

T.C
BAHÇEŞEHİR ÜNİVERSİTESİ

**ÇEVİK YAZILIM GELİŞTİRME SÜRECİNDE
KRİTİK BAŞARI FAKTÖRLERİNİN
BELİRLENMESİ VE ÖNCELİKLENDİRİLMESİNE
YÖNELİK BİR ÖRNEK ÇALIŞMA**

Yüksek Lisans Tezi

ERCAN DÖNMEZ

İstanbul,2009

T.C
BAHÇEŞEHİR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİ TEKNOLOJİLERİ PROGRAMI

ÇEVİK YAZILIM GELİŞTİRME SÜRECİNDE
KRİTİK BAŞARI FAKTÖRLERİNİN
BELİRLENMESİ VE ÖNCELİKLENDİRİLMESİNE
YÖNELİK BİR ÖRNEK ÇALIŞMA

Yüksek Lisans Tezi

ERCAN DÖNMEZ

Tez Danışmanı: DOÇ.DR.ADEM KARAHOCA
Yardımcı Tez Danışmanı: Öğr. Görv. DİLEK KARAHOCA

İstanbul,2009

T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİ TEKNOLOJİLERİ PROGRAMI

Tezin Adı: ÇEVİK YAZILIM GELİŞTİRME SÜRECİNDE
KRİTİK BAŞARI FAKTÖRLERİNİN BELİRLENMESİ
VE ÖNCELİKLENDİRİLMESİNE YÖNELİK BİR
ÖRNEK ÇALIŞMA

Öğrencinin Adı Soyadı: Ercan Dönmez

Tez Savunma Tarihi:

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğu Enstitümüz tarafından onaylanmıştır.

Enstitü Müdürü
Y.Doç.Dr. Tunç Bozbura

Bu Tez tarafımızca okunmuş, nitelik ve içerik açısından bir Yüksek Lisans tezi olarak yeterli görülmüş ve kabul edilmiştir.

Jüri Üyeleri

İmzalar

Unvanı, Adı ve SOYADI

Tez Danışmanı: Doç. Dr. Adem KARAHOCA

Prof.Dr. Nizamettin Aydın

Yard.Doç.Dr. Yalçın Çekiç

ÖNSÖZ

Yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç.Dr.Adem Karahoca'ya en içten dileklerle teşekkür ederim.

Bu çalışma boyunca her türlü desteğini ve koşulsuz güvenini esirgemeyen aileme de teşekkürü bir borç bilirim.

Ağustos, 2009

Ercan DÖNMEZ

İÇİNDEKİLER

ÖNSÖZ.....	5
ŞEKİLLER	10
KISALTMALAR.....	11
ÖZET	12
ABSTRACT.....	13
1. GİRİŞ	14
2. ÇEVİK YÖNTEMLER.....	18
2.1.ÇEVİK YÖNTEMLERİN FARKI.....	22
2.2 ÇEVİK YÖNTEM UYGULAMALARI	24
2.2.1 Ön Proje Analizi Uygulamaları.....	24
2.2.2 Proje Planlama Uygulamaları.....	25
2.2.3 İhtiyaç Mühendisliği Uygulamaları.....	26
2.2.4 Tasarım Uygulamaları.....	27
2.2.5 Kodlama Uygulamaları	29
2.2.6 Test Uygulamaları	31
2.2.7 Geçiş Uygulamaları :	32
2.2.8 Organizasyon Uygulamaları.....	33
2.2.9 Proje Yönetim Metrikleri	35
2.3 ÇY'LER VE CMM.....	35
2.4 ÇY'LER VE DÖKÜMANTASYON	36
2.5 ÇY'LERDE KULLANILAN ARAÇLAR	36
2.6. PROBLEMİN TANIMI.....	37
2.6.1. ÇYG'de Kritik Başarı Faktörleri	38
3. MALZEME VE YÖNTEM	45
3.1.MALZEME	45
3.2 YÖNTEM.....	46
3.3 ARAŞTIRMA VE YÖNTEM.....	52
3.3.1 Chang'in Bulanık AHP'de Mertebe Analizi Yönteminin Adımları	52
4. BULGULAR.....	55
4.1 ÖLÇÜTLER ARASINDA ÖNEM SIRALARININ BELİRLENMESİ.....	55
4.2 İKİLİ KARŞILAŞTIRMA MATRİSİNİN TUTARLILIK İNCELEMESİ	60
4.2.1.Kriter ikili Karşılaştırma Matrisinin Tutarlılığı.....	61
4.2.2 Bulanık Analitik Hiyerarşi Prosesi İçin İkili Karşılaştırmalar Matrisinin Bulunması	64
4.2.3 Bulanık AHP İçin Sentetik Mertebe Değerleri'nin Hesaplanması.....	69
4.2.4. Kriterler için ağırlık vektörünün hesaplanması.....	80
5. TARTIŞMA.....	82
5.1 VERİ ANALİZİ VE SONUÇLARI	82
5.2. ARAŞTIRMA SORULARINA CEVAPLAR	82
5.2.1. Araştırma Sorusu 1	83
5.2.2. Araştırma Sorusu 2	84
5.3. ARAŞTIRMA KISITLARI	86
6. SONUÇLAR VE ÖNERİLER.....	87
6.1. ÇYGP ÖNÜNDEKİ ENGELLER VE YAPILABİLECEKLER.....	91
KAYNAKLAR.....	94
ÖZGEÇMİŞ:	107

TABLOLAR

Tablo No	Açıklama	Sayfa
Tablo 2.1	ÇYGP'de KBF literatür araştırması	40
Tablo 2.2	1999-2002 yılları arasında Literatürdeki ÇYGP'de Kritik Başarı indikatör ve kriterleri	41
Tablo 2.3	2003-2006 yılları arasında Literatürdeki ÇYGP'de Kritik Başarı indikatör ve kriterleri	42
Tablo 2.4	2007 yılında Literatürdeki ÇYGP'de Kritik Başarı indikatör ve kriterleri	43
Tablo 2.5	2008 yılında Literatürdeki ÇYGP'de Kritik Başarı indikatör ve kriterleri	45
Tablo 2.6	Literatürde ÇYGP'de Başarısızlık indikatör ve kriterleri	45
Tablo 3.1	Araştırmada kullanılan likert ölçek	46
Tablo 3.2	Araştırmada kullanılan ana karar kriterleri ve geometrik ortalamaları	47
Tablo 3.3	Araştırmada kullanılan alt karar kriterler ve geometrik ortalamaları	48
Tablo 3.4	Süreçle ilgili alt ve detay karar kriterleri ve geometrik ortalamaları	49
Tablo 3.5	Organizasyonla ilgili alt v detay karar kriterleri ve geometrik ortalamaları	50
Tablo 3.6	Yazılım geliştirme ortamıyla ilgili alt ve detay karar kriterleri ve geometrik ortalamaları	51
Tablo 3.7	Alt yapı ilgili alt ve detay karar kriterleri ve geometrik ortalamaları	51
Tablo 4.1	Temel ölçeğine göre yapılan aralıklandırmalar	55
Tablo 4.2	Ana kriterler için ikili karşılaştırma tablosu	56
Tablo 4.3	Süreç kriterine ait alt kriterler için ikili karşılaştırma tablosu	56
Tablo 4.4	Organizasyon kriterine ait alt kriterler için ikili karşılaştırma tablosu	56
Tablo 4.5	Geliştirme Ortamı kriterine ait alt kriterler için ikili karşılaştırma tablosu	56
Tablo 4.6	Alt yapı kriterine ait alt kriterler için ikili karşılaştırma tablosu	56
Tablo 4.7	Takip Edilen Süreç detay kriterleri için ikili karşılaştırma tablosu	57
Tablo 4.8	İşin Kompleksliği detay kriterleri için ikili karşılaştırma tablosu	57
Tablo 4.9	Yapılan Tasarım detay kriterleri için ikili karşılaştırma tablosu	57
Tablo 4.10	Varolan Standartlar detay kriterleri için ikili karşılaştırma tablosu	58
Tablo 4.11	Çalışanların karakter özellikleri ve motivasyonu detay kriterleri için ikili karşılaştırma tablosu	58
Tablo 4.12	Hedef ve Beklentilerin Netliği detay kriterleri için ikili karşılaştırma tablosu	58
Tablo 4.13	Ekip içi ve müşteriyle etkin işbirliği detay kriterleri için ikili karşılaştırma tablosu	58
Tablo 4.14	İletişim Düzeyi detay kriterleri için ikili karşılaştırma tablosu	59
Tablo 4.15	Kullanılan Yazılım Çatısı detay kriterleri için ikili karşılaştırma tablosu	59
Tablo 4.16	Kullanılan Araçlar detay kriterleri için ikili karşılaştırma tablosu	59
Tablo 4.17	Kullanılan Algoritma detay kriterleri için ikili karşılaştırma tablosu	59
Tablo 4.18	Kullanılan Mimari detay kriterleri için ikili karşılaştırma tablosu	60
Tablo 4.19	Desteklenen Teknik Özellikler detay kriterleri için ikili karşılaştırma tablosu	60
Tablo 4.20	Desteklenen Foksiyonel Özellikler Kullanılan Algoritma detay kriterleri için ikili karşılaştırma tablosu	60
Tablo 4.21	Tesadüfilik göstergeleri	60
Tablo 4.22	AHP ye Göre Süreç ve Organizasyon Ana Kriterler İçin Göreli Ağırlık, Önem Sıra Tutarlılık ve Bilgileri	62
Tablo 4.23	AHP ye Göre Geliştirme Ortamı ve Altyapı Ana Kriterler İçin Göreli Ağırlık, Önem Sıra Tutarlılık ve Bilgileri	63
Tablo 4.24	Örnek alınan kriter bulanık karşılaştırma ölçek ve değerleri	64
Tablo 4.25	Ana Kriterlere ait bulanık ikili karşılaştırma matrisi	64
Tablo 4.26	Süreç bulanık ikili karşılaştırma matrisi	64
Tablo 4.27	Organizasyon bulanık ikili karşılaştırma matrisi	64
Tablo 4.28	Geliştirme Ortamına ait bulanık ikili karşılaştırma matrisi	65
Tablo 4.29	Altyapıya ait bulanık ikili karşılaştırma matrisi	65

Tablo No	Açıklama	Sayfa
Tablo 4.30	Takip edilen süreçte ait bulanık ikili karşılaştırma matrisi	65
Tablo 4.31	Kompleksliğine ait bulanık ikili karşılaştırma matrisi	66
Tablo 4.32	Yapılan tasarım ait bulanık ikili karşılaştırma matrisi	66
Tablo 4.33	Varolan standartlara ait bulanık ikili karşılaştırma matrisi	66
Tablo 4.34	Çalışanların motivasyonuna ait bulanık ikili karşılaştırma matrisi	67
Tablo 4.35	İşbirliği düzeyine ait bulanık ikili karşılaştırma matrisi	67
Tablo 4.36	İşletişim düzeyine ait bulanık ikili karşılaştırma matrisi	67
Tablo 4.37	Kullanılan Yazılım Çatısı'ne ait bulanık ikili karşılaştırma matrisi	67
Tablo 4.38	Kullanılan araçlara ait bulanık ikili karşılaştırma matrisi	68
Tablo 4.39	Kullanılan algoritmaya ait bulanık ikili karşılaştırma matrisi	68
Tablo 4.40	Varolan teknik özelliklere ait bulanık ikili karşılaştırma matrisi	68
Tablo 4.41	Varolan fonksiyonel özelliklere ait bulanık ikili karşılaştırma matrisi	68
Tablo 4.42	Ana Kriterlere Ait Bulanık puan toplamları ve tersleri	70
Tablo 4.43	Süreç Alt Kriterlerine Ait Bulanık puan toplamları ve tersleri	70
Tablo 4.44	Organizasyonel Alt Kriterlere Ait Bulanık puan toplamları ve tersleri	70
Tablo 4.45	Geliştirme Ortamı Alt Kriterlere Ait Bulanık puan toplamları ve tersleri	70
Tablo 4.46	Alt yapı Alt Kriterleri Ait Bulanık puan toplamları ve tersleri	71
Tablo 4.47	Takip Edilen Süreç Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	71
Tablo 4.48	İşin Kompleksliği Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	71
Tablo 4.49	Yapılan Tasarım Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	72
Tablo 4.50	Var olan Standartlar Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	72
Tablo 4.51	Çalışanların Karakter ve Motivasyonu Detay Kriterlerine Ait Bulanık Puan Toplamları ve Tersleri	72
Tablo 4.52	Hedef ve Beklentilerin Netliği Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	73
Tablo 4.53	Ekip içi ve müşteriyle etkin işbirliği Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	73
Tablo 4.54	İletişim Düzeyi Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	73
Tablo 4.55	Kullanılan Yazılım Çatısı Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	73
Tablo 4.56	Kullanılan Araçlar Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	74
Tablo 4.57	Kullanılan Algoritma Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	74
Tablo 4.58	Kullanılan Mimari Algoritma Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	74
Tablo 4.59	Desteklenen Teknik Özellikler Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	74
Tablo 4.60	Desteklenen Fonksiyonel Özellikler Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	75
Tablo 4.61	Ana Kriterlere ait Sentetik Mertebe Değerleri	75
Tablo 4.62	Süreç Alt Kriterine ait Sentetik Mertebe Değerleri	76
Tablo 4.63	Organizasyonel Alt Kriterine ait Sentetik Mertebe Değerleri	76
Tablo 4.64	Geliştirme Ortamı Alt Kriterine ait Sentetik Mertebe Değerleri	76
Tablo 4.65	Altyapı Alt Kriterine ait Sentetik Mertebe Değerleri	76
Tablo 4.66	Takip Edilen Süreç Detay Kriterlerine Ait Sentetik Mertebe Değerleri	77
Tablo 4.67	İşin Kompleksliği Detay Kriterlerine Ait Sentetik Mertebe Değerleri	77
Tablo 4.68	Yapılan Tasarım Detay Kriterlerine Ait Sentetik Mertebe Değerleri	77
Tablo 4.69	Var olan Standartlar Detay Kriterlerine Ait Sentetik Mertebe Değerleri	77
Tablo 4.70	Çalışanların karakter ve motivasyonu Detay Kriterlerine Ait Sentetik Mertebe Değerleri	78
Tablo 4.71	Hedef ve Beklentilerin Netliği Detay Kriterlerine Ait Sentetik Mertebe Değerleri	78

Tablo No	Açıklama	Sayfa
Tablo 4.72	Ekip içi ve müşteriyle etkin işbirliği Detay Kriterlerine Ait Sentetik Mertebe Değerleri	78
Tablo 4.73	İletişim Düzeyi Detay Kriterlerine Ait Sentetik Mertebe Değerleri	78
Tablo 4.74	Kullanılan Yazılım Çatısı Detay Kriterlerine Ait Sentetik Mertebe Değerleri	78
Tablo 4.75	Kullanılan Araçlar Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	79
Tablo 4.76	Kullanılan Algoritma Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	79
Tablo 4.77	Kullanılan MimariAlgoritma Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	79
Tablo 4.78	Desteklenen Teknik Özellikler Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	79
Tablo 4.79	Desteklenen Fonksiyonel Özellikler Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri	79
Tablo 4.80	Normalize edilmiş ağırlıklar tablosu	81

ŞEKİLLER

Şekil No	Açıklama	Sayfa
Şekil 2. 1	ÇM' de Amaç	20
Şekil 2. 2	Yinelemeli ÇY aşamaları	23
Şekil 2. 3	Çevik ve gelenelsel yöntemlerde zamana göre değişim maliyeti	24
Şekil 2. 4	Yayım Planı	26
Şekil 2. 5	Çevik Süreçlerde Yayım ve Yineleme Planı	26
Şekil 2. 6	Test GÜdümlü Geliştirme Süreci	30
Şekil 3. 1	Araştırmada kullanılan karar kriterleri	46
Şekil 3. 2	Bulanık M_2 sayısının M_1 sayısından büyük olabilirlik derecesi	53
Şekil 5.1	Karar Kriterlerinin öncelikleri	84

KISALTMALAR

Kısaltma	İngilizcesi	Türkçesi
YGP	Software Development Projects	Yazılım Geliştirme Projeleri
BT	Information Technologies	Bilgi Teknolojileri
YGO	Software Development Organisations	Yazılım Geliştirme Organizasyonları
YG	Software Development	Yazılım Geliştirme
YGS	Software Development Process	Yazılım Geliştirme Süreçleri
ÇYG	Agile Software Development	Çevik Yazılım Geliştirme
ÇM	Agile Manifesto	Çevik Manifesto
ÇYGP	Agile Software Development Projects	Çevik Yazılım Geliştirme Projeleri
ÇYGY	Agile Software Development Methods	Çevik Yazılım Geliştirme Yöntemleri
KBF	Critical Success Factors	Kritik Başarı Faktörleri
BAHP	Fuzzy Analytic Hierarchy Process	Bulanık Analitik Hiyerarşi Prosesi
ÇY	Agile Methods	Çevik Yöntemler
UP	eXtreme Programming(XP)	Uç Programlama
UYG	Adaptable Software Development	Uyarlanabilir Yazılım Geliştirme
ÖGG	Feature Driven Development	Özellik Güdümlü Geliştirme
DYGM	Dynamic Software Development Model	Dinamik Yazılım Geliştirme Modeli
TGG	Test Driven Development	Test Güdümlü Geliştirme
RBS	Rationale Unified Process	Rational Birleşik Süreç
IDE	Integreted Development Environment	Entegre Geliştirme Ortamı
CMM	Capability Maturity Model	Yetenek Olgunluk Modeli
CMMI	Capability Maturity Model Integration	Tümleşik Yetenek Olgunluk Modeli
DOD	Department of Defense	Amerikan Savunma Bakanlığı
IEEE	The Institute Of Electrical and Electronics Engineers	Elektrik Elektronik Mühendisleri Enstitüsü
UML	Unified Modeling Language	Birleşik Modelleme Dili
OMG	Object Management Group	Nesne Yönetim Grubu
RUP	Rational Unified Process 5.0	Rational Birleşik Süreç
SEI	Software Engineering Institute	Yazılım Mühendisliği Enstitüsü

ÖZET
ÇEVİK YAZILIM GELİŞTİRME SÜRECİNDE
KRİTİK BAŞARI FAKTÖRLERİNİN BELİRLENMESİ VE
ÖNCELİKLENDİRİLMESİNE YÖNELİK BİR ÖRNEK ÇALIŞMA

Dönmez, Ercan

Fen Bilimleri Enstitüsü Bilgi Teknolojileri Pprogramı

Tez Danışmanı: Doç. Dr. ADEM KARAHOCA

Ağustos, 2009, 107 Sayfa

Modern dünyanın tüm alanlarında yazılım çok önemliken, yazılım geliştirme süreci ele alındığında, mükemmel olmadığı görülür. Son zamanlarda yazılım geliştirmeyi geleneksel yöntemlere göre yeni ve farklı şekilde yapan, çevik yazılım geliştirme yöntemleri ortaya çıkmıştır. Çevik yazılım geliştirme süreci, iş süreçlerini karmaşa ve detayları azaltarak desteklemekte böylece hızlı, kolay ve etkin olarak geliştirmeyi hedeflemektedir. Fakat çevik uygulamalarının, geleneksel yöntemlere göre ne kadar etkin ve verimli olduğu ve başarı faktörlerinin neler olduğu pek bilinmemektedir. Çevik yöntemleri kullanan yazılım geliştirme projelerinde başarı hakkında birtakım bulgular olmasına rağmen bu konudaki araştırmalar akademik çevrelerde hala kısıtlı kalmaktadır. Bu tezde kantitatif yaklaşımla, çevik yazılım geliştirmede başarı için değişik öngörüler kullanarak kritik başarı faktörleri ve çevik yazılım geliştirme projelerindeki belli başlı temel uygulamaların kullanımı hakkında bütünleştirilmiş bir resim sağlanması amaçlanmıştır. İlk olarak mevcut literatür ve endüstri deneyimlerine dayanarak, potansiyel kritik başarı faktörleri ve çevik yazılım geliştirme projelerindeki temel uygulamalar tanımlanıp, bir liste halinde derlenmiştir. Daha sonra, bir anket yürütülerek, bu uygulamaları değişik durumlarda faydalı kılan belirli durumlar gözden geçirilerek, Bulanık Analitik Hiyerarşi Prosesi teknikleriyle ağırlıklandırılıp ve önceliklendirilmiş potansiyel kritik başarı faktörleri listesi hazırlanmıştır. Sonuçlar çevik yazılım geliştirme için yedi kritik başarı faktörü olduğunu göstermiştir. Bunlar; a)Sürekli kod entegrasyonu sağlayan entegre geliştirme ortamlarının kullanımı, b)Zengin ve hızlı geliştirme alt yapısı sunan yazılım çatılarının kullanımı, c)Veritabanı entegrasyonu sunan yazılım çatılarının kullanımı, d)Ön mimari modelleme, e)Ön ihtiyaç modelleme , f)Bakım yapılabilir tasarım, g)Evrimsel tasarımıdır. Bu anket Avrupa çapında değişik ülkelerde, değişik projelerde yer almış yazılım geliştirme profesyonelleri arasından toplanan veriler dikkate alınarak hazırlanmıştır. Araştırma sonuçlarına göre, uygulayıcılara kendi yazılım geliştirme süreçlerini geliştirebilmeleri için bazı önerilerde bulunulmuştur. Bu tezin ana bulgularının belli bir uygulamasını uyarlarlarken bütün çevre ve şartlar önemli olduğu dikkate alınmalıdır. Bazı durumlar için, bazı uygulamaların amaca uymayabileceği anlaşılması gerekir. Fakat, belli uygulamaların özel kusurları diğer uygulamaların kombinasyonlarıyla azaltılabilir hatta ortadan kaldırılabilir.

Anahtar Kelimeler: Çevik Yazılım Geliştirme, Kritik başarı faktörleri, Çevik pratikler, Bulanık AHP.

ABSTRACT
DETERMINATION AND PRIORITIZATION OF
THE CRITICAL SUCCESS FACTORS
AGILE SOFTWARE DEVELOPMENT PROCESS: A CASE STUDY

Dönmez, Ercan
The Institute of Science, Information Technologies Graduate Program
Supervisor: Assoc. Prof. Dr. Adem Karahoca

August, 2009, 107 Pages

While software is so important for all facets of the modern world, software development itself is not a perfect process. Agile software development methods have recently emerged as a new and different way of developing software as compared to the traditional methodologies. Agile software development aims at fast, light and effective development that supports customer's business without being chaotic or rigorous. However, little is known about how effective and efficient Agile practices are over the traditional methodologies, and what their success factors are. There have been several disparate anecdotal evidences about the success of software development projects using agile methodologies and research in this subject is still scant in the academic circles. In this thesis, we aimed to provide a consolidated picture of the different predictors of agile software development success and investigated the critical success factors and usage of certain core practises of agile software development projects using quantitative approach. First of all, based on existing literature and industry experiences, a preliminary list of potential critical success factors and core practises of agile projects were identified and compiled. After that, by conducting a survey, we examine what makes these practices beneficial for certain situations and prepared weighted, prioritized list of possible critical success factors by using Fuzzy Analytical Hierarchy Process techniques. The results revealed that identifying seven critical success factors for agile software development projects: a) Using integrated development environments that can provide continuous code integration, b) Using Frameworks that can provide rich and rapid development infrastructure, c) Using Frameworks that can provide database integration, d) Pre-architectural modeling, e) Preliminary requirement modelling, f) Design for maintenance, g) Evolutionary design. This survey was conducted among software development professionals, gathering survey data from different kind of projects and different countries across the Europe. Based on the research results, we finally set up some recommendations for practitioners to reflect upon and improve their own software development process. The main findings of this thesis are important to consider the whole context when implementing a certain practice. For some contexts, certain practices do not fit for the purpose and this has to be realized. However, certain shortcomings of a specific practice might be reduced or even eliminated if implemented in combination with other practices.

Keywords: Agile software development, critical success factors, agile practices, Fuzzy AHP.

1. GİRİŞ

Yazılım, günümüzün modern ve rekabetçi dünyasında çeşitli kişiler ve kurumlar için çok önemli bir role sahip olmasına karşılık, yazılım geliştirme sürecine baktığımızda bunun mükemmel bir süreç olmadığı (Chow & Cao 2007) ve 1950li yıllardan bu yana geliştirilmeye devam edilmesine ve yazılım mühendisliği metodolojilerinin uygulanmasına karşın çoğu Yazılım Geliştirme Projeleri (YGP)'nin müşterinin istediği kalitede ve zamanında teslim edilemediği (Kotonya & Sommerville 1998) bilinmektedir. Problemlerin ancak ileriki aşamalarda tespit edilebilmesi ve bunu düzeltme maliyetinin yüksek olması, proje için yapılan planın bir süre sonra sapmaya başlaması ve aradaki kaybı gidermek için fazla mesai yapılmak zorunda kalınması sebebiyle hata riski ve ekip motivasyonunun giderek düştüğü gözlemlenir. Ekip içi etkin bilgi paylaşımı olmaması dolayısıyla, proje ekibinden ayrılan kişiler proje açısından risk oluşturması kullanıcıların gerçek yaşamdaki ihtiyaçlarını zor olarak karşıladığı, kullanıcılardan yazılımın kullanımının zor olması dolayısıyla sık sık şikayet etmesi (Kotonya & Sommerville 1998), geliştiricilerin yazdıkları kaynak koddan, kodun güncellenmesi durumunda yaşatılmasının güçlüğünden dolayı tatmin olmadıkları (Hall 2001) ortaya çıkar. Proje sponsorlarının yazılım zaman ve bütçesinin aşılması dolayısıyla hayal kırıklığına uğradığı (Klappholz ve diğ. 2003), hatta önceden yürürlüğe konulan yazılım projelerini bile devam eden pahalı bakım ya da düzeltme maliyetlerinden dolayı askıya aldıkları ya da iptal edilebildiği görülmektedir(Chow & Cao 2007).

Yukarıda bahsedilen eksikler genelde Bilgi Teknolojileri(BT)ni ve özelde Yazılım Geliştirme Organizasyonları(YGO)'nı önemli ölçüde etkilemektedir. Buradaki temel soru Yazılım Geliştirme(YG)'yi bahsedilen sorunların getirdiği israftan ve verimsizlikten nasıl kurtarılacağı ve Yazılım Geliştirme Süreçleri(YGS)'nin nasıl iyileştirilebileceğidir.

Geçen 50 yıldan fazla zaman içerisinde YGS'nin karmaşıklığını giderici ve disipline edici birçok yöntem bulunmuştur. Bunların ilk türü; plan yönelimli, tüm ihtiyaçların detaylı olarak belgelendiği ve geliştirme adımlarında belli bir sürece sıkıca bağlı

kalmayı zorunlu kılan bürokratik yöntemler, ağır ya da geleneksel yöntemler olarak adlandırılmaktadır.

Fakat geleneksel yöntemler YG ile ilgili yukarıda bahsedilen sorunlara çözüm bulmada iş dünyasındaki hızlı değişime hızlı tepki vererek değişikliklere adapte olmada yetersiz kalmış ve 2001 yılında 17 yöntem bilimci, eski tecrübe, başarı ve başarısızlık deneyimlerinden yola çıkarak, geleneksel yöntemlerin olumsuzluklarını gidermek için Yazılım Geliştirmedeki gelecek trendleri tartışmış ve Çevik Yazılım Geliştirme(ÇYG) ile ilgili pratik, bürokrasiden uzak bir bakış açısı ile hafif ve yeterli olarak gördükleri ortak prensiplerini Çevik Manifesto(ÇM) adı altında ilan etmişlerdir (Fowler 2001).

Bu manifestoyu imzalayanlar Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas'tır.

Bundan sonra 2005 yılında ise Alistair Cockburn ve Jim Highsmith yönetim uzmanlarını toplayarak bu manifestoya bir ek yazmıştır. Dayanışma bildirgesi (Declaration of Interdependence) adı verilen bu manifesto 6 madde ile Çevik Manifesto'yu günümüze daha iyi uyarlanabilir hale getirmişlerdir.

ÇM, ÇYG'nin ideolojik altyapısını oluşturmaktadır. Cockburn(2001)'a göre, ÇYG'nin çekirdeği, insan ve iletişim odaklı kuralların, hafif fakat yeterli kurallarla proje adımlarında kullanımınıdır. Fowler(2001)'a göre Çevik ve Geleneksel yöntemler arasındaki en önemli fark olarak geleneksel yöntemlerin metotlar üzerinde durması buna karşılık Çevik Yöntemlerin dökümantasyondan çok çalışan kod üzerinde odaklanmasını vurgulamaktadır. Fowler(2001) aynı zamanda, geleneksel yöntemlerin, değişim karşısında dirençli olabilmeye karşılık *çevik yöntemlerin* değişikliğe daha açık olduğunu belirtmektedir.

Coldeway ve diğ.. (2000) göre “çoğu hafif ya da ÇY'lerdeki kişilerarası iletişim ve takım çalışmasının ağır veya geleneksel yöntemlerdeki yoğun dökümantasyon ve biçimselliğin yerini almaktadır.” (Coldeway ve diğ. 2000,131).

Çevik Yazılım Geliştirme Yöntemleri(ÇYGY) yazılımın müşteriyle yakın işbirliği içerisinde uyarlanabilir bir şekilde, hızlıca geliştirilmesini ve uygulamaya alınmasını hedeflemektedir. Böylece iş dünyasındaki değişikliklere karşı çabuk tepki vermesi ve aynı zamanda verimlilik ve etkinliğin sağlanabilmesi sebebiyle YGO tarafından giderek artan oranda kullanılmakta ve kabul görmektedir.

Fakat her organizasyonun kendine özgü yapısı ve farklılıkları sebebiyle uygulamadaki bir takım değişiklikler sebebiyle Çevik Yazılım Geliştirme Projelerinde (ÇYGP) de başarılı ve başarısız örnekler görülebilmektedir.

Bu tez kapsamında, ÇY ilgili literatür tarandıktan sonra ÇYGP’de başarıyı etkileyen ve çeşitli organizasyonlar için ortak nitelik taşıyan Kritik Başarı Faktörleri(KBF), detaylara inen bir anket araştırmasıyla, 4 ana katagoriye ve 22 alt kategoriye ayrıldıktan sonra uygulamada kullanılan 92 ölçüt için, 5-10 yıl arasında yazılım geliştirme projelerinden deneyimli 6 uzmanın görüşleri alınarak puanlanmış ve anket verileri Bulanık Analitik Hiyerarşi Prosesi analiz edilmiştir.

Yapılan analizde ;

a) ”ÇYGP’de başarı için öncelikle dikkat edilmesi gereken faktörlere ve alt faktörler nelerdir?”

b) ”Belirlenen her faktör için ve genel başarı için kritik roldeki en önemli çevik uygulamalar ve bunların öncelik sıraları nelerdir?”

sorularının cevaplanabilmesi için, tüm kriterler ile alt kriterler ve bunların uygulamaları, birbiriyle karşılaştırmalı olarak incelenmiş ve ağırlıklandırılarak önceliklendirilmiştir.

Bu çalışma ile gerek yazılım mühendisliği ve yazılım süreçlerinin iyileştirilmesi ile ilgili araştırmacı ve yazılım firmalarına, gerekse başarılı yazılım geliştirme ve verimlilik için kritik başarı faktörleri hakkında yol gösterici olması ümit edilmektedir. Tezin akışı aşağıdaki gibidir: Tezin ilk bölümünde, tez konusuyla ilgili ön bilgiler verilmiş, ikinci bölümde çevik süreçler ve uygulamaları hakkında ve ÇYGY ve KBF ile ilgili teorik nitelikteki literatür bilgileri verilmiştir. Sonrasında uygulama çalışmasında kullanılan

Bulanık Analitik Hiyerarşi Prosesi(BAHP) yönteminin anlatıldığı üçüncü bölüm izlemiştir. Dördüncü bölümde, yapılan anketten elde edilen veriler doğrultusunda BAHP yöntemiyle en önemli KBF belirlenmesi uygulanması adım adım anlatılmıştır. Beşinci bölümde ise, BAHP yöntemiyle yapılan analiz sonucunda belirlenen bulgular ve araştırma kısıtları tartışılmıştır. Altıncı ve son bölümde de, elde edilen sonuçlar bu konuda çalışma yapan araştırmacılar ve yazılım geliştirme organizasyonlarının bilgisine öneri olarak sunulmuştur.

2. ÇEVİK YÖNTEMLER

Günümüzün küresel rekabet ortamında yazılım şirketleri diğer şirketlerde olduğu gibi giderek daha talepkar olan müşterilere hizmet vermekte ve müşterilerinin iyi kalite, düşük fiyat ve kısa teslim süresi beklentilerini hızla karşılayabilmek için değişik isteklere uyum sağlamak zorunda kalmaktadırlar.

