

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
ENDÜSTRİ MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI**

**ÇOK PROJELİ KAYNAK KISITLI YAZILIM GELİŞTİRME
PROJELERİNİN BENZETİMLE GÖSTERİMİ VE BENZETİM
ENİYİLEMESİ**

HAZIRLAYAN

FERDA GÜVEN

YÜKSEK LİSANS TEZİ

ANKARA - 2021

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
ENDÜSTRİ MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI**

**ÇOK PROJELİ KAYNAK KISITLI YAZILIM GELİŞTİRME
PROJELERİNİN BENZETİMLE GÖSTERİMİ VE BENZETİM
ENİYİLEMESİ**

HAZIRLAYAN

FERDA GÜVEN

YÜKSEK LİSANS TEZİ

TEZ DANIŞMANI

DR. ÖĞR. ÜYESİ BURÇİN ÇAKIR

ANKARA – 2021

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Tezli Yüksek Lisans Programı çerçevesinde Ferda GÜVEN tarafından hazırlanan bu çalışma, aşağıdaki jüri tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 08 / 01 / 2021

Tez Adı: Çok Projeli Kaynak Kısıtlı Yazılım Geliştirme Projelerinin Benzetimle Gösterimi ve Benzetim Eniyilemesi

Tez Jüri Üyeleri (Unvanı, Adı - Soyadı, Kurumu)

İmza

Prof. Dr. Yusuf Tansel İÇ, Başkent Üniversitesi

.....

Dr. Öğr. Üyesi Burçin ÇAKIR, Başkent Üniversitesi

.....

Dr. Öğr. Üyesi Ayyüce AYDEMİR KARADAĞ, Çankaya Üniversitesi

.....

ONAY

Prof. Dr. Faruk ELALDI

Fen Bilimleri Enstitüsü Müdürü

Tarih : ... / ... /

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: 13 / 01 / 2021

Öğrencinin Adı, Soyadı : Ferda GÜVEN

Öğrencinin Numarası : 21710300

Anabilim Dalı : Endüstri Mühendisliği Anabilim Dalı

Programı : Endüstri Mühendisliği Tezli Yüksek Lisans Programı

Danışmanın Unvanı/Adı, Soyadı : Dr. Öğr. Üyesi Burçin ÇAKIR

Tez Başlığı : Çok Projeli Kaynak Kısıtlı Yazılım Geliştirme Projelerinin Benzetimle Gösterimi ve Benzetim Eniyilemesi

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 100 sayfalık kısmına ilişkin, 13 / 01 / 2021 tarihinde şahsım/tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 5'tir. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:

ONAY

Tarih: 13 / 01 / 2021

Dr. Öğr. Üyesi Burçin ÇAKIR

İmza:

TEŐEKKÖR

Bu tez alıřmasının tÖm sÖrecinde emeęi ve katkılarıyla bana destek olan deęerli danıřman hocam Sayın Dr. Öęr. Üyesi Burın AKIR’a,

YÖksek lisans ders ařamasında bilgi ve deneyimlerinden yararlandığım ve desteklerini gördüğÖm Bařkent Üniversitesi EndÖstri MÖhendislięi BÖlÖmÖ’nde alıřan deęerli hocalarıma,

Tez savunma jÖrimde yer alan ve önerileriyle tezime katkıda bulunan hocalarım Sayın Prof. Dr. Yusuf Tansel İ ve Sayın Dr. Öęr. Üyesi AyyÖce AYDEMİR KARADAĞ’a,

Hayatım boyunca bana her konuda destek olan sevgili annem, babam ve ablama,

TeőekkÖrlerimi sunarım.

ÖZET

Ferda GÜVEN

ÇOK PROJELİ KAYNAK KISITLI YAZILIM GELİŞTİRME PROJELERİNİN BENZETİMLE GÖSTERİMİ VE BENZETİM ENİYİLEMESİ

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

2021

Günümüzde iş hayatı ve sosyal hayat başta olmak üzere neredeyse her alanda kullanılan yazılımların üretilmesi ve bireysel ya da toplu kullanıma uygun hale getirilmesi sürecinde, yazılım projelerinin yönetimi oldukça önemlidir. Müşteri memnuniyetinin ön planda olduğu bu dönemde, yazılım firmalarının kısıtlı kaynakları ile farklı boyuttaki ve zorluktaki projelerin ihtiyaçlarını karşılayabilmek için doğru bir planlama yapmaları gerekmektedir. Planlama yaparken, çalışanların farklı yetenek ve tecrübeye sahip olması ve her projenin gerekliliklerinin farklı olması da dikkate alınması gereken unsurlardandır. Bu çalışmada, yazılım geliştirme projelerinin aşamaları analiz edilerek belirlenen performans ölçütlerini iyileştirecek yetenek tabanlı bir personel planlaması problemi dikkate alınmıştır. Çalışmanın amacı, doğru işgücü planlaması ile tamamlanan proje sayısı, projelerin tamamlama süresi, çalışan maliyeti ve toplam kârlılıkta iyileşme sağlamaktır. Bu çelişen amaçlarda en uygun planlamayı yapabilmek için Benzetim modellemesi tekniğinden faydalanılmıştır. Gerçek bir yazılım firmasından uyarlanmış veriler ile, her projede çalışabilecek personel yetenek ve tecrübe seviyesi kısıtları dikkate alınarak, mevcut durumun modellemesi yapılmış, darboğazlar tespit edilerek alternatif sistem önerilerinde bulunulmuştur. Ayrıca Benzetim Eniyilemesi metodu ile yazılım geliştirme projesinin farklı aşamalarında çalışacak en uygun personel sayısı tespit edilmiştir. Sonuçlar incelendiğinde, Benzetim eniyilemesinin, mevcut sistemi, ele alınan performans ölçütleri açısından iyileştirdiği gözlenmiştir.

ANAHTAR KELİMELE: Yazılım Proje Yönetimi, Benzetim, Simülasyon, Deney Tasarımı, Meta-Model, Simülasyon Optimizasyonu, Benzetim Eniyilemesi.

ABSTRACT

Ferda GÜVEN

SIMULATION OF MULTI-PROJECT RESOURCE CONSTRAINED SOFTWARE DEVELOPMENT PROJECTS AND SIMULATION OPTIMIZATION

Başkent University Institute of Science

Department of Industrial Engineering

2021

Today, the management of software projects is crucial in the process of producing software used in almost every field, especially in business life and social life, and making it suitable for individual or collective use. In this period where customer satisfaction is at the forefront, software companies have to make an accurate planning in order to meet the needs of projects of different sizes and difficulties with their limited resources. While planning, the fact that employees have different skills and experience and that the requirements of each project are different are also factors that should be taken into account. In this study, a skill-based personnel planning problem that will improve the performance criteria determined by analyzing the stages of software development projects is taken into account. The aim of the study is to improve the number of completed projects, completion time of projects, employee costs and total profitability with correct workforce planning. Simulation modeling technique has been used to make the most appropriate planning for these conflicting purposes. With data adapted from a real software company, the current situation was modeled, considering the limitations of personnel ability and experience level that can work in each project, bottlenecks were identified and alternative system recommendations were made. In addition, the most suitable number of personnel to work at different stages of the software development project was determined with the simulation optimization method. When the results were examined, it was observed that simulation optimization improved the existing system in terms of the performance criteria considered.

KEY WORDS: Software Project Management, Simulation, Experimental Design, Meta-Model, Simulation Optimization.

İÇİNDEKİLER

TEŞEKKÜR.....	i
ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ	vii
ŞEKİLLER LİSTESİ	ix
SİMGELER VE KISALTMALAR LİSTESİ	x
1. GİRİŞ.....	1
1.1. Yazılım Projeleri ve Yazılım Projelerinde Proje Yönetimi.....	2
1.2. Yazılım Geliştirme Yaşam Döngüsü ve Süreç Modelleri	5
1.3. Yazılım Projelerini Etkileyen Temel Faktörler	8
1.4. Yazılım Geliştirme Projelerinde Kullanılan Çözüm Yöntemleri	10
1.4.1. Çizelgeleme Yöntemi.....	12
1.4.2. Benzetim Yöntemi	13
1.5. Araştırmanın Amacı ve Önemi.....	17
2. İLGİLİ ARAŞTIRMALAR.....	19
2.1. Yazılım Geliştirme Projelerinde Benzetim ile İlgili Yapılan Araştırmalar.....	19
2.2. Yazılım Geliştirme Projelerinde Çizelgeleme ile İlgili Yapılan Araştırmalar	28
3. YÖNTEM.....	33
3.1. Yazılım Firması Sistemi Hakkında Bilgiler.....	33
3.1.1. Süreçler.....	34
3.1.2. Ekiplerin Özellikleri	34
3.1.3. Projelerin Özellikleri.....	36
3.1.4. Projelerin Bedelleri ve Geliş Zamanları.....	38
3.1.5. Çalışanların Saatlik Ücretleri.....	39
3.1.6. Kısıtlar ve Varsayımlar	40
3.1.7. Girdi Parametreleri.....	41
3.1.8. Dikkate Alınan Performans Ölçütleri	41
3.2. Benzetim Modeli ve Benzetim Eniyilemesi	42
3.2.1. Sistemin ARENA Benzetim Programında Gösterilmesi.....	42
3.2.2. Benzetim Eniyilemesi	54
4. BULGULAR VE TARTIŞMA.....	57

4.1.	Benzetim Modelinin Doğruluğu ve Geçerliliği.....	58
4.1.1.	Benzetim Modelinin Doğrulanması	58
4.1.2.	Benzetim Modelinin Geçerliliği.....	58
4.2.	Mevcut Sistemin Benzetim Modeli Çıktıları	58
4.3.	Çıktı Analizi.....	63
4.3.1.	Ortalamalar için Güven Aralığı	63
4.4.	Deney Tasarımı ve Meta-Model Kullanılarak Benzetim Eniyilemesi.....	68
4.4.1.	Deney Tasarımı ve Senaryo Denemeleri	68
4.4.2.	Meta-Model Oluşturma	73
4.4.3.	Karar Modeli ve Optimizasyon.....	85
4.4.4.	Çok Ölçütlü Karar Modeli ve Optimizasyon.....	90
5.	SONUÇ VE ÖNERİLER.....	95
5.1.	Sonuçlar	95
5.2.	Öneriler	97
	KAYNAKLAR.....	98
	EKLER	106
	EK 1: Yalın Proje Türü Geliş Zamanlarına Göre Girdi Analizi	106
	EK 2: Yalın Proje Türü Proje Bedellerine Göre Girdi Analizi.....	106
	EK 3: ARENA Benzetim Modeli Çıktısı.....	107
	EK 4: ARENA Process Analyzer Sonuçları	119
	EK 5: Ortalama Tamamlanan Toplam Proje Sayısı Performans Ölçütüne Göre Oluşturulan Meta-Model Faktöriyel Regresyon Analizi MİNİTAB Sonucu.....	121
	EK 6: Ortalama Tamamlanan Toplam Proje Sayısı Performans Ölçütüne Göre Oluşturulan Meta-Model Regresyon Analizi MİNİTAB Sonucu.....	122
	EK 7: Projelerin Ortalama Toplam Tamamlama Süresi Performans Ölçütüne Göre Oluşturulan Meta-Model Faktöriyel Regresyon Analizi MİNİTAB Sonucu.....	123
	EK 8: Projelerin Ortalama Toplam Tamamlama Süresi Performans Ölçütüne Göre Oluşturulan Meta-Model Regresyon Analizi MİNİTAB Sonucu.....	124
	EK 9: Ortalama Toplam Çalışan Maliyeti Performans Ölçütüne Göre Oluşturulan Meta-Model Faktöriyel Regresyon Analizi MİNİTAB Sonucu.....	125
	EK 10: Ortalama Toplam Çalışan Maliyeti Performans Ölçütüne Göre Oluşturulan Meta-Model Regresyon Analizi MİNİTAB Sonucu	126
	EK 11: Ortalama Toplam Kâr Performans Ölçütüne Göre Oluşturulan Meta-Model Faktöriyel Regresyon Analizi MİNİTAB Sonucu.....	127

EK 12: Ortalama Toplam Kâr Performans Ölçütüne Göre Oluşturulan Meta-Model Regresyon Analizi MİNİTAB Sonucu.....	128
EK 13: Çok Ölçütlü Karar Modeli LINGO çözümü	129



TABLÖLER LİSTESİ

Tablo 1.1. Yazılım projelerini etkileyen temel faktörlerin çıktılar üzerindeki etkileri	9
Tablo 1.2. Literatürdeki bazı çalışmaların amaçları	11
Tablo 3.1. “Gerçekleştirim” süreci aşamaları ve ilgili ekipler	34
Tablo 3.2. Geliştirme ekibinin tecrübe seviyesine göre çalışan dağılımı	35
Tablo 3.3. Test ekibinin tecrübe seviyesine göre çalışan dağılımı	35
Tablo 3.4. Kurulum ekibinin tecrübe seviyesine göre çalışan dağılımı	36
Tablo 3.5. Proje türüne göre süreçlerde çalışabilecek tecrübe seviyesi dağılımı	37
Tablo 3.6. Yalın proje türünde çalışanların tecrübe seviyesine göre saat bazında çalışma süresi dağılımı	37
Tablo 3.7. Orta proje türünde çalışanların tecrübe seviyesine göre saat bazında çalışma süresi dağılımı	38
Tablo 3.8. Kompleks proje türünde çalışanların tecrübe seviyesine göre saat bazında çalışma süresi dağılımı	38
Tablo 3.9. 2020 yılında başlayan projelerin bedelleri ve geliş tarihleri	39
Tablo 3.10. Çalışanları tecrübe seviyesine göre saatlik ücretleri	40
Tablo 4.1. Mevcut sistemdeki ortalama tamamlanan toplam proje sayısı	59
Tablo 4.2. Projelerin ortalama toplam tamamlanma süresi	59
Tablo 4.3. Çalışanların Kullanım Oranları	60
Tablo 4.4. Süreç Kuyruklarında Bekleyen Ortalama Proje Sayısı	60
Tablo 4.5. Süreç Kuyruklarındaki Ortalama Bekleme Süresi	61
Tablo 4.6. Ortalama Toplam Çalışan Maliyeti (TL)	62
Tablo 4.7. Faktörler ve Faktör Düzeyleri	69
Tablo 4.8. Ortalama tamamlanan toplam proje sayısı için faktöriyel deney tasarımı	70
Tablo 4.9. Projelerin ortalama toplam tamamlama süresi için faktöriyel deney tasarımı	71
Tablo 4.10. Ortalama çalışan maliyeti için faktöriyel deney tasarımı	71
Tablo 4.11. Ortalama çalışan maliyeti için faktöriyel deney tasarımı	72
Tablo 4.12. Ortalama tamamlanan toplam proje sayısının x_1 , x_2 , x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi	74

Tablo 4.13. Ortalama tamamlanan toplam proje sayısının $x_1, x_2, x_3, x_4, x_1*x_3, x_1*x_4, x_3*x_4, x_1*x_3*x_4$ etkileşimlerine göre regresyon analizi	75
Tablo 4.14. Ortalama tamamlanan toplam proje sayısı meta-modelinin geçerliliği	76
Tablo 4.15. Projelerin ortalama toplam tamamlama süresinin x_1, x_2, x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi	77
Tablo 4.16. Projelerin ortalama toplam tamamlama süresinin $x_1, x_2, x_3, x_4, x_1*x_3, x_1*x_4, x_3*x_4, x_1*x_3*x_4$ etkileşimlerine göre regresyon analizi	78
Tablo 4.17. Projelerin ortalama toplam tamamlama süresi meta-modelinin geçerliliği	79
Tablo 4.18. Ortalama toplam çalışan maliyetinin x_1, x_2, x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi	80
Tablo 4.19. Ortalama toplam çalışan maliyetinin $x_1, x_2, x_3, x_4, x_1*x_2, x_1*x_3, x_2*x_3, x_2*x_4, x_1*x_2*x_3, x_1*x_2*x_4, x_2*x_3*x_4, x_1*x_2*x_3*x_4$ etkileşimlerine göre regresyon analizi	81
Tablo 4.20. Ortalama toplam çalışan maliyeti meta-modelinin geçerliliği	82
Tablo 4.21. Ortalama toplam kârın x_1, x_2, x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi	83
Tablo 4.22. Ortalama toplam kârın $x_1, x_2, x_3, x_4, x_1*x_3, x_1*x_4, x_3*x_4, x_1*x_3*x_4$ etkileşimlerine göre regresyon analizi	84
Tablo 4.23. Ortalama toplam kâr meta-modelinin geçerliliği	85
Tablo 4.24. Ortalama tamamlanan toplam proje sayısı karşılaştırma tablosu	86
Tablo 4.25. Projelerin ortalama toplam tamamlama süresi karşılaştırma tablosu	88
Tablo 4.26. Ortalama toplam çalışan maliyeti karşılaştırma tablosu	89
Tablo 4.27. Ortalama toplam kârlılık karşılaştırma tablosu	90
Tablo 4.28. Çok ölçütlü karar modelinin optimum değerlerine göre benzetim modeli çıktıları	93
Tablo 4.29. Tek ölçütlü ve çok ölçütlü karar modellerinin benzetim çıktılarının karşılaştırılması	93

ŞEKİLLER LİSTESİ

Şekil 1.1.Yazılım geliştirme yaşam döngüsü	6
Şekil 1.2. Sistem ile çalışma yöntemleri	13
Şekil 1.3. Benzetim yöntemi zaman-maliyet grafiği	14
Şekil 3.1. Kaynakların modelde gösterilmesi	43
Şekil 3.2. Ekiplerin modelde gösterilmesi	44
Şekil 3.3. Kompleks, Orta ve Yalın Proje Türlerinin Gelişleri	45
Şekil 3.4. Kompleks, Orta ve Yalın Proje Türlerinin Geliş Zamanları	45
Şekil 3.5. Kompleks proje atama modülü	46
Şekil 3.6. Orta proje atama modülü	46
Şekil 3.7. Yalın proje atama modülü	47
Şekil 3.8. Kodlama istasyonundan Test istasyonuna geçiş süreci	47
Şekil 3.9. Kodlama istasyonundan Test istasyonuna geçiş süreci	48
Şekil 3.10. Test istasyonundan Yeniden Çalışma 1 istasyonuna geçiş süreci	48
Şekil 3.11. Yeniden Çalışma 1 istasyonundan Entegrasyon istasyonuna geçiş süreci	49
Şekil 3.12. Entegrasyon istasyonundan Doğrulama istasyonuna geçiş süreci	50
Şekil 3.13. Doğrulama istasyonundan Yeniden Çalışma 2 istasyonuna geçiş süreci	50
Şekil 3.14. Yeniden Çalışma 2 istasyonundan Kurulum istasyonuna geçiş süreci	51
Şekil 3.15. Kurulum istasyonundan Süreçlerin Tamamlanması İstasyonuna geçiş süreci	52
Şekil 3.16. Süreçlerin tamamlanması ve atama modülleri	52
Şekil 3.17. Performans Ölçütlerine göre İstatistik modülünde tutulan çıktılar	53
Şekil 3.18. Çalıştırma modülü	53
Şekil 4.1. Mevcut sistemde ortalama tamamlanan toplam proje sayısına göre güven aralığı	63
Şekil 4.2. Mevcut sistemde projelerin ortalama toplam tamamlanma süresine göre güven aralığı	64
Şekil 4.3. Mevcut sistemde ortalama toplam çalışan maliyetine göre güven aralığı	65
Şekil 4.4. Mevcut sistemde ortalama toplam kâra göre güven aralığı	65

SİMGELER VE KISALTMALAR LİSTESİ

KKÇP	kaynak kısıtlı çizelgeleme problemi
PMI	Project Management Institute
PMBOK	Project Management Body of Knowledge
TÇM	toplam çalışma maliyeti
TL	Türk Lirası
TK	toplam kârlılık
TPS	toplam proje sayısı
TTS	toplam tamamlanma süresi
YÇPÇ	yazılım projesi çizelgeleme problemi



1. GİRİŞ

Günümüzde bilimsel, teknolojik, ekonomik gelişmeler tüm dünyada hızla değişmekte ve yayılmaktadır. Dünya genelinde artan rekabet ortamında ülkelerin, bireylerin, kurum ve kuruluşların güçlü olabilmesi için bu gelişim ve değişimleri takip etmek, onlara uyum sağlamak, yeniliklere açık olmak zorunlu hale gelmiştir. Birey, toplum, kurum ve kuruluşların kendilerini geliştirebilmeleri, yeni dünyada rekabetçi olabilmeleri, mevcut kaynaklarını etkili ve verimli kullanabilmeleri için güncel bilimsel bilgiye ulaşma, bilgiyi toplama, kullanma, analiz etme, başka alanlara transfer etme, elde edilen bilgileri mevcut bilgilerle ilişkilendirme becerilerinin olması oldukça önem kazanmıştır. Bu becerilerle birlikte ortaya çıkan yeni fikirler; bireylerin, kurum ve kuruluşların kendilerinin veya başkalarının ihtiyaçlarını karşılamalarına katkı sağlayan projelerin oluşmasını sağlamıştır.

Proje Türk Dil Kurumu Büyük Türkçe Sözlüğü'nde, "Değişik alanlarda önceden plan ve programa alınmış, maliyeti hesaplanmış, kurum ve kuruluşların yönetim organları tarafından onaylanmış, kısa ve uzun vadeye bağlanarak özel kurum veya devlet adına gerçekleştirilmesi kabul edilmiş bilimsel çalışma tasarısı." olarak tanımlanmaktadır [1]. Proje genel olarak, birbiriyle bağlantılı belirli amaçlara ulaşmak, sınırlı kaynaklarla ürün ya da hizmet ortaya koymak için planlanan ve tasarlanan, bireysel ya da ekip olarak yapılan faaliyetler bütünü olup dinamik süreç içerisinde üretilen yeni ve orijinal fikirlerin kullanılmasıyla eyleme geçilmesini sağlayan bir çalışma olarak tanımlanabilir. Bir projenin gerçekleştirilebilmesi için proje hedeflerinin tanımlanması, süresinin belirlenmesi, projenin sürdürülebilir ve iyi yönetilebilmesi amacıyla nerede, nasıl ve kim ya da kimler tarafından yapılacağının planlanması gerekmektedir.

Proje yönetimi, uygulanması düşünülen bir projenin sınırlı süresini, bütçesini ve kısıtlı kaynaklarını etkili ve verimli bir şekilde kullanarak, belirlenen hedeflerine ulaşması ve tamamlanabilmesi için projenin planlanması, organize edilmesi, ihtiyaçların tedarik edilmesi süreçlerine yönelik izlenen yönetim sistemi olarak tanımlanabilir. Literatür incelendiğinde proje yönetimi ile ilgili farklı tanımlamalara da rastlanmaktadır. Qasimov [2], proje yönetimini belirlenen bir amaca ulaşmak üzere bir araya getirilen maddi ve beşeri kaynakların faaliyetlerini planlama, yürütme, düzenleme ve denetleme fonksiyonları topluluğu olarak

tanımlamaktadır. Hazır vd. [3], ise proje yönetimini, proje kapsamındaki işleri, süreçleri sistematik bir şekilde planlamak ve yürütmek için yapılan çalışmalar bütünü şeklinde tanımlamaktadır. Proje yönetimi projenin kavramsal tasarımından sonlandırılmasına kadar olan zaman diliminde süreçlerin tasarımını, organize edilmesini, kontrolünü, koordinasyonunu ve yönlendirilmesini kapsamaktadır. Proje yönetimi, projenin sadece yürütülmesi süreci olmayıp projeye ilgili tüm süreçleri kapsamaktadır. PMI'n (Project Management Institute) resmi yayını olan PMBOK'a (Project Management Body of Knowledge) göre proje yönetimi süreçleri, her biri bir veya daha fazla süreçten oluşan beş grup halinde organize edilebilir: Süreçleri başlatma, planlama süreçleri, süreçleri yürütmek, kontrol süreçleri, kapanış süreçleri [4].

Projenin başından sonunda kadar belirli bilgi alanlarındaki bilgi beceri, yöntem, araç ve tekniklerden yararlanılmaktadır. Alanlar kendine özgü işlevleri olmakla birlikte, birbirlerini doğrudan veya dolaylı bir şekilde etkilemektedir. Alanların proje yönetimine yönelik işlevi bilindiğinde, diğer alanlara ve dolayısıyla proje yönetimine yönelik etkileri de daha net bir şekilde kestirilebilir [5].

1.1. Yazılım Projeleri ve Yazılım Projelerinde Proje Yönetimi

Günümüzde iş hayatının yanında sosyal hayatta da “yazılım” kavramıyla sıklıkla karşılaşılmaktadır. Yazılım kelimesinin Türk Dil Kurumu Büyük Türkçe Sözlüğündeki karşılığı: “Bir bilgisayarda donanımaya hayat veren ve bilgi işlemde kullanılan programlar, yordamlar, programlama dilleri ve belgelerlerin tümü.” olarak tanımlanmıştır [1]. Yazılım, aynı zamanda organizasyonlardaki iş süreçlerini kısaltan, kişi-kişi, kişi-makine ve makine-makine arasındaki iletişimi sağlayan ve verimliliği artıran çeşitli görevler için hazırlanmış programlardır. Gencer ve Kayaca [6], yazılımı, günlük eğlencemizden en karmaşık iş süreçlerine kadar hayatımızın her alanında yerini almış ve etkileri hissedilen bir olgu olarak tanımlamaktadır. Organizasyonlar tarafından ihtiyaç ve rekabet analizi yapılarak geliştirilen yazılımlar, bireysel ya da kurumsal kullanıcıların, kârlılığının yükselmesini ve verimliliğinin artmasını sağlamaktadır. Yazılım geliştirmenin kökleri 1960'lara dayanmaktadır ve o zamandan beri yazılım modern toplumun önemli bir parçası haline gelmiştir. Yazılım

yardımla, bir rapor yazmak, bir köprü kurmak, uzay mekiğini kontrol etmek, internette sosyal ağlar kurmak mümkündür ve genel ekonomide çok önemli bir rol oynamaktadır [7].

Brooks [8], 1986 yılında yaptığı, yazılımın temel parçalarını ele alan çalışmasında yazılımın temel özelliklerini aşağıda kısaca açıklanan, karmaşıklık, uygunluk, değiştirilebilirlik ve görünmezlik olmak üzere 4 şekilde nitelendirmiştir:

- **Karmaşıklık:** Yazılım varlıkları boyutları açısından üretilen birçok sistemin varlıklarından daha karmaşıktır. Yazılımın birbirini aynı şekilde tekrar etmeyen öğelerden oluşması, yazılım sisteminin büyütülmesindeki anlamın öğelerin sadece sayı olarak artışı değil aynı zamanda farklı öğelerin sayısındaki artış olması ve öğelerin birbiriyle doğrusal olmayan bir etkileşimde olması yazılımın temel yapısının karmaşık olmasına sebep olur.
- **Uygunluk:** Bir yazılım üretilirken yazılımın son kullanıcıdaki ya da müşterideki mevcut yazılım ya da yazılımlarla uyumlu olması gerekmektedir. Yazılımlar farklı kişi ya da ekipler tarafından farklı ara yüzlerle ya da farklı yazılım dilleriyle üretildiği için bir araya geldiklerinde çalışmama durumları ortaya çıkabilir. O nedenle üretimden önce mevcut yazılımlarla uyum sağlayabilecek bir tasarımın yapılması, çalışmaların bu tasarıma göre yürütülmesi oldukça önemlidir.
- **Değiştirilebilirlik:** Yazılım ürünü; uygulamalar, kullanıcılar, yasalar ve makine araçlarından oluşan bir kültürel matris içine yerleştirilmiştir. Bunların hepsi sürekli olarak değişir ve değişiklikleri, kaçınılmaz bir şekilde yazılım ürününü değiştirmeye zorlar. Başarılı bir yazılımın sürdürülebilir olması için, yeni ihtiyaçlar ortaya çıktığında ya da teknolojinin gelişmesiyle donanımlar değiştiğinde mevcut yazılımların çalışmasının devam etmesi gerekmektedir. En baştan yazılım üretmek değişim yapmaktan daha maliyetli ve zaman alıcıdır. Bu nedenlerle yazılımın değiştirilebilir olması önemli bir özelliktir.
- **Görünmezlik:** Grafiklerle, diyagramlarla görsel hale getirilmeye çalışılsa da yazılım soyut bir varlıktır. Görselleştirilemediği için zihinde canlanması ve herkes için aynı varlığı ifade etmesi zordur, bu da zihinler arasındaki iletişimin zorlanmasına sebep olur.

Yukarıda açıklamaları yapılan yazılımın temel özellikleri, tüm yazılımlarda bulunan özellikleri kapsamaktadır. Yazılımın temel özelliklerinin yanı sıra, aynı ihtiyaçlarla ilgili yazılımların birbirleriyle kıyaslanması ve daha iyi olanın seçilmek istenmesi durumunda, yazılımın başka özelliklerinin de incelenmesi gerekmektedir. İyi bir yazılım ürünü, işlevsellik ve performans yönünden müşteri beklentisine uygun olmasının yanında güvenilir, anlaşılabilir, izlenebilir, kullanıcı tarafından kabul edilebilir, geliştirilebilir, verimli, doğrulanabilir, tutarlı ve bakımı yapılabilir (ihtiyaçlara göre değişebilir) olmalıdır.

