yazılım değişikliğinin gerçekleştirilmesinde kullanıcı ile bakımcı arasındaki etkileşim gözlemlenir. Ayrıca, problem veya bakım raporları araştırılır.	Ortalama Süre: $T_{ort} = \frac{\text{Toplam}(T_m)}{N}$ $T_m = T_{out} - T_{in}$ $T_{out} = \text{Yazılımın değiştirilerek (veya durum raporu kullanıcıya gönderildiğinde) hata nedenlerinin ortadan kaldırılma zamanı}$ $T_{in} = \text{Hata nedenlerinin ortaya çıkarılma zamanı}$ $N = \text{Kayıtlı ve ortadan kaldırılmış hataların sayısı}$	$0 < T_{ort}$ Oranın küçük olması iyidir. Ancak, büyük sayıdaki hatalar buna dahil edilmemelidir.
<i>Not: En kötü duruma ait en fazla süreyi ve sapmayı gösteren zaman aralığının ölçülmesi önerilmektedir.</i>				
<b>B2 Değiştirilebilirlik İç Metrikleri</b>				
M13 <i>Değişikliklerin kaydedilebilmesi</i>	Program modüllerinin ve özelliklerin değişimi kodda yeterli açıklama satırları ile kaydedildi mi?	Modül değişimine ait oran kaydedilir.	$X = A/B$ $A = \text{Gözden geçirmede onaylanmış fonksiyon/modüllerdeki değişim sayısı}$ $B = \text{Asıl kodda değiştirilmiş toplam fonksiyon/modül sayısı}$	$0 \leq X \leq 1$ Değerin 1'e yakın olması daha fazla kaydedilebilir olması, değişikliğin sıfır olması değişikliğin düşük veya az olduğunu gösterir.

**EK-D ISO 9126 standartına göre değiştirilebilirlik, analiz edilebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler. (Devam)**

Metrik adı	Metriğin amacı	Uygulama metodu	Ölçüm, formül ve veri elemanın hesaplanması	Ölçüm değerinin yorumu	
<b>B4 Sınanabilirlik Dış Metrikleri</b>					
M18	<i>Tekrar sınanabilirliğin verimliliği</i>	Kullanıcı ve bakımcı operasyonel sınamayı ve yazılımın işleme hazır olup olmadığına karar vermeyi kolayca yapabiliyor mu?	Bakım sonrasında yazılım sistemini sınanan kullanıcı veya bakımcının davranışları gözlemlenir.	$X = \text{Toplam}(T)/N$ $T = \text{Rapor edilen hataların çözümlenip çözülemediğinden emin olmak için sınamaya harcanan zaman}$ $N = \text{Çözülmüş hataların sayısı}$	$0 < X$ Değerin küçük olması iyidir.
<i>Not: X, hata sonrasında yapılan sınamanın çözümlenmesi için geçen ortalama zamanı (harcanan çaba) göstermektedir.</i>					
M19	<i>Yerleşik sınaama fonksiyonlarının kullanılması</i>	Kullanıcı ve bakımcı operasyonel sınamayı, ilave sınaama hazırlığı yapmaksızın kolayca yapabiliyor mu?	Bakım sonrasında yazılım sistemini sınanan kullanıcı veya bakımcının davranışları gözlemlenir.	$X = A/B$ $A = \text{Bakımcının yerleşik sınaama fonksiyonlarını doğru şekilde kullanarak gerçekleştirdiği durum sayısı}$ $B = \text{Sınaama fırsatlarının sayısı}$	$0 \leq X \leq 1$ Değerin 1'e yakınlığı ve büyük olması iyidir.
M20	<i>Sınamayı yeniden başlatabilme</i>	Kullanıcı ve bakımcı operasyonel sınamayı, bakım sonrasında denetim noktalarıyla kolaylıkla yapabiliyor mu?	Bakım sonrasında yazılım sistemini sınanan kullanıcı veya bakımcının davranışları gözlemlenir.	$X = A/B$ $A = \text{Bakımcının yürütülen sınaama uygulamasını istediği noktada durdurup ve tekrar başlatarak adım adım denetleme yapabilme isteğine ait durum sayısı}$ $B = \text{Yürütülen sınaama uygulamasının durdurulma sayısı}$	$0 \leq X \leq 1$ Değerin 1'e yakınlığı ve büyük olması iyidir.