Çevik yöntemler(ÇY)'lerin benimsediği ve Çevik Manifesto ile özetlenen ilkeler ve değerler bütünü, yazılım geliştirme dünyası için yukarıda belirtilen müşteri beklentilerine cevap vermek için çare olarak son yıllarda artan oranda kullanılmaya başlanan yeni bir yaklaşım olarak karşımıza çıkmaktadır. Yalın düşünce, müşteriye değer sağlamayan faaliyetleri; zaman, kaynak ve para israfı olarak görür ve sürecin etkinleşmesi için israfın yok edilmesini gerektirir.

Temel ilke olarak ;

- Bireyler ve onlar arasındaki etkileşimi süreç ve araca tercih etmek,
- Çalışan bir yazılımı detaylı bir dokümantasyona tercih etmek,
- Müşteri ile işbirliğini, anlaşma görüşmelerine tercih etmek,
- Değişikliklere istenildiği anda cevap verebilmeyi sınırları belirli bir plana tercih etmek.

Agile Manifesto (2001)'e göre ÇM'in başlıca ilkelerini aşağıdaki gibi sıralayabiliriz:

1. Hızlı, devamlı ve kullanışlı yazılım geliştirerek müşteri memnuniyeti sağlamak amaçlardandır. Bu, projenin ilk aşamalarından itibaren sürekli kod teslimleri ile yapılır ve müşterinin yazılımı çok önceden kullanmaya başlayarak değer sağlamasına olanak sağlar. Günümüzde ÇY'lerin yaygınlaşmasının başlıca nedenlerden biri, yapılan yatırımların hızlı geri dönüşünün olmasıdır .

2. Değişiklik talepleri, projenin ilerleyen aşamalarında oluşsa dahi kabul edilir.

Amaç müşterinin ihtiyaçlarını karşılayan, onlara yarar sağlayacak, gerçek değer katacak yazılımı geliştirmektir ve ihtiyaçlarda meydana gelen değişiklikler projenin sonraki aşamalarında dahi yazılıma aksettirilmelidir. Test güdümlü geliştirme, kapsamlı

otomatik testler, sürekli entegrasyon, basit tasarım, kodun yeniden düzenlenmesi(refactoring) gibi uygulamalar sayesinde değişikliklerin getireceği maliyetler minimuma indirilir ve süreç değişikliklere daha hızlı uygun hale getirilir.

3. Sık sık çalışan yazılım teslim edilmelidir. Bu aralıklar tipik olarak 1-4 hafta arasındadır. Bu sayede sürekli geri beslenim sağlanır ve müşterinin istekleri doğrultusunda yazılım evrimleşerek gelişir.

4. Geliştiriciler ile diğer alan uzmanları arasında günlük ve yakın işbirliği bulunmalıdır. Farklı roller arasında duvarlar örülmez. Rol bazlı ekipler yerine yazılım özelliklerine göre ekipler oluşturulur. Testçi, analist, yazılım geliştirici aynı ekibin içinde çalışır ve sürekli iletişim halindedir.

5. Projeler motive bireyler çevresinde kurulur. Ekip kendi kendine organize olacak yetkiye sahiptir.

6. Yüz yüze görüşme iletişimin en güzel yoludur.

7. Çalışan yazılım parçaları, gelişimin en önemli ölçüsüdür.

8. Sürdürülebilir bir hızı sağlamaya çalışır. Planlamaların sağlıklı olması için ekibin iş teslim hızının güvenilir olması gerekir. Örneğin fazla mesailer gibi yöntemlerle ekibin hızını geçici olarak arttırmak tercih edilen yöntemlerden değildir.

9. Teknik mükemmelliğe katılım ve iyi tasarım çevikliği geliştirir. Teknik açıdan mükemmel, sade çözümler oluşturulmasına özen gösterilir. En iyi tasarım çabuk genişleyebilen tasarımdır

10. Basitlik önemlidir. Sadelik anlayışı akla gelen ilk baştan savma çözümü uygulamak yerine anlaşılması ve sonradan değiştirilmesi kolay, maliyeti en düşük ve o anki gereksinimleri karşılayan çözümü kullanmaktır.

11. En iyi mimariler, ihtiyaçlar ve tasarım kendi kendine organize ekiplerce ortaya çıkarılır.

12. Ekip kısa sürelerle toplanır, çalışma yöntemlerini gözden geçirir. Daha etkin ve etkili çalışmak için geçmişi kapsayan (retrospective) biçimde yapılan toplantılarla gözden geçirir.

ÇY'lerde amaç, operasyonel maliyetleri ve döngü zamanını kısaltıp, kaliteyi arttırmaktır.



Şekil 2.1. ÇM' de Amaç(Agile Manifesto 2001)

ÇY'ler yukardaki manifestoyu kabul eden ve çalışma yöntemlerini çevik bakış açısıyla oluşturulmuş süreçlerdir. Başlıca ÇY'ler şunlardır:

Uç Programlama(UP)-eXtreme Progrmming(XP): Kent Beck tarafından tanımlanan ve son yıllarda oldukça fazla ilgi duyulan ve popüler olan çevik yöntemlerden biridir. UP genel olarak basitlik, iletişim, geribildirim ve cesaret temelleri üzerine kurulmuş olan bir yazılım geliştirme yöntemidir. Detayda ise planlama oyunu, sık sürümler, metafor, basit tasarım, test, yeniden düzenleme, ikili kodlama, ortak sahiplik, sürekli entegrasyon, 40 saat çalışma, müşteri katılımı, kodlama standardı, açık iş alanı gibi uygulamalara sahiptir (Stojanovic ve diğ., 2004).

Uyarlanabilir Yazılım Geliştirme(UYG): UYG, Jim Highsmith tarafından karmaşık ve geniş ölçekli sistemler için geliştirilmiştir. Yöntem artırımlı, tekrarlayan ve değişmez prototiplemeyi kuvvetli bir şekilde cesaretlendirmektedir. Yöntem yeterli destekle projeleri kaosa düşmekten koruyacak ve bir çerçeve sağlamak amaçlanmış ancak bunu yaratıcılığı baskı altına almadan yapması hedeflenmiştir. Bu süreç üç aşamalı çevrimden oluşmaktadır. Bunlar sırasıyla, tahminde bulunma, işbirliği ve öğrenmedir. Artırımlı geliştirme, özellik tabanlı planlama ve müşteri odaklı grup gözden geçirmeleri gibi uygulamaları bulunmaktadır (Abrahamson ve diğ., 2002).

Çevik Modelleme: Çevik Modelleme, Scott W. Ambler tarafından Uç Programlama değerleri göz önüne alınarak geliştirilmiş ve içine alçak gönüllüğün eklenmesi ile son halini almıştır. Altında yatan fikir, geliştiricileri yeterince gelişmiş fakat mümkün olduğunca az model üretmeye cesaretlendirerek belirgin tasarım problemlerinde ve dökümantasyon için desteklemektir. İletişim, Basitlik, Geribildirim, Modellemeyi takım

halinde yapmak, Modelinizi doğru kişilerle inceleme, Modelin uygulanması, Kabul testleri, Cesaret, Alçak Gönüllülük en önemli değerleridir (Ambler &Jeffries, 2002).

Scrum: Scrum, Ken Schawaber tarafından geliştirilen “çevik” süreçtir. Scrum, tasarım ve uygulamaya alma adımları için yazılım geliştirme teknikleri tanımlamak yerine sürekli olarak değişen ortamlarda takım üyelerinin sistemi geliştirmedeki fonksiyonlarının nasıl olması gerektiğiyle ilgilenir (Stojanovic. ve diğ., 2004). Scrumda ana fikir, sistem geliştirme süreci çevresel ve teknik değişkenlerin (ihtiyaçlar, zaman, kaynaklar ve teknoloji) süreç boyunca değişebileceğidir. Bu da değişikliğe karşı cevap verebilmek içinse esnekliği zorunlu kılmaktadır. (Abrahamson ve diğ., 2002). Ön oyun , geliştirme ve oyun sonrası olarak üç aşaması vardır. Scrum’da kendi kendini düzenleyen, yönlendiren takım vardır. Scrum, geliştirmenin 30 günlük çevrimler halinde tamamlanması gereken ve 10 kişiden küçük takımlar için uygun bir yaklaşımdır (Stojanovic ve diğ., 2004).

Özellik Güdümlü Geliştirme(ÖGG) : Sık sık somut bir şekilde çalışan yazılımlar üretilmesi fikrine dayanan yazılım geliştirme sürecidir. Tüm modeli geliştirme, özellik listesi oluşturma, özellik bazlı plan, özellik bazlı tasarım, özellik bazlı geliştirme adımlarından oluşur. Bir özellik, müşteri bazlı <eylem>, <sonuç>, <nesne> formunda kullanıcı ihtiyaçlarından küçültülerek oluşturulmuş fonksiyonlardır. Bunlar önem düzeyine göre düzenlenerek sıralanmalı ve en fazla iki haftalık periyotta tamamlanmalıdır. Bu yaklaşım diğer çevik yaklaşımların aksine çok kritik olan ve 50 ile 250 kişilik projelerde UML kullanılarak uygulanmaktadır (Stojanovic. ve diğ., 2004).

Dinamik Yazılım Geliştirme Modeli (DYGM): DYGM İngiltere’de, DYGM konsorsiyumu tarafından geliştirilen hızlı yazılım geliştirmeye yönelik bir çerçevedir. UP, Scrum gibi süreçlerin işleyişi ile karşılaştırıldığında daha fazla detay içerir ve genelde devlet ihaleleri gibi sabit periyotlu yazılım projelerinde sıklıkla kullanılır. Yazılım geliştirme süreci boyunca aktif kullanıcı katılımı, sık sürümler, güçlendirilmiş takımlar ve test gibi prensiplere sahiptir. Fonksiyonel yineleme, tasarım ve geliştirme yinelemesi ve uygulamaya alma gibi yineleme süreçlerinden oluşur. DYGM açıkça

tanımlanmayan ve sabit olmayan problemi küçük çevrimli seri prototiplerle çözmek amaçlanır. Karmaşık dökümantasyon yerine kısa amaçlar listesi ve kalite kriter listeleri tercih edilir (Stojanovic. ve diğ., 2004).

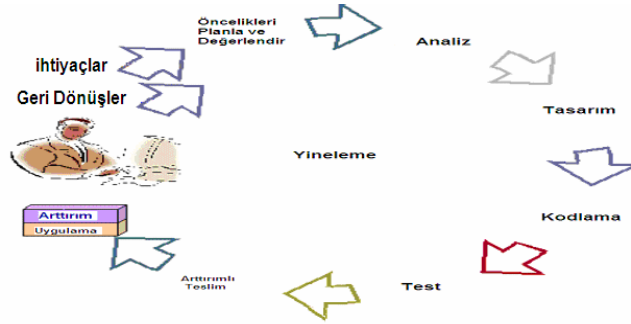
Kristal: Önem durumu ve proje büyüklüğüne göre değişik yöntemlerin uygulandığı bir süreçtir. Önem seviyesi ve büyüklük arttıkça daha ağır yöntemler kullanılır. Buradaki yöntemlerin hepsi en fazla 4 ay, en az 1 ile 3 ay arasında olması tercih edilen arttırımlı çevrimlerden oluşurlar. İletişim ve insanlar arası işbirliği vurgulanmaktadır. Bu süreçlerde duruma göre diğer geliştirme yöntemlerinin uygulamalarının ya da araçlarının kullanımı konusunda bir limit yoktur(Cockburn 2002). UP'deki senaryolar/kullanım durumları, ekran taslakları ve tasarım taslaklarını kolayca kendine uyarlayabilir(Stojanovic ve diğ., 2004).

Bu süreçlerden hangisini kullanalım sorusunun cevabı hiçbiri olacaktır. Önemli olan Çevik Manifesto ile özetlenen bakış açısını ve beraberinde sürekli iyileştirme odaklı kültürel değişikliği kavramaktır. Bu süreçler ancak başlangıç noktası olabilir. Proje ilerledikçe ekip süreci kendisine adapte etmeli ve süreç sürekli iyileştirilmelidir. Çevik süreçlere ait detaylara inmeden önce klasik yöntemlerle arasındaki belirgin farkları incelemek yerinde olacaktır.

2.1.ÇEVİK YÖNTEMLERİN FARKI

Çevik yöntemler, proje tarafları arasında iletişimi cesaretlendirerek, yoğun değişikliklere karşı pazara kısa sürede sürüm, yatırım geri dönüşünde artış ve yüksek kalite sağlayan, çalışan koda odaklandığı kadar ihtiyaçlardaki değişikliklere büyük ölçüde tepki verebilen, yinelemeli ve arttırımlı geliştirme yapabilmeyi vaat etmektedir. Diğer tarafta geleneksel yöntemler iş ihtiyaçlarını çalışan bir yazılıma dönüştürmek için titiz modelleme ve açık bir planı tavsiye eder. Plan güdümlü olarak da anılan bu yöntemlerde, proje başlangıcında tüm ihtiyaçlar büyük ölçüde durağandır. Böylece geliştirme süreci boyunca sabitlenmiş ve iyi tanımlanmış detaylı planlar izlenebilmektedir (Boehm 2002).

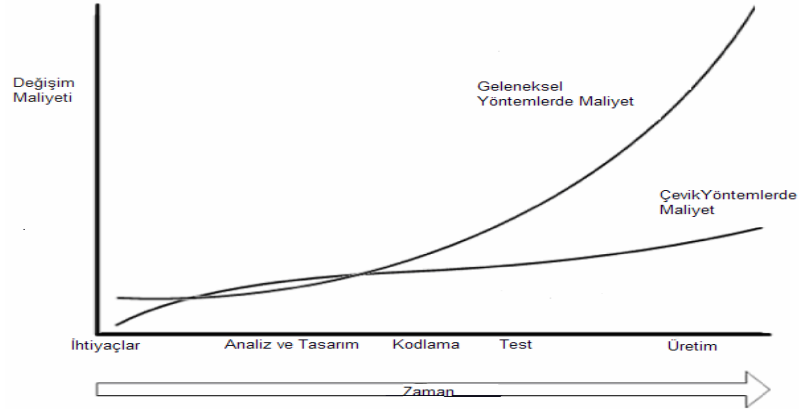
Yazılım geliştirme sürecini; analiz, tasarım, kodlama, test, uygulamaya alma ve bakım olarak aşamalara ayırırsak her iki yöntemde de yazılım geliştirme süreci boyunca ortamdaki belli değişikliklerle baş başa kalınmaktadır. Ancak değişikliğe yaklaşım, çevik süreçlerde projenin başından sonuna aşamasına kadar, ihtiyaçlardaki olası her türlü değişiklikler dikkate alınıp, bir an önce en acilden başlamak üzere müşteri ihtiyaçları küçük yinelemeler halinde hızlı bir şekilde karşılanmaya çalışılmakta eğer sorun var ise yeni bir sürümle hızlı bir şekilde giderilmektedir.



Şekil 2.2 Yinelemeli ÇY aşamaları,
(<http://www.indicthreads.com/images/stories/contentrelated/agile-steps-process.gif>)

Böylece çevik süreçlerle, komple bir sistemi baştan aşağı yeniden gözden geçirmek yerine sadece en önemli parçadaki sorun gözden geçirilerek proje toplamda daha kısa sürede ve daha az maliyetle tamamlanabilmekte ve verimlilik artışı sağlanabilmektedir. Müşteri açısından bakıldığında ise çevik yöntemlerde yazılım hızlı kullanıma alındığı için için yapılan yatırım geri dönüşü de o kadar hızlı olmaktadır.

Geleneksel yöntemlerde ise, müşterinin değişen ihtiyaçlarını karşılamak projenin başlangıç aşamalarında kısıtlı da olsa mümkünken ilerleyen aşamalarda pek mümkün olamamaktadır. Çünkü ihtiyaçlarla ilgili uzun bir analiz, tasarım ve dökümantasyondan sonra geliştirmeye başlanırken ve müşteriden gelecek yeni değişiklik taleplerinin, değişiklik gelene kadar, geçen süreçlerde yapılan işlemleri tekrar gözden geçirip, gerektiğinde en baştan yapmayı gerektirmektedir. Buda tüm proje takviminin sarkarak uzun bir zaman almasını ve dolayısıyla da maliyetinin zaman ilerledikçe artarak başta belirlenen çok fazla olması anlamı taşımaktadır. Her iki yaklaşımın zaman içerisindeki maliyet değişimi aşağıdaki şekil de görülmektedir



Şekil 2.3 Çevik ve geleneksel yöntemlerde zamana göre değişim maliyeti (Agile Advisor, 2009)

2.2 ÇEVİK YÖNTEM UYGULAMALARI

ÇYG Yöntemlerinde kullanılan her uygulama pratiğinin farklı bir sorunun çözümüne odaklanmakta olup, bunların refaktöring yada regresyon testi (Brooks F. P, 1995) gibi bir çoğu Çevik Yöntemlerden önce bulunmuşlardır. ÇYG yöntemleri eski yöntemlerin kombinasyonu yada değiştirilmesiyle oluşturulmuşlardır (Dogs & Klimmer, 2004).

Aşağıda çevik yöntemlerde kullanılan uygulamalar hakkında özet bilgiler süreç adımlarına göre gruplandırılmış olarak bulunabilir.

2.2.1 Ön Proje Analizi Uygulamaları

Fizibilite Analizi: Fizibilite Analizi DYGM'nin yazılım geliştirme projesinin ilk adımıdır. Planlanan proje hedeflerine erişilip erişilemeyeceğinin araştırılması yapılır ve bu bilgiye göre prototiplerin hazırlanması yada uzmanlara danışılabilir ve birkaç haftayı geçmez (Dogs & Klimmer, 2004).

Yazılım Geliştirme Yöntemi Geliştirme: Bu da DYGM nin bir parçasıdır ve fizibilite analiziyle ilişkilendirilir. Burada proje için en uygun geliştirme yöntemi seçilir. Bu da birkaç fazladan fazla sürmez (Dogs & Klimmer, 2004).

2.2.2 Proje Planlama Uygulamaları

Proje planlama uygulamaları, projede yapılması gerekenlerin neler olduğu, hangi sürede nelerin, hangi öncelikle, kimler tarafından yapılacağını belirlenmesini sağlar.

Proje Vizyonu :Proje boyunca ne geliştirilecek, ne işe yarayacak ve nasıl geliştirileceğinin tüm hedeflerin ekiple paylaşıldığı rehberdir.

Hikaye Kartları: Hikaye kartları ÇY’lerde yazılıma eklenecek özellikleri yakalamak için kullanılır. Gereksinim analizi, planlama ve iletişim açısından önemli sistemle ilgili istenilen özellikleri içeren küçük kartlardır. Müşteri ve geliştiricilere istenen özelliğin, ne amaçla, ne zaman nasıl yapılacağı pazarlığında kullanılır(Dogs & Klimmer, 2004).

Mimari altyapı çalışmaları: Proje için teknik açıdan önemli olan ihtiyaçları karşılayacak mimari ve araçların neler olduğunun belirlenmesi için gerekli hazırlıkları kapsar.

Yayın planlaması: ÇY’de en genel planlama olup, yazılım özelliklerinin hangi sırayla teslim edileceği, yayımların tarihleri veya hangi özellikleri içereceğine dair kapsamları içerir.

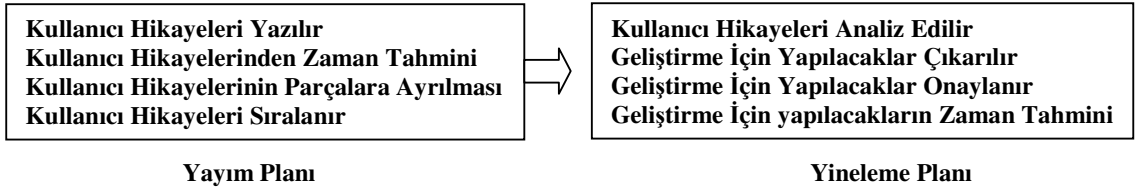
Müşteri ve geliştiriciler hangi özelliğin hangi sürümde uygulanacağını bir arada karar verdikleri, müşterinin sistemle ilgili kendi ihtiyaçlarını yazdıkları hikaye kartlarını geliştiricilere verip , geliştiricilerin bu özelliklerin ne kadar süreceğini tahminde bulunduğu daha sonra müşterinin bu özellikleri önceliklendirdiği ve sonrasında hep birlikte yayımlar halinde gruplandırıldığı pratiktir. (Dogs & Klimmer, 2004).

Aşağıda Şekil 2.4’te müşterilerin istediği özelliklerin müşteri tarafından belirlenen öncelik sırasına konarak hangi periyotta çıkartılacağını karar verildiği ile ilgili örnek bir yayın planlama uygulaması görülmektedir.



Şekil 2.4 Yayım Planı(Lefingwell 2007)

Yineleme planlaması: Yayım planlamasında belirlenen özelliklerle müşteri önceliğine göre tekrar gözden geçirilerek istenen özellikleri karşılamak için teknik olarak yapılması gerekenlerin geliştiriciler tarafından hikaye kartlarına eklendiği ve buna göre ihtiyaç duyulan zaman, kişinin belirlendiği uygulamadır.



Şekil 2.5 Çevik Süreçlerde Yayım ve Yineleme Planı(Beck. & Fowler (2004))

Değişen İhtiyaçların dikkate Alınması: Değişen ihtiyaçların proje yönetiminde dikkate alarak değişen ihtiyaçların uyarlanabildiği böylece projenin çevikliği sağlanabilir. Yayım planında Proje süresince müşteriden gelen yeni ihtiyaçlarda plana eklenip tekrar öncelik sırasına konabilir

Proje Büyüklüğünü Dikkate Alma: (Abrahamsson 2002)'a göre proje büyüklüğünü ve kritikliğini dikkate alan tek metodoloji Kristaldır..Değişik büyüklükteki projelerin nasıl yönetileceğiyle ilgili rehberlik edildiği uygulamadır.

2.2.3 İhtiyaç Mühendisliği Uygulamaları

Kullanıcı Temsilcisi:Bu da DYGM nin bir parçasıdır. Kullanıcı grubundan kullanıcının çalıştığı ortamın bilgilerine sahip ve daha iyi geridönüş yapabilecek aynı zamanda çalıştığı topluluktakilere önemli bilgileri aktaracak ve yeni sistemin potansiyel kullanıcılarından biri olacak kişidir (Dogs & Klimmer, 2004).

Terimler Sözlüğü:Proje daha fazla karmaşık hale geldikçe proje üyelerinin aynı kelime hakkında aynı şeyi düşünmeleri zorlaşır. Zaman zaman yanlış anlaşılmalara olabilir. RBS bunu dikkate alarak terimler sözlüğü pratiğini ortaya çıkarmıştır. Terimler sözlüğünde ekip üyelerinin genel bilgilerini paylaştı ve projede kullanılan önemli terimlerin bulunduğu sözlüktür (Dogs & Klimmer, 2004).

Müşterinin Yerinde Çalışma:XP nin önemli uygulamalarından biridir. Yerinde Müşteri tam zamanlı olarak proje takımıyla aynı ofiste çalışabilir. Amaç müşteri ve proje takımı arasında iletişimi güçlendirip problemleri doğrudan ve hızlıca çözebilmektir (Dogs & Klimmer, 2004).

Prototipler:Prototipleri kullanmak DYGM de bahsedilen pratiklerdendir. Prototiplerle çalışan yazılımı temsil eden ve müşteriyle birlikte test etmekte küçük bir modeldir. Böylece sistemin nasıl çalışması gerektiği gösterilip, ihtiyaçların karşılanıp karşılanamayacağı atlanan bir nokta olup olmadığı araştırılır (Dogs & Klimmer, 2004).

Tüm Süreçlerde İhtiyaç Yönetimi:İhtiyaç Yönetimi uygulaması tüm çevik yöntemlerde kullanılan uygulama olup, değişen ihtiyaçların önceliklendirilmesi ve sistematik olarak birbirleriyle ilişkilendirilmesini içerir.

Kullanım Uzmanları: Kristal yönteminde kullanılan ve kullanımcı temsilcisinden farklı olarak tüm sistem hakkında detaylı bilgiye sahip uzmanların kullanımını içerir (Dogs & Klimmer, 2004).

2.2.4 Tasarım Uygulamaları

CRC Kartları:CRC kartları Ward Cunningham tarafından bulunmuş ve XP de kullanılan pratiklerden biridir. CRC kartı geliştirilecek sistemin küçük bir gösterimidir. Her CRC kartı sınıf ismi, sorumlulukları ve diğer sınıflarla işbirliklerinin yazıldığı

kartlardır bu kartlarla masaya konularak yapılacak sistemin bulmaca gibi çözümlenerek tasarımı yapılır (Beck &Cunningham 1989).

Basit Modeller:Basit Modeller pratiği XP ve Çevik Modelleme tarafından kullanılıp modelin sadece müşterinin gerçekten gerekli gördüğü bilgileri içerir. Bu modellerle karışıklık riski ortadan kalkıp probleme kolaylıkla odaklanma mümkün olur(Dogs & Klimmer, 2004).

Test Edilebilirliğe Dikkate Alma: Yazılımı tasarlarken ,gerçeğe uymayan, tahmin edilemeyen modeller oluşturup, bunları kesfermek çok geç olabilir. Yaratılan modellerin nasıl test edileceği baştan belirtilmelidir. (Ambler 2003)'e göre eğer bir model test edilemezse, bu üretilemez de demektir. Eğer test edilemez bir model varsa bu model yeniden tasarlanmalıdır (Dogs &Klimmer 2004).

Bakım Yapılabilir Tasarım:Bakım Yapılabilir Tasarım RBS tarafından ortaya çıkarılan ve tasarımın daha sonra ortayabilecek değişikliklerin kolayca yapılabilmesini sağlayan basit bir tasarımı öngörmektedir.

Tasarım Uzmanı: Tasarım Uzmanı Kristal metodolojisi tarafından uygulanan ve yazılım tasarımı konusunda iyi deneyimli bir kişinin az deneyimli tasarımcılara yüksek kaliteli tasarım konusunda yardım ettiği rolü içeren pratiktir(Dogs &Klimmer 2004).

Özellik Takımları: Özellik Takımları, ÖGG de kullanılan ve geliştirilmesi gerekli her özelliğin, özel disiplinlerarası bir takım sorumluluğunda geliştirildiği dinamik bir takım pratiğidir(Dogs &Klimmer 2004).

Aynı Modelin Paralel Modellenmesi: Aynı Modelin Paralel Modellenmesi, Çevik modelleme yaklaşımından gelip oluşturulan paralel modellerin en iyisinin seçildiği pratiktir(Dogs &Klimmer 2004).

Açık Model Sunumu: Açık Model Sunumu, Çevik modelleme yaklaşımından gelip kritik önemdeki modellerden en önemlilerinin geliştiricilerin görüp yorum yapabileceği bir ortamda sunulduğu böylece geliştiricilerinde kaydeğer yorumlarından yapılan tasarım modelinin geliştirilmesi hedeflenmektedir(Dogs &Klimmer 2004).

Yeniden Kullanım Sorumlusu: Yeniden Kullanım Sorumlusu, Kristal yaklaşımının bir parçası olup mimar,tasarımcı yada yarı zamanlı bir programcı tarafından yeniden kullanılabilir bileşenlerin belirlenmesi, bunların ticari olanlarının zorunlularının elde edilmesi ile oluşturulan pratiktir.

2.2.5 Kodlama Uygulamaları

Kodlama Standartları:Kodlama Standartları, XP ve diğer yazılım geliştirme yöntemlerince kullanılan bu pratik kaynak kodun genel yapısı ve formatının tanımlandığı ve tüm geliştiricilerin uyması gereken böylece tüm takım üyelerinin kodu anlayabilmesini sağlayan pratiktir(Dogs & Klimmer, 2004).

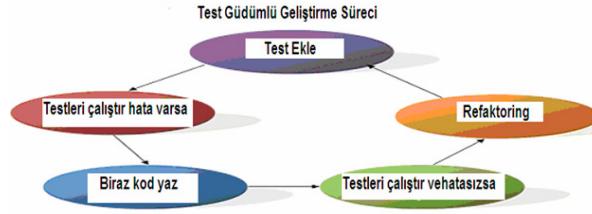
Kolektif Sahiplik:XP nin açıkça belirttiği bir pratiktir. Tüm geliştiriciler kodun kodun sahipliğini üstlenir. Herkes koda değişiklik yapabilir. Böylece kod kalitesinde yükselmek mümkün olabilir(Dogs & Klimmer, 2004).

Konfigurasyon Yönetimi:Konfigurasyon Yönetimi, değişik kaynak kodlu dosyaların yüklenmesi ve bu dosyalarda yapılan değişikliklerin takip edilmesi böylece yazılım eski versiyonunun tekrar çalıştırılabilmesi sağlamak amaçlanır. Genellikle daha çok otomasyon gerektirir(Dogs & Klimmer, 2004).

Eşli Programlama:İki geliştircinin yan yana olduğu ve birinin kod yazma diğerinin yazılan kodu izleme, hatalar ve alternatif yollar konusunda öneride bulunduğu ve sırayla rolleri değiştikleri XP nin en ünlü pratiğidir.

Kod Patternleri:(Abrahamsson 2002)'a göre, bir patern belli bir koşuldaki, mevcut bir probleme daha sonradan tekrar kullanmak için sunulan kanıtlanmış çözümdür.

Test GÜdümlü Geliştirme(TGG): Belli bir işlevi yerine getiren kodu yazmadan önce, bu işlevin yerine getirildiğini test eden test kodunun yazılmasıdır. TGG' de gereksinimler testler olarak ifade edilir ve geliştiricinin yapacağı kodlamayı yönlendirir. Kod yazılmadan önce test kodunda nesnenin arayüzü, diğer nesnelere etkileşimi önceden tasarlanır. Geliştirme yapılırken her ihtiyaç için bir test kodu yazılır, bu testin geçmesi için de gerekli kod yazılır ve testin geçtiği görüldükten sonra bir sonraki ihtiyaca geçilir. Daha önceden Çevik Yöntemler kullanılmadan yazılmış mevcut bir sisteme yeni özellikler eklenirken öncelikle güvenliği sağlamak için üst seviye fonksiyonel testler yazılır. TGG pratiği uyarınca yazılan tüm testler projenin test suitine eklenir ve sürekli entegrasyon kapsamında her kod değişikliğinde çalıştırılır.TGG pratiğinin uygulanışını aşağıdaki diyagram özetlenmiştir.



Şekil 2.6 Test GÜdümlü Geliştirme Süreci (Khamooshi 2008)

Sürekli entegrasyon:Projelerde farkedilmesi ve düzeltilmesi en zahmetli problemler entegrasyon problemleridir.Sürekli entegrasyon sunucusu ekibin yaptığı kod değişikliklerini versiyon kontrol sunucusu yardımıyla izler. Herhangi bir değişikliği olduğunda entegrasyon kurulumu başlatılır. Bu kurulumda basitçe tüm kodlar derlenir, testler çalıştırılır ve sonuçlar yayınlanır ve eğer bir hata varsa kurulum durdurulup sorunun giderilmesi beklenir. Sürekli entegrasyon kodların hatasız, her an kullanıma hazır tutulmasını mümkün kılar.

Refactoring:Benzer ihtiyaçlarda varolan fonksiyonalityi kaybetmeden, yazılan kodun kötü kısımlarının daha iyi bilgi düzeyi vb. dolayısıyla yeniden düzenlenip tekrarların kaldırıldığı, basitleştirilerek iyileştirildiği pratiktir(Dogs & Klimmer, 2004). Refactoring, kod tekrarı olan yerlerde, kodun karmaşık olduğu yerlerde, eski iş mantığının

bozulmadan kodun sade, anlaşılabilir ve esnek hale getirilerek değişiklik maliyetleri en aza indirir. Burada eski fonksiyonlitenin bozulmaması için testler çok önemlidir.

Kısa Sürümler:Kısa Sürümler, çevik hareketin bir planı izleme yerine, değişime tepki değerinin bir sonucu olarak ortaya çıkmış ve esnekliği ve bu esnekliğe yazılımın küçük ve kısa adımlı sürümler haline olmasıyla ulaşılabilir çünkü bu geliştiricilere daha büyük bir adıma geliştirmeye göre daha kolay ve kontrollü bir şekilde daha hızlı ve test edilmesi kolay ve kaliteli sürüm çıkarma mümkün olur.

Kullanıcı Gözden Geçirmeleri: Kullanıcı Gözden Geçirmeleri Kristal Yöntemi tarafından önerilip, kullanıcının zaman zaman mevcut iş sonucu üretilen ürünün hedeflenen şekilde olup olmadığına bakıp uygun olmayanlar üzerinde doğrudan ısrar edebildiği pratiktir.

İş Listesi:İş listesi,Scruml'a gelen ve kalan yapılacak işleri gösteren bir listedir.