Yazılımın temel özellikleri bir yazılımın oluşturulmasının her aşamasında göz önünde bulundurulmalıdır. Organizasyonlar tarafından geliştirilen yazılım türleri; masaüstü yazılımlar, mobil yazılımlar ve web yazılımlar olarak üç tür olarak incelenmektedir. Bu farklı tür yazılımların üretilmesi, bireysel ya da toplu kullanıma uygun hale getirilmesi, yazılım üreten organizasyonların mühendislik ve proje yönetimi süreçlerini etkili ve verimli kullanmalarıyla gerçekleşmektedir. Bu süreçlerin metodoloji seçiminde, üretilecek yazılımın büyüklüğü, karmaşıklığı ve kullanım maksadı gibi karakteristik özellikleri, yazılımı üretecek organizasyon ve proje ekibinin yapısı ve kullanılacak kaynaklar önem arz etmektedir [6]. Bireysel ya da toplu kullanım ihtiyaçlarına uygun yazılımların üretilmesi için üretilecek yazılımın önceden planlanması ve projelendirilmesi gerekmektedir. İhtiyaç ve fikirlerle başlayan yazılımları hayata geçirmek ve bir yazılım ortaya çıkarmak için yapılan projeler yazılım projeleri olarak özelleşmektedir. Yazılım projeleri, ekipman, uygulamalar, hizmetler ve bir organizasyon içinde operasyon, yönetim, analiz ve karar verme işlevlerini desteklemek için bilgi sağlayan temel teknolojiler geniş bir kapsamda uygulanmaktadır [9].

Tüm projelerde olduğu gibi yazılım projelerinde de projenin etkili yönetilmesi, projede uygun ekiplerin yer alması, kaliteli, maliyet etkin çözümlerin ortaya konması oldukça önemlidir. İhtiyaçlara uygun bir şekilde, zamanında, bütçeye uygun yazılımların üretilebilmesi için yazılım geliştirme süreçleri (yazılımları geliştirmek ve idame ettirmek için kullanılan yöntemler, dönüşümler) iyi yönetilmelidir. Yazılım geliştirme süreci kapsamında, süreç boyunca çalışmanın sürdürülmesi ve tamamlanması için proje yönetimi zorunludur. Proje yöneticisinin, belirlenen süre içinde müşteri beklentilerini karşılayabilmesi, bu beklentilerin ne ile ve nasıl karşılanacağını planlamasını yapması ve meydana gelen gelişmeden emin olmak için hem geliştirme ekibi hem de iş paydaşları ile birlikte çalışması gerekmektedir [9]. Yazılım

projelerinin bütçeye uygun olarak zamanında ve kaliteli bir şekilde teslim edilebilmesi için yazılım projelerinin değişen şartlara ve ihtiyaçlara kolay adapte olabilmesi gerekir. Yazılım projelerinin genellikle uzun süreli projeler olması bu süreçte teknolojinin gelişmesiyle birlikte yazılım projelerinde ihtiyaçların farklılaşması, maliyetlerin değişmesi durumları ortaya çıkabilmektedir. Bu sebeple, yazılım projelerinin anlık olarak takip edilmesi gerekmektedir. Sanchez ve Terlizzi [10], bir kuruluşun yazılım geliştirme projelerinin maliyet ve süresini optimize edebilecek proje yönetimi uygulamalarını belirlemeyi amaçladıkları çalışmalarında, yazılım geliştirme proje yönetimi için maliyet ve sürenin önemli bir ölçü olduğunu; projelerin, kararlaştırılan bir kapsam altında maliyet ve zamanı dinamik olarak ele alması gerektiğini belirtmişlerdir.

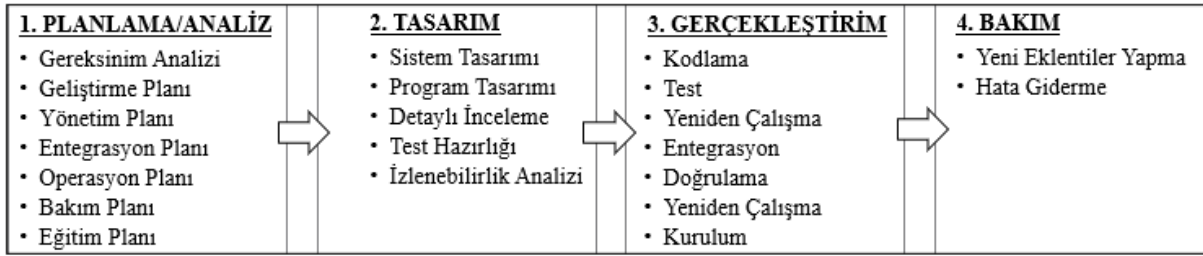
1.2. Yazılım Geliştirme Yaşam Döngüsü ve Süreç Modelleri

Organizasyonların büyümesi ve teknolojinin gelişmesiyle birlikte yazılıma ve yazılım projelerine olan ihtiyaç gittikçe artmaktadır. Ayrıca, birçok yazılım projesi ortaya çıkarılmış ve çıkarılıyor olmasına rağmen tamamlanamayan, geciken, bütçesini aşan birçok yazılım projesi olduğu da bilinmektedir. Erdem ve Younis [9], çalışmalarında, “Standish Group International” tarafından 2004 yılında yapılmış bir çalışmanın sonucuna yer vermişlerdir. Bu çalışmaya göre, yazılım projelerinin %53’ünün gecikmiş ya da bütçesini aşmış, %18’inin tamamlanamamış ve projelerin yalnızca %29’unun zamanında ve ayrılan bütçesine uygun tamamlandığı ortaya konmuştur. Yazılım geliştiricilerin, yazılıma olan yoğun talebi, yazılım projelerinin bütçe ve süresini aşmadan karşılayabilmeleri; yazılım süreçlerini ihtiyaçlara uygun olarak tamamlayabilmeleri için yazılım geliştirmeyi sistematik hale getirmeyi hedefleyen çeşitli süreç modelleri ortaya çıkmıştır. Bu modellerin temel hedefi; projenin başarıyla tamamlanması için yazılım geliştirme yaşam döngüsü boyunca izlenmesi önerilen mühendislik süreçlerinin tanımlanmasıdır.

Yazılım geliştirme yaşam döngüsü, bir yazılım ürünün ihtiyacının ortaya çıkmasından, kullanımdan kalkmasına kadar geçirdiği tüm aşamalar olarak tanımlanmaktadır. Yazılım geliştirme yaşam döngüsü tek yönlü değildir. Döngü içindeki herhangi bir aşamaya geri dönmek ve tekrar ilerlemek mümkündür. Yazılım geliştirme yaşam döngüsünün süreçleri

temel olarak benzer süreçleri içerse de farklı kaynaklarda farklı başlıklar altında toplanmıştır. Örneğin; Mishra ve Dubey [11], çalışmalarında yazılım geliştirme yaşam döngüsünün aşamalarını şu şekilde belirtmişlerdir: Problemi anlama (gereksinimleri toplayarak), çözüm için bir plana karar verme (tasarlamak), planlanan çözümü kodlama, asıl programı test etme, ürünün yerleştirilmesi ve bakımı. Ersoy [12] ise, yazılım geliştirme yaşam döngüsünü 6 adımda sıralamıştır: İhtiyaçlar, tanımlama, tasarım, gerçekleştirme, uygulama, bakım. Sang vd. [13], tarafından yapılan çalışmada ise yazılım geliştirme yaşam döngüsü şu adımlardan oluşmuştur: Planlama, gereksinimler, tasarım, uygulama, entegrasyon, doğrulama.

İncelenen çalışmalarda yazılım geliştirme yaşam döngüsü aşamaları için farklı tanımlamalar kullanmış olsa da aslında tüm çalışmalar genel olarak 4 temel aşamayı kapsamaktadır. Bu çalışmada bu aşamalar; planlama/analiz, tasarım, gerçekleştirim, bakım şeklinde tanımlanmıştır. İlgili aşamalarda gerçekleştirilen süreçler Şekil 1.1.'de gösterilmiştir:



Şekil 1.1.Yazılım geliştirme yaşam döngüsü

Yazılım geliştirme yaşam döngüsünde yer alan 4 temel aşamanın kapsamı aşağıda kısaca açıklanmıştır:

- 1. Planlama/Analiz:** Ortaya çıkarılacak yazılım için ihtiyaçların tanımlandığı, kaynak gereksinimlerinin belirlendiği, fizibilite çalışmasının yapıldığı, mevcut sistemdeki problemlerin tespit edilerek çözümlenmesi konusunda analizlerin yapıldığı aşamadır. Bu aşamada yapılan doğru yatırım ve çalışmalar, projenin sonraki aşamalarında meydana gelebilecek hataların ortaya çıkmasını engellemekte, böylece süre ve maliyet açısından kayıp yaşanmamasını sağlamaktadır.
- 2. Tasarım:** Planlama ve analiz aşamalarıyla belirlenen gereksinim ve çözümleri karşılayabilecek yazılım temel yapısının oluşturulduğu aşamadır.

3. Gerçekleştirim: Kodlama, test, yeniden çalışma, entegrasyon, doğrulama, yeniden çalışma, kurulum süreçlerinin bulunduğu aşamadır. Bu süreçler bazı kaynaklarda farklı şekillerde sıralanabilmektedir. Örneğin, Hanne ve Nickel [14] yaptıkları çalışmada bu sıralamayı; kodlama, denetleme, yeniden çalışma, test ve yeniden çalışma süreçleri olarak sıralamışlardır. Bu çalışmada, yazılım geliştirme yaşam döngüsünün bu aşaması üzerinde yapılan iyileştirme çalışmalarına yer verilmiştir. Bu süreçlerde yapılan işlemler aşağıda açıklanmıştır:

- Kodlama: Yazılım geliştirme ekipleri tarafından yapılan, yazılımın temel parçası olan kod öğelerinin bilgisayar yardımıyla üretilmesi işidir. Öğelerin boyutu ve karmaşıklığı birbirlerinden farklıdır [14].
- Test: Kodlama sürecinden sonra yazılım test ekibi tarafından, yazılan kodlardaki hataları tespit etmek amacıyla yapılan kontrol etme işidir.
- Yeniden Çalışma: Test sürecinden sonra bulunan hataların düzeltilmesi için kodlama ekibinin yeniden çalışması işidir.
- Entegrasyon: Yeniden çalışma sürecinde kodlardaki hatalar düzeltildikten sonra ortaya çıkan yazılımın müşterideki mevcut yazılımlarla entegre ve senkronize bir şekilde çalışmasını sağlamak için yapılan sistemlerin konuşturulması, birleştirilmesi işidir. Kurulum ekibi tarafından yapılmaktadır.
- Doğrulama: Entegrasyon süreci sonrasında yazılım test ekibi tarafından yazılımın çıktılarının, projenin başında tanımlanan koşul ve ihtiyaçları karşılayıp karşılamadığının test edilmesi işidir. Test ekibi tarafından yapılmaktadır.
- Yeniden Çalışma: Doğrulama sürecinde çıkan hataların üzerinde yazılım geliştirme ekibinin yeniden çalışması işidir. Bu süreçte ortaya çıkan hatalar daha az olacağı için bir önceki yeniden çalışma süresinden daha kısa süren bir aşamadır.
- Kurulum: Yeniden çalışma yapıldıktan sonra yazılımın müşteriye kurulumunun yapılması işidir. Kurulum ekibi tarafından yapılmaktadır.

4. Bakım: Değişen teknoloji ve ihtiyaçlara yönelik olarak kurulumu yapılan yazılımın hatalarını giderme, yeni eklentiler yapma aşamasıdır. Yazılımın kullanım ömrü doluncaya kadar süren bir aşamadır.

Yazılım geliştirme yaşam döngüsünün temel adımlarında yer alan süreçlerin takip edeceği düzen ve uygulama yöntemi süreç modelleriyle tanımlanır. Yazılım süreç modellerinin ortaya çıkmasında ilgili dönemin donanım ve yazılım teknolojileri ile sektör ihtiyaçları önemli rol oynamaktadır. Süreç modellerinin uygulanmasıyla daha kaliteli yazılımlar ortaya çıkmakta ve proje yönetimi daha kontrollü bir hale gelmektedir. Süreç modellerinde en sık karşılaşılan modeller şunlardır: Kodla ve Düzelt Modeli (Build & Fix Model), Gelişigüzel Model, Şelale Modeli (Waterfall Model), Sarmal Model (Spiral Model), Hızlı Uygulama Geliştirme Modeli (RAD-Rapid Application Development), Artırımlı Geliştirme Modeli (Incremental Development Model), V-Modeli (V-Shaped Model).

Yazılım geliştirme yaşam döngüsünün çeşitli modelleri olup uygulanacak projeye bağlı olarak her birinin kendine göre avantajları ve dezavantajları bulunmaktadır. Organizasyonlarda mevcut koşullara bağlı olarak kullanılan birçok süreç modeli bulunmaktadır. Yazılım endüstrisinde, tüm bu farklı yazılım geliştirme süreç modelleri tek başlarına kullanılabilse de genellikle birtakım farklılaştırmalarla bu süreçlerin karması kullanılmaktadır. Yazılım endüstrisinde doğru yaşam döngüsü modelinin seçilmesi son derece önemlidir, çünkü yazılımın son teslim tarihine kadar teslim edilmesi ve aynı zamanda istenen kalitede olması gerekmektedir.

1.3. Yazılım Projelerini Etkileyen Temel Faktörler

Yazılım projelerinin girdilerini ve çıktılarını etkileyen birtakım faktörler bulunmaktadır. Bu faktörler proje süresini, maliyetini ve kalitesini etkilemektedir. Rasch ve Tosi (1992), Nidumolu (1995), yazılım projelerini en çok etkileyen temel faktörlerin insan faktörü, teknik faktörler ve yönetim faktörleri olduğunu belirtmişlerdir [15], [16]. Bununla birlikte başka çalışmalar da başka faktörlerin önemli olduğunu vurgulamışlardır. Örneğin, Banker ve Kemerer (1989), proje ekibinin büyüklüğü, ürünün karmaşıklığı, gereksinimlerin değişkenliği, program kısıtlamaları ve kullanılan yazılım araçları, takımlar arası işlevsellik ve takım bağımsızlığı, öğrenme, eğitim ve iletişim genel giderleri faktörlerinin yazılım projelerini etkilediğini ortaya koymuşlardır [17]. Chatzoglou, Theriou ve Aggelides (2007) ise yaptıkları çalışmada yazılım projelerini etkileyen faktörleri araştırmak için proje yöneticilerine çeşitli

sorulardan oluşan bir anket uygulamışlardır. Anket sonuçlarının değerlendirilmesi sonucunda, yazılım projelerini etkileyen temel faktörleri ve zaman, efor, maliyet çıktıları üzerindeki etkilerinin seviyesini Tablo 1.1.'deki gibi göstermişlerdir [18].

Tablo 1.1. Yazılım projelerini etkileyen temel faktörlerin çıktıları üzerindeki etkileri

Faktör	Çıktılara Etkisi
Takım üyelerinin deneyimi	Yüksek
Takım üyelerinin bilgisi	Yüksek
Takım üyelerinin bağlılığı ve sürekliliği	Yüksek
Takım büyüklüğü	Orta
Kullanıcıların katılımı	Orta
Kullanıcıların takım üyeleri ile iletişimi	Yüksek
Kullanıcıların motivasyonu	Orta
Kullanıcıların yetenekleri	Düşük
Kullanıcılar arasındaki fikir ayrılıkları	Yüksek
Geliştirme sırasında kullanıcıyı sistemi anlaması	Yüksek
Müşterinin projeye bağlılığı ve sürekliliği	Orta
Müşterinin taahhüdünü geliştirme	Yüksek
Proje yönetimi	Yüksek
Araç ve tekniklere ulaşabilme	Yüksek
Kaynaklara ulaşabilme	Yüksek
Kaynakları koordine edebilme	Yüksek

Tablo 1.1.'de yer alan yazılım projelerini etkileyen temel faktörler incelendiğinde, projede çalışan ekibin projeye uygun tecrübe ve bilgide olmasının, ekibin uygun araç ve tekniklerle çalışmasının, doğru bir proje yönetimiyle koordine edilmesinin proje için öneminin çok yüksek olduğu görülmektedir. Ayrıca, proje ekibinin sürekliliğinin de yazılım projelerini etkileyen temel faktörlerden olduğu ortaya çıkmaktadır. Proje süresince proje ekibinde değişiklikler yaşanması halinde, önceki çalışan kişilerin yazılım üzerinde yaptıkları çalışmaların yeni kişiler tarafından anlaşılması ve öğrenilmesi konuları ayrıca bir süre

gerektirmektedir. Bu da işlerin zamanında tamamlanamamasına sebep olabilmektedir. O nedenle yazılım projelerinde proje ekiplerinde sık değişiklik yapılması önerilmemektedir [8]. Yazılımın temel özelliği olan karmaşıklık özelliği de ürün kusurlarına, maliyet ve süre aşımına, proje ekibi arasındaki iletişimin zorlaşmasına neden olabilmektedir. Üretilen yazılımın genelini kontrol etmek ve kavramsal bütünlüğü kurgulamak, tamamlanmamış işleri saptamak hem teknik açıdan hem de yönetim açısından zor olduğu için yazılım projelerinde takımı doğru belirlemek ve takımı doğru koordine etmek oldukça önemlidir.

1.4. Yazılım Geliştirme Projelerinde Kullanılan Çözüm Yöntemleri

Yazılımın temel özellikleri ve yazılım projelerini etkileyen temel faktörler dikkate alındığında, yazılım projeleri yaşam döngüsü aşamalarının projelere uygun bir şekilde yönetilmesinin, proje verimliliği açısından önem taşıdığı görülmektedir. Büyük ölçekli yazılım projelerinin tasarımı ve geliştirilmesi karmaşık bir süreç olarak tanımlanmaktadır ve yazılım projeleri genellikle maliyet ve zaman aşımalarıyla birlikte düşük kalite gibi sorunlarla da karşı karşıya kalmaktadır. Son yıllarda yazılım geliştirme projelerinin yönetimi, iyileştirme ve çözüm arayışının temel taşı olarak kabul edilmiştir [19]. Yazılım yaşam döngüsünün Planlama/Analiz, Tasarım, Gerçekleştirim ve Bakım süreçlerini iyileştirmeyi hedefleyen birçok çalışmanın ortak amaçları, temelinde insan kaynağı olan bu özelleşmiş proje türünün süre, maliyet ve kalite açısından en iyiye ulaşmasını sağlamaktır. Yazılım projelerini çeşitli yöntemlerle iyileştirmeyi hedefleyen bazı çalışmaların, projenin hangi çıktı ya da çıktılarını iyileştirmeyi amaçladığı aşağıdaki Tablo 1.2.'de “x” şeklinde işaretlenerek gösterilmiştir:

Tablo 1.2. Literatürdeki bazı çalışmaların amaçları

MAKALE	AMAÇ		
	Süre	Maliyet	Kusur Sayısı (Kalite)
Chang vd., 2008 [20].	x	x	
Di Penta vd., 2011 [21].	x		
Duggan vd., 2004 [22].	x		x
Hanne ve Nickel, 2005 [14].	x	x	x
Jin ve Yao, 2014 [23].	x	x	
Luna vd., 2014 [24].	x	x	
Rodríguez vd., 2011 [25].	x	x	
Shen vd., 2016 [26].	x	x	
Shen vd., 2018 [27].	x	x	
Wen ve Lin, 2008 [28].	x	x	
Xiao vd., 2012 [29].	x	x	
Alba ve Chicano [30]	x	x	

Tablo 1.2.'de görüldüğü üzere, yazılım projeleri süre, maliyet ve kalite çıktılarından sadece birini iyileştirmeye odaklanabildiği gibi birden fazla amaca ulaşmayı da hedefleyebilmektedir. Bu hedeflerin yerine getirilebilmesi için kaynakları yazılım projesine uygun bir şekilde tahsis etmek ve ekiplere görev atamak, proje maliyetini ve tamamlanma süresini etkileyen çok önemli aktivitelerdir. Böyle bir atamaya çözüm bulmak karmaşık olarak tanımlanmaktadır. Bu nedenle proje yöneticilerinin bu şekildeki bir tahsisatı gerçekleştirmek için uygun araçlarla desteklenmesi gerekmektedir. Önerilen yaklaşımlarda, kişisel deneyim, organizasyon yapısı, insan becerileri ve çalışma tutumu bilgisi, bir projeyi etkili bir şekilde planlamak için önem taşımaktadır. Proje yöneticisi, fizibilite ve hesaplanmış çözümlerin sağlamlığı konusunda en yetkili kişi olarak tanımlanmaktadır [21].

1.4.1. Çizelgeleme Yöntemi

Endüstri Mühendisliği'nin birçok alanında atama problemlerinde temel çözüm yaklaşımı olarak kullanılan çizelgeleme yöntemi, yazılım proje yönetimini desteklemek amacıyla projelere kaynak ataması yapılırken de en çok kullanılan yöntemlerden biridir. Yazılım Projesi Çizelgeleme Problemi (YPÇP) literatürde daha çok bilinen Kaynak Kısıtlı Çizelgeleme Probleminin (KKPÇ) özelleşmiş bir halidir ve NP-zor bir problemidir.

KKPÇ ile YPÇP'nin farkları aşağıdaki gibi sıralanabilir:

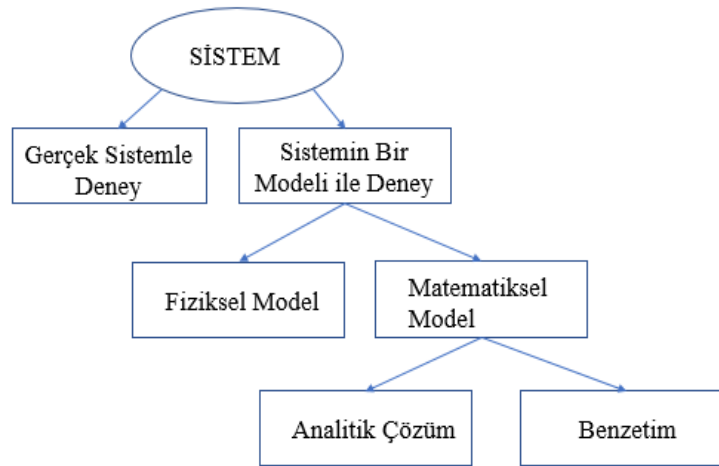
- YPÇP'de kaynaklar çalışanlar olup, bu çalışanların farklı uzmanlık alanları, farklı maaşları ve bir projeye ayırabilecekleri farklı zaman çizelgeleri mevcuttur.
- YPÇP'de proje süresi ile birlikte proje maliyeti de dikkate alınmalıdır.
- YPÇP'de sadece çalışanların uzmanlık alanları ve tecrübelerine göre yapabileceği işler farklılaşmaktadır.
- YPÇP'de çalışanların farklı yetenekleri, KKPÇ'deki farklı kaynak tipleri gibi ölçülebilir değildir.

YPÇP'de dikkate alınan kısıtlar aşağıdaki gibi sıralanabilir:

- Bir çalışan birden fazla projede çalışabilir.
- Proje süresi boyunca yeni görevler ortaya çıkabilmektedir [27].
- Çalışanların farklı uzmanlık alanları ve ilgili uzmanlık alanında farklı tecrübe düzeyleri bulunmaktadır.
- Çalışanların uygunlukları proje süresi içinde değişebilmektedir [27].
- Proje bitimi için bir teslim tarihini olması gerekmektedir [28].
- Görevler farklı zorlukta ve büyüklüktedir [14].
- Projedeki herhangi bir görevi gerçekleştiren çalışan/çalışanlar, o projenin test ekibinde yer alamamaktadır [14].
- Projede kodlama yapan çalışan/çalışanlar o projenin test aşamasında görev alamamaktadır.
- Çalışanların atanmasında fazla mesai ücret politikası uygulanabilmektedir.
- Projelerin erken tesliminde ödül, geç tesliminde ceza uygulanabilmektedir.

1.4.2. Benzetim Yöntemi

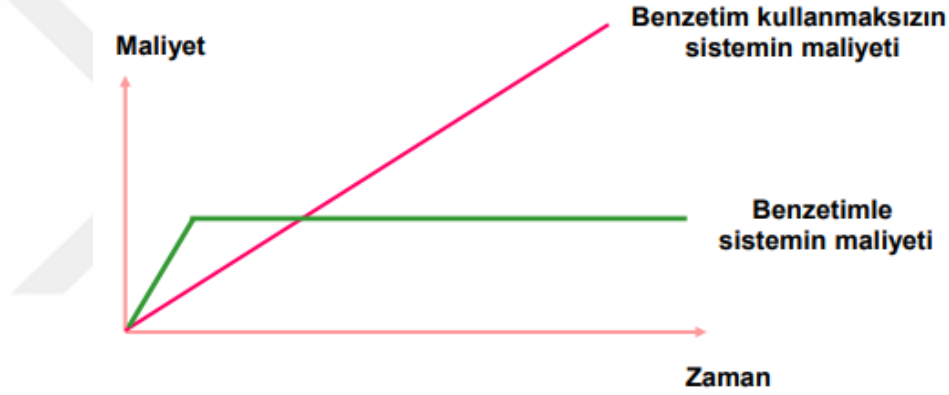
Yazılım projelerinde uygun kaynak ataması yapmak için uygulanan bir diğer yöntem de benzetim (simülasyon) yöntemidir. Benzetim, temel olarak bir sistemin benzeri anlamına gelmektedir. Bir sistemin benzeri olarak tanımlanabilen benzetim, gerçek bir sistemin olasılıklarını görebilmek, karmaşık problemlerin çözülmesini kolaylaştırabilmek amacıyla başvurulan analitik bir yaklaşım olarak gösterilmektedir. Problemlerin büyümesi ve karmaşık hale dönüşmesiyle birlikte gerçek sistemin yansıtılması ve daha kolay çözülebilmesi amacıyla analitik bir yaklaşım olarak kullanılmaktadır. Benzetim, gerçek bir ürün ya da sistem ortaya konmadan önce ürünün ya da sistemin davranışlarının gözlemlenmesi, gerçek hayatta oluşturulmadan önce işleyişi hakkında bilgi sahibi olunması, ürün ya da sistemin avantaj ya da dezavantajlarının değerlendirilebilmesi amacıyla ürün ya da sistemin önceden modellenmesi olarak da tanımlanabilir [31]. Gerçek ya da teorik sistemlerin üzerinde istatistiksel çalışmalar ve hedeflere uygun olarak ne-eğer (what-if) analizleri yapılabilmesi amacıyla benzetim modelleri kullanılmaktadır. Gerçek sistemin olmaması, gerçek sisteme erişimin kolay olmaması, gerçek sistemle deneyin tehlikeli olması veya mümkün olamaması, analitik çözüm tekniklerinin olmaması veya zor olması, sistemin çok yavaş veya çok hızlı olması, gerçek sistemde çalışmanın maliyetinin yüksek olması gibi durumlarda benzetim modelleri üzerinden çalışmak gerekli olmaktadır. Law ve Kelton [32], kitaplarında bir sistem üzerinde çalışma yöntemlerini Şekil 1.2.'deki gibi göstermiştir



Şekil 1.2. Sistem ile çalışma yöntemleri

Şekil 1.2.'de görüldüğü üzere benzetim yöntemi, sistemin bir modeli ile yapılan deneylerde kullanılmaktadır.

Ekonomik açıdan bakıldığında benzetim yönteminin, gerçek sistemler üzerinde yapılacak denemelere kıyasla daha ekonomik bir yöntem olduğu söylenebilmektedir. Benzetim yönteminde daha az zaman ve kaynak ihtiyacı olduğu için gerçek sistem üzerinde yapılacak denemelerden daha az maliyet ortaya çıkmaktadır. Sistem Benzetimi alanında yoğun çalışmaları bulunan Dengiz, 2017 yılında bu alanda verdiği dersin ders notlarında benzetimin ekonomik yönünü açıklamak için Şekil 1.3.'te yer alan zaman-maliyet grafiğini kullanmıştır [33].



Şekil 1.3. Benzetim yöntemi zaman-maliyet grafiği

Şekil 1.3. incelendiğinde sisteme uygulanan benzetim yönteminin ilk yatırım maliyetinin yüksek olduğu ancak uzun dönemde maliyete olan faydasının benzetim kullanılmamasına kıyasla daha fazla olduğu görülmektedir. İlk yatırım maliyetinin yüksek olmasının sebepleri, benzetim yöntemi kullanabilmek için gereken programlarının ya da benzetim dillerinin satın alınması ve bu programları kullanabilecek gerekli çalışanların istihdam edilecek olmasıdır. Bu ilk yatırım maliyeti zaman içinde dengeye gelecek ve benzetim kullanılmayan sistemlere kıyasla daha az maliyetli bir hale dönüşecektir. Benzetim yöntemiyle maliyetlerini azaltan birçok organizasyon bulunmaktadır. Örneğin, giysi, ev ürünleri, yiyecek gibi birçok alanda kullanılan kimyasalların üretimini yapan DuPont firması, tank araba operasyonlarının benzetim modellerini geliştirerek birkaç milyon dolar tasarruf ettiğini açıklamıştır [34]. Bunun dışında IBM, Tropicana, Hallmark Card gibi büyük şirketler de benzetim yöntemi ile tasarruf yaptıklarını açıklamışlardır [31].

Benzetim modeli oluşturulurken girdilerden, modeli değerlendirirken de çıktılarından yararlanılmaktadır. Benzetim modelinde girdilerde değişiklikler yapılarak çıktıların nasıl değişeceği analiz edilmektedir. Çıktıların analiz edebilmek, karmaşık problemleri çözebilmek için için yaklaşık sonuçlar veren, sezgisel yaklaşımlar içeren, sisteme uygun sayısal denemelerin yapılabilmesine imkân tanıyan, çalışma uzunluğunun belirlenip belirlenemediğine dayalı olarak 2 çeşit benzetim türü bulunmaktadır. Bu benzetim türleri aşağıda kısaca açıklanmıştır:

1. Bitişli Benzetim: Her çalışmanın uzunluğunu belirten, doğal bir “E” olayının olduğu simülasyondur. “E” olayı, herhangi bir çalıştırma yapılmadan önce belirtilir ve belirli bir çalıştırma için E'nin oluşma zamanı rastgele bir değişken olabilir. Bu benzetim türünde, bir benzetimin başlangıç koşulları genellikle istenen performans ölçümlerini etkilediğinden, bu koşullar gerçek sistem için olanları temsil etmelidir [35].
2. Bitişli Olmayan (Sürekli) Benzetim: Temel olarak, zamanla birlikte davranışları sürekli değişim gösteren sistemlerin benzetimidir. Bir çalışmanın uzunluğunu belirtmek için E doğal olayının olmadığı bir benzetimdir.