**EK-D ISO 9126 standartına göre değiştirilebilirlik, analiz edilebilirlik, sınanabilirlik altkarakteristiklerine ait metrikler. (Devam)**

Metrik adı	Metriğin amacı	Uygulama metodu	Ölçüm, formül ve veri elemanın hesaplanması	Ölçüm değerinin yorumu	
<b>B4 Sınanabilirlik İç Metrikleri</b>					
M21	Önceden tanımlanmış sınama fonksiyonlarını tanımlığı	Önceden tanımlı (yerleşik) sınama fonksiyonları nasıl halledilecek?	Gerçekleştirilen önceden tanımlı sınama fonksiyonlarının sayısı ile gereksinimi duyulan fonksiyonların sayısı kıyaslanır.	$X=A/B$ A=Gerçekleştirilmiş önceden tanımlı sınama fonksiyonu sayısı B=Gerekli önceden tanımlı fonksiyon sayısı	$0 \leq X \leq 1$ Değerin 1'e yakın olması, tamlığa işaret eder.
M22	Sınanabilirliğin özerkliği	Yazılım ayrı olarak nasıl sınanabilir?	Diğer sistemlere olan bağımlılığının sayısı ile toplam bağımlılık sayısı kıyaslanır.	$X=A/B$ A=Sınamaya ait diğer sistemler ile olan bağımlılık sayısı B=Sınamaya ait diğer sistemler ile olan toplam bağımlılık sayısı	$0 \leq X \leq 1$ Değerin 1'e yakınlığı iyidir.
M23	Sınanabilirlik gelişiminin izlenmesi	Sınama sürecinde tanımlı sınama sonuçlarının görüntülenmesi nasıl yapılır?	Denetleme noktalarının sayısı ile tasarımda belirlenen gerekli denetleme noktalarının sayısı kıyaslanır.	$X=A/B$ A=Gerçekleştirilmiş önceden tanımlı denetleme noktası sayısı B=Hazırlanmış denetleme noktası sayısı	$0 \leq X \leq 1$ Değerin 1'e yakınlığı iyidir.

**EK-E Veri Tabloları****E1 – Mainframe Ortamına Ait Genel Hata Üretme ve Bakım Verileri – Yapılan bakımlar**

AYLARA GÖRE KAPANAN OLAY SAYILARI 01/10/2006-31/12/2006													
		Ocak	Şubat	Mart	Nisan	Mayıs	Haziran	Temmuz	Ağustos	Eylül	Ekim	Kasım	Aralık
Mainframe	Yazılım Destek											211	209
	Yazılım Geliştirme											8	17
	<b>Toplam</b>											<b>219</b>	<b>226</b>

AYLARA GÖRE KAPANAN OLAY SAYILARI 01/01/2007-31/12/2007													
		Ocak	Şubat	Mart	Nisan	Mayıs	Haziran	Temmuz	Ağustos	Eylül	Ekim	Kasım	Aralık
Mainframe	Yazılım Destek	171	208	223	224	214	319	279	191	146	210	271	171
	Yazılım Geliştirme	37	26	16	17	16	27	25	16	15	17	25	21
	<b>Toplam</b>	<b>208</b>	<b>234</b>	<b>239</b>	<b>241</b>	<b>230</b>	<b>346</b>	<b>304</b>	<b>207</b>	<b>161</b>	<b>227</b>	<b>296</b>	<b>192</b>

AYLARA GÖRE KAPANAN OLAY SAYILARI 01/01/2008-31/12/2008													
		Ocak	Şubat	Mart	Nisan	Mayıs	Haziran	Temmuz	Ağustos	Eylül	Ekim	Kasım	Aralık
Mainframe	Yazılım Destek	163	160	189	211	162	226	174	242	321	197	189	160
	Yazılım Geliştirme	24	7	85	39	23	39	69	37	33	29	32	65
	<b>Toplam</b>	<b>187</b>	<b>167</b>	<b>274</b>	<b>250</b>	<b>185</b>	<b>265</b>	<b>243</b>	<b>279</b>	<b>354</b>	<b>226</b>	<b>221</b>	<b>225</b>

AYLARA GÖRE KAPANAN OLAY SAYILARI 01/01/2009-31/12/2009													
		Ocak	Şubat	Mart	Nisan	Mayıs	Haziran	Temmuz	Ağustos	Eylül	Ekim	Kasım	Aralık
Mainframe	Yazılım Destek	182	183	314	150	150	175	211					
	Yazılım Geliştirme	69	60	74	60	44	41	38					
	<b>Toplam</b>	<b>251</b>	<b>243</b>	<b>388</b>	<b>210</b>	<b>194</b>	<b>216</b>	<b>249</b>					