2.2.6 Test Uygulamaları

Geliştirme aşamasından sonra test aşamasına geçilir. Bu aşamada artık geliştiricinin sorumluluğu bitmiştir, ihtiyaç kartı üstünde testçiler çalışmaya başlar.

Testçiler kabul senaryolarını çalıştırır ve kullanıcı gözünden yazılıma eklenen özelliğin kabul kriterlerini karşıladığını doğrular. Hatalar için geliştiricilerin bunları düzeltmeleri sağlanır.

Birim Testleri; programcılar tarafından kod yazım pratiğinin bir parçası olarak yazılır. Birim testi, tekil kodlu modüller veya yazılım ürününü küçük bir bölümün, ihtiyaçları karşılayıp karşılamadığı test edilir(TheFreeDictionary). Burada birim, bir sınıfın metodu yada fonksiyon anlaşılır ve otomatik olması daha faydalıdır. Bir sınıf yazılacaksa sınıfın yerine getireceği sorumluluk test olarak ifade edilir ve sınıfın davranışı bağımlılıklarından izole sahte objeler gibi yöntemlerle test edilir. Birim testleri veritabanı, web servisleri , donanım gibi bağımlılıklardan izole edilmeye çalışılır.

Entegrasyon Testi: Yazılım geliştirirken tüm modüller kendi başlarına doğru çalışabilirler ancak birbiriyle ilişkili modüller bir bütün olarak bakıldığında doğru çalışmayabilir. İşte entegrasyon testi burada devreye girerek birlikte çalışması gerekli modüllerin doğru çalışıp çalışmadığı kontrol edilir. Birden fazla sınıfın rol oynadığı işlevlerin testleri için kullanılır. Bu testler gerçekten veritabanı, web servisleri gibi bağımlılıkları kullanır.

Otomatize Regrasyon Testi:(TheFreeDictionary)' e göre, regrasyon testi ile,düzeltilen hataların tekrar olup olmadığını otomatize olarak kontrol etmek için kullanılır.

Genellikle birim testleri kullanılır.Kodda sık değişiklikler yapıldıkça otomatik regrasyon testi de kısa aralıklarla günde bir yada daha fazla uygulanır(Klasik şelale yönteminde tüm kodun bitiminde olmaktadır.)

Sistem Testi: Sistem testi tüm sistemin en genel testidir ve genellikle performans için fonksiyonel olmayan ihtiyaçlar göz önünde tutulur.

Kabul senaryosu testleri; tamamen kullanıcı gözünden her hikaye kartının kabul kriterlerinin karşılandığını test edilir. Tüm bu testler otomatik çalışacak şekilde hazırlanır ve sürekli entegrasyon sürecine dahil edilir. Bunun yanında fonksiyonel olmayan performans gibi kriterleri için performans testleri yazılır.

2.2.7 Geçiş Uygulamaları :

Test aşaması bittiğinde ürün sahibine bir demo yapılır ve onayı istenir. Onay alındıktan sonra kart 'Bitti Bitti Bitti' aşamasına getirilir. 'Bitti Bitti Bitti' deki ilk Bitti geliştirilmesinin bitişidir. İkinci Bitti testin, üçüncüsü ise ürün sahibinin onayının alınmasını simgeler.

Demo toplantısı: Amacı yineleme kapsamında geliştirilen yeni özellikleri kullanıcılara göstermek ve geri beslenim almaktır. Toplantıya katılanlar yineleme içinde hangi özelliklerin yazılıma eklendiğini görür ve fikirlerini paylaşır. Kullanıcıların kafasındaki belirsizlikler somut yazılım özellikleri gösterilerek aşılır. Ortaya çıkan yeni fikirler toplanıp yazılımın daha yararlı olması adına kullanılır.

Beta Testi:Beta testi Rational Birleşik Süreç(RBS)-(Rationale Unified Process)’ten gelir ve gerçek kullanıcıların gerçek ortamlarında yaptığı ve önceden atlanılma riski bulunan hataların testler kastedilir(Dogs & Klimmer 2004).

Sürüm İçerik Anonsları:Sürüm içerik anonsları Xbreed yöntemi tarafından ortaya atılan, her yeni sürümde bu sürümde olan yenilikler, düzeltilen hatalar yada eklenen özellikler hakkında bilgi vermeyi amaçlayan pratiktir.

2.2.8 Organizasyon Uygulamaları

Haftada 40 saat çalışma: (Beck 20000)’e göre fazla mesai gerekmediği durumlarda haftada 40 saatten fazla çalışmamayı içeren uygulamadır. Fazla mesainin en fazla 2 hafta devam etmesi sonrasında doğacak ihtiyaç için ayarlama yapılmasını önerir. Amaç stresten kaçmak ve takım üyelerine optimal yaratıcı ortam sunmaktır.

İletişim Yeteneklerine Önem:Yüzyüze iletişim diğer iletişim yöntemlerine tercih edilir.

Motive Çalışanlara Önem:Takım üyelerinin iyi iş çıkarabilmeleri için motive edici faaliyetlerde artış sağlanmalıdır.

Teknik Eğitime Önem:Takımın sürekli teknik eğitimin olmasının desteklediği ve eğitimin gereksiz zaman alıcı bir aktivite olmadığı aksine iyi karşılandığı pratiktir.

Öğrenilen Dersler:Birçok durumda proje boyunca edinilen deneyim ve öğrenilen dersler gerek projenin bitmesinden gerekse devam eden başka bir projeden dolayı tekrardan kullanımı zor olmaktadır.Bunun için proje bitiminde tüm takım elde ettiği dersleri paylaşarak bu deneyimin kalıcılığı sağlanabilir.

Gereksiz İşleri Azaltma: Gerçekten çok gerekli olmayan ve kaçınılabilecek kağıt dökümantasyon gibi orta düzeyli işleri ortada kaldırıp iletişimi arttırmayla ilgili pratiktir.

Müşteriyle Düzenli Görüşmeler:Müşteriyle düzenli görüşmeler UYG'nin bir pratiği olup, sık aralıklarla müşteri ve proje takımını yada takım temsilcileriyle buluşup mevcut iş süreciyle ve olası problemleri tartıştığı pratiktir..

Kendi Kendi Organize Ekipler:Takımın kendi sorunlarını kendi yöntemleriyle çözdükleri, hedefe ulaşmak için kendi kurallarını koyabildikleri, proje yöneticisinin emir veren otoriteden çok takıma koçluk eder pozisyonda olduğu pratiktir.

Şablonların ve Standartların Ayarlanması:Şablonların ve Standartların Ayarlanması RBSve Kristal de kullanılan bir tekniktir. Görevleri daha hızlı yapabilmek için önceden tanımlı şablon ve standartların kullanıldığı pratiktir.

Basit Araçlar:Çevik Modellemenin taleplerinden olan basit araçlar, problemleri daha iyi anlayıp çözmek için kullanmayı önerir.

Görsel Modeller:Görsel Modeller RBS ve Çevik Modellemeden gelip, modellerin daha anlaşılabilir olması için uygun tekniklerle grafiksel gösterimini savunur.

Aynı dili konuşmak:Geliştirici, analist, testçi, proje yöneticisi ve müşteri gibi projede farklı roller oynayan insanlar arasında iletişim problemleri olabilir. İletişim eksikliği tasarımda, test senaryolarında eksikliklere yol açabilir. Bu problemlerin önüne geçilmesi için tüm ekibin aynı dili konuşması şarttır.

Günlük ayakta toplantılar:Her sabah aynı saatte tüm ekip üyeleri günlük planlamayı yapmak, son durum hakkında bilgi paylaşımını sağlamak amacıyla ayakta toplantı yapar. Bu toplantıda ekip üyeleri kısaca yaptığı işin son durumunu, karşılaştığı problemleri ve o gün için planlarını anlatır.

Gözden Geçirme Toplantısı:Yineleme bitiminde gözden geçirme toplantısı yapılır. Amaç ekibin yaptığı çalışmalardan elde ettiği dersleri takımla paylaşarak sonraki çalışmalarda ekibin daha verimli çalışmasını sağlamak amaçlıdır. Ekip üyeleri iyi, daha iyi

olabilecek ve onları şaşırtan, kızdıran durumları farklı renklerle kartlara yazıp bunu paylaşımı bu güzel bir örnek olabilir.

2.2.9 Proje Yönetim Metrikleri

Yineleme bitiminde ekibin tamamladığı özelliklerin sayısı, süresi,yayım planından sapmalar, projenin gidişatı Burn Down, Burn Up, Finger gibi farklı grafiklerle görsel olarak izlenerek gerektiğinde müdahale edilir.

2.3 ÇY'LER VE CMM

CMM ve bunun daha geliştirilmiş hali olan CMMI yazılım geliştirme organizasyonları tarafından projelerindeki başarı için iyi tanımlı ve iyi dökümente edilen yazılım geliştirme standartı haline gelmiştir. ÇY'ler, CMM standartlarına benzerliğine rağmen yazılım organizasyonları tarafından geniş ölçüde olarak kabul görmektedir (Awad 2005). CMMI modeli süreçlerin hangi alanlarda pratiklere sahip olması gerektiğini söyler ve buna göre olgunluk seviyeleri tanımlar. Bu uygulamaların neler olacağı, ekibin hangi yöntemlerle çalışacağı konusunda kısıtlar koymaz CMMI ne yapılacak sorusunun cevabıdır. Nasıl sorusunun yanıtını uygulanan sürecin kendisi verir. Günümüzde CMMI sertifikasyonunu ÇY'ler kullanarak gerçekleştiren şirketlerin sayısı artmaktadır. Laurie Williams CMMI yada ISO(İnternational Standardisation Organization) 9000 standartları kullanan birçok firmanın, ÇY'lerle kısmi adoptasyonu düşündüğünü, dikkatli olarak uygulandığında, firmaların verimliliklerini sertifikalarını kaybetmeden arttırabildiklerini belirtmektedir. Bunu Tom DeMarco XP'yi uygulayan organizasyonların 5 ay içerisinde CMMI Seviye 1'den Seviye 4'e 5 ay içerisinde ulaşabileceğini belirterek desteklemektedir(M. A. Awad,2005). Sadece CMMI gereksinimlerini kullanarak süreç iyileştirme yapan şirketler bürokratik, ekibin doğal çalışma haline gelemeyen süreçlerin oluşması riski büyüktür. ÇY adaptasyonu ve sonrasında oluşan sürecin CMMI gereksinimlerine göre paralel eksiklik analizleri çalışması yoluyla, hem etkin bir süreç hemde CMMI sertifikasyon hedefi tutturulabilir.

2.4 ÇY'LER VE DÖKÜMANTASYON

ÇY'ler için dokümantasyon, analiz, planlama gibi aktivitelerin ihmal edildiği gibi önyargılar vardır. Bu aktiviteler klasik süreçlerde projenin başında uzun süren şekilde uygulanır. ÇY'lerde ise aktiviteler proje süresine yayılır ve sürekli yapılır. Dokümantasyon adım adım kartlar geliştirilme aşamasına yaklaştıkça detaylandırılır.

2.5 ÇY'LERDE KULLANILAN ARAÇLAR

Hayatta olduğu gibi doğru seçilmiş araçlarla ÇYGP de de büyük değişiklikler meydana getirilebilir. Normalde gerçekleştirilmesi zahmetli ve uzun süren bir çok görev doğru destek araçlarıyla daha kolay ve basit bir şekilde gerçekleştirilebilir. Piyasada bu konuda faydalanılabilecek bir çok araç mevcuttur(Hunt 2006). Peki ÇYGP'de hangi tip ihtiyaçta hangi araçlara duyulur incelemek istersek;

- Eğer çevik modelleme uyguluyorsak, modelden kodu, koddan modeli kolayca ve en az çabayla eş zamanlı olarak güncelleyebilmeliyiz yeni nesil hafif bir modelleme aracı gerekir. Bu konuda Omondos, Eclipsle uyumlu olduğu için tercih edilebilir.
- Yeni kod geliştirmek gerektiğinde kolay ve hızlıca kod geliştirmeyi sağlayan geliştirme aracı gerekir. Bu konuda ANT, Java tabanlı olduğundan ve eclipsle uyumlu olduğundan tercih edilebilir
- Sık değişken yazılım versiyonlarını kontrol etmek ve eğer işler kötü giderse önceki versiyona dönüp güvenli bir şekilde güncelleyebilmek için Versiyon Kontrol aracı gerekir. Buna örnek olarak Subversion ve CVS verilebilir mümkün olmalıdır .
- Basit ve kolayca refactoring ve arttırımlı geliştirmeyi destekleyen ve kullanımı düşünülen diğer araçlarla entegre çalışabilen bir IDE(Entegre Geliştirme Ortamı) gerekir. Bu konuda Eclipse önde gelen IDE'lere dendir.
- Test suitlerini basitçe tekrar tekrar çalıştırıp, sonuçlarını kolayca ve anında görmek mümkün olmalıdır. Buna örnek olarak JUnit verilebilir.
- Hataların takibi için Jira, Trac gibi araçlar kullanılır

- Çevik proje yönetimi için özel tasarlanmış, ihtiyaçların takibi, yayım ve yineleme planlama, burn down grafikleri gibi kabiliyetleri içeren yazılımlar gerekir. Bunlara örnek olarak Thoughtworks Mingle, Version One yazılımları verilebilir.
- Proje tek bir komut ile kurulabilmelidir. Bunun yanında sürekli entegrasyon için Cruise Control , Hudson gibi araçlar kullanılır. Bunlar Subversion, CVS gibi versiyon kontrol sistemlerini entegre çalışarak kısa aralıklarla otomatik kurulumlar gerçekleştirirler.
- Proje kodlama standartları, potansiyel problemlerin analizi, test kapsam raporları gibi konularda da birçok araç mevcuttur. Bu araçların seçiminde ana kriterlerden biri de otomatik kurulum sürecine dahil olabilme kabiliyetleridir. PMD, FindBugs, Clover, Cobertura, NDepend, FxCop bu tür araçlardır.

2.6. PROBLEMİN TANIMI

Yazılım süreçlerinin iyileştirilmesi ve belirli kalite düzeyinin sağlanması için de yazılımın geliştirme yaşam döngüsünün tüm adımlarında, tanımlı teknik şartlara uygunluğu sağlayacak, sistematik yaklaşımlara ihtiyaç duyulmaktadır. Ancak yazılım geliştirmede pek çok süreç ve pek çok alternatif yaklaşım bulmak mümkündür. Günümüzün hızla gelişen ve değişen dünyasında Yazılım Geliştirme Organizasyonları da bu değişime ayak uydurmak ve verimliliklerini artırıp başarıya ulaşmak için giderek artan oranda ÇY'lere ilgi duymaktadırlar fakat gerek konunun yeni bir konu olması gerekse bu alanda edinilmiş tecrübeler ve yapılan araştırmaların kısıtlı olması sebebiyle ÇY'lere geçişte kendilerini nelerin beklediğini ve projelerinde başarıya giden yolda, nelerden kaçınıp neleri yapmaları gerektiğini tam bilemedikleri için, bu süreçleri kendi projelerine uygulamakta ya çekingen davramakta yada tam olarak uygulayamamakta, böylece ÇY'lerin getirebileceği verimlilik artışı ve diğer faydalarından yeterince faydalanamaktadırlar. Bu tezde de, ÇYG Projelerinde verimlilik ve başarılı olmak için gereken, Kritik Başarı İndikatörleri, ölçütleri ve uygulamaları mevcut literatür taraması ve bizzat kendi tecrübelerimden elde ettiğim sonuçlara göre belirlenmiş daha sonra belirlenen indikatör, ölçütler ve uygulamaların önem sıralarının belirlemek için bir anket

düzenlenmiş ve bu anketten elde edilen veriler, BAHP yöntemiyle analiz edilerek aşağıdaki sorulara yanıtlar aranmaktadır.

1. Çevik sürecin faydaları, kısıtlamaları ve ve kısıtlamalara karşı neler yapılabilir?
2. ÇYG de verimlilik ve başarı için kritik başarı faktörleri nelerdir?
3. ÇYG de verimlilik ve başarı için uygulanması gerekenler pratik uygulamalar nelerdir?
4. ÇYG de verimlilik ve başarı için gereken pratik uygulamaların öncelik sıraları nelerdir?

2.6.1. ÇYG’de Kritik Başarı Faktörleri

Chow & Cao (2007)’a göre ÇYGP’de KBF hakkında yapılan örnek çalışmalar, başarı yada başarısızlık araştırmaları olmasına rağmen akademik ve uygulamalı literatürde ciddi nitelikte araştırma bulunmamaktadır. Olası başarı faktörlerinin ortaya çıkarılmasında, yapılan başarı yada başarısızlık araştırmalarından faydalanmak da mümkündür. Çünkü yapılan hatalardan ders alarak ÇYGP de başarıya ulaşmak mümkün olabilir. Bundan dolayı başarı faktörleri perspektifinden bakıldığında geçmiş deneyimlerden öğrenmenin en etkin yolu, eski projeleri analiz etmektir.

Başarısızlık Araştırması

Reel (1999)’ a göre, başarısızlık yada problem araştırması, tipik olarak çeşitli türdeki projelerden çıkarılan derslerdir. Fakat bu derslerin, yeterli derecede genelleştirilebilir olması için, çoğunlukla benzer nitelikte olması gerekir. Cohn & Ford (2003) kurumların Çevik sürece geçişteki sorunlar üzerine çalışılmıştır. Larman (2004)’de, çevik projelerde karşılaşılan hata ve yanlış anlaşılımları tartışan bir çalışma yapılmıştır. Boehm & Turner (2005) tarafından yapılan çalışmada da, çevik projelerin uygulamaya alınmasındaki yönetsel zorluklara vurgu yapılmıştır. Nerur ve diğ. (2005) tarafından yapılan çalışmada, çevik projelere geçişte problemleri sadece yönetim perspektifinden değil aynı zamanda insan, süreç ve teknoloji boyutlarından da ele alınmıştır. Chow & Cao (2007) tarafından yapılan çalışmada da, organizasyon, insan, süreç ve teknik faktörler boyutlarından başarı önündeki engelleri tanımlayan detaylı bir çalışma yapılmıştır.

Başarı araştırması

Başarı araştırması için, genellikle literatürdeki örnek çalışmalar ve Çevik projeler ve uygulamalarında başarıya etki eden kritik faktörlerle ilgili değerlendirmelerden ve ilgili literatürden faydalanılmıştır. Kritik Başarı Faktörleri(KBF), bir organizasyonun performansını ölçmek ve tanımlamak için ilk olarak Rockhart (1979) tarafından geliştirilmiştir. Daha sonra Bullen & Rockhart (1981) ile Rockhart & Crescenzi (1984) tarafından yapılan çalışmalarda daha sade ve iyi hale getirilmiştir. Reel (1999)' a göre, Yazılım Geliştirme Projeleri(YGP)ndeki KBF, proje yönetim teknikleriyle ilişkilidir. Bytheway (1999)'a göre de, KBF Yazılım mühendisliği ve iş stratejisinin kombinasyonu ile ilişkilidir. Fowler & Highsmith (2001) araştırmasında çevik uygulamalarda ihtiyaçların ve motive çalışanların önemine vurgu yapılmıştır. Fowler (2002) ile Stewe Ambler Workshop (2002)'deki araştırmalarda, yazılım geliştirmede yerel kültür ve sosyal kültürün önemine değinilmiştir. Lindvall ve diğ. (2002) ile Stewe Ambler (Workshop 2002)'deki çalışmalarda ise, insanların bilgi paylaşımı ve sürekli öğrenmenin önemine vurgu yapılmıştır. Lindvall ve diğ. (2002) çalışmasında, kültüre ek olarak, insan ve iletişimi en önemli faktörler olarak tanımlamıştır. Bosghossian(2002) çalışmasında YGP'deki KBF nin çok boyutluluğundan, Highsmith (2002)'te ise, doğrudan çevik uyarlamalardan bahsedilmektedir. Schatz & Abdelshafi (2005)'te Primavera örnek çalışmasından bilgiler sağlanırken, Karlstrom & Runeson (2005)'te ise, Star Gate örnek çalışmasından bilgiler verilmiştir. Boehm & Turner (2003), Augustine ve diğ. (2005), Ceschi ve diğ. (2005) ve Petersen & Wohlin (2009) çalışmalarında da, geneksel ve ÇY'leri karşılaştıran çeşitli çalışmalar yapılmıştır. Reifer ve diğ. (2003), Lindvall ve diğ.(2004) ve Ambler(2006) çalışmalarında ise, geniş organizasyonlar yada projelerde ÇY'lerin uyarlanması üzerine odaklandığı görülmektedir. Koch (2005)'te Çevik uyarlamalarda başarı faktörleriyle ilgili geniş bir derleme çalışması yapılmıştır. (Ambler 2005b) e göre ise, hızlı ve etkin iletişim, çevik yöntemlerin en önemlileri olarak tanımlanmıştır. (Misra ve diğ. 2006) deki çalışmada, ÇYGP'de başarılı adaptasyonla ilgili kriterleri ortaya koyan detaylı bir çerçeve çalışması yapılmıştır. Chow & Cao (2007)'de ise, Çevik Yazılım Projelerine kritik başarı faktörleriyle ilgili organizasyon, insan, süreç, teknik, proje boyutlardan yaklaşan ve önem derecesine göre sıralandığı detaylı bir araştırma gerçekleştirilmiştir. Misra ve diğ. (2008)'da, ise 2006 yılındaki çerçeve çalışmasını sadeleştirerek Çevik Yazılım

Projelerinde başarı faktörleriyle ilgili sadeleştirilmiş bir çerçeve çalışması yapılmıştır. Yapılan bu literatür çalışmaları tablo 2.1’de özetlenmiş olarak görülebilir.

Tablo 2.1 Çevik Yazılım Geliştirme Projelerinde KBF literatür araştırması

Yıl	Yazar	Araştırma Tipi	Yapılan Çalışma
1979	Rockhart	Başarı Araştırması	Organizasyonun performansını ölçmek ve tanımlamak amaçlanmıştır.
1981	Bullen & Rockhart	Başarı Araştırması	Önceki çalışma daha sade ve iyi hale getirilmiştir.
1984	Rockhart & Crescenzi	Başarı Araştırması	Önceki çalışma daha sade ve iyi hale getirilmiştir.
1999	Reel	Başarı Araştırması	KBF proje yönetim teknikleriyle ilişkisi belirtilmiştir.
1999	Reel	Başarısızlık Araştırması	Başarısızlık araştırması tanımlanmıştır.
1999	Bytheway	Başarı Araştırması	KBF Yazılım mühendisliği ve iş stratejisinin kombinasyonu ilişkisi belirtilmiştir.
2001	Fowler & Highsmith	Başarı Araştırması	Çevik uygulamalarda ihtiyaçların ve motive çalışanların önemine vurgu yapılmıştır
2002	Fowler	Başarı Araştırması	Yazılım geliştirmede yerel kültürün önemine değinilmiştir
2002	Ambler	Başarı Araştırması	Yazılım geliştirmede yerel kültürün önemine değinilmiştir
2002	Highsmith	Başarı Araştırması	Doğrudan Çevik uyarlamalardan bahsedilmiştir.
2002	Lindvall ve diğ.	Başarı Araştırması	Yazılım geliştirmede sosyal kültürün önemine değinilmiştir
2002	Lindvall ve diğ.	Başarı Araştırması	Kültüre ek olarak, insan ve iletişimi en önemli faktörler olarak tanımlanmıştır
2002	Bosghossian	Başarı Araştırması	YGP deki KBF nin geliştirme hayat çevriminden, kaynak ve stratejik seviyede planlamaya kadar bir çok boyutu olduğundan bahsedilmiştir.
2003	Boehm & Turner	Başarı Araştırması	ÇY'lerin uygulanmasında iletişim ve müzakerenin önemine değinilmiştir.
2003	Boehm & Turner	Başarı Araştırması	Geneksel ve ÇY'ler karşılaştırılmıştır.
2003	Reifer ve diğ.	Başarı Araştırması	Geneksel ve ÇY'ler karşılaştırılmıştır.
2003	Cohn ve Ford	Başarısızlık Araştırması	Kurumların çevik sürece geçişteki sorunları üzerine çalışılmıştır.
2004	Lindvall ve diğ.	Başarı Araştırması	Geniş projelerde ÇY'lerin uyarlanması üzerine odaklanılmıştır.
2004	Larman	Başarısızlık Araştırması	Çevik projelerde karşılaşılan hata ve yanlış anlaşımaları tartışılmıştır.
2005	Augustine ve diğ.	Başarı Araştırması	Geneksel ve ÇY'leri karşılaştıran çalışmalar yapılmıştır.
2005	Ceschi ve diğ.	Başarı Araştırması	Geneksel ve ÇY'leri karşılaştıran çalışmalar yapılmıştır.
2005	Schatz & Abdelshafi	Başarı Araştırması	Primavera örnek çalışmasından bilgiler sağlanmıştır.
2005	Karlstrom & Runeson	Başarı Araştırması	Star Gate örnek çalışmasından bilgiler vermiştir.
2005	Koch	Başarı Araştırması	Çevik uyarlamalarda başarı faktörleriyle ilgili geniş bir derleme çalışması yapmıştır.
2005	Ambler	Başarı Araştırması	ÇY'lerin en önemlileri olarak biri hızlı ve etkin iletişim olarak tanımlanmıştır.
2005	Boehm & Turner	Başarısızlık Araştırması	Çevik projelerin uygulamaya alınmasındaki yönetimsel zorluklara değinilmiştir.
2005	Nerur ve diğ.	Başarısızlık Araştırması	Çevik projelere geçişte problemleri sadece yönetim perspektifinden değil aynı zamanda insan, süreç ve technology boyutlarından ele alınmıştır.
2006	Ambler	Başarı Araştırması	Geniş projelerde ÇS'lerin uyarlanmasına odaklanılmıştır.
2006	Misra ve diğ.	Başarı Araştırması	ÇYGP'de başarılı adaptasyonla ilgili detaylı bir çerçeve çalışması yapmıştır.
2007	Chow & Cao.	Başarı Araştırması	ÇYP'de KBF ile ilgili kriterleri belirleyen ve önem sırasına sokan şuna kadar yapılmış araştırmaların en detaylısını gerçekleştirmiştir
2008	Misra ve diğ.	Başarı Araştırması	2006 daki çerçeve çalışmasını sadeleştirerek Çevik Yazılım Projelerine Başarı Faktörleriyle ilgili bir çerçeve çalışması yapmıştır.

1999-2002 yılları arasında yapılan çalışmalar incelendiğinde KBF ile ilgili olarak şöyle bir tablo ortaya çıkar;

Tablo 2.2. 1999-2002 yılları arasında Literatürdeki ÇYGP’de Kritik Başarı indikatör ve kriterleri

Yıl	Yazar	Kriter	Alt kriter
1999	Reel	Yönetimle ilgili faktörler	•proje yönetim teknikleri
1999	Bytheway	Süreçle ilgili faktörler Yönetimle ilgili faktörler	•yazılım mühendisliği •iş stratejisi
2001	Fowler &Highsmith	Süreçle ilgili faktörler İnsanla ilgili faktörler	•ihtiyaçların değişkenliği •motive çalışanlar
2002	Fowler	Organizasyonla ilgili faktörler	•yerel kültür
2002	Ambler	İnsanla ilgili faktörler	•sosyal kültür
2002	Lindvall ve diğ..	İnsanla ilgili faktörler	•sosyal kültür •eğitim • birbirinden öğrenmeye isteklilik •dürüstlük • işbirlikçilik •sorumluluk sahibi olmak • takımlar biraz yerel kontrole ihtiyaç duyarlar • takımların adapte edebilme yeteneği de olmalıdır •takımların daha az fakat daha yetenekli takım üyesi •güvenilir olmalı
		Organizasyonla ilgili faktörler	•organizasyon, geliştiricilerin kararlarına uymaya istekli olmalıdır. •üyeleri arasında hızlı iletişim kurmayı sağlayacak ortamlar sunmalıdır • müşteriyle yakın temas •müşteriden hızlı geri dönüş
2002	Bosghossian	Süreçle ilgili faktörler	• geliştirme hayat çevrimi
		Organizasyonla ilgili faktörler	• üst yönetimin tahmini • proje yönetimi • kaynak ve stratejik seviyede planlama

2003-2006 yılları arasında yapılan çalışmalar incelendiğinde KBF ile ilgili olarak şöyle bir tablo ortaya çıkar;

Tablo 2.3. 2003-2006 yılları arasında Literatürdeki ÇYGP’de Kritik Başarı İndikatör ve kriterleri

Yıl	Yazar	Kriter	Alt kriter
2003	Turner & Boehm	İnsanla ilgili faktörler	<ul style="list-style-type: none">• iletişim• müzakere
		Süreçle ilgili faktörler	<ul style="list-style-type: none">• basit tasarım• kısa döngüler
2005	Ambler	Organizasyonla ilgili faktörler	<ul style="list-style-type: none">• hızlı ve etkin iletişim
2006	Misra ve diğ..	İnsanla ilgili faktörler	<ul style="list-style-type: none">• Yetenek• Kişisel Karakteristikler,• İletişim ve Müzakere,• Sosyal Kültür,• Eğitim ve Öğrenme
		Organizasyonla ilgili faktörler	<ul style="list-style-type: none">• Müşteriye Vaatler,• Karar Zamanı,• Takım Dağıtımı,• Takım Büyüklüğü• Şirket Kültürü,• Planlama ve Kontrol,• Kritiklik ve Güvenilirlik,• Dinanizm ve Kesinlik Durumu
		Teknik Faktörler	<ul style="list-style-type: none">• İhtiyaçlar,• Geliştirme• Test

2007 yılında yapılan çalışmalar incelendiğinde KBF ile ilgili olarak şöyle bir tablo ortaya çıkar;

Tablo 2.4. 2007 yılında Literatürdeki ÇYGP’de Kritik Başarı indikatör ve kriterleri

Yıl	Yazar	Kriter	Alt kriter
2007	Chow., & Cao	Organizasyonla ilgili faktörler	<ul style="list-style-type: none"> •Yoğun Yönetim Desteği, Bağlı sponsor yada yönetici, •Birleşik kurum kültürü, •Yüz yüze iletişime büyük önem veren kültür, •Çevik metodolojinin genel kabul gördüğü kurum Ortamı, •Tüm takımın düzenlenmesi, •Çevik stilde geliştirme ortamı araçları, •Çevik yaklaşıma uygun sistemi ödüllendirme
		Kişilerle ilgili faktörler	<ul style="list-style-type: none"> •Yüksek kabiliyet ve uzmana çalışanlar •Yüksek motivasyonlu çalışanlar •Yöneticilerin ÇY hakkındaki bilgisi, •Yöneticilerin uyarlanırlı yönetim tarzı •Bağlı ve kendi kendine organize olabilen takım çalışması •Müşteriyle iyi ilişkiler kurmak
		Süreçle ilgili faktörler	<ul style="list-style-type: none"> • Çevik ihtiyaç yönetimi süreci izlemek • Çevik proje yönetimi süreci izlemek • Çevik konfigürasyon yönetimi süreci izlemek • Yoğun günlük yüz yüze buluşmalarla iletişimi • Fazla mesaisiz, düzenli çalışma çizelgesi • Yoğun müşteri katılımı • Müşterinin tüm otoriteye sahip olması
		Teknik ilgili faktörler	<ul style="list-style-type: none"> •İyi tanımlanmış kodlama standartları •Basit tasarım •Yoğun refactoring faaliyetleri •Yeterli dökümantasyon, •Yazılımın düzenli teslimi •En önemli özelliklerin teslimi •Yeterli entegrasyon testi
		Projeyle ilgili faktörler	<ul style="list-style-type: none"> •Takıma uygun teknik eğitimi •proje doğası yaşamsal önemde olmayan •proje tipi değişken ve acil nitelikte •Dinamik hızlandırılmış çizelgeli projeler •Küçük takımlı projeler •Çoklu ve bağımsız olmayan takımlı projeler •Belirgin Maliyet değerlendirmeli projeler •Belirgin risk analizli projeler

2008 yılında yapılan çalışmalar incelendiğinde KBF ile ilgili olarak şöyle bir tablo ortaya çıkar;

Tablo 2.5 2008 yılında Literatürdeki ÇYGP’de Kritik Başarı indikatör ve kriterleri

Yıl	Yazar	Kriter	Alt kriter
2008	Misra	İnsanla ilgili faktörler	<ul style="list-style-type: none"> •Yetenek •Kişisel Karakteristikler •İletişim ve Müzakere •Sosyal Kültür •Eğitim ve Öğrenme
		Organizasyonla ilgili faktörler	<ul style="list-style-type: none"> •Müşteri Tatmini • Müşteriyle İşbirliği •Müşteriye Vaatler •Karar Zamanı •Takım Dağıtım •Takım Büyüklüğü •Şirket Kültürü •Palnlama •Kontrol
		Diğer faktörler	<ul style="list-style-type: none"> •Hatalardan ders çıkarma •Zamanlama •Diğer Takım karakteristikleri •Araç kullanımı

Yukarıdaki çalışmalar incelendiğinde başarısızlık faktörleriyle ilgili olarak şöyle bir tablo ortaya çıkmaktadır ;

Tablo 2.6 Literatürde ÇYGP’de Başarısızlık indikatör ve kriterleri

Yıl	Yazar	Kriter	Alt kriter
2007	Chow, & Cao	Organizasyonla ilgili faktörler	<ul style="list-style-type: none"> •Üst yönetim sponsorluğu eksikliği •Yönetim vaatleri eksikliği •Organizasyonel kültür çok geleneksel •Organizational kültür çok politik •Organizational büyüklüğü çok geniş •Lojistik düzenlemelerdeki eksiklik
		İnsanla ilgili faktörler	<ul style="list-style-type: none"> •Gerekli yeteneklerdeki eksiklik •Eksik proje yönetim becerileri •Takım çalışmasında eksiklik • Kişi yada grupların direnci •Kötü müşteri ilişkileri
		Süreçle ilgili faktörler	<ul style="list-style-type: none"> • Kötü tanımlanmış proje vizyonu •Kötü tanımlanmış proje ihtiyaçları • Kötü tanımlanmış proje planı •ÇY takip mekanizmasında eksiklik •Müşteri katılımında eksiklik •Kötü tanımlanmış müşteri rolü
		Teknik faktörler	<ul style="list-style-type: none"> •Doğru çevik pratikler setinde eksiklik

ÇYGP’nde başarıyı yakalamak isteyen organizasyonlar için, yukarıda belirtilen başarı ve başarısızlıkla ilgili faktörleri dikkatle uygulanması gerektiği unutmamalıdır.