Bir sistem için benzetim modeli oluştururken öncelikle problem tanımı yapılır ve probleme uygun veriler toplanır. Toplanan veriler yardımıyla rassal değişkenlerin olasılık dağılımları analiz edilir ve böylece modele girilecek girdiler oluşturulur. Girdiler yardımıyla benzetim modeli kurulur ve model kurulduktan sonra modelin geçerliliği, doğruluğu test edilir. Modelin gerçek sistemi doğru yansıttığından emin olduktan sonra modelde yapılabilecek alternatif senaryolar değerlendirilir ve analiz edilecek performans ölçütlerine göre yapılacak deney sayısı, deneyin tekrar sayısı ve modeli çalıştırma süresi gibi bilgileri içeren deney tasarımı oluşturulur. Deney tasarımına uygun olarak senaryo denemeleri gerçekleştirilir ve senaryo denemelerinden çıkan çıktıların çıktı analizi yapılır. Çıktı analizine göre iyileştirilmesi hedeflenen performans ölçütlerinin güven aralıkları belirlenir ve performans ölçütünün en iyi olduğu alternatif sistemler belirlenir. Bu süreçte, belirlenen performans ölçütlerinin en iyi alternatif sisteminin ortaya konulabilmesi için izlenen bir yöntem de Benzetim Eniyilemesi (Simülasyon Optimizasyonu) yöntemidir.

Benzetim modeli, optimizasyon için bir yöntem sağlamamaktadır. Benzetim Eniyilemesi, sistem performansını optimize eden yaklaşım yöntemleriyle bir dizi parametre bulmak için bir prosedür tanımlamaktadır. Stokastik benzetimin eniyilemesi, gradyan tabanlı arama yöntemlerini, stokastik optimizasyonu, yanıt yüzey metodolojisini (Response Surface Methodology-RSM), meta-model uygulamalarını, meta-sezgisel yöntemleri ve istatistiksel yöntemleri içermektedir [36], [37]. Bu çalışmada, benzetim eniyilemesi yöntemi olarak meta-model uygulamaları yöntemi kullanılmıştır. Bir benzetim modeli, girdi değişkenlerinin seçilen bir alt kümesinin dikkate alındığı gerçek sistemin bir soyutlaması olarak bilinmektedir. Benzetimin yanıtı, bu alt kümenin bir fonksiyonu olarak tanımlanmaktadır. Bir meta-model, benzetim girdi değişkenlerinin alt kümesinin sistemi tanımlamak için kullanıldığı başka bir soyutlama anlamına gelmektedir [38]. Benzetim modelinin bir meta-modeli oluşturulmadan önce benzetim modelinin deney tasarımı yapılarak denenebilecek senaryolar test edilmektedir. Denemeler sonrası oluşturulan meta-modelin istatistiksel analizlerle birbirleriyle ilişkisinin anlamlı olduğu görülen karar değişkenlerine bağlı olarak bir regresyon denklemi ortaya çıkmaktadır. Ortaya çıkan regresyon denklemi, sistemin kısıtlarıyla birlikte bir matematiksel karar modeli oluşturur. Karar modelinin sistemin iyileştirilmesine yönelik amaçlarına uygun olarak çözülmesiyle benzetim eniyilemesi tamamlanmış olur. Dengiz vd. [39], bir çalışmalarında, bir elmas takım üretim hattını temsil eden benzetim modeli için regresyon meta-modeli oluşturarak benzetim eniyilemesi tabanlı bir karar destek sistemi sunmuşlardır. Böylece üretim hattında daha az maliyet, süre ve eforla optimum çalışma şartları oluşturmuşlardır.

Yazılım süreci benzetim modellemesi, yazılım geliştirmenin stratejik yönetiminden süreç iyileştirmelerini desteklemeye ve yazılım proje yönetimi eğitime kadar çeşitli konuları ele almak için giderek daha fazla kullanılmaktadır. Yazılım süreci benzetim uygulamalarının kapsamı, yaşam döngüsünün dar odaklı bölümlerinden geniş kurumsal etkilere sahip daha uzun vadeli ürün evrim modellerine kadar uzanmaktadır [40]. Yazılım geliştirme alanında çalışanlar, mevcut süreçleri incelemek ve iyileştirmek için yöntemler aramaktadır. Bu aramalarda başvuru benzetim yöntemi, süreçleri sanal bir ortamda temsil ettiği, görsel olarak izlenebilir bir yapı sunduğu için karar vericiler tarafından sıkça tercih edilmektedir. Karar vericiler ve uygulayıcılar gerçek sisteme geçmeden önce benzetim modeli üzerinden

yaptıkları çalışmalarla verimliliğin artmasını sağlayabilmektedirler. Yazılım geliştirme sürecinin iyileştirilmesi ve organizasyonların karşılaştığı karmaşık soruların üstesinden gelinebilmesi için benzetim yöntemi kullanılması hem akademisyenler hem de uygulayıcılar arasında daha popüler bir hale gelmektedir [41]. Benzetim modellemesi, yazılım ürün geliştirme stratejilerinin tanımlanması, süreç iyileştirme ve eğitimle ilgili karar verme gibi çeşitli konularda, yaşam döngüsünün kısa bir kısmından uzun vadeli ürün evrimine kadar organizasyon çapında etkileri olan bir zaman aralığında destek sunmaktadır [19]. Yazılım projelerini planlamak için bilgisayar destekli teknikler, oldukça rekabetçi yazılım endüstrisindeki yazılım geliştirme sürecinde çok önemli bir adımdır [24]. Bu çalışmada, yazılım geliştirme süreçlerinin benzetim modellemesi yapılarak benzetim eniyilemesi yöntemiyle belirlenen performans ölçütlerinde iyileştirme yapılması sağlanacaktır.

1.5. Araştırmanın Amacı ve Önemi

Bu araştırma, birden fazla yazılım geliştirme projesinin “Gerçekleştirim” süreçlerinde görev alacak çalışanların belirlenmesi, bu çalışanların projelerin türüne, amaçlarına ve kısıtlarına uygun bir şekilde projelere atanması için yazılım proje yöneticisine yardımcı olacak bir benzetim modeli ortaya koymayı ve bu benzetim modeline benzetim eniyilemesi yaparak, belirlenen performans ölçütlerinin hedefleri doğrultusunda mevcut sistemde iyileştirme yapmayı amaçlamaktadır. Yapılan bu çalışma ile yazılım geliştirme projelerinin, kaynakların daha verimli kullanılarak yönetilmesi konularında yapılacak çalışmalara bir katkı sağlamak da hedeflenmiştir. Araştırmadan elde edilecek bulguların yazılım proje yöneticisinin, Benzetim eniyilemesi yöntemi ile projelerin türüne, bütçesine ve süresine uygun bir şekilde kaynak atayabilmesine; kaynakların tecrübe seviyesini, maaşını ve zaman çizelgelerini dikkate alan bir proje yönetimi yapabilmesine katkı sağlayacağı düşünülmektedir. Ayrıca benzetim modeli ve benzetim eniyilemesi kullanılan bu çalışmanın, benzetim modeliyle senaryo analizleri ve benzetim eniyilemesi yapmayı hedefleyen başka çalışmalara da katkı sağlayacağı düşünülmektedir.

Araştırmanın amacı doğrultusunda ortaya konan Yüksek Lisans tez çalışması toplam beş bölümden oluşmaktadır.

Birinci bölümde yazılım, yazılım geliştirme projeleri ve yazılım projelerinde proje yönetimi tanımlamaları yapılmış, yazılım geliştirme yaşam döngüsü süreçleri ve yazılım geliştirme projelerinin iyileştirilmesinde kullanılan çözüm yöntemleri hakkında bilgi verilmiştir.

İkinci bölümde, yazılım geliştirme projelerinde kullanılan Benzetim ve çizelgeleme çözüm yöntemlerine ilişkin literatür araştırmasına yer verilmiştir.

Üçüncü bölümde, çok projeleri kaynak kısıtlı bir yazılım firmasının genel sistemi ve yazılım geliştirme süreçleri ve darboğazları hakkında bilgi verilmiş, mevcut sistemin iyileştirilmesi için yapılan benzetim modellemesi ve benzetim eniyilemesi yöntemi açıklanmıştır.

Dördüncü bölümde, benzetim modelinin doğruluğu ve geçerliliği kontrol edilerek mevcut sistemi yansıtan benzetim modelinin çıktıları analiz edilmiştir. Belirlenen performans ölçütlerinde iyileşme sağlamak amacıyla benzetim eniyilemesi yapılmış, bu kapsamda oluşturulan deney tasarımları, senaryo denemeleri ve meta-modeller açıklanmıştır. Benzetim eniyilemesi yapılmasıyla ortaya çıkan, her performans ölçütü için ayrı belirlenen tek ölçütlü karar modelleri ve tüm performans ölçütlerini aynı anda göz önünde bulunduran çok ölçütlü karar modeli çözülmüş ve çıktıları tartışılmıştır.

Beşinci bölümde çalışmanın sonuçları ve gelecek çalışmalar için önerilere yer verilmiştir.

2. İLGİLİ ARAŞTIRMALAR

Bu bölümde, ülkemizde ve yurtdışında yazılım geliştirme projelerinde Benzetim ve Çizelgeleme Yöntemleri kullanarak yazılım projelerinin yönetimine destek olmayı amaçlayan çalışmalar, ilgili literatür taranarak ele alınmış ve incelenen araştırmalar kronolojik sıraya göre özetlenerek verilmiştir.

2.1. Yazılım Geliştirme Projelerinde Benzetim ile İlgili Yapılan Araştırmalar

Kellner, Madachy ve Raffo (1999), çalışmalarının yer aldığı derginin “Yazılım Süreç Simülasyonu Modellemesi” üzerine çıkardığı özel sayısında yayınlanan ve Birinci Uluslararası Silver Falls Çalıştay'ında (ProSim'98) sunulan makaleleri özetleyen bir çalışma yapmışlardır. Çalışmada, bu alandaki çalışmaları karakterize etmeye yardımcı olan bir çerçeve sağlanmıştır ve bu yeni karakterizasyon şemasını bu özel sayıdaki makalelerin çoğuna uygulanmıştır. Simülasyon kurabilmek için kullanılabilecek soruları ve sorunları (“neden”), sistemi yararlı bir şekilde simüle edilebilecek kapsam ve değişkenleri (“ne”) ve en verimli şekilde kullanılabilecek modelleme yaklaşımlarını ve tekniklerini (“nasıl”) tanımlamışlardır. Benzetimin karar vermede, risk azaltmada ve stratejik, taktiksel, operasyonel düzeylerde yönetime yardımcı olacağını belirten bu çalışmada, yazılım süreçleri simülasyonlarını kullanmanın birçok nedenini aşağıdaki altı amaç kategorisine ayırmışlardır:

- Stratejik Yönetim,
- Planlama,
- Kontrol ve operasyonel yönetim,
- Süreç iyileştirme ve teknolojinin benimsenmesi,
- Anlayış,
- Eğitim ve öğrenim

Ayrıca modelin amacı ile birlikte belirlenen anahtar sorulara bağlı olarak, bir süreç simülasyonunun sonuçları için çok sayıda farklı değişken tasarlanabileceğini anlatan çalışma, yazılım süreci simülasyon modelleri için en sık görülen sonuç değişkenlerini aşağıdaki gibi listelemişlerdir:

- Efor / maliyet,
- Döngü süresi,
- Kusur seviyesi,
- Zaman içindeki personel gereksinimleri
- Personel kullanım oranı,
- Maliyet / fayda, yatırımın geri dönüşü (ROI) veya diğer ekonomik önlemler (genellikle değişiklikler için ilgi çekici)
- İş hacmi / verimlilik
- Kuyruk uzunlukları (biriktirme listeleri)

Yazılım süreç benzetim modelleri geliştirirken, yönetimin ele almak istediği amacı ve soruları/sorunları belirlemenin, model kapsamını ve toplanması gereken verileri tanımlamak için çok önemli olduğunu savunan bu çalışma, pratik uygulama için bir benzetim modelleme yaklaşımının seçilmesinde bir miktar rehberlik sunmuş ve ek araştırma gerektiren bazı konular önermiştir [40].

2003 yılında yaptıkları çalışmada Rus, Neu ve Münch, yazılım geliştirme süreçlerinin ayrık olay tabanlı benzetim modellerinin geliştirilmesine odaklanan ve diğer modelleme yaklaşımlarına da genişletilebilen sistematik bir model önermişlerdir. Modelleme sürecini tanımlamak ve açıklamak için, yazılım mühendisliği uygulama ve ilkelerini uygulamışlardır. Geliştirdikleri modelin, modelleme ve benzetim uygulayan kişiler tarafından kullanılmasını hedeflemişlerdir. Bu çalışmada, benzetim modelinin de bir yazılım ürünü olduğu belirtilmiştir ve bu nedenle benzetim modelinin yaşam döngüsü, yazılım geliştirme yaşam döngüsüne

benzetilmiştir. Yazılım geliştirme yaşam döngüsünün aşağıdaki üç ana aşamadan oluştuğu belirtilmiştir:

1. Geliştirme (Development)
2. Dağıtım (Deployment)
3. İşletim & Bakım

Ayrıca, bu aşamalarındaki faaliyetlerin farklı zaman düzeni ve bağımlılıkları olabileceğini, bu nedenle ortaya çıkan yaşam döngüsünün şelale (waterfall), yinelemeli (iterative) ve çevik (agile) gibi farklı biçimlerde olabileceğini vurgulamışlardır. Bu yaşam döngüsüne göre benzetim modeli oluştururken benzetim modeli geliştirme için gerekli mühendislik faaliyetlerini aşağıdaki gibi listelemişlerdir:

1. Gereksinim tanımlanması ve şartname hazırlanması
2. Modellenen sürecin analizinin yapılması
3. Modelin tasarlanması
4. Modelin uygulanması
5. Geliştirme boyunca modelin doğrulanması ve onaylanması

Araştırmacılar, benzetim modeli oluştururken; müşterinin gereksinimlerindeki değişiklikler, modellenen sürecin tanımındaki değişiklikler, modelin nicel kısmı için verilerin bulunmaması gibi risk faktörleri tanımlamışlardır. Ayrıca, girdi parametreleri ve çıktı değişkenleri tanımlayarak oluşturdukları modelde birtakım ne-eğer kullanım senaryoları denemişlerdir. Bu çalışmada, benzetim yöntemi yardımıyla kullanım senaryolarının oluşturulabildiği ve çıktıların analiz edilebildiği, bu senaryo ve analizlerin de sistemin planlanabilmesi açısından yönetimin karar vermesini desteklediği ortaya koyulmuştur [42].

2003 yılında Tsai, Moskowitz ve Lee tarafından yapılan bir çalışmada, yazılım geliştirme projesinin proje maliyetini, süresini ve riskini azaltmak için projenin planlama aşamasında projeye en uygun proje ekibini oluşturmanın yani kaynak seçiminin çok önemli olduğu vurgulanmıştır. Çalışmalarında, Tayvan'daki Bilgi Endüstrisi Enstitüsü'nden bir veri

alma yazılımı projesine dayanarak, belirsiz bir görev karmaşıklığı ortamında tek ve birden çok uzmanlık için iki ayrı kaynak seçim problemi vakası dikkate alınmıştır. İnsan kaynakları ve görevler arasındaki ilişkiyi göstermek için bir Kritik Kaynak Diyagramı (KKD) kullanılarak, hesaplama açısından verimli, entegre bir yöntem geliştirilmiştir. Geliştirilen bu yöntem sonrasında insan kaynakları Taguchi'nin görev karmaşıklığına dayanıklı parametre tasarımından yararlanılarak seçilmiştir. Taguchi'nin parametre tasarımında, insan kaynakları kontrol edilebilir faktörler olarak görülürken, belirsiz karmaşıklık görevleri kontrol edilemeyen faktörler olarak temsil edilmiştir. Araştırmacılar, Taguchi'nin parametre tasarım yaklaşımını kullanmalarının sebebini, proje maliyeti ve süresine ilişkin verilerin kolayca hesaplanabilir ve kesin olmasına dayandırmışlardır. Ayrıca, görev karmaşıklığı ölçüsünü, her biri iyimser, normal ve kötümser karmaşıklık koşulları altında tahmin edilen yazılı kod satırlarının toplam sayısı olarak belirlemişlerdir. Yaptıkları çalışmada, önerdikleri yaklaşımın güçlü performans elde etmenin yanı sıra proje maliyetini ve süresini önemli ölçüde azaltabilen, insan kaynakları seçimi için verimli ve etkili bir yöntem olduğu ortaya konulmuştur [43].

Neu, Hanne, Münch ve Nickel, 2003 yılında yaptıkları çalışmada, yazılım geliştirme projelerinin denetiminde kullanılmak üzere benzetim tabanlı karar desteği sağlamak için bir kod inceleme modeli oluşturmuşlardır. Yazılım geliştiren kuruluşların, yazılım geliştirmenin kalitesi, maliyeti ve süresiyle ilgili giderek artan zorluklarla karşılaşması nedeniyle, geliştirme sürecini anlamak, belgelemek, görselleştirmek, planlamak ve iyileştirmek için süreç modellerini kullanmasını tavsiye ettikleri bu çalışmalarında oluşturdukları model, denetimlerin başlatılması ve uyarlanması için karar vermeyi desteklemeyi amaçlamaktadır. Bu alanda yapılan daha önceki çalışmalarda önerilen benzetim modellerinin çoğunun sistem dinamikleri yaklaşımına dayandığını ancak, kurumsal sorunların, ürünlerin ve kaynakların daha ayrıntılı bir temsiline izin veren denetim sürecine odaklanan ayrık bir olay simülasyon modeli geliştirilmesi gerektiğini bu çalışmada savunmuşlardır. Çalışmalarına göre, ayrık olay benzetim modelindeki sistem dinamiği yaklaşımının aksine, yazılım ürünleri (kod bileşenleri gibi) ve kaynaklar (geliştiriciler gibi) özniteliklerle (ör. Boyut, beceriler) temsil edilmektedir. Yazılım geliştirmenin yaratıcı ve insan temelli bir faaliyet olduğunu vurgulayan araştırmacılar

alışmalarına, ngrlemeyen etkiler iin becerileri ve bazı stokastik faktrleri dahil etmiřlerdir.

Benzetim modelini kurmak iin ařağıdaki adımların uygulanması gerektiğıne alışmalarında yer vermiřlerdir:

1. Adım: Benzetim Modeli Hedefinin Tanımlanması
2. Adım: Statik Sre Modelinin Geliřtirilmesi
3. Adım: Nitel Bir Modelin Geliřtirilmesi
4. Adım: Nicel Bir Modelin Geliřtirilmesi

alışmalarının sonucunda oluřturdukları modelin birkaç projede denetim yapan mfettiřlere destek olduėunu ortaya koymuřlardır [44].

Kouskouros ve Georgiou, 2006 yılında yaptıkları alışmada, yazılım projelerini yneten proje yneticilerine kontrol ve izleme konusunda yardımcı olmak, aynı zamanda en iyi planlama alternatiflerini belirlemek amacıyla bir ayırık olay benzetim modeli oluřturmuřlardır. ExtendTM benzetim programı aracılığıyla oluřturdukları bu benzetim modeli, teslimat sreleri ve kalite tleri gibi eřitli proje geliřtirme ayrıntılarının tahmin edilmesini saėlayan bir yazılım projesi srecinin temel bir blmne odaklanmıřtır. Model kapsamı, artımlı bir yazılım geliřtirme yařam dngsnn bir blmn ele almıřtır. Bu alışmada kurulan model ařağıdaki senaryolarla, farklı atama alternatiflerinin projenin yrtlmesi zerindeki etkisini incelemiřtir:

Senaryo 1. Bir tasarımcı, bir rnle ilgili tm faaliyetleri yapmak zere atanır.

Senaryo 2. Bir tasarımcı kodlamaya, ikincisi tasarımcı ise dokmantasyon ve birim testine atanır.

Senaryo 3. Bir tasarımcı dokmantasyona, ikinci tasarımcı kodlama ve birim testine atanır.

Senaryo 4. Bir tasarımcı dokmantasyon ve kodlamaya, ikinci tasarımcı birim testine atanır.

Her senaryo deneyinde yeterli sayıda tekrarlamayı belirlemek iin Monte Carlo simlasyonu gerekleřtirilmiř ve sonuların istatistiksel zellikleri (sonu deėiřkenleri) uygun

alıřma sayfasında toplanmıřtır. Bylece kaydedilen sonular, tamamlanma sresi ve rn kalitesi (kusur sayısı) aısından her bir alternatifin proje uygulaması zerinde sahip olacaėı etkiyi tahmin etme fırsatı vermiřtir. Bu alıřma, simlasyonun eřitli deneylerinden elde edilen sonuları karřılařtırarak, her alternatifin artıları ve eksilerinin deėerlendirilebileceėi sonucunu ortaya konmuřtur [19].

Agarwal, 2007 yılında yaptıėı doktora tez alıřmasında yazılım deėiřtirme srecinin ok aracı tabanlı bir simlasyon modeli deėiřtirmiřtir. alıřmasında, yazılım deėiřtirme srelerinin karmařık olması nedeniyle srecin bir blmnde yapılan deėiřikliklerin srecin genel sonucunu nasıl etkileyebileceėini tahmin etmenin zor olduėunu belirten arařtırmacı, yazılım deėiřtirme srecinin benzetim yntemiyle modellenmesinin sreteki deėiřiklikleri izlemede katkı saėlayacaėını vurgulamıřtır. Bu amala, aracılarn simlasyon iindeki gerek yazılım deėiřtiricileri taklit eden varlıklar olduėu bir ajan mimarisine sahip bir yazılım sreci simlasyon modeli deėiřtirmiřtir. Deėiřtirdiėi bu simlasyon modelinin yazılım deėiřtiricilerin ve ėrencilerin yazılım sreci kavramlarını, srecin risklerini ve diėer karar verme etkinliklerini anlayabilecekleri bir yaparak ėrenme modeli olduėunu anlatmıřtır.

Yazılım deėiřtirme sreci simlasyonunun kapsamının genellikle ařaėıdaki kapsamlardan biri olacaėını belirtmiřtir:

- Yařam dngsnn bir kısmı (rneėin, tasarım ařaması, kod incelemesi, testlerin bir kısmı veya tamamı, gereksinim ynetimi)
- Bir deėiřtirme projesi (rneėin, tek rn deėiřtirme yařam dngs)
- Birden ok, eřzamanlı proje (rneėin, bir departman veya blm genelinde)
- Uzun vadeli rn deėiřimi (rneėin, tek bir rnn birden ok, ardıřık srmleri)
- Uzun vadeli organizasyon (rneėin, nemli bir zaman periyodu boyunca birden fazla rnn ardıřık srmlerini kapsayan stratejik organizasyonel hususlar)

Yazılım deėiřtirme sreci simlasyon modelleri iin en yaygın sonu deėiřkenlerini ařaėıdaki řekilde tanımlamıřtır:

- Efor / maliyet
- Döngü süresi (süre, program, pazara sunma süresi, aralık)
- Kusur seviyesi
- Zaman içindeki personel gereksinimleri
- Personel kullanım oranı
- Maliyet / fayda, yatırım getirisi veya diğer ekonomik önlemler
- Çıktı / verimlilik
- Sıra uzunluğu (birikmiş işler)

Bu çalışmada, simülasyon işlemleri modelin yürütülmesi sırasında değiştirilebilir olduğu için simülasyon, her sürenin sonunda kullanıcıdan geri bildirimi kabul edebilir şekilde modellenmiştir. Simülasyon arayüzü, gerçek ve planlanan matrisi görüntülemektedir ve bu karşılaştırma matrislerine dayanarak, kullanıcının, yazılım sürecine projenin genel geliştirme döngüsünü iyileştirebileceğini düşündüğü herhangi bir değişikliği ekleyebileceği şekilde oluşturulmuştur. Kullanıcıya simülasyon kodunu değiştirmek zorunda kalmadan yeni bir yazılım geliştirme aşaması sunan bu simülasyon modelinde araştırmacı, kullanıcının herhangi bir zamanda, performanslarına ve yeteneklerine bağlı olarak geliştiricilerin sayısını değiştirebileceği veya geliştiricileri yazılım geliştirmenin farklı aşamalarına atayabileceği bir simülasyon modeli ortaya koymuştur. Araştırmacı ayrıca, simülasyon modelini yazılım geliştirmenin performansını ve/veya güvenilirliğini daha iyi anlamak ve optimize etmek için bir araç olarak kullanılabileceğini, aynı zamanda yazılım geliştirme süreçlerinin uygulanabilirliğini doğrulamak ve/veya ekip oluşturmada da kullanılabileceğini bu çalışmanın bir faydası olarak tanımlamıştır [41].

Cherif, 2008 yılında yaptığı yüksek lisans tez çalışmasında, çoklu etmen tabanlı benzetim yaklaşımının (MABS-multi-agent based simulation) yazılım süreç benzetim modelleme alanında kullanılmak üzere uygun olup olmadığını araştırmak için bir karşılaştırma çalışması yapmıştır. Karşılaştırma yapabilmek için çoklu etmen tabanlı benzetim simülasyonu

ve sistem dinamiği (SD-System Dynamics) simülasyonu yaklaşımlarının eşit şartlarda uygulanabileceği entegre bir simülasyon çerçevesi tasarlamıştır. Tasarlanan bu modellere aynı girdiler girilerek sonuçları karşılaştırılmıştır. Çalışmada iki yaklaşımın avantajları ve zayıf yönleri tartışılmıştır. MABS yaklaşımı ile SD yaklaşımı karşılaştırıldığında, SD yaklaşımının performansı ve kaliteyi olduğundan fazla tahmin ettiği, süre ve maliyeti olduğundan daha az tahmin etme eğiliminde olduğu gözlemlenmiştir. Çalışmanın sonucunda, çoklu etmen tabanlı benzetim yaklaşımının, yazılım süreç benzetim modelleme alanına sistem dinamiği benzetim yaklaşımından daha uygun olduğu ortaya koyulmuştur [45].

Ruiz, Rodriguez, Riquelme ve Harrison 2011 yılında yaptıkları çalışmada yazılım projeleri proje yöneticilerinin projede çalışacak ilk ekip boyutu için en iyi değerleri bulmasına ve belirli bir proje için tahminlerin planlanmasına yardımcı olmak için simülasyon yoluyla çok amaçlı optimizasyon tekniklerinin kullanılmasından oluşan bir yaklaşımı tanımlamayı ve böylece maliyet, zaman ve verimliliği optimize etmeyi amaçlamaktadır. Çalışmalarında, karar vermeyi optimize etmede kolaylık sağlayan simülasyon yönteminin, proje yöneticileri tarafından uzun yıllardır kullanıldığını ancak, mevcut simülasyon paketlerinin yalnızca tek bir hedefi veya birden çok hedefi tek bir uygunlukta birleştirilen simülasyon optimizasyonunu içerdiğini anlatmışlardır. Araştırmacılara göre, bu tür tek optimizasyon yaklaşımları yararlı olmasına rağmen, birkaç çakışan hedefin optimizasyonunun bir arada olduğu yazılım proje yönetimi gibi kapsamlı bir alanda yeterli görünmemektedir. Bu nedenle bu çalışmada, bir yazılım projesi, Sistem Dinamiği (SD) simülasyon modeliyle modellenmiş, girdi parametrelerinin modele girilmesiyle üretilen veriler, çıktı değişkenlerinin en uygun çözümlerini bulmak için NSGA-II adı verilen çok amaçlı bir optimizasyon algoritması ile birleştirilmiştir.

Çalışılan simülasyon modeline aşağıdaki girdi parametreleri girilmiştir:

- Başlangıçtaki Acemi İş Gücü: Projeye tahsis edilen başlangıçtaki acemi personel sayısı
- İlk Deneyimli İş Gücü: Projeye tahsis edilen ilk deneyimli personel sayısı
- Proje Büyüklüğü: Projenin tahmini boyutu
- Planlanmış Zaman: Projenin tahmini programı

- Çıktı değişkenleri aşağıdaki şekilde tanımlanmıştır:
- Proje Bitiş (Zaman): Projenin son zamanı
- Kümülatif Maliyet (Maliyet): Projenin nihai maliyeti
- Üretkenlik (P-rod): Proje yaşam döngüsü boyunca ekibin ulaştığı ortalama verimlilik
- Tamamlanan Kesir: Simülasyonun herhangi bir anında proje tamamlama yüzdesi
- Etkili İş Gücü: Ekip tarafından gerçekleştirilen etkin iş oranı

Sonuç olarak araştırmacılar, çok amaçlı optimizasyon ve SD simülasyon modellemenin, proje yöneticilerinin karar verme için ihtiyaç duyduğu Pareto cephesini oluşturabilen tamamlayıcı teknikler olduğunu ve proje yöneticilerine karar verme süreçlerinde yardımcı olabildiğini ortaya koymuşlardır [25].