## E2 – Web Ortamına Ait Genel Bakım Verileri

Tarih	Gereksinim (1)	Değişiklik İsteği (2)	Hata (3)	Müşteri Değişiklik İsteği (4)	Müşteri Hata (5)	Toplam (4)+(5)
Ekim'06	38	356	2,039			
Kasım'06		25	450			
Aralık'06		71	257			
Ocak'07	65	57	168			
Şubat'07		58	223			
Mart'07		20	587			
Nisan'07		63	71			
Mayıs'07		50	27			
Haziran'07		3	5			
Temmuz'07		39	104			
Ağustos'07		30	420			
Eylül'07		34	285			
Ekim'07		232	524			
Kasım'07		164	542			
Aralık'07		52	336	40	79	119
Ocak'08		1	77	147	251	398
Şubat'08		11	65	114	252	366
Mart'08	183		19	104	216	320
Nisan'08		6	51	73	136	209
Mayıs'08			40	37	171	208
Haziran'08	260	68	151	43	109	152
Temmuz'08		76	220	54	77	131
Ağustos'08	2	89	85	21	44	65
Eylül'08	10	13	265	39	92	131
Ekim'08	3	16	74	59	84	143
Kasım'08		59	79	57	82	139
Aralık'08		14	29	33	71	104
Ocak'09		23	237	30	144	174
Şubat'09	272	77	95	52	204	256
Mart'09	269	45	133	19	239	258
Nisan'09	49	142	194	17	215	232
Mayıs'09	114	17	39	19	256	275
Haziran'09	125	84	97	7	149	156
Temmuz'09	498	197	218	18	368	386

### E3 – LM Müşteri Değişiklik İstekleri Veri Alanları - Web Ortamı

1	ID	<<tamsayı>>
2	Work Item Type	Web değişiklik isteği
3	State	Analizde, Kapalı, Onay bekliyor, Planlandı, Tamamlandı, Yeni
4	Reason	Faturalanacak süre belirtiniz Faturalanacak süre girildi Hatalı durum değişikliği İptal edildi İşleme alındı Kapsam değişikliği Kapsam dışı Müşteri onayladı Test edilmedi Test ekibi onaylamadı Yeni değişiklik isteği
5	Created by	<<string>>
6	Created date	17.12.2007-20.07.2009 <<tarih+saat>>
7	Analizde Durumuna Getiren Kişi	<<string>>
8	Analizde Durumuna Getirildiği Tarih	boş-24.07.2009 <<tarih+saat>>
9	Onay Bekliyor Durumuna Getiren Kişi	<<string>>
10	Onay Bekliyor Durumu Getirildiği Tarih	boş-29.07.2009 <<tarih+saat>>
11	Onaylandı Durumuna Getiren Kişi	<<string>>
12	Onaylandı Durumuna Getirildiği Tarih	Boş-24.07.2009 <<tarih+saat>>
13	Activated by	<<string>>
14	Activated date	17.12.2007-24.07.2009 <<tarih+saat>>
15	Resolved by	<<string>>
16	Resolved date	23.01.2008-28.07.2009 <<tarih+saat>>
17	Onayı yapan kişi	<<string>>
18	Onay tarihi	23.01.2009-31.07.2009 <<tarih+saat>>
19	Closed by	<<string>>
20	Closed date	24.01.2007-24.07.2009 <<tarih+saat>>
21	Triage	Aktif, Bilgi bekleniyor
22	Title	<<string>>
23	Area path	<<string>>
24	Iteration path	Canlı sistem, Müşteri testi, Şirket içi test
25	Müşteriye etkisi	1 – Acil 2 – Zorunlu 3 – Önemli 4 - Düzeltilmesi İyi Olacak Riskler 5 - İyileştirme
26	Sisteme etkisi	1 - Veri Kaybı 2 - İşlevsellik Kaybı 3 - Giderilebilir İşlevsellik Kaybı 4 - Kısmi İşlevsellik Kaybı 5 - Kozmetik Risk
27	Gerçekleşen süre	boş veya elle girilmiş değer
28	Talep eden müşteri	<<string>>
29	Talep eden şirket	<<string>>
30	Test ekibi tekrar açtı mı?	Evet, Hayır
31	Müşteri tekrar açtı mı?	Evet, Hayır
32	Fatura kesilecek mi?	Evet, Hayır
33	Toplam fatura süresi	boş veya elle girilmiş değer