3. MALZEME VE YÖNTEM

Bu bölümde, öncelikle ÇYGP'nde KBF'nin belirlenmesi için yapılan anket çalışması ve toplanan veriler ve KBF ile ilgili karar verme modelinin oluşturulması için tercih edilen *Bulanık Analitik Hiyerarşi Prosesi (BAHP)* araştırma yöntemi hakkında genel bilgiler verilecektir.

3.1.MALZEME

Bu çalışmada KBF'ne etki eden olası indikatörleri ve faktörleri belirlerken ÇYG'yle ilgili literatürden, yapılan anket çalışmalarından ve mesleki birikimlerden faydalanılmış olup bunlar arasında ortak nitelikteki olanlar bir anket halinde düzenlenmiştir.

Anketin ilk bölümünde öncelikle katılımcıların demografik ve tecrübeleri hakkında sorular sorulmuştur. Anketin ilk bölümünde ÇYGP verimlilik ve başarı için hangi ana kriterler, alt kriterlerin ve uygulamalarıyla ilgili detay kriterlerin önemli olduğu ile ilgili sorular yöneltilmiştir. Anketin üçüncü ve son bölümünde, anket katılımcılarına geçmiş tecrübeleri ve bizzat buldukları Yazılım Geliştirme Projelerinde, ÇY'lerin tercih edilmesinin ya da varsa kullanımının önündeki engeller ile ilgili sorular yöneltilmiştir.

Hazırlanan bu anket yazılım geliştirme konusunda uzman 6 katılımcı tarafından cevaplanmıştır. Katılımcılarının %80'i Türkiye'den ve %20'si Avrupa'dandır. %20'si 0-5 kişi, %60'ı 5-100 kişi, %20'si de 100-1000 toplam çalışana sahip Hızlı Tüketim, Bilgi Teknolojileri, Bankacılık, Telekomünikasyon sektörlerinden firmalarda, Yazılım Geliştirme bölümlerinde, yönetici, geliştirici, analist pozisyonlarda çalışanların hepsi 5-10 yıl arasında deneyime sahip olan uzmanları kapsamış olup, ankete katılanların ortak özelliği tamamının Kurumsal Yazılım Geliştirme projelerinde çalıştıkları görülüp, bunların, %40'ı Kurumsal Yazılımla birlikte E-Ticaret Yazılımlarıyla ilgili projelerde, %20 sinin de Kurumsal Yazılımla birlikte Güvenlik yazılımları alanlarında çalıştıkları görülmüştür.

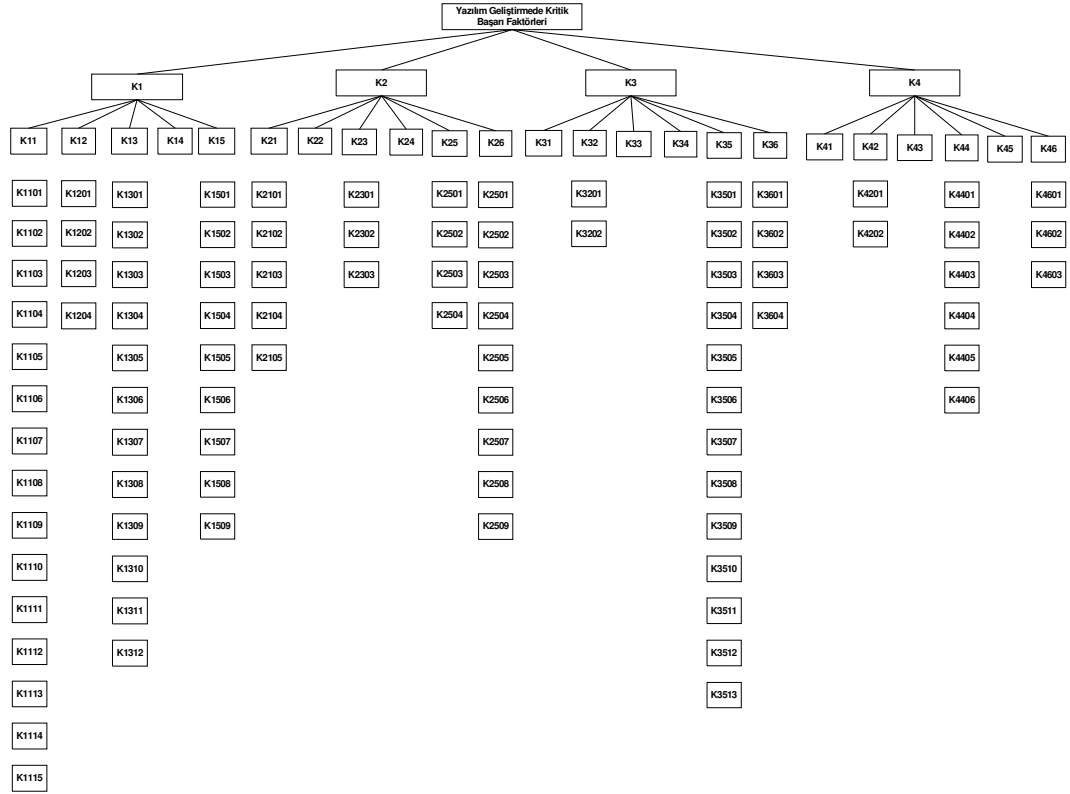
3.2 YÖNTEM

Yukarıda bahsedilen anketle, ÇYGP'nde Verimlilik için kritik başarı faktörlerini ortaya çıkarıp bu süreçlerin esnekleştirilmesinde ÇY'lerin katkıları hakkında mevcut durumu ortaya koyan bir çalışma yapılmış olup, *Bulanık Analitik Hiyerarşi Prosesi* yöntemi ile Verimlilik artışı için kullanılacak kritik başarı faktörleri ile ilgili karar verme modelinin oluşturulması hedeflenmiştir. Tablo 3.1'de belirtildiği gibi 5'li likert ölçeği kullanılmıştır.

Tablo 3.1: Araştırmada kullanılan likert ölçek

Çok olumlu etkiledi	Çok fazla	Tümüyle doğru	Çok önemli	5
Olumlu etkiledi	Büyük ölçüde	Büyük ölçüde doğru	Büyük ölçüde önemli	4
Kısmen olumlu etkiledi	Kısmen	Kararsızım	Kısmen önemli	3
Etkilemedi	Çok az	Büyük ölçüde yanlış	Çok az önemli	2
Olumsuz etkiledi	Hiç	Tümüyle yanlış	Hiç önemli değil	1

Yapılan literatür araştırmasından ve tecrübelerden edinilen bilgilere göre karar değişkenler düzenlenerek aşağıdaki tablolarda gösterilen 4 ana kriter ve 22 alt kriter ve 93 detay kriter elde edilmiştir.



Şekil 3.1 Araştırmada kullanılan karar kriterleri

Bu 22 alt kriter ve 92 detay kriter Bulanık AHP ile Yazılım Gelişirmede Başarı için doğru kriterlerin seçimini yapmak için karar kriterleri olarak kullanılacaktır. Kriterleri temsil eden değerler için Thomas Saaty'nin tavsiye ettiği şekilde geometrik ortalama ölçütü kullanılmıştır. Aşağıdaki tablolarda ÇYGP'de başarıyı etkileyen KBF karar kriterlerinin tanımları ve anket sonucunda elde edilen geometrik ortalamaları görülmektedir.

Tablo 3.2 Araştırmada kullanılan ana karar kriterleri ve geometrik ortalamaları

Ana Kriterler	Geo. Ort.
K1-Süreç iyileştirmeleri	4,57
K2-Organizasyonel iyileştirmeler	3,77
K3-Yazılım geliştirme ortamıyla ilgili iyileştirmeler	5
K4-Altyapısal iyileştirmeler	3,77

Tablo 3.3: Araştırmada kullanılan alt karar kriterler ve geometrik ortalamaları

Ana Kriter	Alt Kriter	Geometrik Ortalama
K1-Süreç	K11-Takip edilen süreç	4,182
K1-Süreç	K12-İşin kompleksliği	4,573
K1-Süreç	K13-Yapılan tasarım	5
K1-Süreç	K14-Desteklenen iş süreçleri	4,781
K1-Süreç	K15- Varolan standartlar	4,317
K2-Organizasyon	K21-Çalışanların karakter(dürüst,sorumluluk sahibi, güvenilir ve müzakereci) ve motivasyonu	4,128
K2-Organizasyon	K22-Ekip yetenek, eğitim ve tecrübesi	4,514
K2-Organizasyon	K23-Hedef ve beklentilerin netliği	4,514
K2-Organizasyon	K24-Yönetim desteği	4,128
K2-Organizasyon	K25-İsbirliği düzeyi	4,317
K2-Organizasyon	K26-İletişim düzeyi(Ekip içi ve müşteriyle yakın,hızlı ve etkin iletişim mümkünse yüz yüze)	4,514
K3-Geliştirme Ortamı	K31- Kullanılan programlama dili	4,781
K3-Geliştirme Ortamı	K32- Kullanılan yazılım çatısı	5
K3-Geliştirme Ortamı	K33- Kullanılan İde	3,68
K3-Geliştirme Ortamı	K34- Kullanılan platform	4,373
K3-Geliştirme Ortamı	K35- Kullanılan araçlar	4,573
K3-Geliştirme Ortamı	K36- Kullanılan algoritma	3,66
K4-Alt Yapı	K41-Veritabanı yapısı	4,317
K4-Alt Yapı	K42- Kullanılan mimari	4,573
K4-Alt Yapı	K43- Kullanılan donanım	2,93
K4-Alt Yapı	K44-Desteklenen teknik özellikler	4,128
K4-Alt Yapı	K45-Desteklenen fonksiyonel özellikler	4,373

Tablo 3.4: Süreçle ilgili alt ve detay karar kriterleri ve geometrik ortalamaları

Alt Kriter	Detay Kriter	Geometrik Ortalama
K11-Takip edilen süreç	K1101- Değişen İhtiyaçları Dikkate alınması	3,29
K11-Takip edilen süreç	K1102- Üst seviye use case kullanımı	3,03
K11-Takip edilen süreç	K1103- Detaylı use case kullanımı	2,49
K11-Takip edilen süreç	K1104- İterasyon planlama	4,18
K11-Takip edilen süreç	K1105- Sürüm planlama	4,37
K11-Takip edilen süreç	K1106- Test planlama	3,95
K11-Takip edilen süreç	K1107- Sürekli tüm süreçler için refactoring yaparak sorunlu noktaları ayıklamak	4,37
K11-Takip edilen süreç	K1108- Test güdümlü geliştirme	3,73
K11-Takip edilen süreç	K1109- Entegrasyon ve Kabul testleri	4,37
K11-Takip edilen süreç	K1110-Bağımsız- Kesif testleri	4,13
K11-Takip edilen süreç	K1111-Gelistirici testleri	4,13
K11-Takip edilen süreç	K1112-Çalışan yazılımın evrimsel paketlenmesi	4,13
K11-Takip edilen süreç	K1113-Tekrarlamalı geliştirme	3,73
K11-Takip edilen süreç	K1114-Sürdürülebilir tempoda ilerleme	3,95
K11-Takip edilen süreç	K1115-Yerel fayda yerine genel faydayı düşünme	3,89
K12-İşin Kompleksliği	K1201-Daha basit ürün geliştirme	4,57
K12-İşin Kompleksliği	K1202-Sadece müşterilerin ihtiyacı olan ve müşteriye değer katan özellikler geliştirmek	3,95
K12-İşin Kompleksliği	K1203-İhtiyaçları küçük senaryolara ayırarak önceliklendirip sıralamak ve en öncelikliden işe başlamak	4,57
K12-İşin Kompleksliği	K1204-Daha kısa sürümlerle daha az özellik geliştirerek müşteriye hızlı geri dönüş	3,95
K13-Yapılan Tasarım	K1301- Detaylı ihtiyaç analizleri yapılması	3,17
K13-Yapılan Tasarım	K1302-Ön Mimari modelleme	3,66
K13-Yapılan Tasarım	K1303-Ön İhtiyac modelleme	3,66
K13-Yapılan Tasarım	K1304-Bilgisayar Destekli Araçlarla modelleme	3,29
K13-Yapılan Tasarım	K1305-Paralel modelle	2,93
K13-Yapılan Tasarım	K1306-Model gözden gecirmeleri	3,1
K13-Yapılan Tasarım	K1307-Ana hatlarıyla mimari tasarım	3,95
K13-Yapılan Tasarım	K1308-Basit tasarım	4,18
K13-Yapılan Tasarım	K1309-Bakım Yapılabilir tasarım	4,37
K13-Yapılan Tasarım	K1310-Evrimsel tasarım	4,37
K13-Yapılan Tasarım	K1311-Tasarım gözden gecirme	3,95
K13-Yapılan Tasarım	K1312-Veri tabanı gözden gecirme	4,18
K13-Yapılan Tasarım	K1313-Arayüz gözden gecirme	4,18
K14-Desteklenen İş Süreçleri		
K15-Varolan Standartlar	K1501-Modelleme standartları	3,29
K15-Varolan Standartlar	K1502-Veri isimlendirme standartları	2,93
K15-Varolan Standartlar	K1503-Kodlama standartları	3,73
K15-Varolan Standartlar	K1504-Arayüz standartları	3,73
K15-Varolan Standartlar	K1505-Test standartları	3,44
K15-Varolan Standartlar	K1506-Versiyon Kurulum standartları	3,89
K15-Varolan Standartlar	K1507-Uyarlama standartları	3,95
K15-Varolan Standartlar	K1508-Problem çözme standartları	3,25
K15-Varolan Standartlar	K1509-Dokümantasyon standartları	2,7

Tablo 3.5: Organizasyonla ilgili alt ve detay karar kriterleri ve geometrik ortalamaları

Alt Kriter	Detay Kriter	Geometrik Ortalama
K21-Çalışan karakter ve motivasyonu	K2101- Küçük bir ekiple çalışmak	3,29
K21-Çalışan karakter ve motivasyonu	K2102-Kollektif Kod sahipliği	2,93
K21-Çalışan karakter ve motivasyonu	K2103-Kendi kendine organize olan takımlar	3,73
K21-Çalışan karakter ve motivasyonu	K2104-Eşli Programlama	3,73
K21-Çalışan karakter ve motivasyonu	K2105-Açık çalışma alanı	3,44
K22-Ekip yetenek,eğitim ve tecrubesi	K2201- Yetenekli ve ÇY konusunda Eğitilmiş ve Deneyimli personelle çalışmak	3,89
K23-Hedef ve beklentilerin netliği	K2301- Proje hedeflerini(neyi, ne zaman, hangi sırada, hangi maliyetle ve riskle) baştan doğru belirleyip tüm taraflarla paylaşmak.	3,95
K23-Hedef ve beklentilerin netliği	K2302- Ana iş süreçlerine odaklanmak	3,25
K23-Hedef ve beklentilerin netliği	K2303- Liderlik için belli birkaç alan seçmek	2,7
K24-Yönetim desteği	K2401-Çevik süreçlerin uygulanması konusunda ekibi teşvik	2,55
K25-İşbirliği Düzeyi	K2501- Üst seviye ihtiyaç analizinin müşteriyle birlikte yapılması	4,18
K25-İşbirliği Düzeyi	K2502- Ekip içi ve müşteriyle etkin işbirliği	4,37
K25-İşbirliği Düzeyi	K2503-Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak.	3,25
K25-İşbirliği Düzeyi	K2504-Proje Değerlendirme\Alınan Dersler' Paylaşım toplantısı	3,73
K26-İletişim Düzeyi	K2601-Tüm süreçlerin arasında entegre bilgi akışı sağlamak	4,37
K26-İletişim Düzeyi	K2602-Ekibin işleri bitirme hızı ,hata oranları ve hataların sebeplerini sürekli izlemek	4,51
K26-İletişim Düzeyi	K2603- Tüm taraflar arasında yüzyüze iletişimi sağlamak	4,51
K26-İletişim Düzeyi	K2604- Müşteriyle yerinde ve etkin iletişim kurmak	3,81
K26-İletişim Düzeyi	K2605-Günlük toplantılar yapma	4,18
K26-İletişim Düzeyi	K2606-Sürekli durum raporlarıyla yapılan, yapılacak ve kalan işlerin takibi	4,51
K26-İletişim Düzeyi	K2607-Hata Raporları	4,37
K26-İletişim Düzeyi	K2608-Hata Eğilim Metrikleri	3,95
K26-İletişim Düzeyi	K2609-Tüm Tarafların Aktif Katılımı	4,37

Tablo 3.6: Yazılım geliştirme ortamıyla ilgili alt ve detay karar kriterleri ve geometrik ortalamaları

Alt Kriter	Detay Kriter	Geometrik Ortalama
K31-Kullanılan Programlama Dili	K3101- Çevik ve Çevik ve Güncel yazılım geliştirme dilleri kullanmak	3,59
K32-Kullanılan Yazılım Çatısı	K3201-Zengin, hızlı geliştirme altyapıları kullanmak	3,95
K32-Kullanılan Yazılım Çatısı	K3202-Süreklili veri tabanı entegrasyonu	4,32
K33-Kullanılan Entegre Geliştirme Ortamı	K3301-Süreklili kod entegrasyonu	4,37
K34-Kullanılan Platform	K3401- Farklı işletim sistemleri(Windows,Unix) ve platform(web,mobil) desteği	3,52
K35-Kullanılan Araçlar	K3501- Xml ile esnek rapor üreticiler kullanmak	3,18
K35-Kullanılan Araçlar	K3502-Görev yönetimi, ihtiyaç analizi ve görevin gelişim durumunu takip için entegre proje yönetim araçları kullanmak	3,64
K35-Kullanılan Araçlar	K3503-Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek	3,73
K35-Kullanılan Araçlar	K3504-Manuel görev tahtası	3,95
K35-Kullanılan Araçlar	K3505-Dijital görev tahtası	3,03
K35-Kullanılan Araçlar	K3506-Otomatize kod geliştirme araçları kullanımı	4,78
K35-Kullanılan Araçlar	K3507-Otomatize arayüz geliştirme araçları	4,13
K35-Kullanılan Araçlar	K3508-Unit test araçları kullanımı	3,95
K35-Kullanılan Araçlar	K3509-Otomatize test araçlarıyla test	3,95
K35-Kullanılan Araçlar	K3510-Özet gant chartlar	3,29
K35-Kullanılan Araçlar	K3511-Detaylı gant chartlar	3,29
K35-Kullanılan Araçlar	K3512-Kanban	3,32
K35-Kullanılan Araçlar	K3513-Otomatize versiyon paketleme ve kurulumu	4,57
K36-Kullanılan Algoritma	K3601- Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak	4,18
K36-Kullanılan Algoritma	K3602-Tekrarlı işleri kaldırılması	4,78
K36-Kullanılan Algoritma	K3603-Bileşenlerin yeniden kullanımı	4,57
K36-Kullanılan Algoritma	K3604-Davranış Güdümlü Programlama	3,32

Tablo 3.7: Alt yapı ilgili alt ve detay karar kriterleri ve geometrik ortalamaları

Alt Kriter	Detay Kriter	Geometrik Ortalama
K41-Veritabanı yapısı	K4101- Farklı veritabanlarını desteklemek	2,86
K42-Kullanılan mimari	K4201- Servis tabanlı mimari kullanmak	2,3
K42-Kullanılan mimari	K4202-Model güdümlü mimari yapı kullanmak	2,49
K43-Kullanılan donanım	K4301- Farklı donanımlarda(PC, Mobil Cihazlar...) çalışabilecek yazılım alt yapısı kullanmak	3,73
K44-Desteklenen teknik özellikler	K4401-Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği	2,49
K44-Desteklenen teknik özellikler	K4402-Web servis uyumlu adapter alt yapısı kullanmak	2,61
K44-Desteklenen teknik özellikler	K4403-Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme	2,64
K44-Desteklenen teknik özellikler	K4404-İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi	2,99
K44-Desteklenen teknik özellikler	K4405-Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak	3,13
K44-Desteklenen teknik özellikler	K4406-Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi	3,25
K45-Desteklenen fonksiyonel özellikler	K4501- Ajax desteğine sahip olmak	2
K45-Desteklenen fonksiyonel özellikler	K4502- Esnek yetkilendirme altyapısına sahip olmak	2,99
K45-Desteklenen fonksiyonel özellikler	K4503-Otomatik hata ayıklama mekanizmaları kullanımı	3,95

3.3 ARAŞTIRMA VE YÖNTEM

Bu bölümde *Bulanık Analitik Hiyerarşi Prosesi* analizi uygulanmıştır. Chang'ın Bulanık AHP'de Mertebe Analizi yöntemi kullanılmıştır. Bu yöntemin tercih sebebi, karmaşık problemlerin modellenmesinde kolaylık sağlaması, problemin yapısına uygun olması , grup karar verme sürecine uygun olması , kullanımının yaygın bir çok nitelikli karar verme tekniği olan Analitik Hiyerarşi Prosesi (AHP) tercih edilmiştir. İkili karşılaştırma yargılarındaki belirsizliği daha iyi ifade etmek amacıyla, Chang'ın Bulanık AHP'de Mertebe Analizi yöntemi kullanılmıştır. Bu yöntem, diğer bulanık AHP tekniklerine kıyasla, kullanımı yaygın, basit olduğu ve adımları AHP yöntemiyle benzerlik gösterdiği için tercih edilmiştir.

3.3.1 Chang'ın Bulanık AHP'de Mertebe Analizi Yönteminin Adımları

Chang(1996)'in yöntemine göre, her bir ölçüt alınır ve her bir hedef için mertebe analizi uygulanır. $x = \{x_1, x_2, \dots, x_n\}$ bir ölçüt kümesi ve $U = \{u_1, u_2, \dots, u_n\}$, bir amaç kümesi olsun. Böylece her bir ölçüt için m tane mertebe analiz değerleri elde edilir. Bu değerler şu şekilde gösterilir.

$M_1, M_2, \dots, M_i = 1, 2, \dots, L$

$M_{g_i}^1, M_{g_i}^2, \dots, M_{g_i}^m \quad (i = 1, 2, \dots, n)$

Burada tüm $M_{g_i}^j$ ($i = 1, 2, \dots, m$) 'ler üçgensel bulanık sayıdır.

Chang'ın mertebe analizinin adımları şu şekilde sıralanabilir:

Adım 1: Ölçüt i 'ye göre bulanık sentetik mertebenin değeri şu şekilde tanımlanır :

$$S_i = \sum_{j=1}^n M_{g_i}^j \otimes \left[\sum_{i=1}^n \sum_{j=1}^n M_{g_i}^j \right]^{-1} \quad (1)$$

Buradaki $\sum_{j=1}^n M_{g_i}^j$ değerini elde etmek için m tane mertebe analiz değerine bulanık

toplama işlemi uygulanır.

$$\sum_{j=1}^m M_{g_i}^j = \left(\sum_{j=1}^m l_i, \sum_{j=1}^m m_i, \sum_{j=1}^m u_i \right) \quad (2)$$

$$\sum_{i=1}^n \sum_{j=1}^m M_{g_i}^j = \left(\sum_{i=1}^n l_i, \sum_{i=1}^n m_i, \sum_{i=1}^n u_i \right) \quad (3)$$

Daha sonra vektörün tersi şu şekilde elde edilir :

$$\left[\sum_{i=1}^n \sum_{j=1}^m M_{g_i}^j \right]^{-1} = \left(\frac{1}{\sum_{i=1}^n u_i}, \frac{1}{\sum_{i=1}^n m_i}, \frac{1}{\sum_{i=1}^n l_i} \right) \quad (4)$$

Adım 2: $M_2 = (l_2, m_2, u_2) \geq M_1 = (l_1, m_1, u_1)$ 'nin olabilirlik derecesi şu şekilde

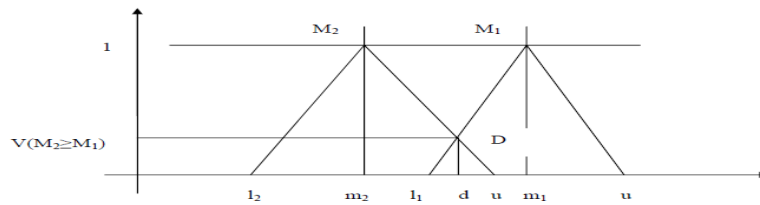
tanımlanır :

$$V(M_2 \geq M_1) = \sup_{y \geq x} [\min \mu_{M_1}(x), \mu_{M_2}(y)] \quad (5)$$

Denk olarak aşağıdaki gibi de ifade edilebilir:

$$V(M_2 \geq M_1) = hgt(M_1 \cap M_2) = \mu_{M_2}(d) = \begin{cases} 1, & m_2 \geq m_1 \\ 0, & l_1 \geq u_2, \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)} & \text{diğer} \end{cases} \quad (6)$$

Denk şekilde $V(M_2 \geq M_1)$ 'i, d , μ_{M_1} ve μ_{M_2} arasındaki en yüksek kesişim noktası D 'nin ordinatı olmak üzere Şekil 3.2'de görüldüğü gibi gösterilmiştir.



Şekil 3.2: Bulanık M_2 sayısının M_1 sayısından büyük olabilirlik derecesi

Kaplan (2007) e göre;

M1 ve M2'yi kıyaslayabilmek için $V(M_2 > M_1)$ ve $V(M_1 \geq M_2)$ değerlerinin her ikisi de gerekmektedir.

Adım 3: Bir konveks bulanık sayının k tane konveks bulanık sayıdan M_i $i = 1, 2, \dots, k$ büyük olmasının olabilirlik derecesi şu şekilde tanımlanır.

$$\begin{aligned} V(M \geq M_1, M_2, \dots, M_k) &= V[(M \geq M_1) \text{ve} (M \geq M_2) \text{ve} \dots \text{ve} (M \geq M_k)] \\ &= \min V(M \geq M_i), \quad i = 1, 2, \dots, k \end{aligned}$$

$$d'(A_i) = \min V(S_i \geq S_k),$$

olduğunu varsayalım, $k = 1, 2, \dots, n$; $k \neq i$ için ağırlık vektörü aşağıdaki gibidir.

$$W' = (d'(A_1), d'(A_2), \dots, d'(A_n))^T$$

Burada A_i $i = 1, 2, \dots, n$ n sayısı kadardır.

Adım 4: Normalize edilmiş ağırlık vektörleri, aşağıdaki gibidir. Burada W , bulanık olmayan bir sayıdır.

$$W = (d(A_1), d(A_2), \dots, d(A_n))^T$$

Chang'ın Bulanık AHP tekniğinde son olarak, her bir karşılaştırma tablosu için elde edilen ağırlıklar AHP tekniğinde olduğu gibi hiyerarşik olarak sentezlenmekte ve nihai alternatif ağırlıkları bulunmaktadır.

4. BULGULAR

Bu bölümde yapılan anket çalışmasından elde edilen verilere, BAHP yönteminin adım adım uygulanması ve böylece ÇYGP’de KBF’nin belirlenmesi anlatılacaktır.

4.1 ÖLÇÜTLER ARASINDA ÖNEM SIRALARININ BELİRLENMESİ

Ölçütler arasında hedefe göre ikili karşılaştırma sonucunda hangi ölçütün daha önemli olduğu bulunacaktır. Bu amaçla ankete katılanlardan ölçütler arasında önem derecesine göre 1 ile 5 arasında likert tipi ölçek ile ağırlıklandırma yapmaları istenmiştir. Bu veriler baz alınarak, çıkan değerlere 1-9 temel ölçeğine göre yapılan aralıklandırma sonucunda değerler atanmış ve ikili karşılaştırmalarda bu değerler kullanılmıştır.

Tablo 4.1: Temel ölçeğine göre yapılan aralıklandırmalar(Köstence N., 2009)

Temel ölçek değerleri	Ölçütler arası oran değerleri		
1	1		
2	1	-	1,5
3	1,5	-	2
4	2	-	2,5
5	2,5	-	3
6	3	-	3,5
7	3,5	-	4
8	4	-	4,5
9	4,5	-	5

Kriterlerin geometrik ortalama değerleri, oranlanarak kriterler arası önem oranları bulunmuştur. Örnek olarak Süreç ile Organizasyonel kriterleri arasındaki oran;

$$\text{Süreç / Organizasyonel} = 4,57 / 3,77 = 1,12$$

$$\text{Organizasyon / Süreç} = 3,77 / 4,57 = 0,88$$

Bunun anlamı; Süreç kriterine verilen puanların geometrik ortalaması ile Organizasyonel kriterlere verilen puanların geometrik ortalamasının 1,12 katıdır veya bir başka ifade ile Organizasyonel kriterlere verilen puanların geometrik ortalaması, Süreç kriterine verilen puanların geometrik ortalamasının 0,88 katıdır.

1,12 oranı Temel ölçüğe göre yapılan aralıklandırmalar tablosunda 2 değerini almaktadır. Bu nedenle, Fonksiyonellik kriteri, Teknik altyapı kriterine göre 2 kat önemlidir diyebiliriz. Diğer karşılaştırmalarda oranlamaya göre aralık değerleri verilerek ikili karşılaştırmalara tablosu elde edilmiştir. Aşağıda Tablo 4.2 ile 4.20 arasında ana kriter, alt kriter ve detay kriterlerin ikili karşılaştırma tabloları görülebilir.