Spasic ve Onggo 2012 yılında bir yazılım ve simülasyon şirketi olan AVL'nin yazılım geliştirme departmanındaki özel bir yazılımın geliştirme süreciyle ilgili bir çalışma yapmışlardır. Çalışmalarında, yazılım geliştirme projelerinin, çeşitli aşamalarındaki yüksek belirsizlik nedeniyle yönetilmesinin zorluğunu ve simülasyonun yazılım proje yöneticileri tarafından proje planları oluşturmak için kullanılan yardımcı araçlardan biri olduğunu vurgulamışlardır. Bu çalışmada yazılım geliştirme süreçleri aşağıdaki 6 aşamadan oluşmuştur:

1. Aşama: Başlangıç
2. Aşama: Analiz ve Şartname Çalışmaları
3. Aşama: Tasarım ve Geliştirme
4. Aşama: Entegrasyon
5. Aşama: Test
6. Aşama: Stabilizasyon

Araştırmacılar, çalışmalarında yazılım geliştirme projesinin tasarım ve geliştirme aşaması için ajan tabanlı simülasyon modeli kurmuşlardır. Yazılım geliştirme sürecinin hangi aşaması için simülasyon modeli kuracaklarına şirkette daha fazla girdi verisi bulabilecekleri

aşamaya göre karar vermişlerdir. Asıl amaçları karmaşıklıktan uzak, ayrıntının çok olmadığı net bir model ortaya koymak olmuştur. Çalışmanın sonucunda aracı tabanlı bir simülasyon modelleme paradigmasının, yazılım geliştirme sürecini açıklamak için uygun olduğu ve görece basit bir ajan tabanlı simülasyon modeliyle bile proje süresinin doğru tahmin edilebildiği görülmüştür. Ayrıca bu çalışma, birçok yazılım geliştirme şirketi/departmanı tarafından kolayca elde edilebilen veriler kullanılarak ajan tabanlı simülasyon modelinin nasıl geliştirilebileceğini, kalibre edilebileceğini ve doğrulanabileceğini göstermiştir [46].

Song ve diğerleri (2018) tarafından yapılan bir çalışmada, belirsizlik altında kaynak kısıtlı çoklu proje planlamasındaki zorlukları ele almak için aracı tabanlı bir simülasyon sistemi önerilmiştir. Sistem, projeler arasındaki iç ve karşılıklı bağımlılıkları, sınırlı kaynaklar için çakışan talepleri ve öngörülemeyen olayların potansiyel zincirleme etkisini dikkate alarak kullanıcıların sağlam programlar geliştirmelerine yardımcı olmayı amaçlamaktadır. Sisteme gömülü interaktif grafik kullanıcı arayüzü ile kullanıcıların proje sürecini araştırması, değiştirilen programı değerlendirmesi, beklenmedik aksaklıklardan kaynaklanan etkiyi tahmin etmesi ve çeşitli ne-eğer analizleri yapılması sağlanmıştır. Öngörülemeyen aksaklıkları gidermek ve çözümlerin sağlamlığını artırmak için simülasyon sisteminde reaktif ve proaktif çizelgeleme algoritmaları geliştirilmiştir. Bu çalışmada önerilen algoritmaların etkinliğini değerlendirmek için teknoloji yoğun bir endüstriyel ürün geliştirme programının gerçek dünya vaka çalışması üzerine deneyler yapılmıştır [47].

2.2. Yazılım Geliştirme Projelerinde Çizelgeleme ile İlgili Yapılan Araştırmalar

2005 yılında T. Hanne ve S. Nickel, bir yazılım geliştirme projesindeki denetimleri ve kodlama, inceleme, test ve yeniden çalışma aşamalarını içeren yeni bir yazılım geliştirme süreci modeli oluşturmuşlardır. Bu model yardımıyla, yazılım geliştirme projesindeki operasyonların ilgili kişilere atanması ve proje çizelgesinin oluşturulmasını hedeflemişlerdir. Oluşturdukları modeli, kalite (kusur sayısı), proje yapım süresi ve maliyetler açısından ele alarak çok amaçlı bir optimizasyon problemi olarak formüle etmişlerdir. Bu problemi çözmek için, bazı ek çizelgeleme buluşsal yöntemlerine dayalı olarak evrimsel bir algoritma tasarlamışlardır. Algoritmanın örnek problemlere uygulanmasının sonuçları tartışılmış ve

algoritmanın güvenilirliğini test etmesi gereken daha ileri bir çalışma için stokastik bir model kullanmaları gerektiği sonucu ortaya konmuştur [14].

Alba ve Chicano, 2007 yılında gerçekleştirdikleri çalışmada yazılım proje yöneticilerinin proje için en uygun kararları almasına yardımcı olmayı amaçlayan genetik algoritmalara dayalı bir araç hazırlamışlardır. Bir projeyi en iyi şekilde bitirmek için bütçe ve insan kaynaklarının en iyi şekilde yönetilmesi gerektiğini vurgulayan araştırmacılar, iyi bir proje yönetimi için proje süresinin ve maliyetinin de en az olacak şekilde olması gerektiğini savunmuşlardır. Genetik algoritmalar yardımıyla oluşturdukları araç, çalışanları proje görevlerine optimal bir şekilde atamak farklı konfigürasyonları denemek için kullanılabilen ve böylece projenin maliyetini, süresini azaltmayı hedeflemiştir. Bu çalışmada, istatistiksel olarak anlamlı çözümler elde etmek için 48 farklı proje senaryosunu çözmüşler ve her test için 100 bağımsız çalışma gerçekleştirmişlerdir. Sonuçlar, daha fazla görev içeren örneklerin çözülmesinin daha zor olduğunu ve çözümlerinin daha pahalı olduğunu göstermiştir. Ayrıca araştırmacılar, daha fazla çalışanı olan projelerin üstesinden gelmenin daha kolay olduğunu ve daha kısa sürede başarılı bir sonuca götürülebildiğini gözlemlemişlerdir [30].

Chang ve diğerleri 2008 yılında yaptıkları bir çalışmada, gerçek yazılım projelerin planlanmasında ortaya çıkan bazı nüansları ve optimum çözümleri bulmak için genetik bir algoritmayı içeren bir çizelgeleme modelini açıklamışlardır. Karmaşık yazılım projelerinin etkili yönetiminin, karmaşık optimizasyon problemlerini çözme becerisine bağlı olduğunu belirten araştırmacılar, yazılım proje yönetimi üzerine yapılan bir çalışmanın, optimal çizelgelemeyi ve kaynakların dikkatli kullanımını gerektiren çalışan-görev ataması yapmasına dikkat edilmesi gerektiğini vurgulamışlardır. Çalışmanın sonucunda, genetik algoritma kullanılan modelin, proje yöneticilerine rehberlik etmeye yardımcı olacak uygun bir araç olduğu ortaya konmuştur [20].

2008 yılında F. Wen ve C.-M. Lin, yazılım geliştirme projelerinde proje süresinin ve maliyetinin en aza inmesi için projedeki insan kaynağına odaklanarak çok aşamalı insan kaynağı atama problemi üzerine bir çözüm yöntemi önermiştir. Önerilen çözüm için çok amaçlı genetik algoritma tasarlanmıştır. Bu algoritmayı çözmek için geliştirilmiş sabit uzunlukta kodlama yöntemi, popülasyon çeşitliliği için uyarlanabilir ağırlıkta bir uygunluk

atama yaklaşımı ve pareto setinde en iyi seçeneğe karar vermek için ağırlık faktörü yöntemi kullanılmıştır. Algoritma üzerinde deneyler yapılmış ve deneyle sonucunda, diğer algoritmalarla karşılaştırıldığında bu çalışmada önerilen algoritmanın daha verimli olduğu ortaya konmuştur [28].

J. F. Gonçaves, J. J. M. Mendes, ve M. G. C. Resende, 2008 yılında yaptıkları çalışmada kaynak kısıtlı çoklu proje çizelgeleme problemi için genetik bir algoritma geliştirmişlerdir. Geliştirdikleri algoritma, 10, 20, 30, 40 ve 50 bir test problemi seti üzerinde test edilmiştir. Hesaplamalı deneylerde, bu algoritmanın karşılaştırılan diğer iki algoritmadan daha iyi sonuçlara sahip olduğu ve optimum değere (sıfır) çok yakın değerler elde ettiği görülmüştür. Çalışmanın sonucunda önerilen algoritmanın etkinliğini doğrulanmıştır. [48].

2011 yılında Yannibelli ve Amandi tarafından yapılan çalışmada, tasarım, kodlama, test, kurulum ve bakım süreçlerini içeren yazılım projelerini planlamak için proje yöneticilerine kaynak atama konusunda yardımcı olmak amacıyla bilgiye dayalı evrimsel bir yaklaşım önerilmiştir. Bu çalışmada önerilen yaklaşım, planlanacak bir yazılım projesi verildiğinde, yaklaşım otomatik olarak proje için uygulanabilir takvimleri tasarlayacak ve tasarlanan her takvimi yöneticiler için öncelikli olan bir optimizasyon hedefine göre değerlendirecek şekilde çalışılmıştır. Çalışmadaki evrimsel yaklaşımın performansını değerlendirmek için, sekiz farklı problem örneği üzerinde geliştirilen hesaplama deneyleri yapılmıştır. Çalışma sonucunda, yaklaşımın bahsedilen sekiz setin yedisinde optimal bir etkinlik seviyesine ve geri kalan sette yüksek bir etkinlik seviyesine ulaştığı gözlemlenmiştir [49].

M. Di Penta, M. Harman, ve G. Antoniol, 2011 yılında yazılım proje planlamasına yönelik arama temelli yaklaşımın kapsamlı bir incelemesini yapmışlardır. Bu inceleme çalışması, Sistem Dinamiği (SD) ve Genetik Algoritma (GA) gibi arama temelli tekniklerin, personel seviyesi ayarlama, personelin ekiplere tahsis etme, farklı uzmanlık alanlarına dayalı ekip oluşturma konularında nasıl kullanılabileceğini göstermiştir. Ayrıca yapılan bu çalışma, arama tekniklerinin iki büyük ölçekli, gerçek dünya bakım ve gelişim projelerine uygulanmasından elde edilen deneysel sonuçları da sunmuştur. Sunulan sonuçlar, hem SA hem de GA'ların yazılım projelerinde karar desteği sağlayan çözümler oluşturabileceğini ortaya koymuştur [21].

J. Xiao, X. T. Ao, ve Y. Tang, 2013 yılında yaptıkları çalışmada yazılım projesi çizelgeleme problemi çözümü için bir karınca kolonisi optimizasyonu yaklaşımı önermişlerdir. Yazılım proje çizelgeleme problemini NP-zor olarak tanımlayan araştırmacılar, farklı sezgisel yöntemlerin performanslarını karşılaştırmışlardır. Karşılaştırmalar sonucunda, önerilen yaklaşımın yazılım projesi çizelgemedeki kullanılan genetik algoritma yaklaşımından daha iyi performans gösterdiği görülmüştür [29].

P. Akyl Kurt, 2018 yılında yaptığı yüksek lisans tez çalışmasında, birden çok projeyi aynı anda yürüten bir yazılım firmasının projelerinin çizelgelenmesi için bir doğrusal karar modeli geliştirmiştir. Geliştirdiği doğrusal karar modeli, kaynak kısıtı altındaki projelerin en az maliyetle, en kısa sürede tamamlanmasını amaçlamaktadır. Modeli oluştururken bu projelerdeki faaliyetlerin öncüllük ilişkilerini de dikkate almıştır. Elde edilen karar modeli firma verileri kullanılarak bir çözücü yardımıyla çözülmüştür. Çalışmanın sonucunda, proje faaliyetlerinin, birden çok projenin en kısa sürede en az maliyetle tamamlanabilmesi için hangi zamanlarda başlaması gerektiği ortaya koyulmuştur [50].

Shen ve diğerleri, 2018 yılında yaptıkları çalışmada, dinamik yazılım projesi çizelgeleme problemi için gerçekçi bir matematiksel model önermişlerdir. Oluşturdukları bu model, beceri yeterliliğinin zamanla gelişebileceğini ve önceki çalışmalardan farklı olarak, bu tür bir iyileştirmenin çalışanların motivasyon ve öğrenme yeteneği özelliklerinden ve becerilerinden etkilendiğini düşünerek oluşturulmuştur. Ayrıca, çalışanların tahsisatla ilgili memnuniyet hedefi ve projenin çeşitli kısıtlamalar altında proje süresi, maliyeti, sağlamlığı ve kararlılığı hedefleri tanımlanmıştır. Yazılım proje planlamasını, çalışanları bir yazılım projesindeki görevlere ayırma sorunu olarak tanımlayan araştırmacılar, yazılım şirketlerinin değişen ortamları nedeniyle yazılım proje planlamasının dinamik bir optimizasyon olduğu anlamına geldiğini belirtmişlerdir. Bu çalışmada, takvimleri dinamik olarak değişen yazılım proje ortamlarına uyarlamak ve sorunu proaktif bir şekilde yeniden planlayarak çözmek için, çok amaçlı iki arşivli memetik algoritma tabanlı Q-öğrenme (multi-objective two-archive memetic algorithm based on Q-learning- MOTAMAQ) önerilmiştir. Memetik algoritma (MA), sezgisel arama yoluyla özel bilgilerle desteklenen mutasyon sürecini taklit eden meta-sezgisel bir yaklaşım olarak tanımlanmıştır. Bu çalışmada önerilen algoritma, gerçek yazılım

projelerinden türetilen 18 dinamik karşılaştırma örneği ve 3 bulut sunucusu üzerinde kapsamlı deneylerle denenmiştir. Çalışmada, Q-öğrenme tabanlı öğrenme mekanizmasının, farklı takvimler için uygun arama operatörlerini seçebildiği sonucu ortaya konmuştur. Ayrıca bu çalışma, yazılım yöneticilerine birçok hedef arasında daha derin bir öngörü sağlayarak, onların daha uygun kararlar almalarını da sağlamıştır [27].

Yukarıda, bu tez çalışması kapsamında yararlanılan, yazılım geliştirme projelerinin iyileştirilmesinde benzetim ve çizelgeleme yöntemi kullanan araştırmalar kısaca özetlenmiştir. Özetlenen araştırmalarla ilgili genel bir değerlendirme yapıldığında, yurt içinde ve yurtdışında yapılan araştırmalarda yazılım geliştirme yaşam döngüsünün farklı aşamalarla tanımlandığı, yazılım geliştirme yaşam döngüsünün farklı aşamalarını benzetim ve çizelgeleme yöntemi gibi farklı yöntemler kullanarak iyileştirmeyi amaçlayan birçok çalışmanın olduğu ve bu çalışmalarının amaçlarının genel olarak, doğru planlamalarla süreyi ve maliyeti azaltmak, kaliteyi ve kârı artırmak olduğu görülmektedir. Ayrıca yazılım geliştirme projelerini yöneten proje yöneticilerine kontrol ve izleme konusunda yardımcı olmak, aynı zamanda en iyi planlama alternatiflerini belirlemek amacıyla yapılan çalışmalarda, yazılım geliştirme projelerinde dikkate alınması gereken temel kısıtın çalışanların tecrübe ve yetenek seviyeleri olduğu, diğer proje türlerinden farklı olarak yazılım projelerinin temelinde insan faktörü bulunan bir proje türü olduğu görülmektedir.

3. YÖNTEM

Bu bölümde, birden çok yazılım geliştirme projesini aynı anda takip eden kaynak kısıtlı bir yazılım firmasının genel sistemi hakkında bilgiler verilmiştir. Sistemin yazılım geliştirme yaşam döngüsünün, “gerçekleştirim” sürecinde gerçekleşen “kodlama”, “test”, “yeniden çalışma”, “entegrasyon”, “doğrulama”, “yeniden çalışma”, “kurulum” aşamalarının mevcut durumunu gösteren bir benzetim modeli oluşturulmuştur. Oluşturulan bu benzetim modelinin kapsamı, amaçları, kısıtları ve bunların benzetim programı üzerinde nasıl gösterildiği konuları üzerinde durulmuştur. Aynı zamanda, benzetim modelinin belirlenen performans ölçütlerini eniyilemek (optimize etmek) amacıyla yapılan çıktı analizinde kullanılan istatistiksel yöntemler ve benzetim eniyilemesi yöntemi konu alınmıştır.

3.1. Yazılım Firması Sistemi Hakkında Bilgiler

Birden çok yazılım geliştirme projesini birlikte yürüten yazılım firmaları, genellikle üzerinde çalıştıkları projelerin bütçelerini, tamamlanması beklenen sürelerini aşmadan, kaliteli bir şekilde müşterilere teslim etmeyi amaçlamaktadır. Ancak yazılımın karmaşıklık, uygunluk, değiştirilebilirlik, görünmezlik gibi temel özellikleri dikkate alındığında bir tane yazılım geliştirme projesini yürütmenin bile birçok açıdan zorlayıcı olduğu düşünülmektedir. Dolayısıyla birden çok yazılım projesini bir arada yürüten bir yazılım firmasının tüm projelere üstten bakabilen, proje adımlarını takip eden, karar vermeye destek sağlayıcı araçlara ihtiyacı bulunmaktadır. Bu çalışmada, tamamlanan proje sayısını en büyükmek, projelerin toplam süresini, çalışan maliyetini en küçükmek ve toplam kârı en büyükmek şeklinde belirlenen dört temel performans ölçütünü iyileştirmeyi hedefleyen çok projeli kaynak kısıtlı bir yazılım firmasının sistemi ele alınmış, firmadan uyarlanmış veriler sistemin girdileri olarak belirlenmiştir. Veriler, başka yazılım firmalarının kendi sistemlerine uyarlayabilecekleri şekilde esnek ve değiştirilebilir veriler olarak tasarlanmıştır.

Bu kapsamda sistem; süreçler, ekiplerin özellikleri, projelerin özellikleri, projelerin bedelleri, çalışanların saatlik ücretleri, kısıtlar ve varsayımlar, dikkate alınan performans ölçütleri başlıklarıyla açıklanmaktadır.

3.1.1. Süreçler

Yazılım firmaları, yazılım geliştirme projeleri yaparken teknik ve idari birçok süreçten geçmektedir. Sistemi iyileştirmek için hangi süreçlerin ele alınacağına baştan karar verip sürece uygun iyileştirmeler yapmak hedeflenmelidir. Bu çalışmada, Planlama/Analiz, Tasarım, Gerçekleştirim ve Bakım süreçlerini içeren yazılım geliştirme yaşam döngüsünün, “Gerçekleştirim” sürecinin aşamalarına odaklanılmıştır. Gerçekleştirim sürecinde gerçekleşen Kodlama, Test, Yeniden Çalışma, Entegrasyon, Doğrulama, Yeniden Çalışma, Kurulum aşamaları bu çalışmanın devamında süreç olarak belirtilecektir. Süreçler proje zorluğuna göre farklı ekiplerdeki farklı tecrübeye sahip çalışanlar tarafından yapılmaktadır. Bu çalışma kapsamında ele alınan süreçler ve süreçlerin yürütülmesinde görev alan ilgili ekipler Tablo 3.1.’de verilmiştir.

Tablo 3.1. “Gerçekleştirim” süreci aşamaları ve ilgili ekipler

Süreç No	Süreç	Ekip
1	Kodlama	Geliştirme Ekibi
2	Test	Test Ekibi
3	Yeniden Çalışma	Geliştirme Ekibi
4	Entegrasyon	Kurulum Ekibi
5	Doğrulama	Test Ekibi
6	Yeniden Çalışma	Geliştirme Ekibi
7	Kurulum	Kurulum Ekibi

Tablo 3.1. incelendiğinde; kodlama, test, yeniden çalışma, entegrasyon, doğrulama, yeniden çalışma, kurulum süreçlerinde geliştirme ekibi, test ekibi ve kurulum ekibi olmak üzere 3 farklı ekibin görev aldığı görülmektedir.

3.1.2. Ekiplerin Özellikleri

Bu sistemde, yazılım geliştirme projelerinin “Gerçekleştirim” sürecinin tamamlanabilmesi için yürütülecek süreçlerde görev alacak ekiplerin, toplamda 30 kişiden oluştuğu varsayılmaktadır. Her ekibin içinde farklı sayıda ve farklı tecrübelerde çalışanlar

bulunmaktadır. Tecrübe seviyesi A-B-C-D-E-F olarak tanımlanmış olup “A” en az tecrübeli, “F” en çok tecrübeli olacak şekilde bir dağılımdan oluşmaktadır. Ekiplerde yer alan farklı tecrübe seviyesine sahip çalışanlar ve sayıları Tablo 3.2., Tablo 3.3. ve Tablo 3.4.’te verilmiştir.

Tablo 3.2. Geliştirme ekibinin tecrübe seviyesine göre çalışan dağılımı

Tecrübe Seviyesi	Çalışan Sayısı
A	2
B	3
C	4
D	3
E	2
F	1
Toplam	15

Tablo 3.2.’de görüldüğü gibi, Geliştirme ekibinde A tecrübe seviyesinden 2, B tecrübe seviyesinden 3, C tecrübe seviyesinden 4, D tecrübe seviyesinden 3, E tecrübe seviyesinden 2, F tecrübe seviyesinden 1 çalışan olmak üzere toplam 15 çalışan görev almaktadır.

Tablo 3.3. Test ekibinin tecrübe seviyesine göre çalışan dağılımı

Tecrübe Seviyesi	Çalışan Sayısı
B	2
C	2
D	3
E	1
Toplam	8

Tablo 3.3.’te görüldüğü gibi, Test ekibinde B tecrübe seviyesinden 2, C tecrübe seviyesinden 2, D tecrübe seviyesinden 3, E tecrübe seviyesinden 1 çalışan olmak üzere toplam 8 çalışan görev almakta olup A ve F tecrübe seviyesinden çalışan bulunmamaktadır.

Tablo 3.4. Kurulum ekibin tecrübe seviyesine göre çalışan dağılımı

Tecrübe Seviyesi	Çalışan Sayısı
B	2
C	2
D	2
F	1
Toplam	7

Tablo 3.4.’te görüldüğü gibi, Kurulum ekibinde B tecrübe seviyesinden 2, C tecrübe seviyesinden 2, D tecrübe seviyesinden 2, F tecrübe seviyesinden 1 çalışan olmak üzere toplam 7 çalışan görev almakta olup A ve E tecrübe seviyesinden çalışan bulunmamaktadır.

3.1.3. Projelerin Özellikleri

Süreçlerin tamamlanmasıyla birlikte projeler ortaya çıkmaktadır. Sistemde yalın, orta ve kompleks olmak üzere 3 tür proje bulunmaktadır. Projenin türü; kapsamının genişliğine, kod satır sayısına, projenin süresine göre belirlenmektedir. Kapsam genişledikçe projelerin karmaşıklığı ve yazılması gereken kod satır sayısı artmaktadır. Kompleks proje türü karmaşıklığın en fazla olduğu proje olarak tanımlanmış proje olup, karmaşıklık özelliği orta ve yalın proje türlerine doğru gittikçe azalmaktadır. Projelerin karmaşıklığının artmasıyla birlikte projelerin tamamlanma süresi de artmaktadır. Yalın proje türünün proje süresi en fazla 1 yıl, Kompleks proje türünün proje süresi en fazla 3 yıl, Orta proje türünün proje süresi en fazla 2 yıl olarak belirlenmiştir. Proje türlerine göre projelerde çalışanların tecrübe düzeyleri de değişmektedir. Kompleks proje türünün tamamlanması daha yüksek tecrübe gerektirdiği için D-E-F tecrübe seviyesindeki çalışanlar görev almaktadır. Bunun yanında “Orta” proje türünde takvimi uygun olan her çalışan çalışabilmektedir. “Yalın” proje türünde ise A-B-C tecrübe seviyesindeki çalışanlar görev alıyor olup proje maliyetini yükseltmemesi için D-E-F tipi çalışanların görev alması tercih edilmemektedir. Tablo 3.5’te Proje Türüne göre süreçlerde hangi tecrübe seviyesindeki çalışanın görev alabileceği gösterilmiştir:

Tablo 3.5. Proje türüne göre süreçlerde çalışabilecek tecrübe seviyesi dağılımı

Proje Türü	Kodlama	Test	Yeniden Çalışma 1	Entegrasyon	Doğrulama	Yeniden Çalışma 2	Kurulum
Kompleks	Geliştirme Ekibindeki D-E-F Tecrübe Seviyesindeki Çalışanlar	Test Ekibindeki D-E Tecrübe Seviyesindeki Çalışanlar	Geliştirme Ekibindeki D-E-F Tecrübe Seviyesindeki Çalışanlar	Kurulum Ekibindeki D-F Tecrübe Seviyesindeki Çalışanlar	Test Ekibindeki D-E Tecrübe Seviyesindeki Çalışanlar	Geliştirme Ekibindeki D-E-F Tecrübe Seviyesindeki Çalışanlar	Kurulum Ekibindeki D-F Tecrübe Seviyesindeki Çalışanlar
Orta	Geliştirme Ekibindeki tüm çalışanlar	Test Ekibindeki tüm çalışanlar	Geliştirme Ekibindeki tüm çalışanlar	Kurulum Ekibindeki tüm çalışanlar	Test Ekibindeki tüm çalışanlar	Geliştirme Ekibindeki tüm çalışanlar	Kurulum Ekibindeki tüm çalışanlar
Yalın	Geliştirme Ekibindeki A-B-C Tecrübe Seviyesindeki Çalışanlar	Test Ekibindeki tüm çalışanlar	Geliştirme Ekibindeki A-B-C Tecrübe Seviyesindeki Çalışanlar	Kurulum Ekibindeki B-C Tecrübe Seviyesindeki Çalışanlar	Test Ekibindeki tüm çalışanlar	Geliştirme Ekibindeki A-B-C Tecrübe Seviyesindeki Çalışanlar	Kurulum Ekibindeki B-C Tecrübe Seviyesindeki Çalışanlar

Tablo 3.5. incelendiğinde Kompleks, Orta ve Yalın Proje Türlerinin; Kodlama, Test, Yeniden Çalışma 1, Entegrasyon, Doğrulama, Yeniden Çalışma 2, Kurulum süreçlerinin farklı tecrübe seviyesindeki çalışanlar tarafından yapıldığı görülmektedir.

Farklı tecrübe seviyesindeki çalışanların projelerin kendileriyle ilgili süreçlerinde çalışma süreleri birbirlerinden farklıdır. Tecrübe seviyesi arttıkça aynı süreç için işi yapma süresi azalmaktadır. Fakat aynı zamanda, farklı tür projelerde görev alabilecek çalışanların proje türüne göre çalışma sürelerinde de değişiklik olmaktadır. Örneğin; Kompleks, Orta ve Yalın Proje türlerinin hepsinde görev alabilecek Test Ekibinden D tecrübe seviyesindeki bir çalışanın her proje türü için ayırdığı süre farklıdır. Kompleks proje türü daha karmaşık olduğu için kompleks projeye daha fazla zaman ayırırken, yalın proje için daha az zaman ayırmak yeterli olacaktır. Bu çerçevede, tüm çalışanların Yalın, Orta ve Kompleks proje türlerinin süreçleri için en az ve en çok ayırdıkları süreler, saat bazında Tablo 3.6., 3.7. ve 3.8.'de yer almaktadır.

Tablo 3.6. Yalın proje türünde çalışanların tecrübe seviyesine göre saat bazında çalışma süresi dağılımı

Süreçler	A TECRÜBE SEVİYESİ		B TECRÜBE SEVİYESİ		C TECRÜBE SEVİYESİ		D TECRÜBE SEVİYESİ		E TECRÜBE SEVİYESİ	
	en az	en çok	en az	en çok	en az	en çok	en az	en çok	en az	en çok
Kodlama	759	810	608	648	547	583				
Test			899	1053	809	948	728	853	655	768
Yeniden Çalışma 1	380	405	304	324	273	292				
Entegrasyon			567	648	510	583				
Doğrulama			810	972	729	875	656	787	590	709
Yeniden Çalışma 2	127	135	101	108	91	97				
Kurulum			219	284	197	255				

Tablo 3.6. incelendiğinde Yalın proje türünde çalışanların çalışma saatlerinin tecrübe seviyesi arttıkça azaldığı görülmektedir.

Tablo 3.7. Orta proje türünde çalışanların tecrübe seviyesine göre saat bazında çalışma süresi dağılımı

Süreçler	A TECRÜBE SEVİYESİ		B TECRÜBE SEVİYESİ		C TECRÜBE SEVİYESİ		D TECRÜBE SEVİYESİ		E TECRÜBE SEVİYESİ		F TECRÜBE SEVİYESİ	
	en az	en çok	en az	en çok	en az	en çok	en az	en çok	en az	en çok	en az	en çok
Kodlama	1519	1620	1215	1296	1094	1166	984	1050	886	945	709	756
Test			1798	2106	1618	1895	1457	1706	1311	1535		
Yeniden Çalışma 1	759	810	608	648	547	583	492	525	443	472	354	378
Entegrasyon			1134	1296	1021	1166	919	1050			661	756
Doğrulama			1620	1944	1458	1750	1312	1575	1181	1417		
Yeniden Çalışma 2	253	270	203	216	182	194	164	175	148	157	118	126
Kurulum			437	567	394	510	354	459			255	331

Tablo 3.7. incelendiğinde Orta proje türünde çalışanların çalışma saatlerinin tecrübe seviyesi arttıkça azaldığı görülmektedir.

Tablo 3.8. Kompleks proje türünde çalışanların tecrübe seviyesine göre saat bazında çalışma süresi dağılımı

Süreçler	D TECRÜBE SEVİYESİ		E TECRÜBE SEVİYESİ		F TECRÜBE SEVİYESİ	
	en az	en çok	en az	en çok	en az	en çok
Kodlama	1476	1575	1329	1417	1063	1134
Test	2185	2559	1966	2303		
Yeniden Çalışma 1	738	787	664	709	531	567
Entegrasyon	1378	1575			992	1134
Doğrulama	1968	2362	1771	2126		
Yeniden Çalışma 2	246	262	221	236	177	189
Kurulum	531	689			383	496

Tablo 3.8. incelendiğinde Kompleks proje türünde çalışanların çalışma saatlerinin tecrübe seviyesi arttıkça azaldığı görülmektedir.