## E4 – LM Müşteri Hata İstekleri Veri Alanları - Web Ortamı

1	ID	<<tamsayı>>
2	Work Item Type	Web hata
3	State	Kapalı, Planlandı, Çözümlendi, Tamamlandı
4	Reason	Aynı hata kayıtlı Hata çözümlendi Hata değildir Test edilmedi İptal edildi İşleme alındı Müşteri onayladı
5	Created by	<<string>>
6	Created date	17.12.2007-16.05.2008 <<tarikh+saat>>
7	Activated by	<<string>>
8	Activated date	18.12.2007-04.07.2008 <<tarikh+saat>>
9	Resolved by	<<string>>
10	Resolved date	18.12.2007-10.10.2008 <<tarikh+saat>>
11	Onayı yapan kişi	<<string>>
12	Onay tarihi	18.12.2007-10.10.2008 <<tarikh+saat>>
13	Closed by	<<string>>
14	Closed date	18.12.2007-06.02.2009 <<tarikh+saat>>
15	Root cause	Arayüz, Dokümantasyon, Entegrasyon, Fonksiyonalite, Güvenlik, Operasyonlar/Sürdürülebilirlik, Performans, Veri Göçü, Veri Kalitesi, Yük/Kapasite
16	Title	<<string>>
17	Area path	<<string>>
18	Iteration path	Canlı sistem, Müşteri testi, Şirket içi test
19	Müşteriye etkisi	1 – Acil 2 – Zorunlu 3 – Önemli 4 - Düzeltilmesi İyi Olacak Riskler 5 – İyileştirme
20	Sisteme etkisi	1 - Veri Kaybı 2 - İşlevsellik Kaybı 3 - Giderilebilir İşlevsellik Kaybı 4 - Kısmi İşlevsellik Kaybı 5 - Kozmetik Risk Düşük, Orta
21	Quick Fix	Evet, Hayır
22	Geçici Çözüm Tipi	Bilgilendirme, Entegrasyon, Kod Değişikliği, Kontrol ve Rapor Değişikliği, Veri Değişikliği, Veri Tekrarı
23	Gerçekleşen süre	boş veya elle girilmiş değer
24	Symptom	<<string>>
25	Talep eden müşteri	<<string>>
26	Talep eden şirket	<<string>>
27	Test ekibi tekrar açtı mı?	Evet, Hayır
28	Müşteri tekrar açtı mı?	Evet, Hayır

**E5 – Temel ölçülere ait aylık ölçüm sonuçları (01.01.2008-31.07.2009)**

		<b>T1 Hata Düzeltilme Süresi</b>	<b>T2 Değişiklik Süresi</b>	<b>T3 Hata sayısı</b>	<b>T4 Değişiklik sayısı</b>	<b>T5 Düzeltilen hata sayısı</b>	<b>T6 Yapılan değişiklik sayısı</b>
1	Ocak'08	431,37	349,20	251	147	194	50
2	Şubat'08	429,14	1.011,49	252	114	233	94
3	Mart'08	453,50	921,81	216	104	208	72
4	Nisan'08	279,87	820,84	136	73	113	53
5	Mayıs'08	713,62	947,80	171	37	167	38
6	Haziran'08	562,07	178,63	109	43	111	21
7	Temmuz'08	204,69	362,93	77	54	71	48
8	Ağustos'08	370,28	229,44	44	21	33	36
9	Eylül'08	784,41	341,49	92	39	115	43
10	Ekim'08	574,27	511,89	84	59	97	66
11	Kasım'08	94,72	318,07	82	57	75	71
12	Aralık'08	189,32	166,41	71	33	45	19
13	Ocak'09	328,10	509,53	144	30	126	31
14	Şubat'09	189,05	103,11	204	52	202	28
15	Mart'09	85,43	94,59	239	19	191	7
16	Nisan'09	860,06	48,32	215	17	232	28
17	Mayıs'09	410,22	46,16	256	19	252	18
18	Haziran'09	239,72	23,08	149	7	124	6
19	Temmuz'09	1.209,18	124,26	368	18	376	13
	<b>Toplam</b>	<b>8.409,02</b>	<b>7.109,05</b>	<b>3.160</b>	<b>943</b>	<b>2.965</b>	<b>742</b>

## EK-F Temel Ölçüler

### Temel Ölçü 1: Hata düzeltme süresi (T1)

İlişki:

- Kalite grubu : Dış kalite ölçümü
- Yazılım ortamı : LM web ortamı
- Ek-E4 veri alanları (Reason, Activated date/Planlandı, Resolved date/Çözümlendi)