Tablo 4.2: Ana kriterler için ikili karşılaştırma tablosu

AnaKriter-Açıklama	K1	K2	K3	K4
K1-Süreç	1	2	1	1
K2-Organizasyon	0,5	1	1	1
K3-Yazılım geliştirme ortamı	1	1	1	2
K4-Altyapı	1	1	0,5	1

Tablo 4.3: Süreç kriterine ait alt kriterler için ikili karşılaştırma tablosu

Alt Kriter-Açıklama	K11	K12	K13	K14
K11-Takip edilen süreç	1	1	1	1
K12-İşin kompleksliği	1	1	1	1
K13-Yapılan tasarım	1	1	1	2
K14-Desteklenen iş süreçleri	1	1	0,5	1
K15- Varolan standartlar	1	0,5	0,5	1

Tablo 4.4: Organizasyon kriterine ait alt kriterler için ikili karşılaştırma tablosu

Alt Kriter-Açıklama	K21	K22	K23	K24	K25	K26
K21-Çalışanların karakter ve motivasyonu	1	1	1	1	1	1
K22-Ekip yetenek, eğitim ve tecrübesi	1	1	1	2	2	1
K23-Hedef ve beklentilerin netliği	1	1	1	2	2	1
K24-Yönetim desteği	1	0,5	0,5	1	1	1
K25-İsbirliği düzeyi	1	0,5	0,5	1	1	1
K26-İletişim düzeyi	1	1	1	1	1	1

Tablo 4.5: Geliştirme Ortamı kriterine ait alt kriterler için ikili karşılaştırma tablosu

Alt Kriter-Açıklama	K31	K32	K33	K34	K35	K36
K31- Kullanılan programlama dili	1	1	2	2	2	2
K32- Kullanılan yazılım kütüphanesi	1	1	1	2	2	2
K33- Kullanılan İde	0,5	1	1	1	1	2
K34- Kullanılan platform	0,5	0,5	1	1	1	2
K35- Kullanılan araçlar	0,5	0,5	1	1	1	2
K36- Kullanılan algoritma	0,5	0,5	0,5	0,5	0,5	1

Tablo 4.6: Alt yapı kriterine ait alt kriterler için ikili karşılaştırma tablosu

Alt Kriter-Açıklama	K41	K42	K43	K44	K45
K41-Veritabanı yapısı	1	1	2	2	1
K42- Kullanılan mimari	1	1	3	2	2
K43- Kullanılan donanım	0,5	0,33	1	1	1
K44-Desteklenen teknik özellikler	0,5	0,5	1	1	1
K45-Desteklenen fonksiyonel özellikler	1	0,5	1	1	1

Tablo 4.7: Takip Edilen Süreç detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K1101	K1102	K1103	K1104	K1105	K1106	K1107	K1108	K1109	K1110	K1111	K1112	K1113	K1114	K1115
K1101- Değişen İhtiyaçları Dikkate alınması	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1
K1102- Üst seviye use case kullanımı	0,5	1	2	1	1	1	1	1	1	1	1	1	1	1	1
K1103- Detaylı use case kullanımı	0,5	0,5	1	1	1	1	1	1	1	1	1	1	1	1	1
K1104- İterasyon planlama	1	1	1	1	1	2	1	2	1	2	2	2	2	2	2
K1105- Sürüm planlama	1	1	1	1	1	2	1	2	1	2	2	2	2	2	2
K1106- Test planlama	1	1	1	0,5	1	1	1	2	1	1	1	1	2	1	2
K1107- Refactoring yaparak sorunlu noktaları açıklamak	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
K1108- Test güdümlü geliştirme	1	1	1	0,5	1	1	1	1	1	1	1	1	1	1	1
K1109- Entegrasyon ve Kabul testleri	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
K1110-Bağımsız- Kesif testleri	1	1	1	0,5	1	1	1	1	1	1	1	2	2	2	2
K1111-Gelistirici testleri	1	1	1	0,5	1	1	1	1	1	1	1	1	2	2	2
K1112-Çalışan yazılımın evrimsel paketlenmesi	1	1	1	0,5	1	1	1	1	1	1	1	1	2	2	2
K1113-Tekrarlamalı geliştirme	1	1	1	0,5	1	1	1	1	1	1	1	1	1	1	1
K1114-Sürdürülebilir tempoda ilerleme	1	1	1	0,5	1	1	1	1	1	1	1	1	1	1	2
K1115-Yerel fayda yerine genel faydayı düşünme	1	1	1	0,5	1	1	1	1	1	1	1	1	1	1	1

Tablo 4.8: İşin Kompleksliği detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K1201	K1202	K1203	K1204
K1201-Daha basit ürün geliştirme	1	2	1	2
K1202-Sadece müşterilerin ihtiyacı olan ve müşteriye değer katan özellikler geliştirmek	0,5	1	1	1
K1203-İhtiyaçları küçük senaryolara ayırarak önceliklendirip sıralamak ve en öncelikliden işe başlamak	1	1	1	2
K1204-Daha kısa sürümlerle daha az özellik geliştirerek müşteriye hızlı geri dönüş	0,5	1	0,5	1

Tablo 4.9: Yapılan Tasarım detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K1301	K1302	K1303	K1304	K1305	K1306	K1307	K1308	K1309	K1310	K1311	K1312	K1313
K1301- Detaylı ihtiyaç analizleri yapılması	1	1	1	1	2	2	1	1	1	1	1	1	1
K1302-Ön Mimari modelleme	1	1	1	2	2	2	1	1	1	1	1	1	1
K1303-Ön İhtiyac modelleme	1	1	1	2	2	2	1	1	1	1	1	1	1
K1304-Bilgisayar Destekli Araçlarla modelleme	1	0,5	0,5	1	2	2	1	1	1	1	1	1	1
K1305-Paralel modelleme	0,5	0,5	0,5	0,5	1	1	1	1	1	1	1	1	1
K1306-Model gözden geçirmeleri	0,5	0,5	0,5	0,5	1	1	1	1	1	1	1	1	1
K1307-Ana hatlarıyla mimari tasarım	1	1	1	1	1	1	1	1	1	1	1	1	1
K1308-Basit tasarım	1	1	1	1	1	1	1	1	1	1	2	1	1
K1309-Bakım Yapılabilir tasarım	1	1	1	1	1	1	1	1	1	1	2	2	2
K1310-Evrimsel tasarım	1	1	1	1	1	1	1	1	1	1	2	2	2
K1311-Tasarım gözden geçirme	1	1	1	1	1	1	1	0,5	0,5	0,5	1	1	1
K1312-Veri tabanı gözden geçirme	1	1	1	1	1	1	1	1	0,5	0,5	1	1	1
K1313-Arayuz gözden geçirme	1	1	1	1	1	1	1	1	0,5	0,5	1	1	1

Tablo 4.10: Varolan Standartlar detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K1501	K1502	K1503	K1504	K1505	K1506	K1507	K1508	K1509
K1501-Veri isimlendirme standartları	0,5	1	1	1	1	1	1	1	2
K1502-Veri isimlendirme standartları	0,5	1	1	1	1	1	1	1	2
K1503-Kodlama standartları	1	1	1	1	2	1	1	2	2
K1504-Arayüz standartları	1	1	1	1	2	1	1	2	2
K1505-Test standartları	1	1	0,5	0,5	1	1	1	2	2
K1506-Versiyon Kurulum standartları	1	1	1	1	1	1	1	2	2
K1507-Uyarılama standartları	1	1	1	1	1	1	1	2	2
K1508-Problem çözme standartları	0,5	1	0,5	0,5	0,5	0,5	0,5	1	2
K1509-Dokumantasyon standartları	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1

Tablo 4.11: Çalışanların karakter özellikleri ve motivasyonu detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K2101	K2102	K2103	K2104	K2105
K2101- Küçük bir ekiple çalışmak	1	1	1	1	1
K2102-Kollektif Kod sahipliği	1	1	1	2	2
K2103-Kendi kendine organize olan takımlar	1	1	1	2	2
K2104-Eşli Programlama	1	0,5	0,5	1	1
K2105-Açık çalışma alanı	1	0,5	0,5	1	1

Tablo 4.12: Hedef ve Beklentilerin Netliği detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K2301	K2302	K2303
K2301- Proje hedeflerini(neyi, ne zaman, hangi sırada, hangi maliyetle ve riskle) baştan doğru belirleyip tüm taraflarla paylaşmak.	1	1	2
K2302- Ana iş süreçlerine odaklanmak	1	1	2
K2303- Liderlik için belli birkaç alan seçmek	0,5	0,5	1

Tablo 4.13: Ekip içi ve müşteriyle etkin işbirliği detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K2501	K2502	K2503	K2504
K2501- Üst seviye ihtiyaç analizinin müşteriyle birlikte yapılması	1	1	1	2
K2502- Ekip içi ve müşteriyle etkin işbirliği	1	1	2	2
K2503-Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak.	1	0,5	1	2
K2504-Proje Degerlendirme\Alinan Dersler' Paylasim toplantisi	0,5	0,5	0,5	1

Tablo 4.14: İletişim Düzeyi detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K26 01	K26 02	K26 03	K26 04	K26 05	K26 06	K26 07	K26 08	K26 09
K2601-Tüm süreçlerin arasında entegre bilgi akışı sağlamak	1	1	1	1	2	1	2	2	1
K2602-Ekibin işleri bitirme hızı ,hata oranları ve hataların sebeplerini sürekli izlemek	1	1	1	1	2	1	2	2	1
K2603- Tüm taraflar arasında yüzyüze iletişimi sağlamak.	1	1	1	1	2	1	2	2	1
K2604- Müşteriyile yerinde ve etkin iletişim kurmak	1	1	1	1	2	1	2	2	1
K2605-Günlük toplantılar yapma	0,5	0,5	0,5	0,5	1	1	1	1	1
K2606-Sürekli durum raporlarıyla yapılan, yapılacak işlerin takibi	1	1	1	1	1	1	2	2	2
K2607-Hata Raporları	0,5	0,5	0,5	0,5	1	0,5	1	1	1
K2608-Hata Eğilim Metrikleri	0,5	0,5	0,5	0,5	1	0,5	1	1	1
K2609-Tum Taraflarin Aktif Katılımı	1	1	1	1	1	0,5	1	1	1

Tablo 4.15: Kullanılan Yazılım Çatısı detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K3201	K3202
K3201-Zengin, hızlı geliştirme altyapıları kullanmak	1	1
K3201-Surekli veri tabani entegrasyonu	1	1

Tablo 4.16: Kullanılan Araçlar detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K35 01	K35 02	K35 03	K35 04	K35 05	K35 06	K35 07	K35 08	K35 09	K35 10	K35 11	K35 12	K35 13
K3501- Xml ile esnek rapor üreticiler kullanmak	1	1	1	1	2	1	1	1	1	1	1	1	1
K3502-Görev yönetimi, İhtiyaç analizi ve görevin gelişim durumunu takip için entegre proje yönetim araçları kullanmak	1	1	1	1	2	1	1	1	1	2	2	2	1
K3503-Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek	1	1	1	1	2	1	1	1	1	2	2	2	1
K3504-Manuel görev tahtası	1	1	1	1	2	1	1	1	1	2	2	2	1
K3505-Dijital görev tahtası	0,5	0,5	0,5	1	1	1	1	1	1	1	1	1	1
K3506-Otomatize kod geliştirme araçları kullanımı	1	1	1	1	1	1	2	2	2	2	2	2	2
K3507-Otomatize arayüz geliştirme araçları	1	1	1	1	1	0,5	1	2	2	2	2	2	1
K3508-Unit test araçları kullanımı	1	1	1	1	1	0,5	0,5	1	1	2	2	2	1
K3509-Otomatize test araçlarıyla test	1	1	1	1	1	0,5	0,5	1	1	2	2	2	1
K3510-Özet gant chartlar	1	0,5	0,5	1	1	0,5	0,5	0,5	0,5	1	1	1	1
K3511-Detaylı gant chartlar	1	0,5	0,5	1	1	0,5	0,5	0,5	0,5	1	1	1	1
K3512-Kanban	1	0,5	0,5	1	1	0,5	0,5	0,5	0,5	1	1	1	1
K3513-Otomatize versiyon paketleme ve kurulumu	1	1	1	1	1	0,5	0,5	1	1	1	1	1	1

Tablo 4.17: Kullanılan Algoritma detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K3601	K3602	K3603	K3604
K3601- Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak	1	1	1	2
K3602-Tekrarlı işleri kaldırılması	1	1	2	2
K3603-Bileşenlerin yeniden kullanımı	1	0,5	1	2
K3604-Davranış Güdümlü Programlama	0,5	0,5	0,5	1

Tablo 4.18: Kullanılan Mimari detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K4201	K4202
K4201- Servis tabanlı mimari kullanmak	1	1
K4202-Model güdümlü mimari yapı kullanmak	1	1

Tablo 4.19: Desteklenen Teknik Özellikler detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K4401	K4402	K4403	K4404	K4405	K4406
K4401-Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği	1	1	1	1	1	1
K4402-Web servis uyumlu adapter alt yapısı kullanmak	1	1	1	1	1	1
K4403-Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme	1	1	1	1	1	1
K4404-İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi	1	1	1	1	1	1
K4405-Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak	1	1	1	1	1	1
K4406-Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi	1	1	1	1	1	1

Tablo 4.20: Desteklenen Funksiyonel Özellikler Kullanılan Algoritma detay kriterleri için ikili karşılaştırma tablosu

Detay Kriter-Açıklama	K4501	K4502	K4503
K4501- Ajax desteğine sahip olmak	1	1	1
K4502- Esnek yetkilendirme altyapısına sahip olmak	1	1	1
K4503-Otomatik hata ayıklama mekanizmaları kullanımı	1	1	1

4.2 İKİLİ KARŞILAŞTIRMA MATRİSİNİN TUTARLILIK İNCELEMESİ

İkili karşılaştırma matrisleri, insan merkezli değerlendirmeler ile subjektif olarak elde edildiğinden dolayı tutarlılığının incelenmesi gerekmektedir. Bu incelemede tutarlılık değerini gösteren tesadüfilik göstergeleri aşağıdaki tablo 4.21'deki gibi literatürde yer almaktadır.

Tablo 4.21: Tesadüfilik göstergeleri

Tesadüfilik matris boyutu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tesadüfilik göstergesi	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49	1,51	1,48	1,56	1,57	1,59

Tutarlılık için $[\lambda(\text{maksimum})-n]/(n-1)$ formülünden hesaplanan deęerin, tesadüfi gösterge deęerine bölünmesi ile elde edilen oranın %10'u aşmaması gerekmektedir.

4.2.1.Kriter ikili Karşılaştırma Matrisinin Tutarlılığı

Kriter ikili karşılaştırma matrisinin tutarlılığını incelemek için matris normalize edilmiştir. İkili karşılaştırmalar matrisi, normalize edilen matrisin görelî önemler vektörü ile çarpılıp toplanmıştır. Bu deęer ana kriterler için $\lambda(\text{maksimum})=4,12$ olarak bulunmuştur. Tutarlılık derecesinin ölçülebilmesi için $[\lambda(\text{maksimum})-n]/(n-1)$ deęerinin hesaplanması gerekmektedir. 4 boyutlu matris için bu deęer $[4,12-4]/10= 0,012$ olarak bulunmuştur. Hesapladığımız deęeri 4 boyutlu matris için verilen 0,9 sayısına böler isek, $0,012/0,9 = 0,013$ olarak bulunur. Dięer bir deyişle %1,3 olarak bulunur. Hesaplanan bu deęer %10 deęerini aşmadığından dolayı ikili karşılaştırmalar matrisinin tutarlı olduęu kanıtlanmıştır. Bu hesaplama dięer alt kriterlere de uygulandığında dięer alt kriterlerin de tutarlı olduęu görülmektedir.

Aşağıda tablo 4.22 ve 4.23'te tüm kriterlere ait tutarlılık deęerleri, ağırlık ve önem sıralıyla birlikte verilmiştir.

Tablo 4.22: AHP ye Göre Süreç ve Organizasyon Ana Kriterler İçin Görelî Ağırlık, Önem Sıra ve Tutarlılık Bilgileri

Ana Kriter	Ağırlık	Sıra	Alt Kriter	Ağırlık	Sıra	λ_{max}	C.I.	Detay Kriter	Ağırlık	Sıra	λ_{max}	C.I.	Genel Ağırlık	Genel Sıra	Genel λ_{max}	Genel C.I.
K1	0,293	1	K11	0,2	2	5,117	0,029	K1101	0,07	3			0	33	4,12	0,040
								K1102	0,07	3			0	33		
								K1103	0,06	4			0	35		
								K1104	0,09	1			0,01	26		
								K1105	0,09	1			0,01	26		
								K1106	0,07	3			0	33		
								K1107	0,08	2			0	29		
								K1108	0,06	4			0	35		
								K1109	0,08	2			0	29		
								K1110	0,07	3			0	33		
								K1111	0,06	4			0	35		
								K1112	0,06	4			0	35		
								K1113	0,05	5			0	37		
								K1114	0,05	5			0	37		
								K1115	0,05	5			0	37		
			K12	0,22	2			K1201	0,34	1	4,1	0,02	0,03	14		
								K1202	0,2	3			0,02	17		
								K1203	0,29	2			0,02	15		
								K1204	0,17	4			0,01	18		
			K13	0,26	1			K1301	0,08	2			0,01	24		
								K1302	0,09	1			0,01	22		
								K1303	0,09	1			0,01	22		
								K1304	0,08	2			0,01	24		
								K1305	0,06	4			0	29		
								K1306	0,06	4			0	29		
								K1307	0,07	3			0,01	25		
								K1308	0,08	2			0,01	24		
								K1309	0,09	1			0,01	22		
								K1310	0,09	1			0,01	22		
								K1311	0,07	3			0,01	25		
								K1312	0,07	2			0,01	25		
								K1313	0,07	3			0,01	25		
			K14	0,1	2			K14	0,095				0,03	11		
			K15	0,13	2			K1501	0,13	1	9,2	0	0	34		
								K1502	0,11	3			0	31		
								K1503	0,13	1			0,01	27		
								K1504	0,13	1			0,01	27		
								K1505	0,11	3			0	31		
								K1506	0,12	2			0	29		
								K1507	0,12	2			0	29		
								K1508	0,07	4			0	38		
K2	0,207	2	K21	0,16	2	6,107	0,022	K2101	0,2	2	5,1	0,019	0,01	22		
								K2103	0,25	1			0,01	21		
								K2104	0,25	1			0,01	21		
								K2105	0,15	3			0,01	26		
								K2106	0,15	3			0,01	26		
			K22	0,21	1			K22	0,21				0,04	6		
			K23	0,21	1			K2301	0,4	1	3	0	0,02	16		
								K2302	0,4	1			0,02	16		
								K2303	0,2	2			0,01	21		
			K24	0,13	2			K24	0,13				0,03	13		
			K25	0,13	2			K2501	0,24	2	4,1	0,04	0,01	23		
								K2502	0,34	2	9,1	0	0,01	20		
								K2503	0,24	1			0,01	23		
								K2504	0,17	1			0	29		
			K26	0,16	2			K2601	0,13	2			0	30		
								K2602	0,13	2			0	30		
								K2603	0,13	3			0	30		
								K2604	0,13	2			0	30		
								K2605	0,08	3			0	38		
								K2606	0,14	1			0	28		
								K2607	0,07	5			0	39		
								K2608	0,07	5			0	39		
								K2609	0,1	4			0	35		

Tablo 4.23: AHP ye Göre Geliştirme Ortamı ve Altyapı Ana Kriterler İçin Görelî Ağırlık, Önem Sıra Tutarlılık ve Bilgileri

Ana Kriter	Ağırlık	Sıra	Alt Kriter	Ağırlık	Sıra	λ_{max}	C.I.	Detay Kriter	Ağırlık	Sıra	λ_{max}	C.I.	Genel Ağırlık	Genel Sıra	Genel λ_{max}	Genel C.I.
K3	0,293	1	K31	0,25	2	6,108	0,022	K31	0,25				0,07	2		
			K32	0,22	1			K3201	0,5	1	2	0	0,03	9		
								K3202	0,5	1			0,03	9		
			K33	0,16	1			K33	0,158				0,05	5		
			K34	0,14	2			K34	0,14				0,04	7		
			K35	0,14	2			K3501	0,08	3			0	36		
								K3502	0,09	2			0	32		
								K3503	0,09	2			0	32		
								K3504	0,09	2			0	32		
								K3505	0,06	3			0	36		
								K3506	0,11	1			0,01	27		
								K3507	0,09	2			0	32		
								K3508	0,08	3			0	36		
								K3509	0,08	3			0	36		
								K3510	0,05	5			0	41		
								K3511	0,05	5			0	41		
								K3512	0,05	5			0	41		
								K3513	0,07	4			0	36		
			K36	0,09	3			K3601	0,28	2	4,1	0	0,07	2		
								K3602	0,34	1			0,09	1		
								K3603	0,24	3			0,06	3		
								K3604	0,14	4			0,04	8		
K4	0,207	2	K41	0,25	2	5,078	0,020	K41	0,251				0,05	4		
			K42	0,31	1			K4201	0,5	1	2	0	0,03	10		
								K4202	0,5	1			0,03	10		
			K43	0,13	1			K43	0,132				0,03	12		
			K44	0,14	1			K4401	0,17	1	6	0	0,01	27		
								K4402	0,17	1			0,01	27		
								K4403	0,17	1			0,01	27		
								K4404	0,17	1			0,01	27		
								K4405	0,17	1			0,01	27		
								K4406	0,17	1			0,01	27		
			K45	0,17	1			K4501	0,33	1	3	0	0,01	19		
								K4502	0,33	1			0,01	19		
								K4503	0,33	1			0,01	19		

AHP' ye göre çıkan sonuçlar incelendiğinde en önemli 10 uygulama şunlardır;

- 1) Tekrarlı işleri kaldırılması, 2) Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak, 3) Bileşenlerin yeniden kullanımı, 4) Sürekli kod entegrasyonu,
- 5) Yetenekli ve ÇY konusunda Eğitimli ve Deneyimli personelle çalışmak, 6) Farklı işletim sistemleri(Windows,Unix) ve platform(web,mobil) desteği, 7) Davranış Güdümlü Programlama, 8) Zengin, hızlı geliştirme altyapıları kullanmak, 9) Sürekli veri tabanı entegrasyonu, 10) Servis tabanlı ve model güdümlü mimari yapı kullanmak.

4.2.2 Bulanık Analitik Hiyerarşi Prosesi İçin İkili Karşılaştırmalar Matrisinin Bulunması

Kullanılan üçgensel sayılardan oluşan kriterler ve bulanık karşılaştırma ölçek ve değerleri tablodaki gibidir.

Tablo 4.24: Örnek alınan kriter bulanık karşılaştırma ölçek ve değerleri

Dilsel Ölçek	Üçgensel Bulanık Ölçek	Üçgensel Bulanık Karşılık Ölçek
Tamamen eşit	(1, 1, 1)	(1, 1, 1)
Eşit Önemde	(1/2, 1, 3/2)	(2/3, 1, 2)
Biraz daha fazla önemli	(1, 3/2, 2)	(1/2, 2/3, 1)
Kuvvetli derecede önemli	(3/2, 2, 5/2)	(2/5, 1/2, 2/3)
Çok kuvvetli derecede önemli	(2, 5/2, 3)	(1/3, 2/5, 1/2)
Tamamıyla önemli	(5/2, 3, 7/2)	(2/7, 1/3, 2/5)

Bulanık ölçek değerleri, 3'lü bulanık sayılar ile tanımlanmış, karşılık ölçek değerleri, her bir bulanık ölçek değerinin tersi olarak hesaplanmıştır. Kriterler için önceki bölümde belirlenen ikili karşılaştırma matrislerinin değerleri yukarıdaki ölçekten yararlanılarak kriterlerin biribiri için görece bulanık önemlerini gösteren bulanık ikili karşılaştırma matrisine çevrilmiştir. Örneğin Süreç Organizasyona göre biraz daha önemde olduğu için ölçek değeri olarak 1, 1.5, 2 e karşılık 0.5, 0.67, 1 alınmıştır. Aşağıda Tablo 4.25 ile 4.41 arasında bulanık ikili karşılaştırma matrisleri ana kriterlerden detaya doğru gösterilmiştir.

Tablo 4.25: Ana Kriterlere ait bulanık ikili karşılaştırma matrisi

Ana Kriterler		K1		K2		K3		K4				
K1-Süreç	1	1	1	1	1.5	2	0.5	1	1.5	0.5	1	1.5
K2-Organizasyon	0.5	0.67	1	1	1	1	0.5	1	1.5	0.5	1	1.5
K3-Yazılım geliştirme ortamı	0.67	1	2	0.67	1	2	1	1	1	1	1.5	2
K4-Altyapı	0.67	1	2	0.67	1	2	0.5	0.67	1	1	1	1

Tablo 4.26: Süreç bulanık ikili karşılaştırma matrisi

Alt Kriter-Açıklama		K11		K12		K13		K14		K15		
K11-Takip edilen süreç	1	1	1	0.5	1	1.5	0.5	1	1.5	0.5	1	1.5
K12-İşin kompleksliği	0.67	1	2	1	1	1	0.5	1	1.5	0.5	1	1.5
K13-Yapılan tasarım	0.67	1	2	0.67	1	2	1	1	1	1.5	2	1
K14-Desteklenen iş süreçleri	0.67	1	2	0.67	1	2	0.5	0.67	1	1	1	0.5
K15- Varolan standartlar	0.67	1	2	0.5	0.67	1	0.5	0.67	1	0.67	1	2

Tablo 4.27: Organizasyon bulanık ikili karşılaştırma matrisi

Alt Kriter-Açıklama		K21		K22		K23		K24		K25		K26	
K21-Çalışanların karakter ve motivasyonu	1	1	1	0.5	1	1.5	0.5	1	1.5	0.5	1	1.5	0.5
K22-Ekip yetenek, eğitim ve tecrübesi	0.67	1	2	1	1	1	0.5	1	1.5	1	1.5	2	0.5
K23-Hedef ve beklentilerin netliği	0.67	1	2	0.67	1	2	1	1	1	1.5	2	1	0.5
K24-Yönetim desteği	0.67	1	2	0.5	0.67	1	0.5	0.67	1	1	1	0.5	1
K25-İsbirliği düzeyi	0.67	1	2	0.5	0.67	1	0.5	0.67	1	0.67	1	2	1
K26-İletişim düzeyi	0.67	1	2	0.67	1	2	0.67	1	2	0.67	1	2	1

Tablo 4.28: Geliştirme Ortamına ait bulanık ikili karşılaştırma matrisi

Alt Kriter-Açıklama		K31		K32		K33		K34		K35		K36
K31- Kullanılan programlama dili	1	1,00	1	0,5	1,00	1,5	0,5	1,00	1,5	0,5	1,00	1,5
K32- Kullanılan Yazılım Çatısı	0,67	1,00	2	1	1,00	1	0,5	1,00	1,5	1	1,50	2
K33- Kullanılan İde	0,67	1,00	2	0,67	1,00	2	1	1,00	1	1	1,50	2
K34- Kullanılan platform	0,67	1,00	2	0,5	0,67	1	0,5	0,67	1	1	1,00	1
K35- Kullanılan araçlar	0,67	1,00	2	0,5	0,67	1	0,5	0,67	1	0,67	1,00	2
K36- Kullanılan algoritma	0,67	1,00	2	0,67	1,00	2	0,67	1,00	2	0,67	1,00	2

Tablo 4.29: Altyapıya ait bulanık ikili karşılaştırma matrisi

Alt Kriter-Açıklama		K41		K42		K43		K44		K45
K41-Veritabanı yapısı	1	1,00	1	0,5	1,00	1,5	1	1,50	2	1
K42- Kullanılan mimari	0,67	1,00	2	1	1,00	1	1,5	2,00	2,5	1
K43- Kullanılan donanım	0,5	0,67	1	0,4	0,50	0,67	1	1,00	1	0,5
K44-Desteklenen teknik özellikler	0,5	0,67	1	0,5	0,67	1	0,67	1,00	2	1
K45-Desteklenen fonksiyonel özellikler	0,67	1,00	2	0,5	0,67	1	0,67	1,00	2	0,67

Tablo 4.30: Takip edilen süreç ait bulanık ikili karşılaştırma matrisi

Detaylı Kriter-Açıklama		K101		K102		K103		K104		K105		K106		K107		K108		K109		K110		K111		K112		K113		K114		K115
K1101- Değişen İhtiyaçları Dikkate alınması	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
K1102- Üst seviye use case kullanımı	0,50	0,67	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
K1103- Detaylı use case kullanımı	0,50	0,67	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
K1104- İterasyon planlama	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1105- Sürüm planlama	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1106- Test planlama	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1107- Refactoring yaparak sorunlu noktaları açıklamak	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1108- Test güdümlü geliştirme	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1109- Entegrasyon ve Kabul testleri	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1110-Bağımsız-Kesif testleri	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1111-Geliştirici testleri	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1112-Çalışan yazılımın evrimsel paketlenmesi	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1113- Tekrarlamalı geliştirme	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1114- Sürdürülebilir tempoda ilerleme	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
K1115-Yerel fayda yerine genel faydayı düşünme	0,67	1,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00

4.34: Çalışanların motivasyonuna ait bulanık ikili karşılaştırma matrisi

Detayt Kriter-Açıklama	K2101			K2102			K2103			K2104			K2105		
K2101- Küçük bir ekiple çalışmak	1	1	1	1	1	1,5	1	1	1,5	0,5	1	1,5	0,5	1	2
K2102-Kollektif Kod sahipliği	0,67	1	2	1	1	1	1	1	1,5	1	2	2	1	2	2
K2103-Kendi kendine organize olan takımlar	0,67	1	2	1	1	2	1	1	1	1	2	2	1	2	2
K2104-Eşli Programlama	0,67	1	2	1	1	1	1	1	1	1	1	1	0,5	1	2
K2105-Açık çalışma alanı	0,67	1	2	1	1	1	1	1	1	0,67	1	2	1	1	1

4.35: İşbirliği düzeyine ait bulanık ikili karşılaştırma matrisi

Detayt Kriter-Açıklama	K2401		K2402		K2403		K2404					
K2501- Üst seviye ihtiyaç analizinin müşterileri birlikte yapılması	1,00	1,00	1,00	0,5	1,00	1,5	0,5	1,00	1,5	1	1,50	2
K2502- Ekip içi ve müşterileri etkin işbirliği	0,67	1,00	2,00	1,00	1,00	1,00	1	1,50	2	1	1,50	2
K2503-Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak.	0,67	1,00	2,00	0,50	0,67	1,00	1,00	1,00	1,00	1	1,50	2
K2504-Proje Degerlendirmel\Alınan Dersler' Paylaşım toplantısı	0,50	0,67	1,00	0,50	0,67	1,00	0,50	0,67	1,00	1,00	1,00	1,00

4.36: İşletişim düzeyine ait bulanık ikili karşılaştırma matrisi

Detayt Kriter-Açıklama	K2501		K2502		K2503		K2504		K2505		K2506		K2507		K2508		K2509		
K2601-Tüm süreçlerin arasında entegre bilgi akışı sağlamak	1	1	1	0,5	1	1,5	0,5	1	1,5	0,5	1	1,5	1	1,5	1,5	1,5	0,5	1	1,5
K2602-Ekibin işleri bitirme hızı ,hata oranları ve hataların sebeplerini sürekli izlemek	0,67	1	2	1	1	1	0,5	1	1,5	0,5	1	1	1,5	1,5	1,5	1,5	2	2	0,5
K2603- Tüm taraflar arasında yüzyüze iletişimi sağlamak	0,67	1	2	0,67	1	1	1	1	1,5	1,5	2	2	0,5	1	1	1,5	1,5	2	0,5
K2604- Müşteriyle yerinde ve etkin iletişim kurmak	0,67	1	2	0,67	1	2	0,67	1	1	1,5	1,5	1,5	1	1,5	2	2	2	2	0,5
K2605-Günlük toplantılar yapma	0,5	0,67	1	0,67	1	1	0,67	1	1	1,5	1,5	1,5	1	1,5	1,5	1,5	2	2	0,5
K2606-Süreklilik durum raporlarıyla yapılan, yapılacak ve kalan işlerin takibi	0,67	1	2	0,67	1	2	0,67	1	1	1,5	1,5	1,5	1	1,5	1,5	1,5	2	2	0,5
K2607-Hata Raporları	0,5	0,67	1	0,67	1	1	0,67	1	1	1,5	1,5	1,5	1	1,5	1,5	1,5	2	2	0,5
K2608-Hata Eğilim Metrikleri	0,5	0,67	1	0,67	1	1	0,67	1	1	1,5	1,5	1,5	1	1,5	1,5	1,5	2	2	0,5
K2609-Tüm Tarafların Aktif Katılımı	0,67	1	2	0,67	1	2	0,67	1	1	1,5	1,5	1,5	1	1,5	1,5	1,5	2	2	0,5

4.37: Kullanılan Yazılım Çatısına ait bulanık ikili karşılaştırma matrisi

Detayt Kriter-Açıklama	K3201		K3202			
K3201-Zengin, hızlı geliştirme altyapıları kullanmak	1,00	1,00	1,00	0,5	1,00	1,5
K3201-Surekli veri tabani entegrasyonu	0,67	1,00	2,00	1,00	1,00	1,00

4.38: Kullanılan araçlara ait bulanık ikili karşılaştırma matrisi

Detay Kriter-Açıklama	K3501	K3502	K3503	K3504	K3505	K3506	K3507	K3508	K3509	K3510	K3511	K3512	K3513
K3501- Xml ile esnek rapor üreticiler kullanmak	1,00	1,00	1,00	0,5	1,5	1,5	1,5	1,00	1,00	1,00	1,00	1,00	1,5
K3502-Görev yönetimi, İhtiyaç analizi ve görevin gelişim durumunu takip için entegre proje yönetim araçları kullanmak	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3503-Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3504-Manuel görev tahtası	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3505-Dijital görev tahtası	0,50	0,67	1,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3506-Otomatize kod geliştirme araçları kullanımı	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3507-Otomatize arayüz geliştirme araçları	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3508-Unit test araçları kullanımı	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3509-Otomatize test araçlarıyla test	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3510-Özet gant chartlar	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3511-Detaylı gant chartlar	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3512-Kanban	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5
K3513-Otomatize versiyon paketleme ve kurulumu	0,67	1,00	2,00	0,67	1,00	1,00	1,00	1,00	1,00	1,50	1,50	1,50	1,5

4.39: Kullanılan algoritmaya ait bulanık ikili karşılaştırma matrisi

Detay Kriter-Açıklama	K3601	K3602	K3603	K3604
K3601- Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak	1,00	1,00	1,00	1,50
K3602-Tekrarlı işleri kaldırılması	0,67	1,00	2,00	1,50
K3603-Bileşenlerin yeniden kullanımı	0,67	1,00	2,00	1,50
K3604-Davranış Güdümlü Programlama	0,50	0,67	1,00	1,00

4.40: Varolan teknik özelliklere ait bulanık ikili karşılaştırma matrisi

Detay Kriter-Açıklama	K4401	K4402	K4403	K4404	K4405	K4406
K4401-Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği	1	1	1	1	1	1
K4402-Web servis uyumlu adapter alt yapısı kullanmak	1	1	1	1	1	1
K4403-Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme	1	1	1	1	1	1
K4404-İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi	1	1	1	1	1	1
K4405-Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak	1	1	1	1	1	1
K4406-Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi	1	1	1	1	1	1

4.41: Varolan fonksiyonel özelliklere ait bulanık ikili karşılaştırma matrisi

Detay Kriter-Açıklama		K4601			K4602			K4603	
K4501- Ajax desteğine sahip olmak	1	1	1	1	1	1	1	1	1
K4502- Esnek yetkilendirme altyapısına sahip olmak	1	1	1	1	1	1	1	1	1
K4503-Otomatik hata ayıklama mekanizmaları kullanımı	1	1	1	1	1	1	1	1	1

4.2.3 Bulanık AHP İçin Sentetik Mertebe Değerleri'nin Hesaplanması

Sentetik mertebe değerleri hesaplanırken,

a)İlk olarak, yukarıda bulunan bulanık ikili karşılaştırmalar matrisinin her bir satırdaki aynı bulanık sayıya(l,m,u) karşılık gelen ilk değerler toplanır.

b)Sonra kriterler için her bir sütununun toplamları alınır.

c)Daha sonra bunların matematiksel olarak tersi alınır.

d)Son olarak her bir kriterin her bir l,m,u değerini aynı sütundaki ters değerle çarparsak sentetik mertebe değerlerine ulaşmış oluruz.Örnek olarak ana kriterler için hesaplanmasını görmek istersek;

a)Her bir kriter için l,m,u değerleri şöyle hesaplanır;

$$\begin{aligned}l_{(\text{Süreç})} &= 1+1+0,5+0,5=3 \quad ; m_{(\text{Süreç})} = 1+1,5+1+1=4,5 \quad ; u_{(\text{Süreç})} = 1+2+1,5+1,5=6 \\l_{(\text{Org.})} &= 0,5+1+0,5+0,5=2,5; m_{(\text{Org.})} = 0,7+1+1+1=3,7 \quad ; u_{(\text{Org.})} = 1+1+1,5+1,5=5 \\l_{(\text{Gel.Ort.})} &= 0,7+0,7+1+1 =3,4; m_{(\text{Gel.Ort.})} = 1+1+1+1,5=4,5 \quad ; u_{(\text{Gel.Ort.})} = 2+2+1+2 =7 \\l_{(\text{Altyapı})} &= 0,7+0,7+0,5+1=2,9; m_{(\text{Altyapı})} = 1+1+0,7+1=3,7 ; u_{(\text{Altyapı})} = 1+2+1,5+1,5=6\end{aligned}$$

b)Daha sonra ana kriterler için her bir sütundaki tüm l,m,u değerlerinin toplamları şöyle bulunur;

$$\sum l = 3+2,5+3,34+2,9 = 11,7$$

$$\sum m = 4,5+3,7+4,5+3,7 = 16,3$$

$$\sum u = 6+5+7+6 = 24 \quad \text{olarak bulunur.}$$

Yukarıda belirtilen hesaplama yöntemi tüm kriterlere uygulanmıştır ve ana kriterlerden detay kriterlere doğru bulunan bulanık puanlar(l,m,u), bunların toplamları ve terslerinin değerleri aşağıda Tablo 4.42 ile 4.60 arasında gösterilmiştir.