3.1.4. Projelerin Bedelleri ve Geliş Zamanları

Projeler başlamadan önce müşterinin ihtiyacına yönelik olarak geliştirilecek yazılımın, geliştirme aşamasında ihtiyaç duyacağı işçilik, malzeme, taşeron gibi giderlerin fizibilite çalışmaları yapılmaktadır. Yapılan fizibilite çalışması sonucunda ortaya çıkan maliyet, firmanın kârlılığı ve müşterinin bütçesi gözetilerek satış fiyatına dönüşür. Bu fiyat proje bedeli olarak adlandırılmaktadır. Proje türüne göre proje bedelleri farklılık göstermektedir. Yalın

proje türünden Kompleks proje türüne gidildikçe projelerin bedelleri artmaktadır. Bu çalışmada; Kompleks proje türünün proje bedeli 15.000.000 TL – 20.000.000 TL arasında, Orta proje türünün proje bedeli 6.000.000 TL – 14.000.000 TL arasında, Yalın proje türünün proje bedeli 400.000 TL – 5.000.000 TL arasında olacak şekilde belirlenmiştir. Tablo 3.9.’da 2020 yılında başlayan projelerin proje bedelleri ve geliş tarihleri yer almaktadır. Tabloda yer alan proje bedelleri ve geliş tarihleri varsayımlar üzerine yazılmıştır.

Tablo 3.9. 2020 yılında başlayan projelerin bedelleri ve geliş tarihleri

No	Kompleks Proje Türü		Orta Proje Türü		Yalın Proje Türü	
	Proje Bedeli	Geliş Zamanı	Proje Bedeli	Geliş Zamanı	Proje Bedeli	Geliş Zamanı
1	₺20.000.000	20.01.2020	₺12.000.000	9.03.2020	₺5.000.000	6.01.2020
2	₺15.000.000	13.07.2020	₺10.000.000	20.04.2020	₺3.000.000	3.02.2020
3	₺17.000.000	7.11.2020	₺9.000.000	15.06.2020	₺1.000.000	24.02.2020
4			₺11.000.000	3.08.2020	₺500.000	20.04.2020
5			₺7.000.000	26.10.2020	₺2.000.000	11.05.2020
6			₺12.000.000	16.11.2020	₺3.000.000	11.05.2020
7			₺8.000.000	7.12.2020	₺4.000.000	22.06.2020
8					₺5.000.000	13.07.2020
9					₺1.000.000	3.08.2020
10					₺600.000	24.08.2020
11					₺400.000	28.09.2020
12					₺1.500.000	12.10.2020
13					₺4.000.000	9.11.2020
14					₺500.000	9.11.2020
15					₺3.000.000	14.12.2020
Ortalama	₺17.333.333		₺9.857.143		₺2.300.000	

Tablo 3.9. incelendiğinde 2020 yılında Kompleks proje türünden 3 adet, Orta proje türünden 7 adet ve Yalın proje türünden 15 adet geldiği görülmektedir.

3.1.5. Çalışanların Saatlik Ücretleri

Çalışanların saatlik ücretleri, çalıştıkları proje türünden bağımsız olarak tecrübe seviyesine göre değişmektedir. Daha yüksek tecrübe seviyesindeki çalışanların saatlik ücreti daha yüksektir. Tablo 3.10.’da çalışanların tecrübelerine göre aldıkları saatlik ücretler yer almaktadır. Tabloda belirtilen ücretler, varsayımlar üzerine yazılmıştır.

Tablo 3.10. Çalışanları tecrübe seviyesine göre saatlik ücretleri

Ekip	A Tecrübe Seviyesi	B Tecrübe Seviyesi	C Tecrübe Seviyesi	D Tecrübe Seviyesi	E Tecrübe Seviyesi	F Tecrübe Seviyesi
Geliştirme Ekibi	₺17	₺23	₺28	₺37	₺48	₺63
Test Ekibi		₺23	₺28	₺37	₺48	₺63
Kurulum Ekibi		₺23	₺28	₺37		₺63

Tablo 3.10. incelendiğinde tüm ekiplerdeki aynı tecrübe seviyesindeki çalışanların aynı saatlik ücreti aldığı görülmektedir.

3.1.6. Kısıtlar ve Varsayımlar

Çok projeli bir yazılım firmasının sistemini oluştururken sistemde yer alan bazı kısıtlar varsayımlar belirlenmiştir. Bu kısıtlar ve varsayımlar aşağıdaki şekilde listelenmiştir:

- Mesai Saatleri 08.00-17.00 arasında, öğle saatleri 12.00-13.00 arasındadır.
- Mesai saatleri dışında ve öğle arası saatlerinde çalışma yapılmayacaktır.
- Bir çalışan birden fazla projede çalışabilmektedir.
- Her projede; Kodlama, Test, Yeniden Çalışma-1, Entegrasyon, Doğrulama, Yeniden Çalışma-2 ve Kurulum süreçleri aynı süreç sırasıyla uygulanmaktadır.
- Her süreç en az 1 kişi tarafından yapılacaktır.
- Bir süreci yerine getiren çalışanlar o sürecin gerektirdiği ekiplerden seçilecektir.
- Ekiplerin öncelikle en az tecrübeli çalışanlarına görev atanacak, onların takvimi doluyorsa diğer çalışanlara atama yapılacaktır.
- Bir projenin Kodlama sürecini yürüten çalışanla, Yeniden Çalışma 1- Yeniden Çalışma 2 süreçlerini yürüten çalışan aynı olacaktır.
- Kompleks proje türünün Kurulum aşamasında Kurulum ekibinden 2 çalışan görev alacaktır.

- Karmaşık proje en fazla 3, orta proje en fazla 2, yalın proje en fazla 1 yılda tamamlanacaktır.
- Proje gecikmesi halinde günlük gecikme maliyeti 100 TL olacaktır.
- Projenin erken bitmesi halinde erken bittiği gün sayısı kadar günlük 50 TL gelir sağlanacaktır.

3.1.7. Girdi Parametreleri

Sistemin girdi parametreleri aşağıdaki şekilde belirtilmiştir:

- Kompleks, Orta ve Yalın proje türlerinin geliş aralıkları
- Firmaya gelen projelerin dağılımı
- Kodlama, Test, Yeniden Çalışma-1, Entegrasyon, Doğrulama, Yeniden Çalışma-2 ve Kurulum süreçlerinde çalışan kişi sayısı
- Kodlama, Test, Yeniden Çalışma-1, Entegrasyon, Doğrulama, Yeniden Çalışma-2 ve Kurulum süreçlerindeki işlem süreleri

3.1.8. Dikkate Alınan Performans Ölçütleri

Çok projeleri yazılım firması aşağıdaki performans ölçütlerini dikkate almaktadır. Sistemde yapılacak iyileştirme çalışmaları bu performans ölçütlerini iyileştirmek üzere yapılacaktır.

1. Ortalama Tamamlanan Toplam Proje Sayısı
2. Ortalama Toplam Proje Tamamlanma Süresi
3. Ortalama Toplam Çalışan Maliyeti
4. Ortalama Toplam Kârlılık

3.2. Benzetim Modeli ve Benzetim Eniyilemesi

Son yıllarda, benzetim karmaşık sistemleri incelemek, anlamak ve kontrol etmek için daha önemli bir araç haline gelmiştir. Karmaşık sistemlerin doğasının karmaşıklığı ve belirsizliği, bilgi yönetimi, matematiksel modelleme, benzetim, karar destek sistemleri vb. dahil olmak üzere karmaşık çözüm araçları veya hibrit yaklaşımlar gerektirmektedir [36]. Bu çalışmada, “Yazılım Firması Sistemi Hakkında Bilgiler” başlığı altında ayrıntılarıyla tanımlanan sistemin mevcut durumu, “ARENA Benzetim Programı” yardımıyla Benzetim Modeli oluşturularak gösterilmiştir. Mevcut durumu inceledikten sonra oluşan darboğazları tespit ederek, dikkate alınan performans ölçütlerini eniyilemek amacıyla benzetim modelinin çıktıları analiz edilecektir. Çıktı analizinin devamında sistemin performans ölçütlerini en çok etkileyen ekipler ve çalışanlar tespit edilecek ve benzetim eniyilemesi yöntemi ile ortalama tamamlanan toplam proje sayısını en büyük, ortalama toplam proje tamamlanma süresini en küçük, ortalama toplam çalışan maliyetini en küçük, ortalama toplam kârlılığı en büyük yapan ekip sayısı önerileri her performans ölçütü için ayrı ayrı değerlendirilecektir. Bu değerlendirmelere “Bulgular ve Tartışma” bölümünde ayrıntılı ve sistematik olarak yer verilecektir. Bu bölümde ise, mevcut sistemi tanımlamak için ARENA Benzetim Programı aracılığıyla oluşturulan benzetim modeli detaylı bir şekilde açıklanacak ve Benzetim Eniyilemesi yöntemi hakkında bilgi verilecektir.

3.2.1. Sistemin ARENA Benzetim Programında Gösterilmesi

Bu bölümün amacı, belirlenen performans ölçütleri dahilinde proje geliştirme ayrıntılarının tahmin edilmesini sağlayan bir yazılım proje geliştirme sürecinin temel bir bölümünü simüle eden bir model uygulamaktır. Modelin amacı, proje yöneticilerine kontrol ve izleme konusunda yardımcı olmak ve aynı zamanda en iyi planlama alternatiflerini belirlemektir. Model kapsamı, yazılım geliştirme yaşam döngüsünün kodlama, test, yeniden çalışma, entegrasyon, doğrulama, yeniden çalışma, kurulum süreçlerini içeren “Gerçekleştirim” bölümünü kapsamaktadır.

Bu bölümün amacı, belirlenen performans ölçütleri dahilinde proje geliştirme ayrıntılarının tahmin edilmesini sağlayan bir yazılım proje geliştirme sürecinin temel bir

bölümünü simüle eden bir model uygulamaktır. Modelin amacı, proje yöneticilerine kontrol ve izleme konusunda yardımcı olmak ve aynı zamanda en iyi planlama alternatiflerini belirlemektir. Model kapsamı, yazılım geliştirme yaşam döngüsünün kodlama, test, yeniden çalışma, entegrasyon, doğrulama, yeniden çalışma, kurulum süreçlerini içeren “Gerçekleştirim” bölümünü kapsamaktadır. Sistemin ARENA Benzetim Programında girişleri ve kodları aşağıda başlıklar halinde verilmiştir.

a. Kaynaklar ve Kaynak Setleri

Sistemde Geliştirme Ekibi, Test Ekibi ve Kurulum Ekibi olmak üzere 3 farklı ekip bulunmaktadır. Bu ekipler A-B-C-D-E-F tecrübe seviyesinde çalışanlar içermektedir. Bu kapsamda, ARENA Benzetim Programı’nda kaynaklar “Resource” modülü içinde Şekil 3.1.’deki gibi tanımlanmıştır. Kaynakların gösterimi için Geliştirme Ekibi çalışanı için “Developer”, Test Ekibi çalışanı için “Tester”, Kurulum Ekibi çalışanı için “Deployer” kelimeleri kullanılmış ve tecrübe seviyeleri yanlarında belirtilmiştir. Örneğin, Developer_A Geliştirme Ekibinin A tecrübe seviyesindeki çalışanları anlamına gelmektedir.

Resource - Basic Process					
	Name	Type	Schedule Name	Schedule Rule	Busy / Hour
1	Developer_A	Based on Schedule	Sch Dev_A	Preempt	17
2	Developer_B	Based on Schedule	Sch Dev_B	Preempt	23
3	Developer_C	Based on Schedule	Sch Dev_C	Preempt	28
4	Developer_D	Based on Schedule	Sch Dev_D	Preempt	37
5	Developer_E	Based on Schedule	Sch Dev_E	Preempt	48
6	Developer_F	Based on Schedule	Sch Dev_F	Preempt	63
7	Tester_B	Based on Schedule	Sch Test_B	Preempt	23
8	Tester_C	Based on Schedule	Sch Test_C	Preempt	28
9	Tester_D	Based on Schedule	Sch Test_D	Preempt	37
10	Tester_E	Based on Schedule	Sch Test_E	Preempt	48
11	Deployer_B	Based on Schedule	Sch Dep_B	Preempt	23
12	Deployer_C	Based on Schedule	Sch Dep_C	Preempt	28
13	Deployer_D	Based on Schedule	Sch Dep_D	Preempt	37
14 ►	Deployer_F	Based on Schedule	Sch Dep_F	Preempt	63

Şekil 3.1. Kaynakların modelde gösterilmesi

Şekil 3.1.’deki görüldüğü gibi kaynaklar “Based on Schedule” yani tanımlanan çizelgelerine göre hareket edecek şekilde modellenmişlerdir. Çizelgeleri “Schedule (Çizelge)” modülünde, çalışma saatleri 08.00-17.00 saatleri arasında, 12.00-13.00 saatlerinde 1 saatlik öğle arası olacak şekilde tanımlanmıştır. Aynı şekle bakıldığında “Schedule Rule (Çizelge Kuralı)” kolonunda “Preempt (öncelik)” seçili olduğu görülmektedir. Bu gösterim, kaynak

seçiminde daha az tecrübeli olanın öncelikli olduğu kuralını göstermektedir. Aynı zamanda “Busy/Hour” kolonuna da her çalışanın Tablo 3.10.’da yer alan saatlik ücretlerinin girildiği görülmektedir.

Benzetim modelinde ekipler “Set” modülü içinde kaynak setleri olarak tanımlanmıştır. Bu kapsamda, Geliştirme Ekibi için “Developer Team”, Geliştirme Ekibinin Kompleks projelerdeki çalışanlarının bulunduğu ekip için “Developer Team_Complex”, Geliştirme Ekibinin Yalın projelerdeki çalışanlarının bulunduğu ekip için “Developer Team_Lean”; Test Ekibi için “Test Team”, Test Ekibinin Kompleks projelerdeki çalışanlarının bulunduğu ekip için “Test Team_Complex”; Kurulum Ekibi için “Deployer Team”, Kurulum Ekibinin Kompleks projelerdeki çalışanlarının bulunduğu ekip için “Deployer Team_Complex”, Kurulum Ekibinin Yalın projelerdeki çalışanlarının bulunduğu ekip için “Deployer Team_Lean” şeklinde tanımlama yapılmıştır. “Set” modülünün içeriği Şekil 3.2.’de yer almaktadır.

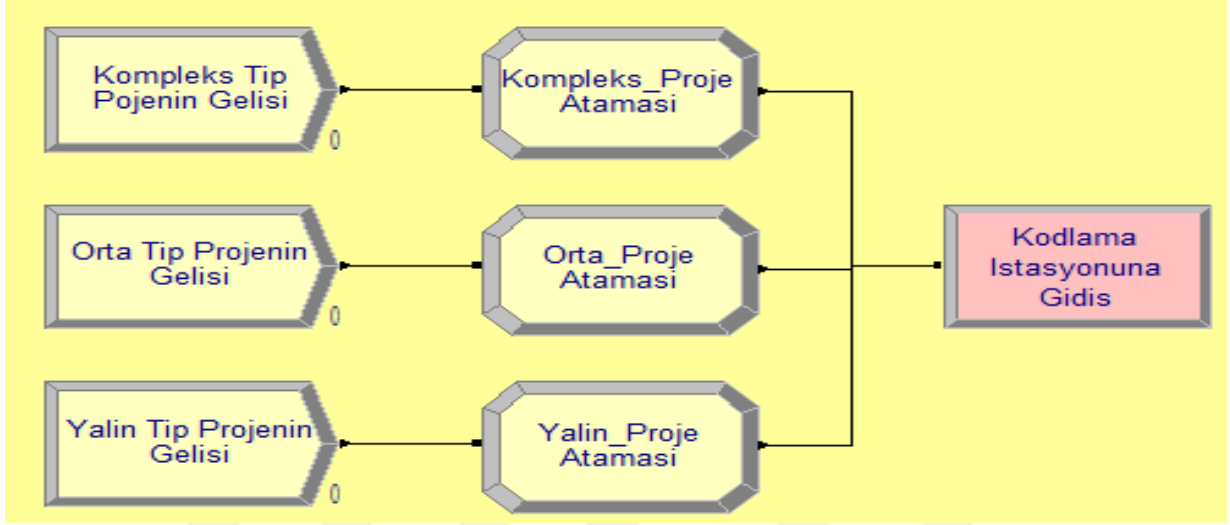
Set - Basic Process			
	Name	Type	Members
1	Developer Team	Resource	6 rows
2	Test Team	Resource	4 rows
3	Deployer Team	Resource	4 rows
4	Developer Team_Complex	Resource	3 rows
5	Developer Team_Lean	Resource	3 rows
6	Test Team_Complex	Resource	2 rows
7	Deployer Team_Complex	Resource	2 rows
8 ▶	Deployer Team_Lean	Resource	2 rows

Şekil 3.2. Ekiplerin modelde gösterilmesi

Şekil 3.2’de yer alan “Members” kolonundaki satır sayıları ilgili ekipte çalışabilecek ilgili kaynakları göstermektedir. Örneğin, Kurulum Ekibinin Yalın proje türünde görev alabilecek çalışanlar B ve C tecrübe seviyesindeki çalışanlar olduğu için “Deployer Team_Lean” ekibinin karşısındaki “Members” sütununda yazan 2 rows (satır) bu iki çalışan grubunu temsil etmektedir.

b. Projelerin Gelişleri

Kompleks, Orta ve Yalın olmak üzere 3 proje türünün gelişleri proje Tablo 3.9.’daki geliş tarihlerine göre “Create (Oluştur)” modülüne eklenmiştir. Benzetim modelinin görüntüsü Şekil 3.3.’te yer almaktadır.



Şekil 3.3. Kompleks, Orta ve Yalın Proje Türlerinin Gelişleri

Şekil 3.3.’te görüldüğü gibi projeler, projelerin aynı anda gelme ihtimalleri göz önünde bulundurularak 3 farklı “create” modülünde gelecek şekilde tanımlanmıştır.

Projelerin geliş tarihleri göz önünde bulundurularak Kompleks projenin gelişi 120 günde bir, Orta projenin gelişi 50 günde bir şeklinde modele girilmiştir. Yalın projenin yıllık geliş sayısının fazla olması sebebiyle ARENA Benzetim Programındaki “Input Analyzer” yardımıyla girdi dağılımı belirlenmiş ve üçgen dağılımda geldiği görülmüştür. Yalın projelerin geliş zamanlarına göre girdi analizi EK 1’de yer almaktadır. Projelerin geliş zamanları Şekil 3.4.’de gösterilmektedir.

Create - Basic Process									
	Name	Entity Type	Type	Value	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
1	Kompleks Tip Projenin Gelişi	proje_kompleks	Random (Expo)	120	1	Days	1	Infinite	0.0
2	Orta Tip Projenin Gelişi	proje_orta	Random (Expo)	50	1	Days	1	Infinite	0.0
3	Yalın Tip Projenin Gelişi	proje_yalin	Expression	90	TRIA(0.5,15.3,57.5)	Days	1	Infinite	0.0

Şekil 3.4. Kompleks, Orta ve Yalın Proje Türlerinin Geliş Zamanları

Şekil 3.3.’te görüldüğü gibi gelen projelere “Assign (Atama)” modülü ile özellikleri atanmıştır. Bu kapsamda modüllere girilen kodlamalar; Kompleks Proje türü için Şekil 3.5.’te, Orta Proje Türü için Şekil 3.6.’da ve Yalın Proje Türü için 3.7.’de yer almaktadır.

Assign

Name: Kompleks_Proje Atamasi

Assignments:

- Attribute, Project Type, 1
- Entity Picture, Picture.Red Page
- Attribute, ProjectValue, DISC(0.33, 15000000, 0.67, 17000000, 1, 20000000)
- Attribute, ArrivalTimeProject, TNDW
- Attribute, MaxTimePeriod, 3*12*22*9
- <End of list>

Add... Edit... Delete

OK Cancel Help

Şekil 3.5. Kompleks proje atama modülü

Şekil 3.5.’te görüldüğü gibi Kompleks proje; Proje Tipi 1 olarak tanımlanmıştır. Tablo 3.9.’daki proje bedelleri “ProjectValue” şeklinde girilmiş ve en fazla 3 yıl içinde tamamlanması şartının sağlanabilmesi için “MaxTimePeriod” şeklinde bir özellik atanarak süresiyle ilgili bilgi tutulması sağlanmıştır.

Assign

Name: Orta_Proje Atamasi

Assignments:

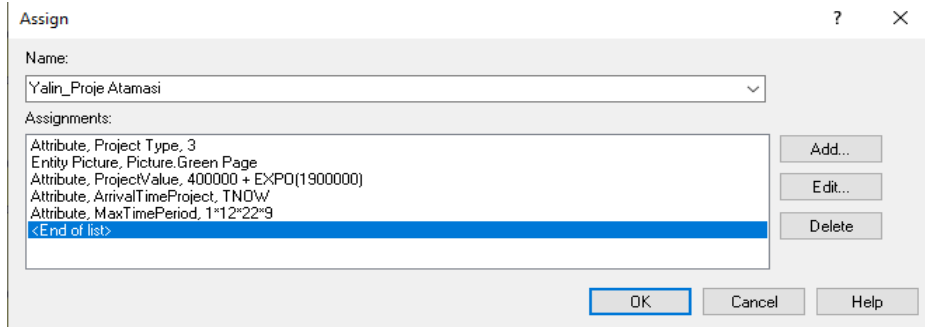
- Attribute, Project Type, 2
- Entity Picture, Picture.Yellow Page
- Attribute, ProjectValue, UNIF(7000000, 12000000)
- Attribute, ArrivalTimeProject, TNDW
- Attribute, MaxTimePeriod, 2*12*22*9
- <End of list>

Add... Edit... Delete

OK Cancel Help

Şekil 3.6. Orta proje atama modülü

Şekil 3.6.’da görüldüğü gibi Orta proje; Proje Tipi 2 olarak tanımlanmıştır. Tablo 3.9.’daki proje bedelleri “ProjectValue” şeklinde Uniform dağılıma göre girilmiş ve en fazla 2 yıl içinde tamamlanması şartının sağlanabilmesi için “MaxTimePeriod” şeklinde bir özellik atanarak süresiyle ilgili bilgi tutulması sağlanmıştır.



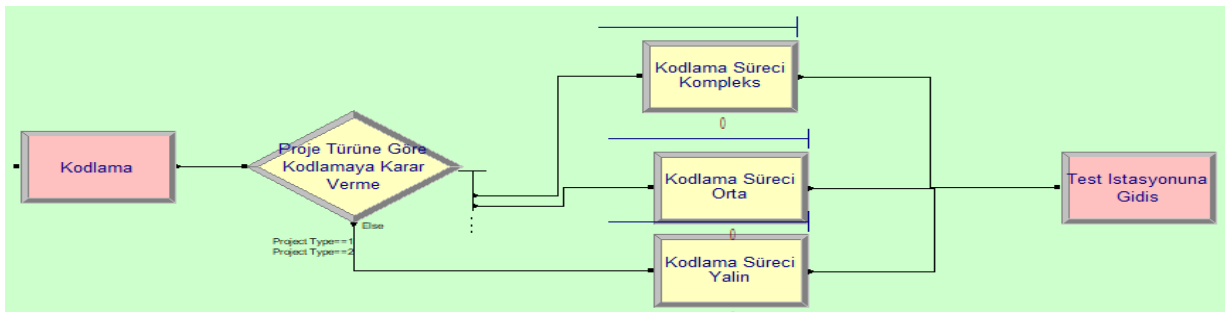
Şekil 3.7. Yalın proje atama modülü

Şekil 3.7.’de görüldüğü gibi Yalın proje; Proje Tipi 3 olarak tanımlanmıştır. Tablo 3.9.’daki proje bedelleri, Yalın projenin yıllık geliş sayısının fazla olması sebebiyle ARENA Input Analyzer yardımıyla analiz edilmiş ve girdi analizinin sonucunda ortaya çıkan dağılıma göre girilmiştir. Yalın projelerin proje bedeline göre girdi analizi EK 2’de yer almaktadır. En fazla 1 yıl içinde tamamlanması şartının sağlanabilmesi için “MaxTimePeriod” şeklinde bir özellik atanarak süresiyle ilgili bilgi tutulması sağlanmıştır.

Projelerin gelmesi ve ilgili atamalarının yapılmasıyla projeler, Kodlama istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

c. Kodlama İstasyonu

Sisteme gelen projeler öncelikle kodlama sürecini yaşamaktadır. ARENA Benzetim Programında “Gerçekleştirim” aşamasındaki her süreç bir istasyon olarak tanımlanmıştır. Şekil 3.8’de projelerin Kodlama istasyonundan Test istasyonuna gidişleri sırasında geçirdikleri süreçleri gösterilmiştir.



Şekil 3.8. Kodlama istasyonundan Test istasyonuna geçiş süreci

Şekil 3.8.’de görüldüğü gibi Kodlama İstasyonuna gelen projeler “Decision (Karar Verme)” modülü ile proje türlerine göre farklı kodlama süreçlerine yönlendirilmişlerdir. Her proje kendi özellikleri doğrultusunda izlemesi gereken süreci takip etmektedir. Süreç modüllerinde her proje türü için çalışabilecek çalışanlar kaynak olarak atanmıştır. Süreç modülleri içine girilen bilgiler Şekil 3.9.’de yer almaktadır.

Process - Basic Process									
	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Expression
13	Kodlama Süreci Kompleks	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	DeveloperTeamComplexServiceTimes_Coding(DeveloperIndexComplex)
14	Kodlama Süreci Orta	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	DeveloperTeamServiceTimes_Coding(DeveloperIndexModerate)
15	Kodlama Süreci Yalın	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	DeveloperTeamLeanServiceTimes_Coding(DeveloperIndexLean)

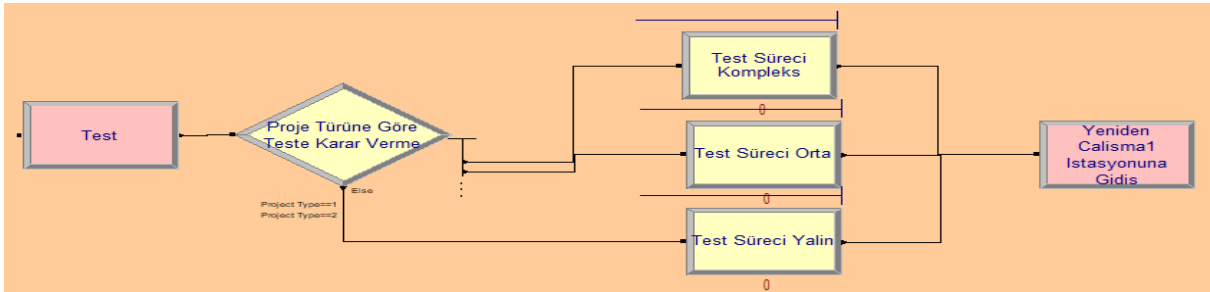
Şekil 3.9. Kodlama istasyonundan Test istasyonuna geçiş süreci

Şekil 3.9.’da görüldüğü gibi Proje Türlerine göre Kodlama süreçlerine ilgili ekiplerden seçim yapılmaktadır. “Expression” kolonunda yer alan bilgiler ilgili çalışanın kodlama süreci için çalışma süresini ifade etmektedir.

Kodlama süreci tamamlanan projeler, Test istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

d. Test İstasyonu

Şekil 3.10.’da görüldüğü gibi Test İstasyonuna gelen projeler “Decision (Karar Verme)” modülü ile proje türlerine göre farklı test süreçlerine yönlendirilmişlerdir. Her proje kendi özellikleri doğrultusunda izlemesi gereken süreci takip etmektedir. Süreç modüllerinde her proje türü için çalışabilecek çalışanlar kaynak olarak atanmıştır.

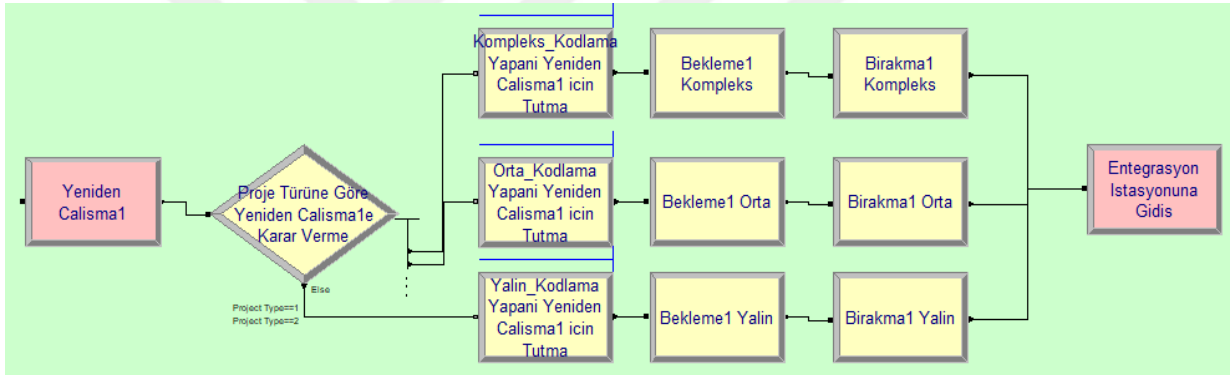


Şekil 3.10. Test istasyonundan Yeniden Çalışma 1 istasyonuna geçiş süreci

Şekil 3.10.'da görüldüğü gibi Test süreci tamamlanan projeler, Yeniden Çalışma 1 istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

e. Yeniden Çalışma 1 İstasyonu

Şekil 3.11.'de görüldüğü gibi Yeniden Çalışma 1 İstasyonuna gelen projeler “Decision (Karar Verme)” modülü ile proje türlerine göre farklı süreçlere yönlendirilmişlerdir. Bu istasyonun diğer istasyonlardan en büyük farkı, “Kodlama” hangi çalışan tarafından yapıldıysa “Yeniden Çalışma 1” sürecinin de aynı çalışan tarafından yapılıyor olmasıdır. “Seize-delay-release” modülleri içinde yapılan kaynak seçimiyle, kodlama yapan kişiyi yeniden çalışma için tekrar seçebilme özelliği modele eklenmiştir.