Ölçüm yöntemi:

- Bakımcılar tarafından hataların giderilme süreleri, çalışma saatleri esas alınarak hesaplanır
- Resolve date: Hatanın ortadan kaldırılma zamanı
- Activated date: Hatanın ortaya çıkarılma zamanı
- $T1 = \text{Hesapla}[(01.01.2008 \leq \text{Toplam}(\text{Resolved date} - \text{Activated date}) \leq 31.07.2009)] - \text{Planlandı}$

Ölçüm birimi: Gün

Özellik: Bakıma ait hata düzeltme faaliyetlerinin gerçekleştirilme süresi

Kısıt: Bakımcının, hataları düzeltmesi sırasındaki duraklamaları hesaplanamamaktadır.

### Temel Ölçü 2: Değişiklik süresi (T2)

İlişki:

- Kalite grubu : Dış kalite ölçümü
- Yazılım ortamı : LM web ortamı
- Ek-E3 veri alanları (Reason, Activated date/Planlandı, Resolved date/Çözümlendi)

Ölçüm yöntemi:

- Bakımcılar tarafından değişikliğin gerçekleştirilme süreleri, çalışma saatleri esas alınarak hesaplanır
- Resolve date: Değişikliğin tamamlanma zamanı
- Activated date: Değişikliğin gerçekleştirilmeye başlama zamanı
- $T2 = \text{Hesapla}[(01.01.2008 \leq \text{Toplam}(\text{Resolved date} - \text{Activated date}) \leq 31.07.2009)] - \text{Planlandı}$

Ölçüm birimi: Gün

Özellik: Bakım işlemlerine ait değişikliklere ait temel faaliyetlerinin gerçekleştirilme süresi

Kısıt: Bakımcının, değişiklik sırasındaki duraklamaları hesaplanamamaktadır.

**EK-F Temel Ölçüler (Devam)****Temel Ölçü 3: Hata sayısı (T3)****İlişki:**

- Kalite grubu : Dış kalite ölçümü
- Yazılım ortamı : LM web ortamı
- Ek-E4 veri alanları (Reason, Created date)

**Ölçüm yöntemi:**

- Müşteri tarafından bildirilen hatalı durumlar sayılır
- T3=Say [(01.01.2008≤Created date ≤31.07.2009)- Aynı hata kayıtlı  
- Hata değildir - İptal edildi]

**Ölçüm birimi: Hata**

**Özellik: Bakıma neden olan hataların sayısı**

**Temel Ölçü 4: Değişiklik sayısı (T4)****İlişki:**

- Kalite grubu : Dış kalite ölçümü
- Yazılım ortamı : LM web ortamı
- Ek-E3 veri alanları (Reason, Created date)

**Ölçüm yöntemi:**

- Müşteri tarafından bildirilen değişiklik istekleri sayılır
- T4=Say[(01.01.2008≤Created date≤31.07.2009)-Hatalı durum değişikliği  
-İptal edildi -Kapsam dışı]

**Ölçüm birimi: Değişiklik**

**Özellik: Bakıma neden olan değişiklik isteklerinin sayısı**



**EK-F Temel Ölçüler (Devam)****Temel Ölçü 5: Düzeltilen hata sayısı (T5)**

İlişki:

- Kalite grubu : Dış kalite ölçümü
- Yazılım ortamı : LM web ortamı
- Ek-E4 veri alanları (Reason, Onay tarihi/Tamamlandı)

Ölçüm yöntemi:

- Düzeltilen hatalar sayılır
- $T5 = \text{Say}[01.01.2008 \leq \text{Onay tarihi} \leq 31.07.2009]$

Ölçüm birimi: Hata

Özellik: Bakım işleminin gerçekleştirilmesi sonucunda çözümlenen hata sayısı

**Temel Ölçü 6: Yapılan değişiklik sayısı (T6)**

İlişki:

- Kalite grubu : Dış kalite ölçümü
- Yazılım ortamı : LM web ortamı
- Ek-E3 veri alanları (Reason, Onay tarihi/Tamamlandı)

Ölçüm yöntemi:

- Yapılan değişiklikler sayılır
- $T6 = \text{Say}[01.01.2008 \leq \text{Onay tarihi} \leq 31.07.2009]$

Ölçüm birimi: Değişiklik

Özellik: Bakım işleminin gerçekleştirilmesi sonucunda çözümlenen değişiklik sayısı