Tablo 4.42: Ana Kriterlere Ait Bulanık puan toplamları ve tersleri

Ana Kriterler	l	m	u
K1-Süreç	3	4,5	6
K2-Organizasyon	2,5	3,67	5
K3-Yazılım geliştirme ortamı	3,34	4,5	7
K4-Altyapı	2,84	3,67	6
Toplam	11,7	16,3	24
Ters	0,09	0,06	0,04

Tablo 4.43: Süreç Alt Kriterlerine Ait Bulanık puan toplamları ve tersleri

Alt Kriter-Açıklama	l	m	u
K11-Takip edilen süreç	3	5	7
K12-İşin kompleksliği	3,67	5,5	8
K13-Yapılan tasarım	4,34	6	9
K14-Desteklenen iş süreçleri	3,34	4,67	7,5
K15- Varolan standartlar	3,34	4,34	7
Toplam	17,7	25,5	38,5
Ters	0,06	0,04	0,03

Tablo 4.44: Organizasyonel Alt Kriterlere Ait Bulanık puan toplamları ve tersleri

Alt Kriter-Açıklama	l	m	u
K21-Çalışanların karakter ve motivasyonu	3,5	6	8,5
K22-Ekip yetenek, eğitim ve tecrübesi	4,67	7	10
K23-Hedef ve beklentilerin netliği	4,84	7	10,5
K24-Yönetim desteği	3,67	5,34	8
K25-İsbirliği düzeyi	3,84	5,34	8,5
K26-İletişim düzeyi	4,35	6	11
Toplam	24,9	36,7	56,5
Ters	0,04	0,03	0,02

Tablo 4.45: Geliştirme Ortamı Alt Kriterlere Ait Bulanık puan toplamları ve tersleri

Alt Kriter-Açıklama	l	m	u
K31- Kullanılan programlama dili	3,5	6	8,5
K32- Kullanılan Yazılım Çatısı	4,67	7	10
K33- Kullanılan İde	4,84	7	10,5
K34- Kullanılan platform	3,67	5,34	8
K35- Kullanılan araçlar	3,84	5,34	8,5
K36- Kullanılan algoritma	4,35	6	11
Toplam	24,9	36,7	56,5
Ters	0,04	0,03	0,02

Tablo 4.46: Alt yapı Alt Kriterleri Ait Bulanık puan toplamları ve tersleri

Alt Kriter-Açıklama	l	m	u
K41-Veritabanı yapısı	4	6	8
K42- Kullanılan mimari	5,17	7	9,5
K43- Kullanılan donanım	2,9	4,17	5,67
K44-Desteklenen teknik özellikler	3,17	4,34	6,5
K45-Desteklenen fonksiyonel özellikler	3,51	4,67	8
Toplam	18,8	26,2	37,67
Ters	0,05	0,04	0,03

Tablo 4.47: Takip Edilen Süreç Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Detay Kriter-Açıklama	l	m	u
K1101- Değişen İhtiyaçları Dikkate alınması	9	16	23
K1102- Üst seviye use case kullanımı	8,5	15,2	22
K1103- Detaylı use case kullanımı	8	14,3	21
K1104- İterasyon planlama	12,5	19	27,5
K1105- Sürüm planlama	12,7	19	28
K1106- Test planlama	10	15,8	24
K1107- Refactoring yaparak sorunlu noktaları ayıklamak	12	18	28
K1108- Test güdümlü geliştirme	8,68	14	22,5
K1109- Entegrasyon ve Kabul testleri	10,9	16,5	27,5
K1110-Bağımsız- Kesif testleri	10,9	15,7	24,5
K1111-Gelistirici testleri	10,5	15,2	24,5
K1112-Çalışan yazılımın evrimsel paketlenmesi	10,5	14,9	24
K1113-Tekrarlamalı geliştirme	9,02	13	22
K1114-Sürdürülebilir tempoda ilerleme	9,52	13,2	22
K1115-Yerel fayda yerine genel faydayı düşünme	8,85	12	20
Toplam	152	232	360,5

Tablo 4.48: İşin Kompleksliği Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Detay Kriter-Açıklama	l	m	u
K1201-Daha basit ürün geliştirme	3,5	5	6,5
K1202-Sadece müşterilerin ihtiyacı olan ve müşteriye değer katan özellikler geliştirmek	2,5	3,67	5
K1203-İhtiyaçları küçük senaryolara ayırarak önceliklendirip sıralamak ve en öncelikliden işe başlamak	3,34	4,5	7
K1204-Daha kısa sürümlerle daha az özellik geliştirerek müşteriye hızlı geri dönüş	2,67	3,34	5
Toplam	12	16,5	23,5
Ters	0,08	0,06	0,043

Tablo 4.49: Yapılan Tasarım Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Yapılan Tasarım Detay Kriterleri	l	m	u
Detay Kriter-Açıklama	8	14	20
K1301- Detaylı ihtiyaç analizleri yapılması	8,67	14,5	21
K1302-Ön Mimari modelleme	8,84	14,5	21,5
K1303-Ön İhtiyac modelleme	8,17	13,3	19,5
K1304-Bilgisayar Destekli Araçlarla modelleme	7	11,7	17
K1305-Paralel modelle	7,17	11,7	17,5
K1306-Model gözden gecirmeleri	8,02	13	22
K1307-Ana hatlarıyla mimari tasarım	8,69	13,5	23
K1308-Basit tasarım	9,86	14,5	24,5
K1309-Bakım Yapılabilir tasarım	10	14,5	25
K1310-Evrimsel tasarım	8,19	12	21
K1311-Tasarım gözden gecirme	8,53	12,3	22,5
K1312-Veri tabanı gözden gecirme	8,7	12,3	23
Toplam	110	172	277,5
Ters	0,01	0,01	0,004

Tablo 4.50: Var olan Standartlar Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

	l	m	u
Detay Kriter-Açıklama			
K1502-Veri isimlendirme standartları	6,5	10,5	14,5
K1503-Kodlama standartları	6	9,17	12,5
K1504-Arayüz standartları	6,84	10,5	15,5
K1505-Test standartları	7,01	10,5	16
K1506-Versiyon Kurulum standartları	6,34	9,34	14
K1507-Uyarılama standartları	6,85	10	16,5
K1508-Problem çözme standartları	7,02	10	17
K1509-Dokümantasyon standartları	5,67	7,52	11
Detay Kriter-Açıklama	5	6,36	9
Toplam	57,2	83,9	126
Ters	0,02	0,01	0,008

Tablo 4.51 Çalışanların Karakter ve Motivasyonu Detay Kriterlerine Ait Bulanık Puan Toplamları ve Tersleri

Çalışanların karakter ve motivasyonu Detay Kriterleri	l	m	u
K2101- Küçük bir ekiple çalışmak	3	5	7
K2102-Kollektif Kod sahipliği	4,17	6	8,5
K2103-Kendi kendine organize olan takımlar	4,34	6	9
K2104-Eşli Programlama	3,17	4,34	6,5
K2105-Açık çalışma alanı	3,34	4,34	7
Toplam	18	25,7	38
Ters	0,06	0,04	0,026

Tablo 4.52: Hedef ve Beklentilerin Netliği Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Hedef ve Beklentilerin Netliği Detay Kriterleri	l	m	u
K2301- Proje hedeflerini(neyi, ne zaman, hangi sırada, hangi maliyetle ve riskle) baştan doğru belirleyip tüm taraflarla paylaşmak.	2,5	3,5	4,5
K2302- Ana iş süreçlerine odaklanmak	2,67	3,5	5
K2303- Liderlik için belli birkaç alan seçmek	2	2,34	3
Toplam	7,17	9,34	12,5
Ters	0,14	0,11	0,08

Tablo 4.53: Ekip içi ve müşteriyle etkin işbirliği Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Ekip içi ve müşteriyle etkin işbirliği Detay Kriterleri	l	m	u
K2501- Üst seviye ihtiyaç analizinin müşteriyle birlikte yapılması	3	5	6
K2502- Ekip içi ve müşteriyle etkin işbirliği	3,67	5	7
K2503- Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak.	3,17	4,17	6
K2504-Proje Degerlendirme\Alinan Dersler' Paylasim toplantisi	2,5	3,01	4
Toplam	12,3	17,2	23
Ters	0,08	0,06	0,043

Tablo 4.54: İletişim Düzeyi Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

İletişim Düzeyi Detay Kriterleri	l	m	u
K2601-Tüm süreçlerin arasında entegre bilgi akışı sağlamak	6,5	10,5	14,5
K2602-Ekibin işleri bitirme hızı ,hata oranları ve hataların sebeplerini sürekli izlemek	6,5	4	14,5
K2603- Tüm taraflar arasında yüzyüze iletişimi sağlamak.	6,67	10,5	15
K2604- Müşteriyle yerinde ve etkin iletişim kurmak	6,84	10,5	15,5
K2605-Günlük toplantılar yapma	7,01	10,5	16
K2606-Sürekli durum raporlarıyla yapılan, yapılacak ve kalan işlerin takibi	5	5	11
K2607-Hata Raporları	7,35	10,5	17
K2608-Hata Egilim Metrikleri	5,67	7,35	10,5
K2609-Tum Taraflarin Aktif Katilimi	5,34	7,35	11,5
Toplam	56,9	76,2	125,5
Ters	0,02	0,01	0,008

Tablo 4.55: Kullanılan Yazılım Çatısı Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Kullanılan Yazılım Çatısı Detay Kriterleri	l	m	u
K3201-Zengin, hızlı geliştirme altyapıları kullanmak	1,5	2	2,5
K3202-Surekli veri tabani entegrasyonu	1,67	2	3
Toplam	3,17	4	5,5
Ters	0,32	0,25	0,182

Tablo 4.56: Kullanılan Araçlar Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Kullanılan Araçlar Detay Kriterleri	l	m	u
K3501- Xml ile esnek rapor üreticiler kullanmak	7,5	4	19,5
K3502-Görev yönetimi, İhtiyaç analizi ve görevin gelişim durumunu takip için entegre proje yönetim araçları kullanmak	9,17	15	21,5
K3503-Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek	9,34	15	22
K3504-Manuel görev tahtası	9,51	15	22,5
K3505-Dijital görev tahtası	7	11,7	17
K3506-Otomatize kod geliştirme araçları kullanımı	11,4	16,5	25
K3507-Otomatize arayüz geliştirme araçları	10,4	15,2	23,5
K3508-Unit test araçları kullanımı	9,35	13,8	22
K3509-Otomatize test araçlarıyla test	9,52	13,8	22,5
K3510-Özet gant chartlar	7,51	5	17,5
K3511-Detaylı gant chartlar	7,34	10,4	16
K3512-Kanban	7,51	10,4	16,5
K3513-Otomatize versiyon paketleme ve kurulumu	8,7	12,3	23
Toplam	114	158	268,5
Ters	0,01	0,01	0,004

Tablo 4.57: Kullanılan Algoritma Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Kullanılan algoritma Detay Kriterleri	l	m	u
K3601- Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak	3	5	6
K3602-Tekrarlı işleri kaldırılması	3,67	5	7
K3603-Bileşenlerin yeniden kullanımı	3,17	4,17	6
K3604-Davranış Güdümlü Programlama	2,5	3,01	4
Toplam	12,3	17,2	23
Ters	0,08	0,06	0,043

Tablo 4.58: Kullanılan Mimari Algoritma Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Kullanılan Mimari Detay Kriterleri	l	m	u
K4201- Servis tabanlı mimari kullanmak	1,5	2	2,5
K4202-Model güdümlü mimari yapı kullanmak	1,67	2	3
Toplam	3,17	4	5,5
Ters	0,32	0,25	0,182

Tablo 4.59: Desteklenen Teknik Özellikler Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Desteklenen teknik özellikler Detay Kriterleri	l	m	u
K4401-Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği	6	6	6
K4402-Web servis uyumlu adapter alt yapısı kullanmak	6	6	6
K4403-Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme	6	6	6
K4404-İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi	6	6	6
K4405-Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak	6	6	6
K4406-Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi	6	6	6
Toplam	36	36	36
Ters	0,028	0,028	0,028

Tablo 4.60: Desteklenen Fonksiyonel Özellikler Detay Kriterlerine Ait Bulanık puan toplamları ve tersleri

Desteklenen fonksiyonel özellikler	l	m	u
K4501- Ajax desteğine sahip olmak	3	3	3
K4502- Esnek yetkilendirme altyapısına sahip olmak	3	3	3
K4503-Otomatik hata ayıklama mekanizmaları kullanımı	3	3	3
Toplam	9	9	9
Ters	0,11	0,11	0,111

c) Daha sonra ana kriterler için her bir sütundaki tüm l,m,u değerlerinin toplamlarının, matematiksel olarak tersini alınmış ve

$$1/\sum l = 1/11,8 = 0,09$$

$$1/\sum m = 1/16,4 = 0,06$$

$$1/\sum u = 1/24 = 0,04 \text{ olarak bulunur.}$$

d) Son olarak ana kriterin l,m,u değerini aynı sütundaki ters değerle çarpımı sonucu sentetik mertebe değerlerine aşağıdaki gibi ulaşılmış oluruz.

$$l(\text{süreç}) : 3 \times 0,09 = 0,27 ; m(\text{süreç}) : 4,5 \times 0,06 = 0,27 ; u(\text{süreç}) : 6 \times 0,04 = 0,24$$

$$l(\text{Org.}) : 2,5 \times 0,09 = 0,23 ; m(\text{Org.}) : 3,7 \times 0,06 = 0,22 ; u(\text{Org.}) : 5 \times 0,04 = 0,20$$

$$l(\text{Gel.Ort.}) : 3,4 \times 0,09 = 0,36 ; m(\text{Gel.ort.}) : 4,5 \times 0,06 = 0,27 ; u(\text{Gel.Ort.}) : 7 \times 0,04 = 0,28$$

$$l(\text{Altyapı}) : 2,9 \times 0,09 = 0,26 ; m(\text{Altyapı}) : 3,7 \times 0,06 = 0,22 ; u(\text{süreç}) : 6 \times 0,04 = 0,24$$

Aşağıdaki tablolarda tüm kriterlere ait Sentetik Mertebe Değerleri mevcuttur.

Yukarıda belirtilen hesaplama yöntemi tüm kriterlere uygulanmıştır ve ana kriterlerden detay kriterlere doğru bulunan sentetik mertebe değerleri aşağıda Tablo 4.61 ile 4.79 arasında gösterilmiştir.

Tablo 4.61: Ana Kriterlere ait Sentetik Mertebe Değerleri

Ana Kriter-Açıklama	l	m	u
K1-Süreç iyileştirmeleri	0,27	0,27	0,24
K2-Organizasyonel iyileştirmeler	0,23	0,22	0,20
K3-Yazılım geliştirme ortamıyla ilgili iyileştirmeler	0,36	0,27	0,28
K4-Altyapısal iyileştirmeler	0,26	0,22	0,24

Tablo 4.62:Süreç Alt Kriterine ait Sentetik Mertebe Değerleri

Alt Kriter-Açıklama	l	m	u
K11-Takip edilen süreç	0,17	0,2	0,18
K12-İşin kompleksliği	0,21	0,2	0,21
K13-Yapılan tasarım	0,25	0,2	0,23
K14-Desteklenen iş süreçleri	0,19	0,2	0,2
K15- Varolan standartlar	0,19	0,2	0,18

Tablo 4.63:Organizasyonel Alt Kriterine ait Sentetik Mertebe Değerleri

Alt Kriter-Açıklama	l	m	u
K21-Çalışanların karakter(dürüst,sorumluluk sahibi, güvenilir ve müzakereci) ve motivasyonu	0,14	0,2	0,15
K22-Ekip yetenek, eğitim ve tecrübesi	0,19	0,2	0,18
K23-Hedef ve beklentilerin netliği	0,2	0,2	0,19
K24-Yönetim desteği	0,15	0,1	0,14
K25-İsbirliği düzeyi	0,15	0,1	0,15
K26-İletişim düzeyi(Ekip içi ve müşteriyle yakın,hızlı ve etkin iletişim mümkünse yüz yüze)	0,18	0,2	0,2

Tablo 4.64:Geliştirme Ortamı Alt Kriterine ait Sentetik Mertebe Değerleri

Alt Kriter-Açıklama	l	m	u
K31- Kullanılan programlama dili	0,14	0,2	0,15
K32- Kullanılan Yazılım Çatısı	0,19	0,2	0,18
K33- Kullanılan İde	0,2	0,2	0,19
K34- Kullanılan platform	0,15	0,1	0,14
K35- Kullanılan araçlar	0,15	0,1	0,15
K36- Kullanılan algoritma	0,18	0,2	0,2

Tablo 4.65:Altyapı Alt Kriterine ait Sentetik Mertebe Değerleri

Alt Kriter-Açıklama	l	m	u
K41-Veritabanı yapısı	0,21	0,2	0,21
K42- Kullanılan mimari	0,28	0,3	0,25
K43- Kullanılan donanım	0,16	0,2	0,15
K44-Desteklenen teknik özellikler	0,17	0,2	0,17
K45-Desteklenen fonksiyonel özellikler	0,19	0,2	0,21

Tablo 4.66: Takip Edilen Süreç Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K1101- Değişen İhtiyaçları Dikkate alınması	0,06	0,07	0,06
K1102- Üst seviye use case kullanımı	0,06	0,07	0,06
K1103- Detaylı use case kullanımı	0,05	0,06	0,06
K1104- İterasyon planlama	0,08	0,08	0,08
K1105- Sürüm planlama	0,08	0,08	0,08
K1106- Test planlama	0,07	0,07	0,07
K1107- Sürekli tüm süreçler için refactoring yaparak sorunlu noktaları ayıklamak	0,08	0,08	0,08
K1108- Test güdümlü geliştirme	0,06	0,06	0,06
K1109- Entegrasyon ve Kabul testleri	0,07	0,07	0,08
K1110-Bağımsız- Kesif testleri	0,07	0,07	0,07
K1111-Gelistirici testleri	0,07	0,07	0,07
K1112-Çalışan yazılımın evrimsel paketlenmesi	0,07	0,06	0,07
K1113-Tekrarlamalı geliştirme	0,06	0,06	0,06
K1114-Sürdürülebilir tempoda ilerleme	0,06	0,06	0,06
K1115-Yerel fayda yerine genel faydayı düşünme	0,06	0,05	0,06

Tablo 4.67: İşin Kompleksliği Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K1201-Daha basit ürün geliştirme	0,29	0,3	0,28
K1202-Sadece müşterilerin ihtiyacı olan ve müşteriye değer katan özellikler geliştirmek	0,21	0,22	0,21
K1203-İhtiyaçları küçük senaryolara ayırarak önceliklendirip sıralamak ve en öncelikliden işe başlamak	0,28	0,27	0,3
K1204-Daha kısa sürümlerle daha az özellik geliştirerek müşteriye hızlı geri dönüş	0,22	0,2	0,21

Tablo 4.68: Yapılan Tasarım Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K1301- Detaylı ihtiyaç analizleri yapılması	0,07	0,08	0,07
K1302-Ön Mimari modelleme	0,08	0,08	0,08
K1303-Ön İhtiyac modelleme	0,08	0,08	0,08
K1304-Bilgisayar Destekli Araçlarla modelleme	0,07	0,08	0,07
K1305-Paralel modelle	0,06	0,07	0,06
K1306-Model gözden gecirmeleri	0,07	0,07	0,06
K1307-Ana hatlarıyla mimari tasarım	0,07	0,08	0,08
K1308-Basit tasarım	0,08	0,08	0,08
K1309-Bakım Yapılabilir tasarım	0,09	0,08	0,09
K1310-Evrimsel tasarım	0,09	0,08	0,09
K1311-Tasarım gözden gecirme	0,07	0,07	0,08
K1312-Veri tabanı gözden gecirme	0,08	0,07	0,08
K1313-Arayuz gözden gecirme	0,08	0,07	0,08

Tablo 4.69: Var olan Standartlar Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K1501-Modelleme standartları	0,11	0,13	0,12
K1502-Veri isimlendirme standartları	0,1	0,11	0,1
K1503-Kodlama standartları	0,12	0,13	0,12
K1504-Arayüz standartları	0,12	0,13	0,13
K1505-Test standartları	0,11	0,11	0,11
K1506-Versiyon Kurulum standartları	0,12	0,12	0,13
K1507-Uyarılma standartları	0,12	0,12	0,13
K1508-Problem çözme standartları	0,1	0,09	0,09
K1509-Dokümantasyon standartları	0,09	0,08	0,07

Tablo 4.70: Çalışanların karakter ve motivasyonu Detay Kriterlerine Ait Sentetik Merteye Değerleri

Detay Kriter-Açıklama	l	m	u
K2101- Küçük bir ekiple çalışmak	0,17	0,19	0,18
K2102-Kollektif Kod sahipliği	0,23	0,23	0,22
K2103-Kendi kendine organize olan takımlar	0,24	0,23	0,24
K2104-Eşli Programlama	0,18	0,17	0,17
K2105-Açık çalışma alanı	0,19	0,17	0,18

Tablo 4.71: Hedef ve Beklentilerin Netliği Detay Kriterlerine Ait Sentetik Merteye Değerleri

Detay Kriter-Açıklama	l	m	u
K2301- Proje hedeflerini(neyi, ne zaman, hangi sırada, hangi maliyetle ve riskle) baştan doğru belirleyip tüm taraflarla paylaşmak.	0,35	0,37	0,36
K2302- Ana iş süreçlerine odaklanmak	0,37	0,37	0,4
K2303- Liderlik için belli birkaç alan seçmek	0,28	0,25	0,24

Tablo 4.72: Ekip içi ve müşteriyle etkin işbirliği Detay Kriterlerine Ait Sentetik Merteye Değerleri

Detay Kriter-Açıklama	l	m	u
K2401-Çevik süreçlerin uygulanması konusunda ekibi teşvik	0,24	0,29	0,26
K2501- Üst seviye ihtiyaç analizinin müşteriyle birlikte yapılması	0,3	0,29	0,3
K2502- Ekip içi ve müşteriyle etkin işbirliği	0,26	0,24	0,26
K2503-Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak.	0,2	0,18	0,17

Tablo 4.73: İletişim Düzeyi Detay Kriterlerine Ait Sentetik Merteye Değerleri

Detay Kriter-Açıklama	l	m	u
K2601-Tüm süreçlerin arasında entegre bilgi akışı sağlamak	0,11	0,14	0,12
K2602-Ekibin işleri bitirme hızı ,hata oranları ve hataların sebeplerini sürekli izlemek	0,11	0,05	0,12
K2603-Tüm taraflar arasında yüzyüze iletişimi sağlamak	0,12	0,14	0,12
K2604- Müşteriyle yerinde ve etkin iletişim kurmak	0,12	0,14	0,12
K2605-Günlük toplantılar yapma	0,12	0,14	0,13
K2606-Sürekli durum raporlarıyla yapılan, yapılacak ve kalan işlerin takibi	0,09	0,07	0,09
K2607-Hata Raporları	0,13	0,14	0,14
K2608-Hata Eğilim Metrikleri	0,10	0,10	0,08
K2609-Tüm Tarafların Aktif Katılımı	0,09	0,10	0,09

Tablo 4.74: Kullanılan Yazılım Çatısı Detay Kriterlerine Ait Sentetik Merteye Değerleri

Detay Kriter-Açıklama	l	m	u
K3201-Zengin, hızlı geliştirme altyapıları kullanmak	0,47	0,5	0,45
K3202-Sürekli veri tabanı entegrasyonu	0,53	0,5	0,55

Tablo 4.75: Kullanılan Araçlar Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K3501- Xml ile esnek rapor üreticileri kullanmak	0,07	0,03	0,07
K3502-Görev yönetimi, İhtiyaç analizi ve görevin gelişim durumunu takip için entegre proje yönetim araçları kullanmak	0,08	0,09	0,08
K3503-Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek	0,08	0,09	0,08
K3504-Manuel görev tahtası	0,08	0,09	0,08
K3505-Dijital görev tahtası	0,06	0,07	0,06
K3506-Otomatize kod geliştirme araçları kullanımı	0,1	0,1	0,09
K3507-Otomatize arayüz geliştirme araçları	0,09	0,1	0,09
K3508-Unit test araçları kullanımı	0,08	0,09	0,08
K3509-Otomatize test araçlarıyla test	0,08	0,09	0,08
K3510-Özet gant chartlar	0,07	0,03	0,07
K3511-Detaylı gant chartlar	0,06	0,07	0,06
K3512-Kanban	0,07	0,07	0,06
K3513-Otomatize versiyon paketleme ve kurulumu	0,08	0,08	0,09

Tablo 4.76: Kullanılan Algoritma Detay Kriterlerine Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K3601- Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak	0,24	0,29	0,26
K3602-Tekrarlı işleri kaldırılması	0,3	0,29	0,3
K3603-Bileşenlerin yeniden kullanımı	0,26	0,24	0,26
K3604-Davranış Güdümlü Programlama	0,2	0,18	0,17

Tablo 4.77: Kullanılan Mimari Algoritma Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K4201- Servis tabanlı mimari kullanmak	0,47	0,5	0,45
K4202-Model güdümlü mimari yapı kullanmak	0,53	0,5	0,55

Tablo 4.78: Desteklenen Teknik Özellikler Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K4401-Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği	0,17	0,17	0,17
K4402-Web servis uyumlu adapter alt yapısı kullanmak	0,17	0,17	0,17
K4403-Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme	0,17	0,17	0,17
K4404-İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi	0,17	0,17	0,17
K4405-Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak	0,17	0,17	0,17
K4406-Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi	0,17	0,17	0,17

Tablo 4.79: Desteklenen Fonksiyonel Özellikler Detay Kriterlerine Ait Sentetik Mertebe Değerleri

Detay Kriter-Açıklama	l	m	u
K4501- Ajax desteğine sahip olmak	0,33	0,33	0,33
K4502- Esnek yetkilendirme altyapısına sahip olmak	0,33	0,33	0,33
K4503-Otomatik hata ayıklama mekanizmaları kullanımı	0,33	0,33	0,33

4.2.4. Kriterler için ağırlık vektörünün hesaplanması

Bu aşamada her ikili sentetik merteye değeri için;

$\mu_2 = (l_2, m_2, u_2) \geq \mu_1 = (l_1, m_1, u_1)$ formülünün olabirlik dereceleri $V(\mu_2 \geq \mu_1)$ değerleri hesaplanmıştır.

$$V(\mu_2 \geq \mu_1) = \begin{cases} 1, & \text{eger} \dots m_2 \geq m_1 \\ 0, & \text{eger} \dots l_1 \geq u_2 \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & \text{diger .durumlarda} \end{cases}$$

Örnek olarak $V(S_{\text{Süreç}} \geq S_{\text{Organizasyon}})$ için; $m_2 \geq m_1$ ($0,27 \geq 0,23$) olduğu için $V=1$ olarak bulunmuştur. Tüm kriter karşılaştırmaları için bu işlem uygulanarak, ayrı ayrı ağırlık vektörü hesaplanmış ve aşağıdaki tablolarda ayrı ayrı gösterilmiştir.

$$V(S_{\text{Süreç}} \geq S_{\text{Organizasyon}}) = 1,$$

$$V(S_{\text{Süreç}} \geq S_{\text{Geliştirme Ortamı}}) = 1,$$

$$V(S_{\text{Süreç}} \geq S_{\text{Altyapı}}) = 1,$$

$$V(S_{\text{Organizasyon}} \geq S_{\text{Gelistirme Ortamı}}) = 0,$$

$$V(S_{\text{Organizasyon}} \geq S_{\text{Altyapı}}) = 1,$$

$$V(S_{\text{Organizasyon}} \geq S_{\text{Süreç}}) = 0,$$

$$V(S_{\text{Geliştirme Ortamı}} \geq S_{\text{Süreç}}) = 1$$

$$V(S_{\text{Gelistirme Ortamı}} \geq S_{\text{Organizasyon}}) = 1$$

$$V(S_{\text{Gelistirme Ortamı}} \geq S_{\text{Altyapı}}) = 1$$

$$V(S_{\text{Altyapı}} \geq S_{\text{Süreç}}) = 0$$

$$V(S_{\text{Altyapı}} \geq S_{\text{Organizasyon}}) = 1$$

$$V(S_{\text{Altyapı}} \geq S_{\text{Gelistirme Ortamı}}) = 0$$

$$d(\text{Süreç}) = \min(1, 1, 1) = 1$$

$$d(\text{Organizasyon}) = \min(0, 1, 0) = 0$$

$$d(\text{Geliştirme Ortamı}) = \min(1, 1, 1) = 1$$

$$d(\text{Altyapı}) = \min(0, 1, 0) = 0$$

Ana Kriterler için normalize edilmiş ağırlıklar vektörü

$W'=(1/(1+1),0/(1+1), 1/(1+1),0/(1+1))$ e göre $W'(1/2,0/2,1/2,0/2) =(0.5, 0. 0.5, 0)$ şeklinde elde edilmiştir. Buna Göre Süreç ve Geliştirme Ortamıyla ilgili yapılacak iyileştirmeler eşit derecede %50, %50 gibi yüksek önem seviyesindedir.