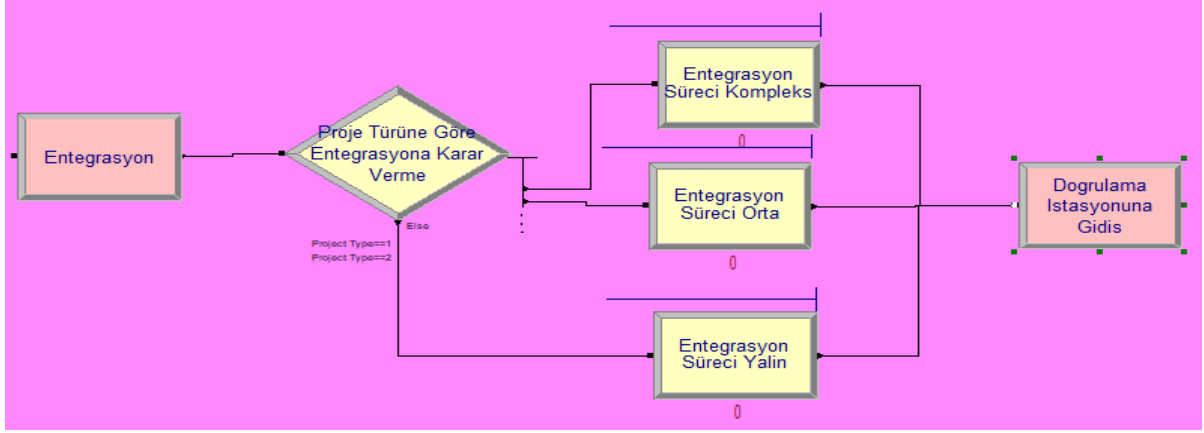


Şekil 3.11. Yeniden Çalışma 1 istasyonundan Entegrasyon istasyonuna geçiş süreci

Şekil 3.11.'de görüldüğü gibi Yeniden Çalışma 1 süreci tamamlanan projeler, Entegrasyon istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

f. Entegrasyon İstasyonu

Şekil 3.12.'de görüldüğü gibi Entegrasyon İstasyonuna gelen projeler “Decision (Karar Verme)” modülü ile proje türlerine göre farklı entegrasyon süreçlerine yönlendirilmişlerdir. Her proje kendi özellikleri doğrultusunda izlemesi gereken süreci takip etmektedir. Süreç modüllerinde her proje türü için çalışabilecek çalışanlar kaynak olarak atanmıştır.

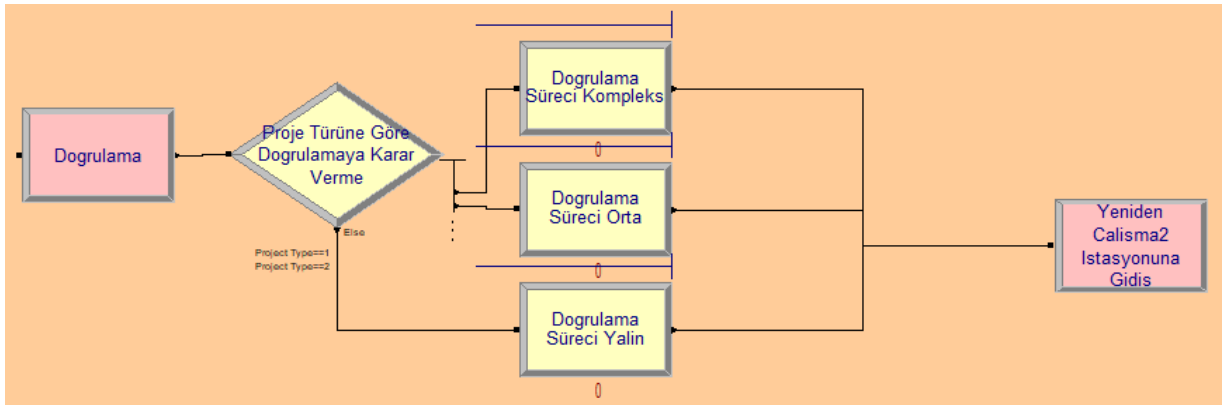


Şekil 3.12. Entegrasyon istasyonundan Doğrulama istasyonuna geçiş süreci

Şekil 3.12.’da görüldüğü gibi Entegrasyon süreci tamamlanan projeler, Doğrulama istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

g. Doğrulama İstasyonu

Şekil 3.13.’te görüldüğü gibi Doğrulama İstasyonuna gelen projeler “Decision (Karar Verme)” modülü ile proje türlerine göre farklı doğrulama süreçlerine yönlendirilmişlerdir. Her proje kendi özellikleri doğrultusunda izlemesi gereken süreci takip etmektedir. Süreç modüllerinde her proje türü için çalışabilecek çalışanlar kaynak olarak atanmıştır.

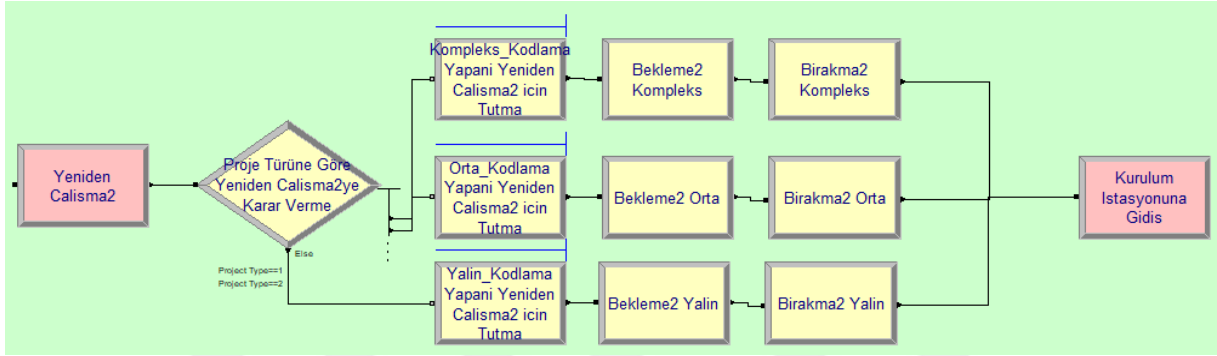


Şekil 3.13. Doğrulama istasyonundan Yeniden Çalışma 2 istasyonuna geçiş süreci

Şekil 3.13.’te görüldüğü gibi Doğrulama süreci tamamlanan projeler, Yeniden Çalışma 2 istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

h. Yeniden Çalışma 2 İstasyonu

Şekil 3.14.'te görüldüğü gibi Yeniden Çalışma 2 İstasyonuna gelen projeler “Decision (Karar Verme)” modülü ile proje türlerine göre farklı süreçlere yönlendirilmişlerdir. Bu istasyonun diğer istasyonlardan en büyük farkı, “Kodlama” ve “Yeniden Çalışma 2” süreçleri hangi çalışan tarafından yapıldıysa “Yeniden Çalışma 1” sürecinin de aynı çalışan tarafından yapılıyor olmasıdır. “Seize-delay-release” modülleri içinde yapılan kaynak seçimiyle, kodlama yapan kişiyi yeniden çalışma için tekrar seçebilme özelliği modele eklenmiştir.

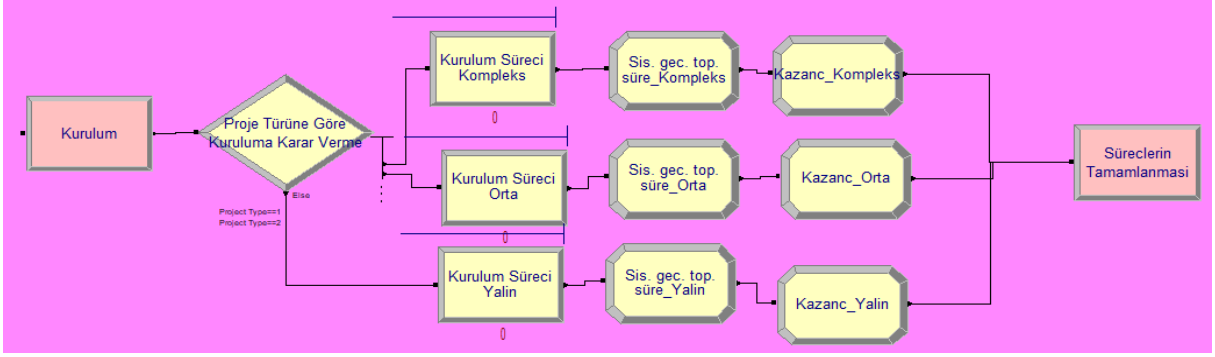


Şekil 3.14. Yeniden Çalışma 2 istasyonundan Kurulum istasyonuna geçiş süreci

Şekil 3.14.'te görüldüğü gibi Yeniden Çalışma 1 süreci tamamlanan projeler, Entegrasyon istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

i. Kurulum İstasyonu

Şekil 3.15.'te görüldüğü gibi Kurulum İstasyonuna gelen projeler “Decision (Karar Verme)” modülü ile proje türlerine göre farklı Kurulum süreçlerine yönlendirilmişlerdir. Her proje kendi özellikleri doğrultusunda izlemesi gereken süreci takip etmektedir. Süreç modüllerinde her proje türü için çalışabilecek çalışanlar kaynak olarak atanmıştır.

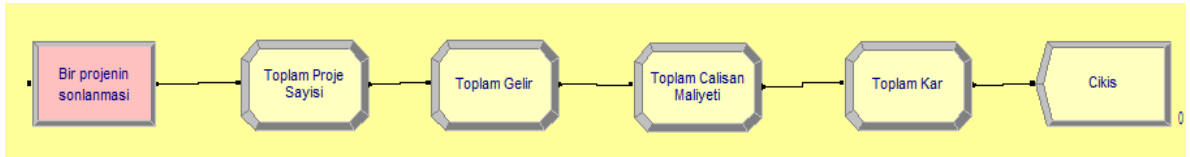


Şekil 3.15. Kurulum istasyonundan Süreçlerin Tamamlanması İstasyonuna geçiş süreci

Şekil 3.15.’te görüldüğü gibi Kurulum süreci tamamlanan projeler, Süreçlerin Tamamlanması istasyonuna gitmek üzere “Route” modülü ile rotalanmışlardır.

j. Süreçlerin Tamamlanması

Şekil 3.16’da görüldüğü gibi Süreçlerin Tamamlanması istasyonuna gelen projeler “Çıkış” modülüne ilerlemektedir. Bu süreçte, belirlenen performans ölçütlerine göre özelliklerin atandığı atama modülleri eklenmiştir. Bu atama modülleri aracılığıyla Toplam Proje Sayısı, Toplam Gelir, Toplam Çalışan Maliyeti ve Toplam Kâr çıktılarına ulaşabilmek için gerekli bilgiler modele tanımlanmıştır.



Şekil 3.16. Süreçlerin tamamlanması ve atama modülleri

Şekil 3.16’da görüldüğü gibi performans ölçütlerinin amaçlarının eniyilemesi için, proje türlerini ayırmaksızın her proje türü için toplam verilerin tanımlanması öncelikli olarak hedeflenmiştir. Çok projeli kaynak kısıtlı bir firmanın performans ölçütlerini genel olarak iyileştirmek hedeflendiği için her proje türü için ayrı veri tutulmamıştır.

k. Performans ölçütlerine göre istatistik modülünde tutulan çıktılar

Benzetim modelinin İstatistik modülü, belirlenen performans ölçütlerinin eniyilemesi için gerekli çıktıların tutulmasını sağlamaktadır. Bu kapsamda toplam proje sayısı, toplam tamamlanma süresi, toplam maliyet, toplam kâr bilgileri bu İstatistik modülü yardımıyla tutulmuştur. İstatistik modülü girdileri Şekil 3.17’de yer almaktadır.

Statistic - Advanced Process					
	Name	Type	Expression	Report Label	Output File
1	Statistic_Hour_PreferredNumOut	Output	EntitiesOut(proje_kompleks) + EntitiesOut(proje_orta) + EntitiesOut(proje_yalin)	Statistic_Hour_PreferredNumOut	hour_preferred_NumOut.DAT
2	Statistic_Hour_PreferredTAVG	Output	TAVG(proje_kompleks.TotalTime) + TAVG(proje_orta.TotalTime) + TAVG(proje_yalin.TotalTime)	Statistic_Hour_PreferredTAVG	hour_preferred_TAVG.DAT
3	Statistic_Hour_PreferredUseCost	Output	ResBusyCost(Deployer_B) + ResBusyCost(Deployer_C) + ResBusyCost(Deployer_D) + ResBusyCost(Deployer_F) + ResBusyCost(Developer_A) + ResBusyCost(Developer_B) + ResBusyCost(Developer_C) + ResBusyCost(Developer_D) + ResBusyCost(Developer_E) + ResBusyCost(Developer_F) + ResBusyCost(Tester_B) + ResBusyCost(Tester_C) + ResBusyCost(Tester_D) + ResBusyCost(Tester_E)	Statistic_Hour_PreferredUseCost	hour_preferred_UseCost.DAT
4	Statistic_Hour_PreferredProfit	Output	TotalProjectRevenue - Total Employee Cost	Statistic_Hour_PreferredProfit	hour_preferred_Profit.DAT

Şekil 3.17. Performans Ölçütlerine göre İstatistik modülünde tutulan çıktılar

Şekil 3.17.’de yer alan veriler incelendiğinde, tutulması hedeflenen istatistiklerin türünün “çıktı” olduğu görülmektedir. Bu çıktılar benzetim modelinin çıktıları içinde ayrıca yer almış ve bu çıktıların “Bulgular ve Tartışma” bölümünde ayrıntılı incelemesi yapılmıştır.

l. Modeli Çalıştırma

Kurulan benzetim modeli, 10 yıl için 5 tekrar sayısı ile çalıştırılmıştır. “Çalıştırma” modülü girdileri Şekil 3.18.’de yer almaktadır.

Run Setup

Run Speed | Run Control | Reports | Project Parameters

Replication Parameters | Array Sizes | Arena Visual Designer

Number of Replications: 5

Initialize Between Replications

☒ Statistics ☒ System

Start Date and Time: ☒ 1 Ocak 2021 Cuma 08:00:00

Warm-up Period: 0.0 Time Units: Hours

Replication Length: 3650 Time Units: Days

Hours Per Day: 9

Base Time Units: Hours

Terminating Condition:

Şekil 3.18. Çalıştırma modülü

Şekil 3.18.'de görüldüğü gibi model 1 Ocak 2021 saat 08.00 itibariyle 3650 gün yani 10 yıl için 5 tekrarla saat bazında çalıştırılmıştır. Ayrıca projelerin gecikmesi halinde günlük 100 TL zarar ve erken tesliminde günlük 50 TL gelir olması da model içinde tanımlanmıştır.

3.2.2. Benzetim Eniyilemesi

Benzetim modeli oluşturma, karmaşık sistemlerin dinamik davranışlarını analiz etmek ve değerlendirmek için kullanılan kullanışlı bir yöntem olarak bilinmektedir. Benzetim modelleri genel olarak, deneme yanılma yoluyla seçenek sayısı ve "ne-eğer" yanıtını vermektedir. Benzer şekilde, optimizasyon modelleri, belirli koşullar altında "en iyisi nedir" sorusuna yanıt vermektedir. Ancak, yalnızca benzetim veya optimizasyon teknikleriyle uygun bir çözüm elde etmek pek olası değildir ve bu nedenle benzetim ve optimizasyon modellerinin birlikte kullanılması esastır [51]. Benzetim tabanlı optimizasyon, sistem davranışını geçerli bir benzetim modeli kullanarak yorumladıktan sonra optimizasyon tekniklerini entegre etmektedir. Bu nedenle optimum çözüme ulaşmak için, benzetim modeli, problemi optimum çözüme yaklaşmak için yinelemeli olarak çalıştırılır. Tüm girdi değişkenleri için benzetim deneyleri çalıştırmak mümkünse, optimum çözüm verimli bir şekilde bulunabilmektedir. Ancak, uygulanması her zaman pratik bir yöntem değildir. Bu tür durumlarda amaç, tüm olası değerleri denemek yerine sınırlı bir girdi değişkenleri kümesi için modeli çalıştırmaktır. Bu da deney tasarımlarıyla ve meta-modellerle mümkün olmaktadır. Öncelikle benzetim modelinin çıktı analizleriyle modeli etkileyen bağımsız değişkenler (faktörler) belirlenir. Bu faktörlerin faktör düzeylerine göre deney tasarımları yapılır. Deney tasarımı yapmak için kullanılan farklı yöntemler bulunmaktadır ancak 2^k tam faktöriyel deneysel tasarımı en çok kullanılan yöntemlerden biridir [52]. Faktör sayısı "k" olarak gösterilmektedir. Modeli çalıştırma tekrar sayısı ve 2^k faktöriyel tasarım sayısı kadar çıktı elde edilmiş olur. Böylece çok fazla sayıda olan çıktılar, azalmış olmaktadır. Her performans ölçütü için oluşturulan deney tasarımları üzerinde senaryo denemeleri yapılmaktadır. Senaryo denemeleri sonucunda performans ölçütlerini eniyileyecek faktör düzeylerini bulmak için modelin özet modeli olan meta-modeller kurulmaktadır. Meta-modeller yardımcı model olarak da tanımlanabilmektedir. Karmaşık sistemlerin benzetim modelleri yerine kullanılmaktadır [53]. Oluşturulan meta-modeller, regresyon analizleri ile istatistiksel olarak analiz edilir. Analizler sonucunda ortaya

ıkan regresyon meta-model eřitlikleri, performans ltlerinin ama fonksiyonları olarak tanımlanmaktadır. Genel olarak, meta-modeldeki faktrler arası etkileřimlere Eřitlik 3.1'deki gibi bir polinom fonksiyonu belirlenebilmektedir.

$$y = \beta_0 + \sum_{n=1}^m \sum_{i=1}^k \beta_{ni} x_i^n + \sum_{n=1}^m \sum_{l=1}^m \sum_{i=1}^k \sum_{j=1}^k \beta_{nkij} x_i^n x_j^l \quad (3.1)$$

Burada “y” tahmini ıktı; x_i girdiler; k girdi sayısı; m polinomun sırası ve β regresyon katsayılarıdır. Bu gsterimle benzetim, orijinal benzetim sresinin yalnızca bir kısmını almakta ve bylece nemli hesaplama tasarruflarına izin vermektedir. Benzetim ve optimizasyon tekniğinin kombinasyonu, optimizasyon problemleri iin etkili bir ara saėlamaktadır [54]. Meta-modelin geerliliğinin kontrol edilmesinden sonra regresyon meta-modeli kısıtlarla birleřtirilerek matematiksel karar modeline dnřr ve matematiksel model czlerek ama fonksiyonun en iyi deėeri ve en iyi deėeri veren faktr dzeyleri belirlenmiř olur.

Meta-model oluřturmanın gvenilirliėi, Friedman vd. [55] tarafından yapılan bir alıřmayla test edilmiřtir.  farklı benzetim modeli kullanan 30 benzetim deneyinden oluřan bir alıřmaya dayanarak, benzetim meta-modelinin, benzetim sonrası analizde kullanılmak zere gvenilir ve geerli bir teknik olduėu ve muhtemelen temel aldıėı benzetim modeli kadar iyi olduėu sonucuna varmıřlardır.

Benzetim eniyilemesi yntemi kullanarak retim, hizmet sektr gibi farklı sektrlerin, farklı ihtiyalarının iyileřtirilmesine rnek olabilecek birok alıřma bulunmaktadır. rneėin, Dengiz vd. [56] tarafından yapılan bir alıřmada, Trkiye'deki bir otomotiv firmasının boyahane blm iin retim kontrol modeli geliřtirilmiřtir. Gerek bir vaka alıřması olarak, otomotiv řirketinin boyahane retim hattının optimum alıřma ayarı, benzetim eniyilemesi yaklařımı kullanılarak belirlenmiřtir. alıřmanın eniyileme ařamasında, deney tasarımı yapılarak senaryolar denenmiř, meta-model oluřturulmuř ve oklu doėrusal regresyon analizi ile sistemin kritik deėiřkenleri tanımlanmıřtır. Meta model doėrulanmıř ve sonu olarak benzetim eniyilemesi yaklařımından elde edilen bulgular, retkenliėini artırmak iin mevcut retim sisteminin yeniden tasarlanması ve kontrol iin řirkete destek olmuřtur.

Benzetim eniyilemesi yöntemiyle yapılan başka bir çalışma ise, sipariş üzerine tekerlekli sandalye üretimi yapan bir firmanın verimliliğini artırmak amacıyla yapılmıştır. Üretim planlama ve kontrol sistemi konularında iyileştirme yapmak isteyen firmaya, mevcut sisteme alternatif sistem önerilmiştir. Mevcut sistem ve alternatif sistem için ARENA Benzetim Programı ile benzetim modeli kurulmuş ve iki sistem karşılaştırılmıştır. Karşılaştırmaya göre alternatif sistemin firma için daha uygun olduğu sonucu ortaya çıkmıştır. Böylece önerilen sistemin en iyi çalışma koşullarını analiz etmek için benzetim eniyilemesi yöntemi uygulanmış ve sonuç olarak firmaya zaman, maliyet ve emek konularında daha iyi bir çalışma sistemi önerilmiş olmuştur [57].

Bu çalışmada benzetim eniyilemesi yöntemi, yazılım geliştirme projelerinin “Gerçekleştirim” sürecinin, dikkate alınan performans ölçütlerinin amaçları doğrultusunda iyileştirilmesi için kullanılacaktır. Uygulanan yöntem “Bulgular ve Tartışma” bölümünde ayrıntılarıyla yer almaktadır.

4. BULGULAR VE TARTIŞMA

Bu bölümde, çalışmanın “Yöntem” bölümünde açıklanan çok projeli kaynak kısıtlı bir yazılım firmasının mevcut sistemini temsil eden, ARENA Benzetim Programı’nda kurulan Benzetim modelinin 5 replikasyonla 10 yıl için çalıştırılmasıyla elde edilen bulgulara yer verilmiştir. Bunun için öncelikle benzetim modelinin doğruluğu ve geçerliliği kontrol edilmiştir. Kontrol sonrasında mevcut sistemin benzetim modelinin çıktıları tablolarda gösterilerek incelenmiştir. Mevcut sistemi, tamamlanan proje sayısını en büyükmek, projelerin toplam süresini, çalışan maliyetini en küçükmek ve toplam kârı en büyükmek şeklinde belirlenen dört temel performans ölçütü doğrultusunda ayrı ayrı ve aynı anda daha iyi bir noktaya taşıyabilmek amacıyla mevcut çıktılar analiz edilmiştir. Çıktı analizi süreci öncelikle, ortalamalar için güven aralığının belirlenmesiyle başlamış ve güven aralığında yer alan ortalamalar için çıktı analizi devam etmiştir. Belirlenen performans ölçütlerinin en iyiye ulaşması için Benzetim Eniyilemesi yöntemi kullanılan bu çalışmada, sistemin temel girdisi olan çalışan sayılarının sisteme olan etkilerini gözlemlemek amacıyla deney tasarımları kurulmuş ve projeyi etkileyen temel kaynakların sayısına yönelik farklı senaryolar denenmiştir. Senaryo denemelerinin sonuçları “Deney Tasarımları ve Senaryo Denemeleri” başlığı altında yer almaktadır. Her performans ölçütü için ayrı oluşturulan deney tasarımları üzerinde yapılan senaryo denemeleri sonucunda performans ölçütlerini eniyileyecek çalışan sayılarını bulmak için modelin özet modeli olan meta-modeller kurulmuştur. Kurulan meta-modeller ve çıktıları “Meta-Model Oluşturma” başlığı altında bulunmaktadır. Oluşturulan meta-modeller, çeşitli istatistiksel analizler gerçekleştirmek amacıyla kullanılan MİNİTAB İstatistik Programı yardımıyla istatistiksel olarak analiz edilmiş ve analizler sonucunda ortaya çıkan denklemler, performans ölçütlerinin amaç fonksiyonları olarak tanımlanmıştır. Tanımlanan bu amaç fonksiyonları, sistemin kapasite kısıtları ile birlikte matematiksel karar modellerinin oluşturulmasını sağlamıştır. Oluşturulan karar modelleri belirlenen performans ölçütlerinin hedefleri doğrultusunda Excel Çözücü yardımıyla çözülmüş, meta-model ve Benzetim modelinin çıktıları karşılaştırılmış, işlem adımları ve sonuçları “Karar Modeli ve Optimizasyon” başlığı altında paylaşılmıştır. Ayrıca tüm ölçütlerin aynı anda değerlendirildiği bir çok ölçütlü karar modeli geliştirilmiş ve Lingo programı yardımıyla çözülmüştür.

4.1. Benzetim Modelinin Doğruluğu ve Geçerliliği

Bu bölümde, mevcut sistemi temsil eden, ARENA Benzetim Programı yardımıyla kurulan benzetim modelinin doğruluğu ve geçerliliği kontrol edilmektedir.

4.1.1. Benzetim Modelinin Doğrulanması

ARENA Benzetim Programı üzerinde kurulan model 5 tekrar sayıyla 10 yıl için çalıştırılmış ve doğru kodlandığı kontrol edilmiştir. Kaynaklar ve kaynak setleri gibi girdi parametrelerinin, olayların oluş zamanlarının, süreçlerin işleyişinin modelin mantıksal yapısına uygun olduğu görülmüştür. Benzetim modelinin gerçek sistemi tam ve doğru bir şekilde temsil ettiğini göstermek için girdi parametrelerinde değişiklikler yapılmış ve çıktı değişimi farklı kişiler tarafından incelenmiştir. Böylece çalışmada kurulan model doğrulanmıştır.

4.1.2. Benzetim Modelinin Geçerliliği

Çalışılan sistemin gerçek çıktıları bilinmediği için hazırlanan modelle gerçek sistem arasında kıyaslama yapılamamış ve geçerliliği test edilememiştir.

4.2. Mevcut Sistemin Benzetim Modeli Çıktıları

“Yöntem” bölümünde ayrıntılı olarak tanımlanan mevcut sistemin verileri ARENA Benzetim Programı’na girilmiş ve model 5 tekrar sayısı ile 10 yıl için saat bazında çalıştırılmıştır. Modelin çalışma başlangıç günü 1 Ocak 2021 olarak belirlenmiştir. ARENA Benzetim modelinin çıktısı EK 3’te yer almaktadır.

Benzetim modelinin çıktıları, Ortalama Tamamlanan Toplam Proje Sayısı, Projelerin Ortalama Toplam Tamamlanma Süresi, Çalışanların Kullanım Oranları, Süreçlerin Kuyruklarında Bekleyen Ortalama İş Sayısı, Süreçlerin Kuyruklarındaki Ortalama Bekleme Süresi, Ortalama Toplam Çalışan Maliyeti, Ortalama Toplam Gelir ve Ortalama Toplam Kâr çıktıları temel alınarak incelenmiş ve bu kapsamda aşağıdaki çıktılar elde edilmiştir.

a. Ortalama Tamamlanan Toplam Proje Sayısı

Ortalama Tamamlanan Toplam Proje Sayısı = 54.6 olup proje türlerine göre dağılımları Tablo 4.1.'de yer almaktadır.

Tablo 4.1. Mevcut sistemdeki ortalama tamamlanan toplam proje sayısı

Proje Türü	Sisteme Gelen Proje Sayısı	Tamamlanan Proje Sayısı
Yalın	150,8	36,0
Orta	83,2	15,2
Kompleks	31,4	3,4
Toplam	265,4	54,6

Tablo 4.1.'de görüldüğü üzere, projelerin türlerine göre sisteme en çok gelen ve en çok sayıda tamamlanan proje türü Yalın proje türü olup, bunu sırasıyla Orta ve Kompleks proje türleri izlemektedir.

b. Projelerin Ortalama Toplam Tamamlanma Süresi

Projelerin Ortalama Toplam Tamamlanma Süresi = 55.672,08 saat olup proje türlerine göre dağılımları Tablo 4.2.'de yer almaktadır.

Tablo 4.2. Projelerin ortalama toplam tamamlanma süresi

Projelerin Ortalama Toplam Tamamlanma Süresi		
Yalın Proje Türü	Orta Proje Türü	Kompleks Proje Türü
14.153,7	18.146,4	23.372,0

Tablo 4.2.'de görüldüğü üzere projelerin türlerine göre ortalama tamamlanma süresi en çok olan proje türü Kompleks proje türü olup, bunu sırasıyla Orta ve Yalın proje türleri izlemektedir.

c. Çalışanların Kullanım Oranları

Geliştirme Ekibi, Test Ekibi ve Kurulum Ekibi çalışanlarının tecrübe seviyelerine göre projelerdeki kullanım oranları Tablo 4.3.'te yer almaktadır.

Tablo 4.3. Çalışanların Kullanım Oranları

Çalışanların Kullanım Oranları			
Çalışan Türü	İlgili Ekip		
	Geliştirme Ekibi	Test Ekibi	Kurulum Ekibi
A	88%		
B	78%	98%	85%
C	43%	97%	83%
D	51%	95%	83%
E	18%	84%	
F	8%		80%
Genel Ortalama	47,66%	93,5%	82,75%

Tablo 4.3.'te görüldüğü üzere genel kullanım oranı en yüksek olan ekip Test Ekibi olup, bunu sırasıyla Kurulum Ekibi ve Geliştirme Ekibi izlemektedir.

d. Süreçlerin Kuyruklarında Bekleyen Ortalama Proje Sayısı

Proje türüne göre Kodlama, Test, Yeniden Çalışma 1, Entegrasyon, Doğrulama, Yeniden Çalışma 2, Kurulum süreçlerinde bekleyen ortalama proje sayısı Tablo 4.4.'te yer almaktadır.

Tablo 4.4. Süreç Kuyruklarında Bekleyen Ortalama Proje Sayısı

Süreç Kuyruklarında Bekleyen Ortalama Proje Sayısı				
Süreçler	Yalın Proje	Orta Proje	Kompleks Proje	Genel Ortalama
Kodlama	0,02	0,00	0,00	0,01
Test	31,62	17,88	7,02	18,84
Yeniden Çalışma 1	1,48	0,66	0,40	0,85
Entegrasyon	0,70	0,17	0,18	0,35
Doğrulama	15,54	9,61	2,76	9,30
Yeniden Çalışma 2	0,86	0,35	0,18	0,46
Kurulum	0,45	0,08	0,68	0,40
Genel Ortalama	7,24	4,11	1,60	

Tablo 4.4.'te görüldüğü üzere süreç kuyruklarında bekleyen ortalama proje sayısı, Yalın proje türünde en fazla olup bunu Orta ve Kompleks proje türü izlemektedir. Ayrıca, tüm proje türleri için Test ve Doğrulama süreçlerinde bekleyen ortalama proje sayısı en fazladır. Yalın ve Orta proje türlerinin bu süreçlerinde bekleyen proje sayısı, Kompleks proje türünün bu süreçlerinde bekleyen proje sayısından daha fazladır.

e. Süreçlerin Kuyruklarındaki Ortalama Bekleme Süresi

Proje türüne göre Kodlama, Test, Yeniden Çalışma 1, Entegrasyon, Doğrulama, Yeniden Çalışma 2, Kurulum süreçlerindeki ortalama bekleme süresi Tablo 4.5.'te yer almaktadır.

Tablo 4.5. Süreç Kuyruklarındaki Ortalama Bekleme Süresi

Süreç Kuyruklarındaki Ortalama Bekleme Süresi				
Süreçler	Yalın	Orta	Kompleks	Genel Ortalama
Kodlama	4,3	0,1	1,0	1,8
Test	6.792,5	6.859,4	6.603,2	6.751,7
Yeniden Çalışma 1	605,7	503,0	1.059,0	722,56
Entegrasyon	303,7	141,6	476,9	307,4
Doğrulama	7.162,2	8.440,1	8.516,9	8.039,73
Yeniden Çalışma 2	726,8	704,4	1.249,3	893,5
Kurulum	399,8	173,2	4.225,0	1.599,33
Genel Ortalama	2.285	2.403,11	3.161,61	

Tablo 4.5.'te görüldüğü üzere süreç kuyruklarındaki ortalama bekleme süresi, Kompleks proje türünde en fazla olup bunu Orta ve Yalın proje türü izlemektedir. Bu proje türlerinin süreçlerindeki işlem sürelerinin birbirinden farklı olması bu farkın temel sebebidir. Ayrıca, Test, Doğrulama ve Kurulum süreçlerindeki ortalama bekleme süresi en fazladır.

f. Ortalama Toplam Çalışan Maliyeti

5 replikasyon sayısı ile 10 yıllık Ortalama Toplam Çalışan Maliyeti 20.458.262,08 Türk Lirası (TL) olup ekip ve çalışan türlerine göre dağılımları aşağıdaki Tablo 4.6.'da yer almaktadır. Toplam Çalışan Maliyeti = Her çalışanın çalıştığı süre x saatlik ücreti olarak hesaplanmaktadır.

Tablo 4.6. Ortalama Toplam Çalışan Maliyeti (TL)

Ortalama Toplam Çalışan Maliyeti (TL)				
Çalışan Türü	İlgili Ekip			Genel Ortalama
	Geliştirme Ekibi	Test Ekibi	Kurulum Ekibi	
A	943.020,37			943.020,37
B	1.680.851,17	1.381.272,30	1.216.030,20	1.426.051,22
C	1.676.157,01	1.656.877,73	1.489.171,70	1.607.402,15
D	1.555.624,51	3.303.426,00	1.921.704,41	2.260.251,64
E	473.505,17	1.326.579,99		900.042,58
F	172.352,33		1.661.689,19	917.020,76
Genel Ortalama	1.083.585,09	1.917.039,01	1.572.148,88	

Tablo 4.6.'da görüldüğü üzere ortalama çalışan maliyeti en çok olan ekip Test Ekibi olup bunu Kurulum Ekibi ve Geliştirme Ekibi izlemektedir. Çalışan maliyetinin en yüksek olduğu tecrübe seviyesinin ise D, C, B, A, F, E şeklinde sıralandığı görülmektedir.

g. Ortalama Toplam Gelir

Ortalama Toplam Gelir = Tamamlanan Projelerin Proje Bedeli + Erken Bitirme Günlük Geliri olup 361.509.547,912 TL'dir. 5 replikasyon sayısı ile 10 yıl için çalıştırılan benzetim modelinin sonucunda çıkan değerdir.

h. Ortalama Toplam Kâr

Ortalama Toplam Kâr = Toplam Gelir – Toplam Çalışan Maliyeti – Gecikme Kaynaklı Zarar olup 341.372.420,829 TL'dir. 5 replikasyon sayısı ile 10 yıl için çalıştırılan benzetim modelinin sonucunda çıkan değerdir..

Mevcut sistemin çıktıları, Ortalama Tamamlanan Toplam Proje Sayısı, Projelerin Ortalama Toplam Tamamlanma Süresi, Çalışanların Kullanım Oranları, Süreçlerin Kuyruklarında Bekleyen Ortalama İş Sayısı, Süreçlerin Kuyruklarındaki Ortalama Bekleme Süresi, Ortalama Toplam Çalışan Maliyeti, Ortalama Toplam Gelir ve Ortalama Toplam Kâr çıktıları temel alınarak incelenmiştir. Mevcut sistemin temel alınan performans ölçütleri doğrultusunda iyileştirilebilmesi için mevcut çıktılarının çıktı analizi yapılmaktadır.

4.3. Çıktı Analizi

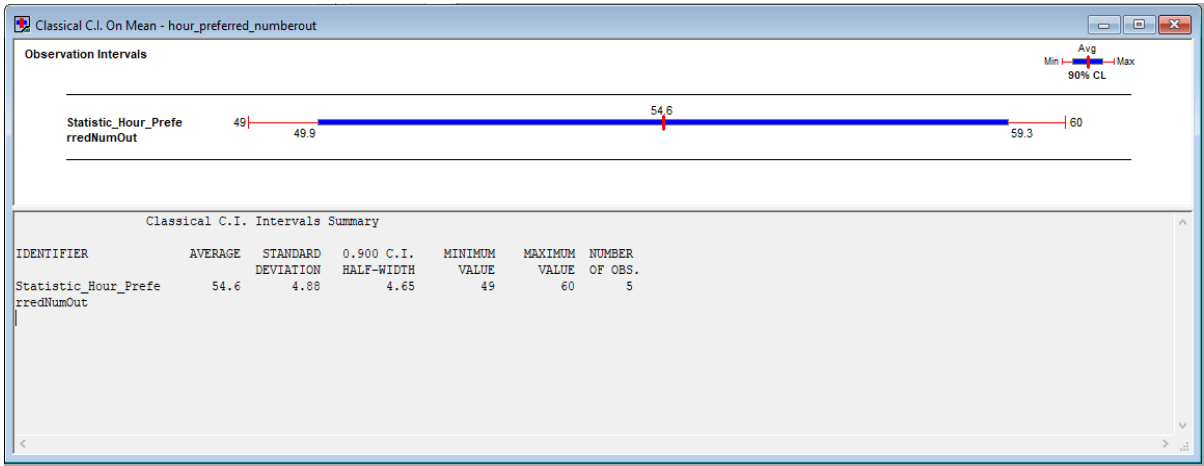
Mevcut sistem 9 saatlik mesai saatleri içermektedir ve benzetim modeli 10 yıl için çalıştırılmıştır, çalışmanın tamamlanacağı süre ve süreçlerin tamamlanma süreleri belirlidir. Bu nedenle bitişli çıktı analizi yapılır. Öncelikle performans ölçütlerine göre ortalamalar için güven aralığı kontrol edilir.

4.3.1. Ortalamalar için Güven Aralığı

ARENA Benzetim Programı'nda çıktı analizi yapabilmek için kullanılan "Output Analyzer" aracı ile; ortalama tamamlanan toplam proje sayısı, projelerin ortalama toplam tamamlanma süresi, ortalama toplam çalışan maliyeti ve ortalama toplam kâr çıktıları göz önünde bulundurularak 4 temel çıktı için çıktı analizi yapılmıştır. Sonuçları aşağıda ayrı ayrı açıklanmış ve ARENA Output Analyzer'dan alınan ekran fotoğrafları ilgili başlıklar altına eklenmiştir.

a. Ortalama Tamamlanan Toplam Proje Sayısına Göre Güven Aralığı

Ortalama Tamamlanan Toplam Proje Sayısına göre "ARENA Output Analyzer" yardımıyla belirlenen güven aralığı Şekil 4.1'de gösterilmiştir.



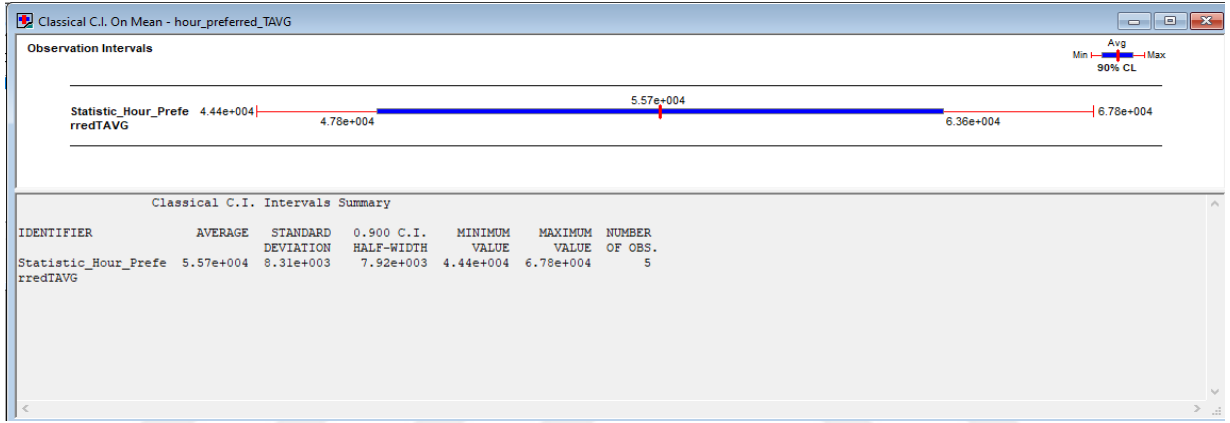
Şekil 4.1. Mevcut sistemde ortalama tamamlanan toplam proje sayısına göre güven aralığı

Şekil 4.1.'de yer alan Ortalama Tamamlanan Toplam Proje Sayısına göre güven aralığına bakıldığında, mevcut sistemde %90 güvenlik düzeyinde ortalama 49,9 ile 59,3 adet

arasında tamamlanan proje olacağı görülmektedir. Mevcut sistemdeki tamamlanan toplam proje sayısı ortalaması 54,6 olup güven aralığı içinde yer almaktadır.

b. Projelerin Ortalama Toplam Tamamlanma Süresine Göre Güven Aralığı

Projelerin Ortalama Toplam Tamamlanma Süresine göre “ARENA Output Analyzer” yardımıyla belirlenen güven aralığı Şekil 4.2’de gösterilmiştir.

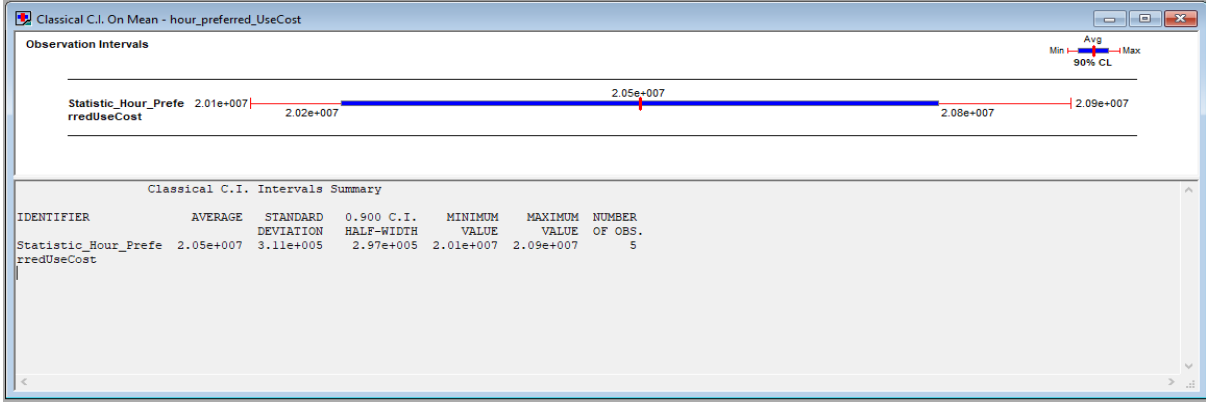


Şekil 4.2. Mevcut sistemde projelerin ortalama toplam tamamlanma süresine göre güven aralığı

Şekil 4.2.’de yer alan projelerin ortalama toplam tamamlanma süresine göre güven aralığına bakıldığında, mevcut sistemdeki projelerin ortalama toplam tamamlanma süresinin, %90 güvenlik düzeyinde güven aralığı içinde yer aldığı görülmektedir.

c. Ortalama Toplam Çalışan Maliyetine Göre Güven Aralığı

Ortalama toplam çalışan maliyetine göre “ARENA Output Analyzer” yardımıyla belirlenen güven aralığı Şekil 4.3’te gösterilmiştir.

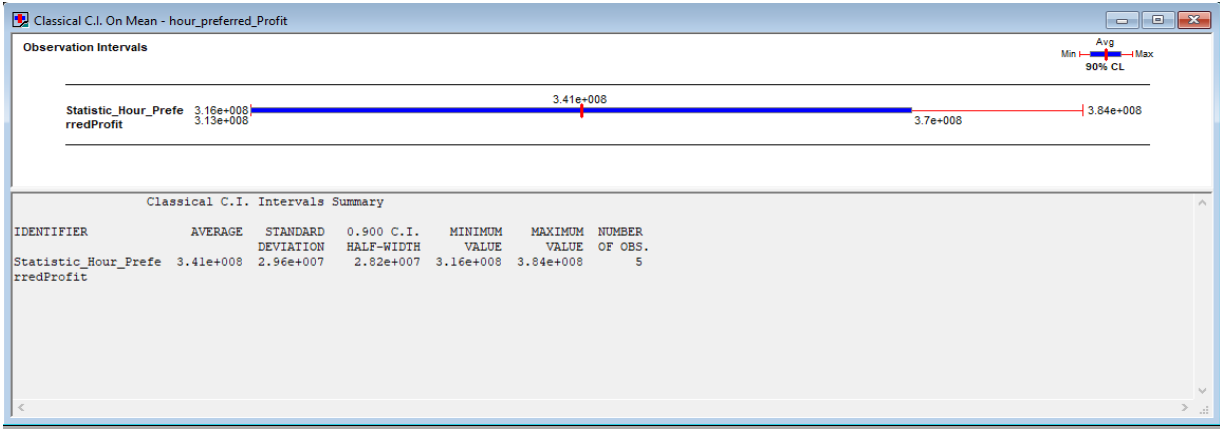


Şekil 4.3. Mevcut sistemde ortalama toplam çalışan maliyetine göre güven aralığı

Şekil 4.3.'te yer alan, ortalama toplam çalışan maliyetine göre güven aralığına bakıldığında, mevcut sistemdeki ortalama toplam çalışan maliyetinin, %90 güvenlik düzeyinde güven aralığı içinde yer aldığı görülmektedir.

d. Ortalama Toplam Kâra Göre Güven Aralığı

Ortalama toplam kâra göre “ARENA Output Analyzer” yardımıyla belirlenen güven aralığı Şekil 4.4’te gösterilmiştir.



Şekil 4.4. Mevcut sistemde ortalama toplam kâra göre güven aralığı

Şekil 4.4.'te yer alan, ortalama toplam kâra göre güven aralığına bakıldığında, mevcut sistemdeki ortalama toplam kârın, %90 güvenlik düzeyinde güven aralığı içinde yer aldığı görülmektedir.

Tüm performans ölçütleri için analiz edilen güven aralıkları ayrı ayrı incelendiğinde kurulan benzetim modelinin temel alınan dört farklı çıktısının, %90 güvenlik düzeyinde güven aralığı içinde yer aldığı görülmektedir.

Modelin çıktılarını incelediğimizde, proje çalışanlarının kullanım oranlarının genellikle %70'in üzerinde ve yüksek olduğu görülmektedir. Genellikle Kompleks ve Orta projelerde görev alan D-E-F tecrübe seviyesindeki çalışanların projelerde çalışma oranlarının, diğer çalışanlara kıyasla daha düşük olduğu, çıktılarda gözlemlenebilmektedir. E ve F tecrübe seviyesindeki çalışanların aynı zamanda yönetim görevlerinin de olması sebebiyle firma içinde farklı görev ve sorumlukları da olabilmektedir. Bu nedenle E ve F tecrübe seviyesindeki çalışanların tüm projelerde çok yoğun bir şekilde çalışmaları firma için öncelikli beklenti değildir. Firmanın asıl beklentisi; en az maliyetle, projeleri zamanında ya da zamanından önce tamamlayarak daha yüksek kâr elde edebilmek ve daha fazla projeyi tamamlayarak alanında prestij sağlamaktır. Öte yandan, proje maliyetlerinin düşük tutulması ve kârlılığın yüksek olması hedefleri ile projelerin projeleri aksatmadan tamamlayabilecek daha az maaşa sahip, daha az tecrübeli çalışanlar tarafından yapılmasına öncelik veriliyor olması, A-B-C tecrübe seviyesindeki çalışanların daha çok projede görev almasına sebep olmaktadır. Dolayısıyla bu tecrübe seviyelerine sahip çalışanların projelerdeki iş yükleri oldukça yüksek olmaktadır. Projelerdeki iş yüklerinin fazla olmasıyla birlikte, A-B-C tecrübe seviyesindeki çalışanların, görev aldıkları projelerde kazandıkları tecrübenin dışında kazanmaları gereken başka tecrübeler de bulunmaktadır. Bu çalışanların başka eğitimlere ve kişisel gelişimlerine zaman ayırabilmeleri de ayrıca önemlidir. Bu nedenle ideal çalışma ortamını sağlamak için firmada iş paylaşımı yapılırken, çalışanların iş yoğunluklarının karar vericiler tarafından gözetilmesi ve iş yoğunluğu çok yüksek olan çalışanların işlerini kolaylaştırabilecek çözümlerin oluşturulması gerekli olmaktadır. Karar vericilere destek olmak amacıyla bu çalışmada kurulan benzetim modeli yardımıyla, karar vericiler dikkate aldıkları performans ölçütlerini eniyilemek için senaryolar oluşturabilir ve senaryoların sonuçlarına göre yeni kararlar alabilirler.

Benzetim modelinin diğer bir çıktısı olan süreçlerin kuyruklarında bekleyen ortalama proje sayısı da daha yoğun olan süreçlerle ilgili bilgi vermektedir. Bu kapsamda Test

Ekibindeki çalışanların görev aldığı Test ve Doğrulama süreçlerinde kuyrukta bekleyen proje sayısının en fazla olduğu görülmektedir. Özellikle Yalın ve Orta proje türlerinde bu sayının daha çok olduğu söylenebilmektedir. Test Ekibindeki çalışan sayısının diğer ekiplerdeki çalışan sayısına kıyasla daha az olması da işlerin aksamasına ve çalışanlarının yoğunluklarının artmasına sebep olmaktadır. Proje türleri karşılaştırıldığında ise bekleyen proje sayısının en çok, Yalın projelerde daha sonra Orta projelerde ve en son da Kompleks projelerde olduğu çıktılarından anlaşılabilmektedir. Diğer proje türleriyle karşılaştırıldığında, Yalın Proje Türünün Entegrasyon sürecinde de bekleyen iş sayısının diğer proje türlerinin aynı sürecine kıyasla daha fazla olduğu fark edilmektedir. Çalışan kullanım oranları göz önünde bulundurulduğunda, Yalın projelerde görev alan B ve C tecrübe seviyesindeki Kurulum ekibi çalışanlarının iş yüklerinin fazla olması da Entegrasyon sürecindeki bekleyen iş sayısının fazla olması çıktısını desteklemektedir.

Oluşturulan benzetim modelinin çıktıları süreçlerin kuyruklarındaki bekleme süresi açısından da incelendiğinde en fazla bekleme süresinin Test Ekibinin görev aldığı Test ve Doğrulama süreçlerinde olduğu görülmektedir.

Benzetim modelinin temel çıktıları genel olarak değerlendirildiğinde; mevcut sistemde en çok iş yoğunluğuna sahip ekibin Test Ekibi, daha sonra Kurulum Ekibi ve en son Geliştirme Ekibi olduğu görülmektedir. Çalışan kullanım oranları çıktısı, süreçlerin kuyruklarında bekleyen proje sayısı ve süreçlerin kuyruklarındaki bekleme süresi çıktılarıyla birlikte ele alındığında, projelerin tamamlanmasını etkileyen temel ekibin Test ve Kurulum ekipleri olduğu ve bu ekiplerin de Yalın projelerde görev alan çalışanlarının projelerdeki bekleme sürelerinde en etkili olan çalışanlar olduğu sonucu görülmektedir. Bu nedenle, Test ve Kurulum ekiplerindeki B ve C tecrübe seviyesindeki çalışanların sayılarının temel kaynaklar olduğu ve sistemden daha çok proje çıkmasında etkili olduğu söylenebilmektedir. Bu çalışmanın devam eden kapsamında, Test ve Kurulum Ekiplerinin B ve C tecrübe seviyesindeki çalışanları üzerinde durulacak ve kritik olan bu çalışanlar üzerinden değerlendirmeler yapılarak, tamamlanan proje sayısını en büyükmek, projelerin toplam süresini ve çalışan maliyetini en küçükmek, toplam kârı en büyükmek şeklinde belirlenen performans ölçütlerinin en iyiye ulaşması için benzetim eniyilemesi yapılmaktadır. Benzetim

Eniyilemesi yapmak için öncelikle benzetim modeli çıktıları üzerinde senaryo denemeleri gerçekleştirilmektedir. Senaryo denemelerinin gerçekleştirilebilmesi için deney tasarımı oluşturulmaktadır. Oluşturulan deney tasarımı üzerinde yapılan deneme senaryolar incelenmekte ve denemeler sonucunda en iyi ekip sayısını bulmak için meta-model kurularak, meta-modelin çıktıları değerlendirilmektedir. Kurulan meta-modeller, karar modeli olarak belirlenerek, amaçlarına uygun bir şekilde çözülerek amaca göre en iyi ekip sayısı bulunmaktadır.

4.4. Deney Tasarımı ve Meta-Model Kullanılarak Benzetim Eniyilemesi

Benzetim eniyilemesi yöntemi, oluşturulan benzetim modeline göre en iyi tasarımı elde etmek için uygulanmaktadır. Belirlenen performans ölçütünün amaçlarına göre en iyi çıktıların elde edilebilmesi için gereken en iyi girdi setinin belirlenmesine yardımcı olmaktadır. Bu bağlamda, belirlenen performans ölçütlerine göre meta-modeller geliştirilerek benzetim eniyilemesi yapılmıştır.

4.4.1. Deney Tasarımı ve Senaryo Denemeleri

ARENA Benzetim Programı yardımıyla mevcut sistemin Tamamlanan Toplam Proje Sayısı, Toplam Proje Tamamlanma Süresi, Çalışanların Kullanım Oranları, Süreçlerin Kuyruklarında Bekleyen Ortalama Proje Sayısı ve Bekleme Süresi, Ortalama Çalışan Maliyet, Ortalama Gelir ve Ortalama Kâr gibi çıktıları incelendiğinde Test ve Kurulum Ekiplerinin B ve C tecrübe seviyesindeki kaynaklarının projelerin tamamlanmasını etkileyen temel kaynaklar olduğu görülmektedir. Tamamlanan proje sayısını ve toplam kârı en büyükmek, projelerin toplam süresini ve çalışan maliyetini en küçükmek şeklinde belirlenen performans ölçütlerinin en iyiye ulaşması için ilgili ekiplerin B ve C tecrübe seviyesindeki çalışan sayıları faktör olarak belirtilmiştir. Performans ölçütlerinin amaçlarının sağlanabilmesi için, bu faktörlerde sınırsız sayıda yeni alım yapılamayacağı için, projelerin ilgili süreçlerinin tamamlanabilmesi amacıyla en az ve en çok sayıda kaç çalışana gerek duyulduğu, Tablo 4.7.'de faktör düzeyleri olarak gösterilmektedir. Faktör düzeylerinden alt düzey, süreçlerin ve projelerin tamamlanması için gerekli en az çalışan sayısını; üst düzey ise süreçlerin ve

projelerin tamamlanması için, organizasyonun maddi ve fiziki koşulları göz önünde bulundurularak çalışabilecek en fazla çalışan sayısını ifade etmektedir.

Tablo 4.7. Faktörler ve Faktör Düzeyleri

Faktörler	Faktör Düzeyleri	
	Alt Düzey	Üst Düzey
Test Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	2	5
Test Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı	2	4
Kurulum Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	2	5
Kurulum Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı	2	4

Tablo 4.7.'de yer alan faktörlerin faktör düzeylerine bakıldığında Test Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı için 4, Test Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı için 3, Kurulum Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı için 4, Kurulum Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı için 4 farklı olasılığın olduğu görülmektedir. Bu olasılıkların birbirleriyle kombinasyonuna bakıldığında ise, performans ölçütlerini en iyiye ulaştırabilmek için $4 \times 3 \times 4 \times 3 = 144$ çıktının ayrı ayrı değerlendirilmesi ve en iyiye veren sonucun izlenmesi gerekmektedir. Fakat bu çalışmanın tek tek yapılması vakit almakla beraber karışıklıkların ve karar vermeyi zorlaştıran etmenlerin ortaya çıkmasına da sebep olabilmektedir. O nedenle faktöriyel deney tasarımı yapılarak çok sayıda olan bu verilerin daha az sayıda veriye dönüştürülmesi sağlanmaktadır. Bu çalışmadaki deney tasarımında, düzey sayısı 2 olduğu için 2^k faktöriyel tasarım yapılmaktadır.

k: faktör sayısı iken $k=4$ olduğu için Kombinasyon Sayısı $= 2^4 = 16$ olarak bulunur. Her bir deney için en az 5 deneme yapıldığı için elimizde en az $16 \times 5 = 80$ çıktı bulunmaktadır. Bu çalışmada, 4 farklı performans ölçütüne göre deney tasarımları yapılmıştır. Aşağıdaki ilgili başlıklar altında gösterilen deney tasarımı tablolarında yazan çıktılar, ARENA Benzetim Programı'nın süreç analizi yapmaya yarayan "Process Analyzer" aracı ile girdiler ve replikasyon sayıları değiştirilerek elde edilmiştir. "Process Analyzer" aracı yardımıyla elde edilen sonuçlar EK 4'te verilmiş olup özet tabloları Tablo 4.8., Tablo 4.9., Tablo 4.10. ve Tablo 4.11.'de gösterilmiştir.

a. Ortalama Tamamlanan Toplam Proje Sayısı Performans Ölçütüne Göre Deney Tasarımı

Ortalama tamamlanan toplam proje sayısına için hazırlanan 2^4 faktöriyel deney tasarımı Tablo 4.8.'de yer almaktadır.

Tablo 4.8. Ortalama tamamlanan toplam proje sayısı için faktöriyel deney tasarımı

Senaryo	Ortalama Tamamlanan Proje Sayısı (Adet)								
	Faktörler				Replikasyon Sayısı				
	Test Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	Test Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı	1	2	3	4	5
1	2	2	2	2	60,0	55,0	56,0	56,0	54,6
2	2	2	2	4	70,0	60,5	59,0	62,3	62,2
3	2	2	5	2	70,0	63,5	60,7	61,5	60,4
4	2	2	5	4	70,0	62,5	60,3	60,3	59,6
5	2	4	2	2	60,0	55,0	56,0	56,0	54,6
6	2	4	2	4	70,0	73,0	66,3	64,3	65,6
7	2	4	5	2	70,0	63,5	60,7	61,5	60,4
8	2	4	5	4	70,0	62,5	60,3	60,3	59,6
9	5	2	2	2	58,0	52,5	55,0	54,3	56,2
10	5	2	2	4	82,0	79,5	77,0	75,8	77,2
11	5	2	5	2	79,0	76,5	77,0	77,5	78,8
12	5	2	5	4	80,0	73,0	75,3	74,8	77,6
13	5	4	2	2	58,0	52,5	55,0	54,3	56,2
14	5	4	2	4	82,0	79,5	77,0	75,8	77,2
15	5	4	5	2	79,0	76,5	77,0	77,5	78,8
16	5	4	5	4	80,0	73,0	75,3	74,8	77,6

Tablo 4.8.'de görüldüğü üzere 4 faktörün sayılarında değişiklik yapılarak 16 farklı senaryo denenmiş ve 5 replikasyon sayısı ile toplamda 80 adet çıktı elde edilmiştir.

b. Projelerin Ortalama Toplam Tamamlama Süresi Performans Ölçütüne Göre Deney Tasarımı

Projelerin ortalama toplam tamamlama süresi için hazırlanan 2^4 faktöriyel deney tasarımı Tablo 4.9.'da yer almaktadır.