Yukarıdaki yöntem diğer tüm kriterlere uygulandığında aşağıdaki Tablo 4.68 ile gösterilen normalize edilmiş ağırlıklar tablosu ortaya çıkmakta ve ağırlık değerleri kırmızı renkle gösterilmektedir.

Tablo 4.80: Normalize edilmiş ağırlıklar tablosu

ind	K1					K2						K3						K4				
W'	0,5					0						0,5						0				
	K11	K12	K13	K14	K15	K21	K22	K23	K24	K25	K26	K31	K32	K33	K34	K35	K36	K41	K42	K43	K44	K45
	0	0	1	0	0	0	0,5	0,5	0	0	0	0	0,5	0,5	0	0	0	0	1	0	0	0
01	0	0,84	0	1	0,23	0	1	0,5	0,44	0,2	1	1	0,5	1	1	0	0,42	1	0,5	1	0,167	0,33
	0	0	0	0	0	0	0	0	0	0	0	0	0,125	0,25	0	0	0	0	0	0	0	0
02	0	0	0,25		0	0,5		0,5	0,44	0			0,5			0	0,42		0,5		0,167	0,33
	0	0	0,125		0	0		0	0	0			0,125			0	0		0		0	0
03	0	0,16	0,25		0,23	0,5		0	0,22	0,2						0,19	0,16				0,167	0,33
	0	0	0,125		0	0		0	0	0						0	0				0	0
04	0,5	0	0		0,23	0			0	0,2						0,27	0				0,167	
	0	0	0		0	0			0	0						0	0				0	0
05	0,5	0	0		0	0				0,2						0					0,167	
	0	0	0		0	0			0	0						0					0	0
06	0	0	0		0,16					0						0,27					0,167	
	0	0	0		0					0						0					0	0
07	0	0	0		0,16					0,2						0,27						
	0	0	0		0					0						0						
08	0	0	0		0					0						0						
	0	0	0		0					0						0						
09	0	0	0,25		0					0						0						
	0	0	0,125		0					0						0						
10	0	0	0,25													0						
	0	0	0,125													0						
11	0	0	0													0						
	0	0	0													0						
12	0	0	0													0						
	0	0	0													0						
13	0	0	0													0						
	0	0	0													0						
14	0	0	0													0						
	0	0	0													0						
15	0	0	0													0						
	0	0	0													0						

Yukarıdaki tablo 4.68' e bakıldığında, Organizasyonel ve alt yapıyla ilgili yapılacak iyileştirmeler ağırlık vektöründe diğer kriterlere göre 0 değerini almış olduğu görülmektedir, bu kriterlerin önemsiz olduğunu değil diğer kriterlere göre daha az önem taşıdığını göstermektedir.

5. TARTIŞMA

Bu bölümde, önceki bölümde anket verilerinin BAHP yöntemiyle analizi sonucunda elde edilen bulgular tartışılacak ve araştırma kısıtları ortaya konulacaktır.

5.1 Veri Analizi ve Sonuçları

Günümüzde karar vericiler karar verme sürecinde, deneyim ve öznel algılardan kaynaklanan belirsizlikler ile karşı karşıya kalırlar. Çok kriterli karar sistemleri, farklı alanlarda uzmanların kullanımına daha çok ihtiyaç duymaktadır. Geleneksel AHP yöntemi, karar vericilerin gereksinimlerini tam olarak karşılayamadığından ve karar vericilerin alternatifleri değerlendirirken subjektif olup, kesin yargılar verememelerinden dolayı, bu çalışmada yazılım tercihleri alternatiflerini değerlendirmede BAHP yaklaşımı ele alınmıştır. BAHP yaklaşımı ile verilerin değerlendirilmesinde yer alan belirsizlik etkili bir şekilde temsil edilebilir ve daha etkin bir karar verilmesine yardımcı olur.

ÇYGP başarısına olumlu etki eden faktörlerin hangileri olduğunu karar vermeyi amaçlayan bu tezde de, başarı faktörlerinin başarıya çoklu etkileri ve başarıya katkı ağırlıkları BAHP yöntemiyle analizi, çeşitli sektörlerdeki Yazılım Geliştirme Projelerinden katılımcıların cevapladığı anket soruları üzerinden gerçekleştirilmiştir.

5.2. Araştırma Sorularına Cevaplar

Yukarıdaki analizler sonucunda tezin başında hedeflenen araştırma sorularına cevaplar verebilir. Ana kriter ve alt kriterlerin ve detay kriterlerin kendi içinde gruplu olarak ağırlıklarının belirlenmesine rağmen sonuçta esas amacımız olan ÇYGP için KBF'ni bulmak için genel ağırlıklar dikkate alınmıştır.

Aşağıda araştırma soruları ve buna verilen cevaplar görülebilir;

5.2.1. Araştırma Sorusu 1

İlk araştırma sorusu “ÇYGP’de başarı için öncelikle dikkat edilmesi gereken faktörlere ve alt faktörler nelerdir?” idi. Yukarıda elde edilen araştırma sonuçları bize olası faktörlerin hepsinin değil sadece bazılarının önemli olduğu ve bunlara öncelikle dikkat etmek gerektiğini göstermiştir. Buna göre;

Ana Faktörler Dikkate alındığında ;

Süreç ve Geliştirme ortamı ile ilgili faktörlere öncelikle verilecek önem diğer Organizasyon ve Alt yapı faktörlerine göre verimliliğe daha fazla katkı sağlayacağı ve başarıya ulaşmada esas öneme sahip olduğu görülmüştür.

Alt Faktörler dikkate alındığında;

Süreçle ilgili faktörler dikkate alındığında verimlilik ve başarıya en çok(%100) Yapılan Tasarımın katkı sağladığı ve dolayısıyla tasarıma ilk olarak önem verilmesi gerektiği,

Geliştirme Ortamıyla ilgili faktörler dikkate alındığında en çok Kullanılan Yazılım Çatısı(%50) ve kullanılan Entegre Geliştirme Ortamı’nın (%50)etkilediği ve dolayısıyla Yazılım Çatısı ve Entegre Geliştirme Ortamı’nın doğru seçilmesi gerektiği sonucuna varılabilir,

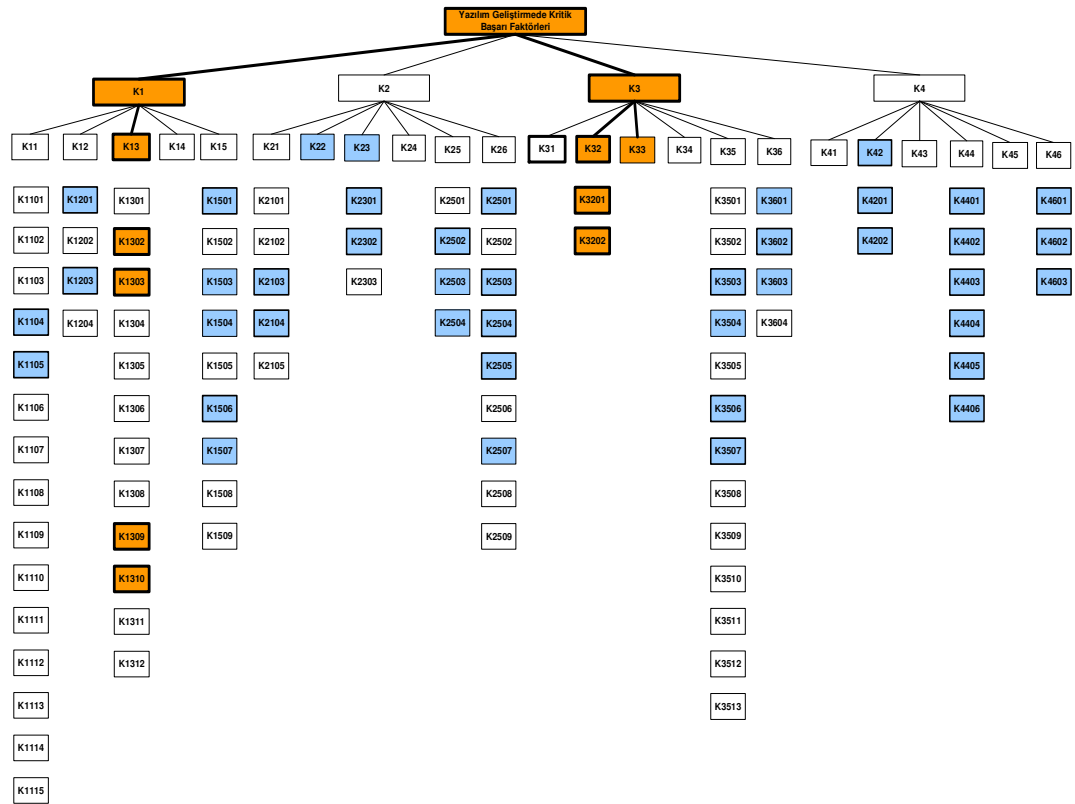
Organizasyonla ilgili faktörler her ne kadar süreç ve geliştirme ortamı kadar etkilemese de kendi içinde bakıldığında en çok Ekip yetenek, eğitim ve tecrübesi(%50) ve Hedef ve Beklentilerin Netliği(%50) etkilediği ve bundan dolayı verimlilik için organizasyonel düzeyde öncelikle ekip üyelerinin yetenekli, eğitilmiş ve tecrübeli olması ve doğru hedeflerle belirleyip beklenenler anlaşarak geliştirme projesine başlamak gerektiği,

Alt yapı ile ilgili faktörler her ne kadar süreç ve geliştirme ortamı kadar etkilemese de kendi içinde bakıldığında en çok Kullanılan Mimari(%100) nin etkilediği ve bundan dolayı verimlilik için doğru mimari kurgu üzerinden geliştirme projesine başlamak gerektiği sonucuna varılabilir.

5.2.2. Araştırma Sorusu 2

İkinci araştırma sorusu b) ”Belirlenen her faktör için ve genel başarı için kritik roldeki en önemli çevik uygulamalar ve bunların öncelik sıraları nelerdir?” İdi.

Önceki bölümde yapılan analiz çalışması sonucunda, ÇYGP’de verimlilik ve başarı için dikkate alınması gereken KBF belirlenerek önem vermesi gerekli, başlıca faktörleri sıralı bir şekilde sunarak yardımcı olmak için kullanılmış ve kriterlerin ağırlıklarına ve tercihlere bağlı sonuçlar aşağıda Şekil 5.1. üzerinde renklendirilerek gösterilmiştir.



Şekil 5.1 Karar Kriterlerinin öncelikleri

Not:Şekil 5.1 de gösterilen Turuncu renkli kriterler birincil öncelikteki kriterlerdir. Mavi Renkli olanlar 2. öncelikli olan kriterlerdir

Normalize edilmiş önem tablosuna göre elde edilen sıralama ilk planda etki eden faktörler aşağıdaki Tablo 5.1’ de ve ikinci planda etki eden faktörlerde ağırlık ve alt kriterler içindeki sırası aşağıdaki tablo 5.2’de gösterilmiştir.gösterilmiştir.;

Tablo 5.1: KBF ağırlık ve genel sıraları

Kriter-Açıklama	Genel Ağırlık	Genel Sıra
K3301-Kullanılan entegre geliştirme ortamı	0,5	1
K1302-Ön Mimari modelleme	0,125	2
K1303-Ön İhtiyac modelleme	0,125	2
K1309-Bakım Yapılabilir tasarım	0,125	2
K1310-Evrimsel tasarım	0,125	2
K3201-Zengin, hızlı geliştirme altyapıları kullanmak	0,125	2
K3202-Surekli veri tabanı entegrasyonu	0,125	2

Tablo 5.2: 2.derecedeki faktörlerin ağırlık ve alt kriterler içi sıraları

Kriter-Açıklama	Ağırlık	Sıra
K1104-İterasyon planlama	0,5	1
K1105- Sürüm planlama	0,5	1
K1201-Daha basit ürün geliştirme	0,84	1
K1203-İhtiyaçları küçük senaryolara ayırarak önceliklendirip sıralamak ve öncelikliden işe başlamak	0,16	2
K1501-Modelleme standartları	0,23	1
K1503-Kodlama standartları	0,23	1
K1504-Arayüz standartları	0,23	1
K1506-Versiyon Kurulum standartları	0,16	2
K1507-Uyarlama standartları	0,16	2
K2103-Kendi kendine organize olan takımlar	0,5	1
K2104-Eşli Programlama	0,5	1
K2301-Proje hedeflerini(neyi, ne zaman, hangi sırada, hangi maliyetle ve riskle)belirleyip herkesle paylaşmak.	0,5	1
K2302-Ana iş süreçlerine odaklanmak	0,5	1
K2501-Üst seviye ihtiyaç analizinin müşteriyle birlikte yapılması	0,44	1
K2502-Ekip içi ve müşteriyle etkin işbirliği	0,44	1
K2503-Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak.	0,22	2
K2601-Tüm süreçlerin arasında entegre bilgi akışı sağlamak	0,2	1
K2603-Tüm taraflar arasında yüzyüze iletişimi sağlamak	0,2	1
K2604-Müşteriyle yerinde ve etkin iletişim kurmak	0,2	1
K2605-Günlük toplantılar yapma	0,2	1
K2607-Hata Raporları	0,2	1
K3503-Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek	0,19	1
K3504-Manuel görev tahtası	0,27	1
K3506-Otomatize kod geliştirme araçları kullanımı	0,27	1
K3507-Otomatize arayüz geliştirme araçları	0,27	2
K3601-Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak	0,42	1
K3602-Tekrarlı işleri kaldırılması	0,42	1
K3603-Bileşenlerin yeniden kullanımı	0,16	2
K4201-Servis tabanlı mimari kullanmak	0,5	1
K4202-Model güdümlü mimari yapı kullanmak	0,5	1
K4401-Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği	0,167	1
K4402-Web servis uyumlu adapter alt yapısı kullanmak	0,167	1
K4403-Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme	0,167	1
K4404-İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi	0,167	1
K4405-Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak	0,167	1
K4406-Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi	0,167	1
K4501-Ajax desteğine sahip olmak	0,33	1
K4502- Esnek yetkilendirme altyapısına sahip olmak	0,33	1
K4503-Otomatik hata ayıklama mekanizmaları kullanımı	0,33	1

5.3. ARAŞTIRMA KISITLARI

Anketten toplanan verilere göre, var olan kısıtlamalar şunlardır:

İlk olarak, Anketi yanıtlayan 6 katılımcının hepsi de, yazılım geliştirme projelerinde 5-10 yıl arasında tecrübeye sahip olmasına rağmen, bunların çoğu, çevik yöntem uygulamalarını ya kısmi olarak kullanmakta ya da yeni kullanmayı düşünmekte olması sebebiyle bu konudaki tecrübelerin ve kullandıkları uygulamaların olgunluğa ulaşması için biraz daha zamana ihtiyacı duyulacağını söyleyebiliriz.

İkinci olarak, anketi yanıtlayanların %66,7'i bizzat geliştirici, % 33,3'ü analist olduğu için ÇYGP'de KBF'ye etki eden faktörleri ile ilgili sorulara daha çok geliştirici bakış açısıyla yorumlamış olma ihtimalidir.

Üçüncü olarak; BAHP yönteminin ağırlık vektöründen normalize edilmiş ağırlıklar vektörünü bulurken ikili sentetik mertebe değerinin olabilirlik derecelerini bulurken

$\mu_2 (l_2, m_2, u_2) \geq \mu_1 = (l_1, m_1, u_1)$ formülünün olabilirlik dereceleri $m_2 \geq m_1$ değil ve $l_1 \geq u_2$ ise olabilirlik değerinin 0 atanmasından dolayı 0 olarak bulunan faktörlerin ağırlık değerinin 0 olarak hesaplanması dolayısıyla bazı faktörlerin genel önem sırasının bulunması olasılığı kısıtlayıcı faktörler olarak ele alınabilir.

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, BHAP yöntemi ile anket verilerinin kantitatif analizi ile ÇYGP'de KBF'ni ortaya koymaya çalışılmıştır. Veriler, çeşitli büyüklük ve tipteki kurumlarda çalışan, çeşitli roldeki katılımcıların Yazılım Geliştirme Projelerindeki geçmiş deneyimlerini göz önüne alarak yaptıkları değerlendirmelere göre oluşturulmuştur.

Tezde ilk olarak Çevik projeleri etki eden faktörlerle ilgili yapılan çalışmalarla elde edilen içerisinden proje başarısına etki edenleri derlenip özetlenmiş ve bu faktörler 4 ana kriter 22 alt kriter altında 93 uygulama pratiği ÇYGP'de KBF olası karar kriterleri olarak belirlenmiş ve BAHP yöntemiyle analiz edilerek bunlar içinden 7si birinci derecede etki eden Kritik başarı faktörleri olarak belirlenmiştir. Bunlardan;

- a) Sürekli entegrasyon sağlayabilen entegre geliştirme ortamlarının kullanımı,
- b) Zengin ve hızlı geliştirme alt yapıları sunan Yazılım Çatısı uygulamaları kullanımı, ,
- c) Sürekli veri tabanı ile entegrasyon sağlayabilen Yazılım Çatısı uygulamaları kullanımı,
- d) Ön mimari modelleme,
- e) Ön ihtiyaç modelleme,
- f) Bakım yapılabilir tasarım,
- g) Evrimsel tasarım

ilk olarak uygulanması gereken çevik uygulamalar olduğu sonucuna varılabilir. Ancak bu diğer uygulamaların önemsiz olduğu anlamına gelmez. Bunun sebebi diğer detay uygulamaların belirli bir ağırlığı olan fakve diğ.t faktör yada ana faktörlerin görece ağırlığı sıfır çıkmasından dolayı genel ağırlığı sıfır çıkmasıdır.

İkincil önemde olan, yani ağırlığı sıfırdan farklı fakat ana yada alt faktörünün ağırlığı sıfır olan çevik uygulamalara bakıldığında ise;

Takip edilen süreçle ilgili olarak ;

- a) İterasyon planlama
- b) Sürüm planlama

İşin kompleksliği ile ilgili olarak;

- a) Daha basit ürün geliştirme
- b) İhtiyaçları küçük senaryolara ayırarak önceliklendirip sıralamak ve en öncelikliden işe başlama

Var olan Standartlar ile ilgili olarak;

- a) Modelleme standartları,
- b) Kodlama Standartları
- c) Arayüz Geliştirme Standartları
- d) Versiyon Kurulum standartları

Ekip üyelerinin karakter ve motivasyonu ile ilgili olarak;

- a) Kollektif Kod sahipliği,
- b) Kendi kendine organize olan takımlar

Hedef ve beklentilerin netliği ile ilgili olarak;

- a) Proje hedeflerini(neyi, ne zaman, hangi sırada, hangi maliyetle ve riskle) baştan doğru belirleyip tüm taraflarla paylaşmak.
- b) Ana iş süreçlerine odaklanmak

İşbirliği düzeyi ile ilgili olarak;

- a) Üst seviye ihtiyaç analizinin müşteriyle birlikte yapılması
- b) Ekip içi etkin bilgi paylaşımı
- c) Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak

İletişim düzeyi ile ilgili olarak;

- a) Tüm süreçlerin arasında entegre bilgi akışı sağlamak
- b) Tüm taraflar arasında yüzyüze iletişimi sağlamak
- c) Müşteriyle yerinde ve etkin iletişim kurmak
- d) Günlük toplantılar yapma
- e) Hata Raporları

Kullanılan araçlar ile ilgili olarak;

- a) Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek
- b) Manuel görev tahtası
- c) Otomatize kod geliştirme araçları kullanımı
- d) Otomatize arayüz geliştirme araçları

Kullanılan algoritma ile ilgili olarak;

- a) Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak
- b) Tekrarlı işleri kaldırılması
- c) Bileşenlerin yeniden kullanımı.

Kullanılan mimari ile ilgili olarak;

- a) Servis tabanlı mimari kullanmak
- b) Model güdümlü mimari yapı

Desteklenen teknik özellikler ile ilgili olarak;

- a) Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği
- b) Web servis uyumlu adapter alt yapısı kullanmak
- c) Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme
- d) İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi
- e) Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak
- f) Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi

Desteklenen fonksiyonel özellikler ile ilgili olarak;

- a) Ajax desteğine sahip olmak
- b) Esnek yetkilendirme altyapısına sahip olmak
- c) Otomatik hata ayıklama mekanizmaları kullanımı

Uygulamaları, ikincil öncelikli olarak kullanmak gerektiği sonucuna varılabilir.

Yapılan araştırma sonucunda yukarıda belirlenen KBF'leri ile uygulayıcılara ÇYGP'de rehberlik edecek birinci ve ikinci derecede önemli faktörler ortaya konulmaya çalışılmıştır. Ancak unutulmaması gereken bu faktörlerin gerçek hayatta kullanılırken, içinde bulunulan projeye göre belli bir adaptasyondan geçmesi gerekebileceğidir. Yani her projenin kendine özgü organizasyon, teknoloji, alt yapı, süreç yapısı mutlaka dikkate alınarak uygulamaya geçilmesi gerekmektedir. Projenin tüm süreçlerinde proje yapısına en uygun çevik uygulamalar seçildikten sonra bunları dikkatli bir şekilde

uygulamış ve ortaya çıkan sorunlar anında çözüldükten sonra ilerlenmişse başarıya ulaşılmaması için bir sebep yoktur. Ancak çevik yöntemler tüm problemleri çözecek ve proje başarısının garantisi olacak bir sihirli değnek değildir. Problemler varolmasına rağmen problemler çözülmüyorsa proje çevik yöntemleri kullansın veya kullanmasın başarısız olacaktır.

6.1. ÇYGP ÖNÜNDEKİ ENGELLER VE YAPILABİLECEKLER

Yönetim İnadı: ÇYG'ye karşı yönelim genellikle geliştiriciler yada geliştirme takımları tarafından olmaktadır. Birçok durumda yönetim belirgin bir direç gösterebilir. Bunun ;

1. ÇYG' Yöntemlerine yabancı olmak,
2. UP ve Çevik Modellemenin uygulamalarının yanlış anlaşılması,
3. Kontrolü kaybetme duygusu,
4. Geliştirme işlemlerine uzaklık,
5. Tüm projeyi planlamış olma hissi ihtiyacı,
6. Geleneksel yöntemin hedeflerini tamamlamada daha yardımcı olacağını düşünme,
7. Araştırmalar aksini gösterse de eşli programlamanın verimliliği düşüreceği görüşü,
8. Yönetimin normalde alışılında farklı bir yaklaşım uygulayıp, başarısız olma riskini alamaması
9. Bilinmeyen korkusu gibi çeşitli sebepleri olabilir. Bu direçleri kırmanın yegane yolu eğitimden ve bu yaklaşımı az önemde bir projeye adapte etmek ve daha sonra faydalarını gördükçe adım adım tüm projelere yaygınlaştırmak önerilebilir.

Başarısız Proje Sendromu: Başarısız örnekler ÇY'lere karşı ters tepki yaratıp bunu uygulamanın vakit kaybı olacağını düşündürebilir. Deneyimlere bakılınca başarılı örnekler başarısız örneklerden daha çoktur ve başarısız örnekler istisna olarak görülebilir. Fakat başarısız projenin sebeplerine bakılıp bunlardan ders almak gerekir. Burada önemli olan doğru projede doğru uygulamaların kullanılıp kullanılmadından emin olmak gerektiğidir..

Geliştirici Direnci: Çoğu geliştirici mevcut geliştirme yaklaşımlarından gayet memnundurlar ve bunu değiştirmek istemezler. Ayrıca birçok geliştirici eşli

programlama yada takım halinde tasarım yada modellemeyi rahatsız edici bulurlar ve işbirliğine karşı dururlar. Bazı durumlarda da yönetim eşli programlamayı yada takım çalışmasını yasaklamakta yada kısıtlı olarak izin vermektedir(Hunt 2006). Fakat unutulmamalıdır ki çoğu durumda bir kişinin göremediği basit bir problemi başka bir kişi görebilir yada kişilerin ayrı ayrı ürettiği çözümler birlikte geliştirilen çözümlerden daha hızlı ve fazla değerde olabilmektedir.

Müşterinin Karşı Çıkması: Müşteriler ÇYG yaklaşımına çeşitli sebeplerle karşı çıkabilirler.Bunlar;

- 1.Yazılımın ne yaptığı ile ilgili bilgilerinin olmaması,
2. Takip edebilecekleri detaylı bir planın olmaması,
3. Katılım ve tahhüt düzeyleri
- 4- Çevik yaklaşımın müşterilere getireleri hakkında bilgilerinin olmaması gibi sebeplerdir(Hunt 2006). Burada önemli olan müşterilere yada onların temsilcilerine elde edecekleri getiriler hakkında bilgi verip ikna etmektir.

Sözleşme Zorlukları: Sözleşmeler genel olarak geleneksel şelale yaklaşımına göre yapılır ve özellikle olması gereken ihtiyaçlar belirtilir fakat tanımlı özelliklerin karşılanmadığı görülünce sözleşme şartlarına uyulmadığı fikri oluşur ve bu sabit fiyatlı projelerde müşterinin sorunu daha da şiddetlendirir. Bu yüzden yazılım tedarikçisinin istenilen fonksiyonalitye sağlamak için isteyeceği kesin tutarı belirtmesi gerekir.Bu işlem belki yazılım geliştiricilerin olmadığı bir ortamda gerçekleşiyor da olabilir. Değişebilecek ihtiyaçlar için önceden kesin zaman ve maliyet çıkarmak ÇY'lerde gerçekten aşılması gereken bir zorluktur (Hunt 2006). Burada yapılabilecek öncelikle, ilk çevrim sonunda müşterinin önemli ihtiyaçları için bir takvim ve bütçede istenen yazılımı teslim ederek, müşterinin güvenini kazanmak, sonrasında ise, müşterinin ihtiyacı olduğu değişken ihtiyaçları belirleyip bunlar için gerekli zamanı ve bütçeyi ayrı yaparak projeyi tamamlamak düşünülebilir. Burada müşteri güveni çok önemlidir.

Çevikliğe Aışkanlık: Diğer bir engel de ÇYGP'nin nasıl başlayacağı ve yürütüleceğiyle ilgili bilgi eksikliğidir. Hatta bir yada daha fazla ÇYGP'de bulunana kadar

- Yazılımın maliyetini projenin başında nasıl tahmin edeceğiz?

- Kaç çevrim yapmanın iyi olduğuna nasıl karar vereceğiz?
- Her çevrimde nelerin bulunması gerektiğine nasıl karar vereceğiz?
- Her çevrimin süresine nasıl karar vereceğiz?

gibi önemli sorular bizi engelleyebilir(Hunt 2006)

Aslında bu soruların, proje ekibi ve müşteriyle birlikte yapılacak toplantılarda istenen özellikler, önceliklerin ve işlerin tahmini sürelerinin belirlendiği planlama aşamasında kolayca cevaplanabileceği görülebilir.

Yetenekli Kişi İhtiyacı: Alistair Cockburn ve Jim Highsmith' e göre proje başarısı için en önemli insan faktörlerin arasında olan yaartıcı ve yetenekli geliştiriciler gerekmektedir. Fakat Boehm'in öne sürdüğü gibi dünyadaki geliştiricilerin %50 si orta düzeyin altında geliştirici olması ÇY'lerin başarısında önemli bir kısıt olarak durmaktadır. Alistair Cockburn de ÇY'lerdeki başarıda insan faktörünün önemli bir değişken olduğunu kabul etmekle birlikte aslında bunun sadece ÇYG için değil diğer yazılım geliştirme yaklaşımlarında da olduğunu ve eğer projelerdeki insanlar yeterince iyi ise verilen görevleri başarılı bir şekilde yerine getirebileceklerini eğer insanlar yeterince iyi değilse hiçbir süreç kişilerin yeteneksizliklerini örtüp başarıyı sağlayamayacağını belirtmektedir(Awad 2005).

Sonuçta bu araştırmada ÇYGP'de verimlilik için belirlenen kritik başarı faktörleri ve bu başarı faktörlerini uygulanmasında dikkat edilecek noktalar hakkındaki öneriler ile, Çevik Yazılım Geliştirme ile ilgilenen araştırmacı ve bu konuda çalışan yada çalışmayı düşünen uygulacılara rehberlik ederek, ülkemizdeki yazılım geliştirme projelerinde çevik yöntemlerin kullanımının tartışmaya açılarak çevik yöntemlerin daha yaygın ve bilinçli bir şekilde kullanımının sağlanması ile yazılım geliştirme projelerinde verimlilik artışı sağlanarak, yazılım sektörünün ülkemizdeki gelişimine dolaylı da olsa katkı sağlayabilmek, en önemli beklentimdir.

KAYNAKLAR

Kitaplar

Ambler, S.W., 2002. *Agile modeling: Effective practices for extreme programming and the unified process*, New York: John Wiley & Sons Publishing.

Ambler, S.W., 2002b. *Agile Modeling*. John Wiley & Sons.

Beck, K. (2000). *Extreme programming explained – Embrace change change..* Reading, MA: Addison-Wesley Longman.

Beck, K., & Fowler, M., 2004, *Planning extreme programming*. Upper Saddle River, NJ: Addison-Wesley

Brooks, F. P., 1995. *The mythical man-month: essays on software engineering*, Addison Wesley Longman, Inc., USA

Cockburn A., 2000, *Agile Software Development Draft Version*, <http://zsiie.icis.pcz.pl/ksiazki/Agile%20Software%20Development.pdf>

Cockburn A., 2006, *Agile Software Development: The Cooperative Game*, Second Edition, Addison Wesley Professional, USA

Cockburn, A., 2002. *Agile Software Development*, Addison-Wesley, Boston, Massachusetts.

Cummins F.A., 2009, *Building the agile enterprise with soa,bpm and mbm*, Elsevier Inc., USA

Lomas, C. D. W., Wilkinson, J., Maropoulos, P. G. & Matthews, P. C., 2007. *Implementing Digital Enterprise Technologies for Agile Design in the virtual enterprise*, Springer, USA, http://ltodi.est.ips.pt/det2006/papers/Distributed/f116_D1.pdf, [01.03.2008]

Highsmith, J., 2002. *Agile Software development ecosystems*. Addison- Wesley, Boston, Massachusetts.

Hunt J., 2006, *Agile Software Construction*, Springer, London, UK, <http://www.springeronline.com>, [20.05.2008]

Jacobson I., Booch, G. & Rumbaugh, J. (1999). *The unified software development process*. Reading, MA: Addison-Wesley.

Stojanovic Z., Dahanayake A. & Sol H., 2004, *Agile development methods and component-orientation: A review and analysis*, Ed. Siau K.,2004. *Advanced topics in database research volume 3*, P1-22, Idea Group Publishing, USA

Tate K., Forward by Highsmith J., 2005, *Sustainable software development: An agile perspective*, Addison-Wesley Professional , USA, TATEch01_p001-012,10,11,2005,

Sürelî Yayınlar

Augustine, S., Payne, B., Sencindiver, F., Woodcock, S., 2005. *Agile project management: steering from the edges*. Communications of the ACM, 48 (12), 85–89.

Aoyama M., 1998., *Web-based agile software development*,IEEE, November/December 98, <http://rockfish.cs.unc.edu/COMP290-agile/Aoyama-98.pdf>,[01.02.2008].

Boehm, B., Turner, R., 2005. *Management challenges to implement agile processes in traditional development organizations*. IEEE Software 22 (5), 30–39.

Beck, K. & Cunningham W,1989. *A Laboratory for Object-Oriented Thinking*, in Conference Proceedings of OOPSLA'89, New Orleans, Louisiana

Boehm B. 2002. Get ready for agile methods, with care.IEEE Computer ,35,1,Pp. 64-69

Bytheway, A.J., 1999. *Successful software projects and how to achieve them*. IEEE Software, 16 (3), 15–17.

Chang D. Y., (1996), *Applications of the extent analysis method of fuzzy ahp*
European Journal of Operational Research, 95, pp. 649-655

Ceschi, M., Sillitti, A., Succi, G., Panfilis, S.D., 2005. *Project management in plan-based and agile companies*, IEEE Software 22 (3), 21–27.

Chow, T., & Cao, D. B. (2008). *A survey study of critical success factors in agile software projects*. Journal of Systems and Software, 81(6), 961-971

Cohn, M., Ford, D., 2003. *Introducing an agile process to an organization*. IEEE Computer 36 (6), 74–78.

Boehm, B., Turner, R., 2003. *Using risk to balance agile and plan-driven methods*. IEEE Computer 36(6), 57–66. Pepperdine University, Malibu, California. Sloan School of Management, Center for Information Systems Research, Cambridge, Massachusetts.

Boehm B. and Turner R., 2003. *Observations on balancing discipline and agility*. Proceedings of the Agile Development Conference (ADC'03). June. Salt Lake City, Utah, USA.

Clean book: A handbook of agile software craftsmanship, Prenticehall, USA
<http://www.flazx.com/preview/0132350882,1>, [20.05.2008]

Concas G., Damiani E., Scotto M. & Succi G. (Eds.), 2007. *Agile processes in software engineering and extreme programming*, 8th International Conference, XP 2007, June 18-22, 2007 Proceedings, Springer, UK

Dove R. & Turkington G., 2008. *Relating agile development to agile operations*, Conference on Systems Engineering Research (CSER), University of Southern California, Redondo Beach, April, 2008
<http://www.parshift.com/Files/PsiDocs/Pap080404Cser2008DevOpsMigration.pdf>, [20.05.2008]

Karlstrom, D., Runeson, P., 2005. *Combining agile methods with star-gate project management*. IEEE Software 22 (3), 43–49.

Kelly A., 2008. *Changing software development: Learning to be agile*, John Wiley & Sons Ltd, UK

Kotonya G. & Sommerville I., 1998. *Requirements engineering*, John Wiley & Sons, Chichester, UK

Larman, C., 2004. *Agile & iterative development*. Addison-Wesley, Boston, Massachusetts.

Lefingwell, D., 2007. *Software Agility: Best Practices for Large Enterprises* (Agile Software Development Series) (Paperback), Addison-Wesley Professional; 1st edition

Klappholz, D., Bernstein, L., D. Port. 2003 *Assessing attitude towards, knowledge of, and ability to apply, software development process*, 16th Conference on Software Engineering Education and Training (CSEET'03), IEEE, pp. 268-278

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., et al., 2004. *Agile software development in large organizations*. IEEE Computer 37 (12), 26–34.

Hall T., Rainer A., Baddoo N., Beecham S., 2001, “*An empirical study of maintenance issues within process improvement programmes in the software industry*”, in Proceedings of the IEEE International Conference on Software Maintenance, 2001, pp. 422-430.

Reifer, D.J., Maurer, F., Erdogmus, H., 2003. *Scaling agile methods*. IEEE Software 20 (4), 12–14.

Rockart, J. (1979). *Chief Executives Define Their Own Information Needs*. In: *Harvard Business Review*, March/April 1979, 81-92.

Petersen, K., Wohlin, C. ,2009, *A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case*. The Journal of Systems and Software(2009), Elsevier Inc , doi:10.1016/j.jss.2009.03.036

Nerur, S., Mahapatra, R.K., Mangalaraj, G., 2005. *Challenges of migrating to agile methodologies*. Communications of the ACM 48 (5), 72–78.

Rockart, J. (1982). *The Changing Role of the Information Systems Executive: A Critical Success Factors Perspective*. In: *Sloan Management Review*, 23(1), 3-13.

Rockhart, J.F., Crescenzi, A.D., 1984. *Engaging top management in information technology*. Sloan Management Review 25 (4), 3–16.

Subhas C. Misra, Ph.D., 2008, Adopting Agile Software Project Management Practices: Success Factors , Changes Required, and Challenges , ICIM 2008 - International Conference on Innovation and Management <http://www.merit.unu.edu/ICIM2008/docs/SubhasMisra.pdf>, [10.03.2009]

Reel, J.S., 1999. *Critical success factors in software projects*. IEEE Software 16 (3), 18–23.

Schatz, B., Abdelshafi, I., 2005. *Primavera gets agile: A successful transition to agile development*. IEEE Software 22(3), 36–42.

Diğer Yayınlar

Abrahamson P., Salo O., Ronkainen J. & Warsta J.,2002.*Agile software development methods: review and analysis*, University Of Olulu, VTT Publications 478, Finland, <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf> , [01.02.2008]

Agile Manifesto,2001. *Manifesto for agile software development development*, <http://www.agilemanifesto.org>, [01.02.2008]

Agile Advisor, 2009. <http://www.agileadvisor.com/2009/01/yagni-and-cost-of-change-curve.html> [01.03.2009]

Agile Alliance. 2001. *Manifesto for agile software development development*, <http://www.agilealliance.org>, [01.02.2008]

Ambler. S.W., 2005a. *Where is the proof that agile methods work*. <http://www.agilemodeling.com/essays/proof.htm>, [01.02.2008]

Ambler. S.W., 2005b. *Communication on aAgile software projects*. <http://www.agilemodeling.com/essays/communication.htm>, [01.02.2008]

Ambler. S.W. Ambler. 2005c. *Remaining Agile*, <http://www.agilemodeling.com/essays/remainingAgile.htm>, [01.02.2008].

Ambler. S.W., 2006. *Supersize me: Scaling agile software development*, 14 (3), 46–48. <http://www.agilealliance.org/show/1442>, [01.02.2008].

Alam,P. N., 2003. *Agile process recommendations for a market-driven company*, *Master's Thesis*, MSE-2003:16, Blekinge Institute Of Technology, Dept. of Software Engineering and Computer Science, Ronneby ,Sweden [http://www.bth.se/fou/cuppsats.nsf/all/297c3d409be04b92c1256d43007a70f9/\\$file/Payam%20-%20Report.pdf](http://www.bth.se/fou/cuppsats.nsf/all/297c3d409be04b92c1256d43007a70f9/$file/Payam%20-%20Report.pdf), [01.02.2008]

Awad M.A., 2005, *A Comparison between aAgile and traditional software development methodologies*, *Report*, School of Computer Science and software Engineering, The University of Western Australia,

Beedle, B, *Xbreed*, <http://www.xbreed.net/description.html>, [24.03.08]

Bosghossian, Z.J., 2002. *An investigation into the critical success factors of software development process, time, and quality*, Ph.D. Thesis, Pepperdine University, Malibu, California.

Bullen, C.V., Rockhart, J.F., 1981. *A primer on critical success factors* (Working Paper No. 69), Massachusetts Institute of Technology, Sloan School of Management, Center for Information Systems Research, Cambridge, Massachusetts.

Derby E., Larsen D., 2006, *Agile retrospectives making good teams great*, Pragmatic Bookshelf, Dallas, USA <http://www.pragmaticprogrammer.com>, [20.03.2008]

Dogs, C.& Klimmer, T., 2004. *An evaluation of the uage of agile core practices:How they are used in industry and what we can learn from their usage*, Master Thesis, Software Engineering, School of Engineering Blekinge Institute of echnology , Thesis no: MSE-2004:07 [20.03.2008]

DSDM. 2003. *Dynamic systems development modeling*,DSDM consortium, <http://www.dsdm.org>, [20.03.2008]

eWorkshop, 2002. *Summary of the first eworkshop on agile methods*, <http://fcmd.umd.edu/projects/Agile/Summary/SummaryPF.htm>, [20.05.2008] .

TheFreeDictionary.com, <http://encyclopedia.thefreedictionary.com>, [15.04.2008]

Fowler M., 2005, *The New Methodology*, <http://martinfowler.com/articles/newMethodology.html>, [20.05.2008]

Hussain N., 2007. *Unwired enterprise systems*, Master's Thesis, 2007-111, IT University of Gothenburg, Department of Applied Information Technology, Gothenburg, Sweden, http://gupea.ub.gu.se/dspace/bitstream/2077/10497/1/gupea_2077_10497_1.pdf, [20.05.2008]

Kalissery, B., 2007. *Managing agile information technology infrastructure*, Master's Thesis, Massacahuttes Institute Of Technology, Master Of Science in System Design and Management Program, Massacahuttes,USA, <http://dspace.mit.edu/handle/1721.1/42363>,[20.05.2008]

Kalermo J. & Rissanen J., 2002. *Agile software development in theory and practise* , Master's Thesis, University Of Jyvasckyla, Department Of Computer Science and Information Systems”, Jyvakyala, Finland http://www.cs.jyu.fi/sb/Publications/KalermoRissanen_MastersThesis_060802.pdf, [20.05.2008]

Kaplan, S., 2007. *Hava savunma sektörü tezgah yatırım projelerinin bulanık ahp ile değerlendirilmesi, Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Bölümü

Khamooshi P., 2008, *Test driven development*, <http://pooya.khamooshi.com/Pages/MyArticles/Index.aspx>,[12.10.2008]

Koch, A.S., 2005. *Agile software development: Evaluating the methods for your organizations*. Artech House, Northwood, Massachusetts.

Köstence N., 2009, *Kurumsal kaynak planlaması yazılım paketleri ve kuruma Özel yazılımların seçim aşamasında karşılaştırılması, Yüksek Lisans Tezi*, Bahçeşehir Üniversitesi, FBE, Bilgi Teknolojileri Programı

M. A. Awad,, 2005. *A comparison between agile and traditional software development methodologies, Report*, School of Computer Science and software Engineering, The University of Western Australia

M. Fowler and J. Highsmith. 2001. *The Agile Manifesto*. Software Development Magazine. August,
<http://www.sdmagazine.com/documents/s=844/sdm0108a/0108a.htm>, [30.05.2008.].

Martin F., 2001. *The agile manifesto: where it came from and where it may go*,
<http://martinfowler.com/articles/agileStory.html>, [30.05.2008.].

Saarnak S. & Gustafsson B., 2003 , *A comparison of lifecycles - Agile software processes vs. projects in non-Agile software companies*, Master's Thesis, MSE-2003:12, Department of Software Engineering and Computer Science, Blekinge Institute of Technology, Ronneby ,Sweden

Subhas C. Misra, Vinod Kumar, and Uma Kumar, 2006, *Success Factors of Agile Software Development*, Carleton University, Ottawa, Canada,
<http://www1.ucmss.com/books/LFS/CSREA2006/SER5088.pdf>, [10.05.2008]

Tsun C. & Cao D.B., 2007. *A survey study of critical success factors in agile software projects*, School of Business and Technology, Capella University, Minneapolis,USA

Murray C., 2008. *Lean and agile software development: a case study*,
Master's Thesis, Massacahuttes Institute Of Technology, Master Of Science in Management and Technology, Massacahuttes, USA
<http://dspace.mit.edu/handle/1721.1/43176?show=full>, [11.036.2009]

Murauskaite A. & Adomaskas A., 2008. *Bottlenecks in agile software development identified using theory of constraints (toc) principles*, Master's Thesis, MSE-2008:14, Department of Computer Science, University of Gothenburg, Gothenburg ,Sweden
http://gupea.ub.gu.se/dspace/bitstream/2077/10457/1/gupea_2077_10457_1.pdf, [11.03.2009]

EK 1: Anket Soruları

Ad Soyad :
Telefon :
Firma Adı :
Hangi sektörde çalışıyorsunuz? :

Şirketinizde toplam kaç kişi çalışıyor	5 kişiye kadar	100 kişiye kadar	1000 kişiye kadar	5000 kişiye kadar	5000 kişiden fazla
	[]	[]	[]	[]	[]

Şirketinizde kaç yıldır çalışıyorsunuz?	1 yıla kadar	3 yıla kadar	5 yıla kadar	10 yıla kadar	10 yıldın büyük
	[]	[]	[]	[]	[]

Hangi rolde çalışıyorsunuz?	Yönetici	Geliştirici	Analist	Testçi	Diğer
	[]	[]	[]	[]	[]

Hangi tip projelerde çalışıyorsunuz?	Kurumsal Yazılımlar	E-ticaret Yazılımları	Güvenlik Yazılımları	Sistem Yazılımları	Diğer
	[]	[]	[]	[]	[]

Projenizde kaç kişi çalışıyor	5 kişiye kadar	10 kişiye kadar	30 kişiye kadar	50 kişiye kadar	50 kişiden fazla
	[]	[]	[]	[]	[]
Projelerinizde Çevik Geliştirme Süreçleri kullanıyor yada kullanmayı düşünmekte misiniz?	evet	[]		Hayır	[]

Kaç yıldır Çevik Geliştirme Süreçleri kullanmaktasınız?	1 yıla kadar	2 yıla kadar	3 yıla kadar	5 yıla kadar	Hiç Biri
	[]	[]	[]	[]	[]

A. Başarılı Bir Yazılım Projesi	Tümüyle doğru	Büyük ölçüde doğru	Kararsızım	Büyük ölçüde yanlış	Tümüyle yanlış
A1- İhtiyaçları tam karşılayandır	[]	[]	[]	[]	[]
A2- Belirlenen sürede tamamlanandır	[]	[]	[]	[]	[]
A3- Belirlenen Maliyetle Tamamlanandır	[]	[]	[]	[]	[]
A4- Performans kriterlerini sağlayandır	[]	[]	[]	[]	[]
A5- Değişikliklere hızlı cevap verebilendir	[]	[]	[]	[]	[]

B. Yazılım Geliştirmede Verimlilik İçin Yapılması Gereken iyileştirmeler	Çok önemli	Büyük ölçüde önemli	Kısmen önemli	Çok az önemli	Hiç önemli değil
B1-Süreç iyileştirmeleri	[]	[]	[]	[]	[]
B2-Organizasyonel iyileştirmeler	[]	[]	[]	[]	[]
B3-Yazılım geliştirme ortamıyla ilgili iyileştirmeler	[]	[]	[]	[]	[]
B4-Altyapısal iyileştirmeler	[]	[]	[]	[]	[]

C. Yazılım Projelerinizde Verimliliği Eteleyen Ana Faktörler	Çok önemli	Büyük ölçüde önemli	Kısmen önemli	Çok az önemli	Hiç önemli değil
C1- Kullanılan programlama dili	[]	[]	[]	[]	[]

C2- Kullanılan Yazılım Çatısı	[]	[]	[]	[]	[]
C3- Kullanılan İde	[]	[]	[]	[]	[]
C4- Kullanılan platform	[]	[]	[]	[]	[]
C5- Kullanılan araçlar	[]	[]	[]	[]	[]
C6- Kullanılan mimari	[]	[]	[]	[]	[]
C7- Kullanılan donanım	[]	[]	[]	[]	[]
C8- Kullanılan algoritma	[]	[]	[]	[]	[]
C9- Varolan standartlar	[]	[]	[]	[]	[]
C10-Veritabanı yapısı	[]	[]	[]	[]	[]
C11-Yapılan tasarım	[]	[]	[]	[]	[]
C11-Desteklenen iş süreçleri	[]	[]	[]	[]	[]
C12-Desteklenen teknik özellikler	[]	[]	[]	[]	[]
C13-Desteklenen fonksiyonel özellikler	[]	[]	[]	[]	[]
C14-Takip edilen süreç	[]	[]	[]	[]	[]
C16- Ekip yetenek, eğitim ve tecrübesi	[]	[]	[]	[]	[]
C17-Hedef ve beklentilerin netliği	[]	[]	[]	[]	[]
C18-Yönetim desteği	[]	[]	[]	[]	[]
C19-İsbirligi düzeyi	[]	[]	[]	[]	[]
C20-İletişim düzeyi	[]	[]	[]	[]	[]
C21-İşin kompleksliği	[]	[]	[]	[]	[]
C22- Çalışanların karakter ve motivasyonu	[]	[]	[]	[]	[]

D.Yazılım Geliştirmede Verimliliği Arttıran Altyapısal Özellikler	Çok önemli	Büyük ölçüde önemli	Kısmen önemli	Çok az önemli	Hiç önemli değil
D1- Zengin ve hızlı geliştirme altyapıları kullanmak	[]	[]	[]	[]	[]
D2- Servis tabanlı mimari kullanmak	[]	[]	[]	[]	[]
D3- Farklı veritabanlarını desteklemek	[]	[]	[]	[]	[]
D4- Farklı donanımlarda(PC, Mobil Cihazlar...) çalışabilecek yazılım alt yapısı kullanmak	[]	[]	[]	[]	[]
D5- Farklı işletim sistemleri(Windows,Unix..) ve platform(web,mobil..) desteği	[]	[]	[]	[]	[]
D6- Çevik ve Çevik ve Güncel yazılım geliştirme dilleri kullanmak	[]	[]	[]	[]	[]
D7- Xml ile esnek rapor üreticiler kullanmak	[]	[]	[]	[]	[]
D8- Ajax desteğine sahip olmak	[]	[]	[]	[]	[]
D9- Esnek yetkilendirme altyapısına sahip olmak	[]	[]	[]	[]	[]
D10-Çizelgeleme altyapısı ile arka planda çalışma ve çizelgeleme özelliği	[]	[]	[]	[]	[]
D11-Model güdümlü mimari yapı kullanmak	[]	[]	[]	[]	[]
D12-Web servisi uyumlu adapter alt yapısı kullanmak	[]	[]	[]	[]	[]
D13-Görev yönetimi, İhtiyaç analizi ve görevin gelişim durumunu takip için entegre proje yönetim araçları kullanmak	[]	[]	[]	[]	[]
D14-Özelleştirme altyapısı kullanarak farklı dillerde çalıştırabilme	[]	[]	[]	[]	[]
D15-İş akış sistemi alt yapısı ile iş süreçlerinin esnek yönetilebilmesi	[]	[]	[]	[]	[]
D16-Kural tabanlı dinamik yazılım geliştirme alt yapısına sahip olmak	[]	[]	[]	[]	[]
D17-Mesajlaşma sistemiyle farklı sunucular arasında veri alışverişi	[]	[]	[]	[]	[]
D18-Yazılım geliştirirken otomatik kod üreticilerden faydalanmak	[]	[]	[]	[]	[]
D19-Arayüz geliştirirken otomatik arayüz üreticilerden faydalanmak	[]	[]	[]	[]	[]
D20-Otomatik test araçlarından faydalanmak	[]	[]	[]	[]	[]
D21-Otomatik versiyon kurum araçlarından faydalanmak	[]	[]	[]	[]	[]

D22-Hata ayıklama mekanizmaları kullanmak	[]	[]	[]	[]	[]
D23-Sistem monitör araçları kullanarak sistem çalışırkenki performansını izlemek	[]	[]	[]	[]	[]
E.Yazılım geliştirme Süreç ve Organizasyonu ile ilgili faktörler	Çok önemli	Büyük ölçüde önemli	Kısmen önemli	Çok az önemli	Hiç önemli değil
E1- Proje hedeflerini(neyi, ne zaman, hangi sırada, hangi maliyetle ve riskle) baştan doğru belirleyip tüm taraflarla paylaşmak.	[]	[]	[]	[]	[]
E2- Ana iş süreçlerine odaklanmak	[]	[]	[]	[]	[]
E3- Liderlik için belli birkaç alan seçmek	[]	[]	[]	[]	[]
E4- Küçük bir ekiple çalışmak	[]	[]	[]	[]	[]
E5- Kendi kendine organize olabilen bir ekiple çalışmak	[]	[]	[]	[]	[]
E6- Ekip içi etkin bilgi paylaşımı	[]	[]	[]	[]	[]
E7- Deneyimli personelle çalışmak	[]	[]	[]	[]	[]
E8- Geliştirme adımlarını basitleştirip gereksizlerini kaldırmak	[]	[]	[]	[]	[]
E9- Tekrarlı işleri ortadan kaldırmak	[]	[]	[]	[]	[]
E10- Basit ürünler ve ürün aileleri üretmek	[]	[]	[]	[]	[]
E11-Başarılı ürün süreç ve bileşenleri tekrar tekrar kullanmak	[]	[]	[]	[]	[]
E12-Sadece müşterilerin ihtiyacı olan ve müşteriye değer katan özellikler geliştirmek	[]	[]	[]	[]	[]
E13-İhtiyaç analizi ve tasarımı müşteriyle birlikte yapmak	[]	[]	[]	[]	[]
E14-İhtiyaçları küçük senaryolara ayırarak önceliklendirip sıralamak ve en öncelikliden işe başlamak	[]	[]	[]	[]	[]
E15-Tüm süreçlerin arasında entegre bilgi akışı sağlamak	[]	[]	[]	[]	[]
E16-Ekibin işleri bitirme hızı ,hata oranları ve hataların sebeplerini sürekli izlemek	[]	[]	[]	[]	[]
E17-Sürekli tüm süreçler için refactoring yaparak sorunlu noktaları ayıklamak	[]	[]	[]	[]	[]
E18-Daha kısa sürümlerle daha az özellik geliştirerek müşteriye hızlı geri dönüş	[]	[]	[]	[]	[]
E19-Sorunların çözümü ve yeni gelişim fırsatlarını belirlemek için beyin fırtınası faaliyetlerini yapmak.	[]	[]	[]	[]	[]

F.Çevik Yazılım Geliştirme Süreçleri Kullanmaya Teşvik Eden Sebepler	Çok fazla	Büyük ölçüde	Kısmen	Çok az	Hiç
F1- Proje çevrim süresi	[]	[]	[]	[]	[]
F2- Yazılımın karmaşıklığı	[]	[]	[]	[]	[]
F3- İhtiyaçların dengeye gelmesi için	[]	[]	[]	[]	[]
F4- Proje takımının büyüklüğü	[]	[]	[]	[]	[]
F5- Projenin büyüklüğü	[]	[]	[]	[]	[]
F6- Güvenlik ve kritiklik seviyesi	[]	[]	[]	[]	[]
F7- Tekrar kullanılabilir parçaların inşası için	[]	[]	[]	[]	[]
F8- Dağıtık geliştirme	[]	[]	[]	[]	[]
F9- Dış hizmet kullanımı	[]	[]	[]	[]	[]
F10-Kişisel ilgi	[]	[]	[]	[]	[]
F11-Tartışma grupları	[]	[]	[]	[]	[]
F12-Şirketiniz	[]	[]	[]	[]	[]
F13-Profesyonel organizasyonlar	[]	[]	[]	[]	[]
F14-Eğitim seminerleri	[]	[]	[]	[]	[]
F15-Ticari yayınlar	[]	[]	[]	[]	[]

G. Çevik Yazılım Geliştirme Süreç Uygulamalarının Verimliliğe Etkisi.	Çok olumlu etkiledi	Olumlu etkiledi	Kısmen olumlu etkiledi	Etkilemedi	Olumsuz etkiledi
G1- Müşteriyle yerinde iletişim kurmak	[]	[]	[]	[]	[]
G2- Üst seviye ihtiyaç analizinin müşteriyle birlikte yapılması	[]	[]	[]	[]	[]
G3- Detaylı ihtiyaç analizleri yapılması	[]	[]	[]	[]	[]
G4- Değişen İhtiyaçları Dikkate alınması	[]	[]	[]	[]	[]
G5- Üst seviye use case kullanımı	[]	[]	[]	[]	[]
G6- Detaylı use case kullanımı	[]	[]	[]	[]	[]
G7- İterasyon planlama	[]	[]	[]	[]	[]
G8- Sürüm planlama	[]	[]	[]	[]	[]
G9- Test planlama	[]	[]	[]	[]	[]
G10-modelleme standartları	[]	[]	[]	[]	[]
G11-Veri isimlendirme standartları	[]	[]	[]	[]	[]
G12-Kodlama standartları	[]	[]	[]	[]	[]
G13-Arayüz standartları	[]	[]	[]	[]	[]
G14-Test standartları	[]	[]	[]	[]	[]
G15-Versiyon Kurulum standartları	[]	[]	[]	[]	[]
G16-Uyarılama standartları	[]	[]	[]	[]	[]
G17-Problem çözme standartları	[]	[]	[]	[]	[]
G18-Dokümantasyon standartları	[]	[]	[]	[]	[]
G19-Ön Mimari modelleme	[]	[]	[]	[]	[]
G20-Ön İhtiyac modelleme	[]	[]	[]	[]	[]
G20-Bilgisayar Destekli Araçlarla modelleme	[]	[]	[]	[]	[]
G21-Paralel modelle	[]	[]	[]	[]	[]
G22-Model gözden gecirmeleri	[]	[]	[]	[]	[]
G23-Detaylı mimari tasarım	[]	[]	[]	[]	[]
G24-Ana hatlarıyla mimari tasarım	[]	[]	[]	[]	[]
G25-Basit tasarım	[]	[]	[]	[]	[]
G26-Bakım Yapılabilir tasarım	[]	[]	[]	[]	[]
G27-Evrimsel tasarım	[]	[]	[]	[]	[]
G28-Tasarım gözden gecirme	[]	[]	[]	[]	[]
G29-Veri tabanı gözden gecirme	[]	[]	[]	[]	[]
G30-Arayüz gözden gecirme	[]	[]	[]	[]	[]
G31-Manuel görev tahtası	[]	[]	[]	[]	[]
G32-Dijital görev tahtası	[]	[]	[]	[]	[]
G33-Tekrarlı işleri kaldırılması	[]	[]	[]	[]	[]
G34-Daha basit ürün geliştirme	[]	[]	[]	[]	[]
G35-Bileşenlerin yeniden kullanımı	[]	[]	[]	[]	[]
G36-Otomatize kod geliştirme araçları kullanımı	[]	[]	[]	[]	[]
G37-Otomatize arayüz geliştirme araçları	[]	[]	[]	[]	[]
G38-Sürekli kod entegrasyonu	[]	[]	[]	[]	[]
G39-Sürekli veri tabanı entegrasyonu	[]	[]	[]	[]	[]
G40-Test güdümlü geliştirme	[]	[]	[]	[]	[]
G41-Unit test araçları kullanımı	[]	[]	[]	[]	[]
G42-Kabul testleri	[]	[]	[]	[]	[]
G43-Bağımsız- Kesif testleri	[]	[]	[]	[]	[]
G44-Gelistirici testleri	[]	[]	[]	[]	[]
G45-Otomatize test araçlarıyla test	[]	[]	[]	[]	[]
G46-Otomatik hata ayıklama mekanizmaları kullanımı	[]	[]	[]	[]	[]
G47-Günlük toplantılar yapma	[]	[]	[]	[]	[]

G48-Sürekli durum raporlarıyla yapılan, yapılacak ve kalan işlerin takibi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G49-Özet gant chartlar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G50-Detaylı gant chartlar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G51-Proje Degerlendirme\Alinan Dersler' Paylasim toplantisi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G52-Hata Raporlari	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G53-Hata Egilim Metrikleri	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G54-Tum Taraflarin Aktif Katilimi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G55-Kollektif Kod sahipliği	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G56-Kendi kendine organize olan takimler	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G57-Eşli Programlama	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G58-Açık çalışma alanı	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G59-Davranış Güdümlü Programlama	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G60-Kanban	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G61-Otomatize versiyon paketleme ve kurulumu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G62-Çalışan yazılımın evrimsel paketlenmesi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G63-Küçük sürümlerle çalışma	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G64-Tekrarlamalı geliştirme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G65-Sürdürülebilir tempoda ilerleme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G66-Yerel fayda yerine genel Faydayı düşünme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cevik süreçlerin faydalarını firmanız ve müşterileriniz açısından derecelendiriniz?

	Çok olumlu etkiledi	Olumlu etkiledi	Kısmen olumlu etkiledi	Etkilemedi	Olumsuz etkiledi
<i>H.Şirketinize katkıları</i>					
H1- Proje görünürliğünde artış	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H2- Verimlilikte artış	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H3- Yazılım kalitesinde gelişme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H4- Maliyetlerde düşüş	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H5- Geliştirme sürecindeki basitlik	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H6- Mühendislik disiplninde artış	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H7- Yazılım bakımı ve genişletilmesinde artış	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H8- Yazılım ve iş sistemleri arasında artan uyum	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H9- Risklerde azalma	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H10- Pazara çıkış hızında artış	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H11-Değişen öncelikleri daha kolay yönetebilme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H12-Takım moralinde yükselme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H13-Bakım ve geliştirmede kolaylık	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H14-Daha fazla rekabet avantajı	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H15-Müşteri portföyünde artış	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>I.Müşterilere katkıları</i>					
I1- Daha fazla deger sunar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2- Müşteriye esneklik sağlar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I3- Operasyonel verimliliği artırır	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I4- Müşteri Tatminini artırır	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I5- Risklerde azalma	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I6- Pazara çıkış hızında artış	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I7- Maliyetlerde Azalma	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

J.Çevik Yazılım Geliştirme Süreçlerinde kullanılan araçları verimliliğe etkisi açısından derecelendiriniz.	Çok olumlu etkiledi	Olumlu etkiledi	Kısmen olumlu etkiledi	Etkilemedi	Olumsuz etkiledi
J1- Proje yönetim araçları	[]	[]	[]	[]	[]
J2- Hesap tablosu	[]	[]	[]	[]	[]
J3- Hata ayıklayıcı	[]	[]	[]	[]	[]
J4- Unit test aracı	[]	[]	[]	[]	[]
J5- Otomatize kabul testi aracı	[]	[]	[]	[]	[]
J6- Kod değerlendirme aracı	[]	[]	[]	[]	[]
J7- Görev masası aracı	[]	[]	[]	[]	[]
J8- ihtiyaç yönetimi aracı	[]	[]	[]	[]	[]
J9- Sürekli entegrasyon araçları	[]	[]	[]	[]	[]
J10-Otomatize kod geliştirme araçları	[]	[]	[]	[]	[]
J11-Otomatize arayüz geliştirme araçları	[]	[]	[]	[]	[]
J12-Kanban kartı	[]	[]	[]	[]	[]
J13-Versiyon yönetim araçları	[]	[]	[]	[]	[]
J14-Entegre Yazılım Çatısı araçları	[]	[]	[]	[]	[]
J15-Performans izleme araçları	[]	[]	[]	[]	[]
J16-Yardım masası araçları	[]	[]	[]	[]	[]
K.Çevik araçların seçiminde etkili olan faktörler	Çok fazla	Büyük ölçüde	Kısmen	Çok az	Hiç
K1- Hızlı ve daha kolay	[]	[]	[]	[]	[]
K2- İzlenebilirlik	[]	[]	[]	[]	[]
K3- Üst seviye görev takibi	[]	[]	[]	[]	[]
K4- Çevik sürece uygunluk	[]	[]	[]	[]	[]
K5- Dağıtık organizasyon	[]	[]	[]	[]	[]
K6- Çoklu takımlar için ölçeklenebilir	[]	[]	[]	[]	[]
K7- Geliştirmede esneklik	[]	[]	[]	[]	[]
K8- Düşük geliştirme maliyetleri	[]	[]	[]	[]	[]
K9- Artan kalite seviyesi	[]	[]	[]	[]	[]
K10-Müşteri ihtiyaçlarına daha iyi uyum	[]	[]	[]	[]	[]
K11-Verimlilik Artışı	[]	[]	[]	[]	[]
K12-Risklerde düşme	[]	[]	[]	[]	[]
L.Çevik Yöntemlerin başarılı bir şekilde uygulanmasının önündeki engeller	Çok fazla	Büyük ölçüde	Kısmen	Çok az	Hiç
L1- Organizasyonel kültürü değiştirme zorluğu	[]	[]	[]	[]	[]
L2- Değişime karşı genel direnç	[]	[]	[]	[]	[]
L3- Çevik pratikler konusunda tecrübeli personel eksikliği	[]	[]	[]	[]	[]
L4- Eksik yönetim desteği	[]	[]	[]	[]	[]
L5- Proje karmaşıklığı yada ölçeği	[]	[]	[]	[]	[]
L6- Çevik yöntemleri uygulama konusundaki cesaret	[]	[]	[]	[]	[]
L7- Çevik dönüşümü kavrama için gereken zaman	[]	[]	[]	[]	[]
L8- Uzun döngü zamanı	[]	[]	[]	[]	[]
L9- Bütçe sabitleri	[]	[]	[]	[]	[]
L10-İsteksizlik	[]	[]	[]	[]	[]
L11-Şirket çalışma yapısı(dış hizmet kullanım oranı yüksek)	[]	[]	[]	[]	[]
L12-Süreç bağımsızlığı	[]	[]	[]	[]	[]
L13-Planlamada eksiklik	[]	[]	[]	[]	[]
L14-Yönetimsel kontrolde kayıp riski	[]	[]	[]	[]	[]
L15-Eksik Dökümantasyon	[]	[]	[]	[]	[]
L16-Eksik tahmin edilebilirlik	[]	[]	[]	[]	[]
L17-Eksik mühendislik disiplini	[]	[]	[]	[]	[]

ÖZGEÇMİŞ:

Bu tezin yazarı, Ercan DÖNMEZ, 18-07-1977'de İzmir'in Ödemiş ilçesinde doğdu. Ödemiş'te aldığı İlk ve orta öğretimi ardından,1994 yılında başladığı Balıkesir Üniversitesi Endüstri Mühendisliği Bölümü'nden 1998 yılında birincilikle Lisans derecesi aldı. Mezuniyet sonrası çeşitli özel şirketlerde çalıştı. 2006 yılında Bahçeşehir Üniversitesi Fen Bilimleri Enstitüsü Bilgi Teknolojileri Yüksek Lisans Programı'na başladı.

2001 yılından buyana çalıştığı Şahinler Holding Bilgi İşlem Bölümünde ERP(Kurumsal Kaynak Planlaması) yazılım projesi çerçevesinde 10 dan fazla projenin sistem analiz, tasarım, test ve uygulamaya alma aşamalarında görev aldı.Ve halen sistem analist ve ERP danışmanı olarak yeni projelerin geliştirilmesi çalışmalarına devam etmektedir.