Tablo 4.9. Projelerin ortalama toplam tamamlama süresi için faktöriyel deney tasarımı

Senaryo	Projelerin Ortalama Toplam Tamamlama Süresi (saat)								
	Faktörler				Replikasyon Sayısı				
	Test Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	Test Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi B Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi C Tecrübe Seviyesindeki Çalışan Sayısı	1	2	3	4	5
1	2	2	2	2	44.362,2	56.074,1	55.646,3	55.513,7	55.672,1
2	2	2	2	4	42.817,6	49.119,0	50.739,1	50.825,7	51.369,0
3	2	2	5	2	46.924,7	53.059,4	51.522,5	51.222,7	51.529,4
4	2	2	5	4	42.993,3	50.808,7	51.229,5	51.225,7	51.324,6
5	2	4	2	2	44.362,2	56.074,1	55.646,3	55.513,7	55.672,1
6	2	4	2	4	42.817,6	51.867,7	44.037,1	40.392,6	43.925,7
7	2	4	5	2	46.924,7	53.059,4	51.522,5	51.222,7	51.529,4
8	2	4	5	4	42.993,3	50.808,7	51.229,5	51.225,7	51.324,6
9	5	2	2	2	27.265,1	29.492,6	29.410,0	29.459,6	29.816,1
10	5	2	2	4	49.884,0	51.804,4	51.118,6	50.995,0	48.840,3
11	5	2	5	2	47.959,1	52.595,5	54.549,4	52.328,3	49.597,9
12	5	2	5	4	47.559,4	39.689,0	42.506,9	43.540,8	43.405,6
13	5	4	2	2	27.265,1	29.492,6	29.410,0	29.459,6	29.816,1
14	5	4	2	4	49.884,0	51.804,4	51.118,6	50.995,0	48.840,3
15	5	4	5	2	47.959,1	52.595,5	54.549,4	52.328,3	49.597,9
16	5	4	5	4	47.559,4	39.689,0	42.506,9	43.540,8	43.405,6

Tablo 4.9.'da görüldüğü üzere 4 faktörün sayılarında değişiklik yapılarak 16 farklı senaryo denenmiş ve 5 replikasyon sayısı ile toplamda 80 adet çıktı elde edilmiştir.

c. Ortalama Toplam Çalışan Maliyeti Performans Ölçütüne Göre Deney Tasarımı

Ortalama Çalışan Maliyeti için hazırlanan 2^4 faktöriyel deney tasarımı Tablo 4.10.'da yer almaktadır.

Tablo 4.10. Ortalama çalışan maliyeti için faktöriyel deney tasarımı

Senaryo	Ortalama Toplam Çalışan Maliyeti (TL)								
	Faktörler				Replikasyon Sayısı				
	Test Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	Test Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi B Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi C Tecrübe Seviyesindeki Çalışan Sayısı	1	2	3	4	5
1	2	2	2	2	20.070.428,665	20.477.483,524	20.426.805,575	20.478.919,196	20.458.262,103
2	2	2	2	4	19.603.451,866	19.938.846,540	20.190.502,653	19.966.656,431	19.956.237,753
3	2	2	5	2	19.127.431,122	19.663.890,879	19.866.555,871	19.912.638,651	19.901.581,531
4	2	2	5	4	20.096.550,623	20.395.937,664	20.337.110,924	20.311.474,426	20.388.083,972
5	2	4	2	2	20.070.428,665	20.477.483,524	20.426.805,575	20.478.919,196	20.458.262,103
6	2	4	2	4	19.603.451,866	22.558.399,719	22.672.370,124	23.011.938,513	23.104.361,356
7	2	4	5	2	19.127.431,122	19.663.890,879	19.866.555,871	19.912.638,651	19.901.581,531
8	2	4	5	4	20.096.550,623	20.395.937,664	20.337.110,924	20.311.474,426	20.388.083,972
9	5	2	2	2	22.386.074,013	22.857.984,603	22.630.023,842	22.547.274,661	22.532.359,876
10	5	2	2	4	21.890.927,936	22.684.164,445	22.738.533,114	22.822.711,934	22.990.616,965
11	5	2	5	2	22.407.893,976	22.747.353,670	23.054.997,936	23.035.010,245	22.924.151,693
12	5	2	5	4	22.951.310,185	23.364.231,525	23.432.336,869	23.352.231,352	23.425.571,160
13	5	4	2	2	22.386.074,013	22.857.984,603	22.630.023,842	22.547.274,661	22.532.359,876
14	5	4	2	4	21.890.927,936	22.684.164,445	22.738.533,114	22.822.711,934	22.990.616,965
15	5	4	5	2	22.407.893,976	22.747.353,670	23.054.997,936	23.035.010,245	22.924.151,693
16	5	4	5	4	22.951.310,185	23.364.231,525	23.432.336,869	23.352.231,352	23.425.571,160

Tablo 4.10.'da görüldüğü üzere 4 faktörün sayılarında değişiklik yapılarak 16 farklı senaryo denenmiş ve 5 replikasyon sayısı ile toplamda 80 adet çıktı elde edilmiştir.

d. Ortalama Toplam Kâr Performans Ölçütüne Göre Deney Tasarımı

Ortalama Toplam Kâr için hazırlanan 2^4 tam faktöriyel deney tasarımı Tablo 4.11.'de yer almaktadır.

Tablo 4.11. Ortalama çalışan maliyeti için faktöriyel deney tasarımı

Senaryo	Faktörler					Kârlılık (TL)				
						Replikasyon Sayısı				
	Test Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	Test Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı	Kurulum Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı		1	2	3	4	5
1	2	2	2	2		315.806.460,466	319.891.673,034	341.285.958,819	346.117.211,632	341.372.420,829
2	2	2	2	4		384.442.476,385	365.872.202,308	368.291.689,829	382.505.633,526	388.028.458,710
3	2	2	5	2		371.528.349,867	385.485.881,451	360.490.338,973	366.849.114,417	358.023.363,813
4	2	2	5	4		350.022.990,907	370.429.397,150	357.878.044,501	361.328.470,311	359.316.582,615
5	2	4	2	2		315.806.460,466	319.891.673,034	341.285.958,819	346.117.211,632	341.372.420,829
6	2	4	2	4		384.442.476,385	393.648.764,913	351.424.604,955	343.390.790,041	355.609.144,479
7	2	4	5	2		371.528.349,867	385.485.881,451	360.490.338,973	366.849.114,417	358.023.363,813
8	2	4	5	4		350.022.990,907	370.429.397,150	357.878.044,501	361.328.470,311	359.316.582,615
9	5	2	2	2		320.649.763,745	284.642.006,956	288.941.180,348	278.135.639,658	287.257.201,170
10	5	2	2	4		409.525.919,003	457.849.643,099	437.789.106,972	436.203.783,167	438.372.138,621
11	5	2	5	2		478.720.495,966	498.914.897,079	499.605.775,040	473.452.063,703	464.523.965,415
12	5	2	5	4		439.027.578,528	424.519.431,774	436.698.128,127	421.682.244,843	438.926.304,673
13	5	4	2	2		320.649.763,745	284.642.006,956	288.941.180,348	278.135.639,658	287.257.201,170
14	5	4	2	4		409.525.919,003	457.849.643,099	437.789.106,972	436.203.783,167	438.372.138,621
15	5	4	5	2		478.720.495,966	498.914.897,079	499.605.775,040	473.452.063,703	464.523.965,415
16	5	4	5	4		439.027.578,528	424.519.431,774	436.698.128,127	421.682.244,843	438.926.304,673

Tablo 4.11.'de görüldüğü üzere 4 faktörün sayılarında değişiklik yapılarak 16 farklı senaryo denenmiş ve 5 replikasyon sayısı ile toplamda 80 adet çıktı elde edilmiştir.

Genel olarak incelendiğinde; 4 farklı performans ölçütünün iyileştirilmesi amacıyla belirlenen 4 faktör için 2^4 faktöriyel deney tasarımının oluşturulduğu ve toplam 16 senaryo için 5 replikasyonla benzetim modelinin çalıştırıldığı görülmektedir. Bu benzetim modelinin çalıştırılmasıyla her performans ölçütü için 80 adet çıktı elde edilmiştir. Buna göre, oluşturulan deney tasarımlarının, performans ölçütüne göre en iyi sonucu veren senaryolarını bulmak için meta-modeller oluşturulmuş ve oluşturulan meta-modellerin benzetim eniyilemesi yoluyla en iyi sonucu veren çıktılarına ulaşılmıştır. Her bir performans ölçütüne ait meta-modeller, 4.4.2. Meta-Model Oluşturma başlığı altında verilmiştir.

4.4.2. Meta-Model Oluşturma

Meta-model olarak adlandırılan soyut model, benzetim modelinin yerini almak için kullanılmaktadır. Meta-modeller, benzetim sonuçlarının istatistiksel özetlenmesi için bir uygulama sağlayarak, çalışma süresinin gereksinimlerini azaltmaktadır. Meta-modeller, sistem sınırlarını dikkate alarak benzetim sürecinde amaç işlevi olarak rol almaktadır [38], [58]. Dört farklı performans ölçütünün meta-modelleri, aşağıda ilgili performans ölçütlerinin başlıkları altında gösterilmiştir. Deney tasarımı faktör olarak tanımlanan değişkenler, meta-modelin karar değişkenleri konumuna gelmektedir. Kurulan meta-modelin karar değişkenleri ve kısıtları aşağıda gösterilmiştir.

Karar Değişkenleri

Meta-modellerde kullanılacak karar değişkenleri aşağıda tanımlanmıştır:

x_1 : Test Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı

x_2 : Test Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı

x_3 : Kurulum Ekibi - B Tecrübe Seviyesindeki Çalışan Sayısı

x_4 : Kurulum Ekibi - C Tecrübe Seviyesindeki Çalışan Sayısı

Kısıtlar

Meta-modellerin kısıtları aşağıda tanımlanmıştır:

$$2 \leq x_1 \leq 5 \quad (4.1)$$

$$2 \leq x_2 \leq 4 \quad (4.2)$$

$$2 \leq x_3 \leq 5 \quad (4.3)$$

$$2 \leq x_4 \leq 4 \quad (4.4)$$

$$x_i \geq 0 \text{ ve tamsayı, } \forall i \quad (4.5)$$

a. Ortalama Tamamlanan Toplam Proje Sayısı Performans Ölçütüne Göre Oluşturulan Meta-Model

Tablo 4.8.'deki faktöriyel deney tasarımındaki veriler MINITAB İstatistik programı ile analiz edilmiştir. Meta-modelde x_1 , x_2 , x_3 ve x_4 olarak tanımlanan karar değişkenlerinin meta-modele olan etkilerinin ve kendi aralarındaki ilişkinin anlaşılması için öncelikle faktöriyel regresyon analizi yapılmaktadır. Ayrıntılı analiz sonuçları EK 5'te yer almakta olup etkileri görebilmek için öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.12.'de verilmektedir.

Tablo 4.12. Ortalama tamamlanan toplam proje sayısının x_1 , x_2 , x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi

Factorial Regression: Proje_Sayisi versus x1; x2; x3; x4

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
3,03168	91,38%	89,36%	86,53%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	15	6235,38	415,69	45,23	0,000
Linear	4	3869,18	967,29	105,24	0,000
x1	1	1999,33	1999,33	217,53	0,000
x2	1	7,96	7,96	0,87	0,356
x3	1	752,35	752,35	81,86	0,000
x4	1	1109,54	1109,54	120,72	0,000
2-Way Interactions	6	2051,11	341,85	37,19	0,000
x1*x2	1	7,96	7,96	0,87	0,356
x1*x3	1	332,11	332,11	36,13	0,000
x1*x4	1	216,49	216,49	23,55	0,000
x2*x3	1	7,96	7,96	0,87	0,356
x2*x4	1	7,96	7,96	0,87	0,356
x3*x4	1	1478,64	1478,64	160,88	0,000
3-Way Interactions	4	307,13	76,78	8,35	0,000
x1*x2*x3	1	7,96	7,96	0,87	0,356
x1*x2*x4	1	7,96	7,96	0,87	0,356
x1*x3*x4	1	283,25	283,25	30,82	0,000
x2*x3*x4	1	7,96	7,96	0,87	0,356
4-Way Interactions	1	7,96	7,96	0,87	0,356
x1*x2*x3*x4	1	7,96	7,96	0,87	0,356
Error	64	588,23	9,19		
Total	79	6823,61			

Tablo 4.12.'de yer alan değerleri analiz etmek için çıktıların P-value değerlerine bakılmaktadır. P-value değeri ≤ 0.05 olan karar değişkenlerinin ve etkileşimlerin istatistiksel olarak anlamlı olduğu kabul edilmektedir. Buna göre bu performans ölçütü için MINITAB programı yardımıyla yapılan faktöriyel tasarım analizinden elde edilen P-value değerlerinin; $Pvalue(x_1) = Pvalue(x_3) = Pvalue(x_4) = Pvalue(x_1*x_3) = Pvalue(x_1*x_4) = Pvalue(x_3*x_4) = Pvalue(x_1*x_3*x_4) = 0 \leq 0,05$ olması sebebiyle; tekli etkileşimlerden sadece x_1 , sadece x_3 , sadece x_4 ; ikili etkileşimlerden x_1*x_3 , x_1*x_4 , x_3*x_4 ; üçlü etkileşimlerden $x_1*x_3*x_4$ etkileşimlerinin istatistiksel olarak anlamlı olduğu, dördü etkileşimlerin anlamlı olmadığı sonucu ortaya çıkmaktadır. Meta-model oluştururken meta-model denklemine istatistiksel olarak anlamlı olan etkileşimler katılmaktadır, fakat bu performans ölçütü için x_2 istatistiksel olarak anlamlı olmasa bile karar değişkeni olduğu için meta-model denkleminde yer almaktadır.

Meta-Model denkleminde yer alacak x_1 , x_2 , x_3 , x_4 , x_1*x_3 , x_1*x_4 , x_3*x_4 , $x_1*x_3*x_4$ etkileşimleri için Regresyon Analizi yapılır. Ayrıntılı analiz sonuçları EK 6'da yer almakta olup etkileri görebilmek için öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.13.'te verilmektedir.

Tablo 4.13. Ortalama tamamlanan toplam proje sayısının x_1 , x_2 , x_3 , x_4 , x_1*x_3 , x_1*x_4 , x_3*x_4 , $x_1*x_3*x_4$ etkileşimlerine göre regresyon analizi

ANALİZ					
Regression Analysis: Proje_Sayisi versus x1; x2; x3; x4; x1*x3; x1*x4; x3*x4; x1*x3*x4					
Model Summary					
	S	R-sq	R-sq(adj)	R-sq(pred)	
	3,01158	90,56%	89,50%	88,02%	
Coefficients					
Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	59,15	6,94	8,53	0,000	
x1	-11,91	1,80	-6,61	0,000	64,44
x2	0,315	0,337	0,94	0,352	1,00
x3	-1,31	1,80	-0,73	0,470	64,44
x4	-0,33	2,17	-0,15	0,880	41,53
x1*x3	3,414	0,473	7,22	0,000	118,89
x1*x4	4,024	0,570	7,06	0,000	99,53
x3*x4	0,061	0,570	0,11	0,915	99,53
x1*x3*x4	-0,836	0,150	-5,59	0,000	148,53

Tablo 4.13'teki Coefficients (Katsayılar) dikkate alınarak elde edilen meta-modelin Ortalama Tamamlanan Toplam Proje Sayısını (TPS) tahmin eden regresyon modeli aşağıdaki eşitlikte gösterilmiştir.

$$\begin{aligned} \text{TPS} = & 59,15 - 11,91 x_1 - 0,315 x_2 - 1,31 x_3 - 0,33 x_4 + 3,414 x_1 * x_3 + 4,024 x_1 * x_4 \\ & + 0,061 x_3 * x_4 - 0,836 x_1 * x_3 * x_4 \end{aligned}$$

Meta-modelin Geçerliliği

Kurulan meta-modelin benzetimi ne derece iyi yansıttığını görmek için meta-modelin geçerliliği kontrol edilmiştir. Bunun için Mutlak Göreceli Hata ("Absolute Relative Error"- (ARE)) değerleri kullanılmıştır. Bu yöntemde sistemin girdileri rasgele seçilir ve benzetim modeli ile meta-modelin bu girdilere göre çıktıları karşılaştırılır.

$\text{ARE}(\text{so}, \text{mo}) = |(\text{so}-\text{mo})/\text{so}|$ eşitliği olup so:benzetim çıktısının değeri, mo: meta-model çıktısının değeri olarak tanımlanmıştır [39], [59].

Meta-modelin geçerliliğini kontrol etmek için kullanılan ARE değerleri Tablo 4.14'te yer almaktadır.

Tablo 4.14. Ortalama tamamlanan toplam proje sayısı meta-modelinin geçerliliği

Regresyon Meta-Modelinin Geçerliliği						
x ₁	x ₂	x ₃	x ₄	mo	so	ARE Değeri
2	3	3	3	59,594	56,5	0,055
5	4	5	4	73,92	73	0,013
4	3	4	3	67,851	65,5	0,036
2	2	5	2	61,616	60	0,027
2	4	5	2	60,986	63,5	0,040
Ortalama ARE=						0,034

Meta-modelin ortalama ARE değerleri %3,4'tür. Dört karar değişkeni (x₁, x₂, x₃ ve x₄) arasındaki tüm olası kombinasyonları keşfetmek için hesaplama açısından yeterince verimli olduğu ve kabul edilebilir bir hata payı olduğu için meta modelin, benzetim modelinin soyutlama modeli olarak kullanılabileceği sonucuna varılmıştır meta-model geçerlidir.

b. Projelerin Ortalama Toplam Tamamlama Süresi Performans Ölçütüne Göre Oluşturulan Meta-Model

Tablo 4.9.'daki faktöriyel deney tasarımındaki veriler MINITAB İstatistik programı ile analiz edilmiştir. Meta-modelde x_1 , x_2 , x_3 ve x_4 olarak tanımlanan karar değişkenlerinin meta-modele olan etkilerinin ve kendi aralarındaki ilişkinin anlaşılması için öncelikle faktöriyel regresyon analizi yapılmaktadır. Ayrıntılı analiz sonuçları EK 7'de yer almakta olup etkileri görebilmek için öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.15.'te verilmektedir.

Tablo 4.15. Projelerin ortalama toplam tamamlama süresinin x_1 , x_2 , x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi

Factorial Regression: Proje_Suresi versus x1; x2; x3; x4

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
3109,49	87,51%	84,58%	80,48%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	15	4335689265	289045951	29,89	0,000
Linear	4	1194179515	298544879	30,88	0,000
x1	1	861291999	861291999	89,08	0,000
x2	1	5956714	5956714	0,62	0,435
x3	1	290801454	290801454	30,08	0,000
x4	1	36129348	36129348	3,74	0,058
2-Way Interactions	6	1601037274	266839546	27,60	0,000
x1*x2	1	5956714	5956714	0,62	0,435
x1*x3	1	281471335	281471335	29,11	0,000
x1*x4	1	570958692	570958692	59,05	0,000
x2*x3	1	5956714	5956714	0,62	0,435
x2*x4	1	5956714	5956714	0,62	0,435
x3*x4	1	730737104	730737104	75,58	0,000
3-Way Interactions	4	1534515762	383628940	39,68	0,000
x1*x2*x3	1	5956714	5956714	0,62	0,435
x1*x2*x4	1	5956714	5956714	0,62	0,435
x1*x3*x4	1	1516645618	1516645618	156,86	0,000
x2*x3*x4	1	5956714	5956714	0,62	0,435
4-Way Interactions	1	5956714	5956714	0,62	0,435
x1*x2*x3*x4	1	5956714	5956714	0,62	0,435
Error	64	618809698	9668902		
Total	79	4954498962			

Tablo 4.15.'te yer alan değerleri analiz etmek için çıktıların P-value değerlerine bakılmaktadır. P-value değeri $\leq 0,05$ olan karar değişkenlerinin ve etkileşimlerin istatistiksel olarak anlamlı olduğu kabul edilmektedir. Buna göre bu performans ölçütü için MINITAB programı yardımıyla yapılan faktöriyel tasarım analizinden elde edilen P-value değerlerinin; $Pvalue(x_1) = Pvalue(x_3) = Pvalue(x_1*x_3) = Pvalue(x_1*x_4) = Pvalue(x_3*x_4) = Pvalue(x_1*x_3*x_4) = 0 \leq 0,05$ olması sebebiyle; tekli etkileşimlerden sadece x_1 , sadece x_3 ; ikili etkileşimlerden x_1*x_3 , x_1*x_4 , x_3*x_4 ; üçlü etkileşimlerden $x_1*x_3*x_4$ etkileşimlerinin istatistiksel olarak anlamlı olduğu, dördü etkileşimlerin anlamlı olmadığı sonucu ortaya çıkmaktadır. Meta-model oluştururken meta-model denklemine istatistiksel olarak anlamlı olan etkileşimler katılmaktadır, fakat bu performans ölçütü için x_2 ve x_4 istatistiksel olarak anlamlı olmasa bile karar değişkeni olduğu için meta-model denkleminde yer almaktadır.

Meta-Model denkleminde yer alacak x_1 , x_2 , x_3 , x_4 , x_1*x_3 , x_1*x_4 , x_3*x_4 , $x_1*x_3*x_4$ etkileşimleri için Regresyon Analizi yapılır. Ayrıntılı analiz sonuçları EK 8'de yer almakta olup etkileri görebilmek için öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.16.'da verilmektedir.

Tablo 4.16. Projelerin ortalama toplam tamamlama süresinin x_1 , x_2 , x_3 , x_4 , x_1*x_3 , x_1*x_4 , x_3*x_4 , $x_1*x_3*x_4$ etkileşimlerine göre regresyon analizi

Regression Analysis: Proje_Suresi versus x_1 ; x_2 ; x_3 ; x_4 ; x_1*x_3 ; x_1*x_4 ; x_3*x_4 ; $x_1*x_3*x_4$

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
3050,07	86,67%	85,17%	83,07%

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	127756	7024	18,19	0,000	
x_1	-30767	1825	-16,86	0,000	64,44
x_2	-273	341	-0,80	0,426	1,00
x_3	-15921	1825	-8,72	0,000	64,44
x_4	-22215	2198	-10,11	0,000	41,53
x_1*x_3	6639	479	13,85	0,000	118,89
x_1*x_4	8554	577	14,82	0,000	99,53
x_3*x_4	4758	577	8,24	0,000	99,53
$x_1*x_3*x_4$	-1935	152	-12,77	0,000	148,53

Tablo 4.16'daki Coefficients (Katsayılar) dikkate alınarak elde edilen meta-modelin Projelerin Ortalama Toplam Tamamlama Süresini (TTS) tahmin eden regresyon modeli aşağıdaki eşitlikte gösterilmiştir.

$$\text{TTS} = 127756 - 30767 x_1 - 273 x_2 - 15921 x_3 - 22215 x_4 + 6639 x_1 * x_3 + 8554 x_1 * x_4 + 4758 x_3 * x_4 - 1935 x_1 * x_3 * x_4$$

Meta-modelin Geçerliliği

Meta-modelin geçerliliğini kontrol etmek için kullanılan ARE değerleri Tablo 4.17'de yer almaktadır.

$\text{ARE}(\text{so}, \text{mo}) = |(\text{so}-\text{mo})/\text{so}|$ eşitliğidir (so=benzetim çıktısının değeri, mo: meta-model çıktısının değeri)

Tablo 4.17. Projelerin ortalama toplam tamamlama süresi meta-modelinin geçerliliği

Regresyon Meta-Modelinin Geçerliliği							
x ₁	x ₂	x ₃	x ₄	mo	so	ARE	
2	4	2	4	46.520,00	47.959,07	0,030	
2	3	3	3	50.145,00	48.005,20	0,045	
5	4	2	4	50.261,00	47.559,42	0,057	
2	4	2	4	46.520,00	47.959,07	0,030	
2	4	5	4	49.247,00	47.959,07	0,027	
Ortalama ARE=						0,038	

Meta-modelin ARE değerlerinin ortalaması %3,8'dir. Kabul edilebilir bir hata payı olduğu için meta-model geçerlidir.

c. Ortalama Toplam Çalışan Maliyeti Performans Ölçütüne Göre Oluşturulan Meta-Model

Tablo 4.10.'daki faktöriyel deney tasarımındaki veriler MINITAB İstatistik programı ile analiz edilmiştir. Meta-modelde x₁, x₂, x₃ ve x₄ olarak tanımlanan karar değişkenlerinin meta-modele olan etkilerinin ve kendi aralarındaki ilişkinin anlaşılması için öncelikle faktöriyel regresyon analizi yapılmaktadır. Ayrıntılı analiz sonuçları EK 9'da yer almakta olup etkileri

görebilmek için öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.18.'de verilmektedir.

Tablo 4.18. Ortalama toplam çalışan maliyetinin x_1 , x_2 , x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi

Factorial Regression: Calisan_Maliyeti versus x_1 ; x_2 ; x_3 ; x_4

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
443541	92,17%	90,33%	87,76%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	15	1,48183E+14	9,87888E+12	50,22	0,000
Linear	4	1,28770E+14	3,21924E+13	163,64	0,000
x_1	1	1,22807E+14	1,22807E+14	624,25	0,000
x_2	1	1,59466E+12	1,59466E+12	8,11	0,006
x_3	1	3,37827E+11	3,37827E+11	1,72	0,195
x_4	1	4,03003E+12	4,03003E+12	20,49	0,000
2-Way Interactions	6	1,27180E+13	2,11967E+12	10,77	0,000
x_1*x_2	1	1,59466E+12	1,59466E+12	8,11	0,006
x_1*x_3	1	6,99528E+12	6,99528E+12	35,56	0,000
x_1*x_4	1	7,67833E+11	7,67833E+11	3,90	0,053
x_2*x_3	1	1,59466E+12	1,59466E+12	8,11	0,006
x_2*x_4	1	1,59466E+12	1,59466E+12	8,11	0,006
x_3*x_4	1	1,70919E+11	1,70919E+11	0,87	0,355
3-Way Interactions	4	5,10080E+12	1,27520E+12	6,48	0,000
$x_1*x_2*x_3$	1	1,59466E+12	1,59466E+12	8,11	0,006
$x_1*x_2*x_4$	1	1,59466E+12	1,59466E+12	8,11	0,006
$x_1*x_3*x_4$	1	3,16811E+11	3,16811E+11	1,61	0,209
$x_2*x_3*x_4$	1	1,59466E+12	1,59466E+12	8,11	0,006
4-Way Interactions	1	1,59466E+12	1,59466E+12	8,11	0,006
$x_1*x_2*x_3*x_4$	1	1,59466E+12	1,59466E+12	8,11	0,006
Error	64	1,25906E+13	1,96728E+11		
Total	79	1,60774E+14			

Tablo 4.18.'de yer alan değerleri analiz etmek için çıktıların P-value değerlerine bakılmaktadır. P-value değeri ≤ 0.05 olan karar değişkenlerinin ve etkileşimlerin istatistiksel olarak anlamlı olduğu kabul edilmektedir. Buna göre bu performans ölçütü için MINITAB programı yardımıyla yapılan faktöriyel tasarım analizinden elde edilen P-value değerlerinin;

$Pvalue(x_1) = Pvalue(x_2) = Pvalue(x_4) = Pvalue(x_1*x_2) = Pvalue(x_1*x_3) = Pvalue(x_2*x_3) = Pvalue(x_2*x_4) = Pvalue(x_1*x_2*x_3) = Pvalue(x_1*x_2*x_4) = Pvalue(x_2*x_3*x_4) = Pvalue(x_1*x_2*x_3*x_4) = 0 \leq 0,05$ olması sebebiyle; tekli etkileşimlerden sadece x_1 , sadece x_2 , sadece x_4 ; ikili etkileşimlerden x_1*x_2 , x_1*x_3 , x_2*x_3 , x_2*x_4 ; üçlü etkileşimlerden $x_1*x_2*x_3$, $x_1*x_2*x_4$, $x_2*x_3*x_4$; dördü etkileşim olan $x_1*x_2*x_3*x_4$ etkileşimlerinin istatistiksel olarak anlamlı olduğu sonucu ortaya çıkmaktadır. Meta-model oluştururken meta-model denklemine istatistiksel olarak anlamlı olan etkileşimler katılmaktadır, fakat bu performans ölçütü için x_3 istatistiksel olarak anlamlı olmasa bile karar değişkeni olduğu için meta-model denkleminde yer almaktadır.

Meta-Model denkleminde yer alacak x_1 , x_2 , x_3 , x_4 , x_1*x_2 , x_1*x_3 , x_2*x_3 , x_2*x_4 , $x_1*x_2*x_3$, $x_1*x_2*x_4$, $x_2*x_3*x_4$, $x_1*x_2*x_3*x_4$ etkileşimleri için Regresyon Analizi yapılır. Ayrıntılı analiz sonuçları EK 10'da yer almakta olup etkileri görebilmek için öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.19.'da verilmektedir.

Tablo 4.19. Ortalama toplam çalışan maliyetinin x_1 , x_2 , x_3 , x_4 , x_1*x_2 , x_1*x_3 , x_2*x_3 , x_2*x_4 , $x_1*x_2*x_3$, $x_1*x_2*x_4$, $x_2*x_3*x_4$, $x_1*x_2*x_3*x_4$ etkileşimlerine göre regresyon analizi

Regression Analysis: Calisan_Maliyeti versus x_1 ; x_2 ; x_3 ; x_4 ; x_1*x_2 ; x_1*x_3 ; x_2*x_3 ; x_2*x_4 ; $x_1*x_2*x_3$; $x_1*x_2*x_4$; $x_2*x_3*x_4$; $x_1*x_2*x_3...$

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
492399	89,90%	88,09%	85,27%

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	16361560	1237516	13,22	0,000	
x_1	1307243	294628	4,44	0,000	64,44
x_2	299884	513516	0,58	0,561	87,01
x_3	437929	294628	1,49	0,142	64,44
x_4	-199111	174089	-1,14	0,257	10,00
x_1*x_2	-72739	128425	-0,57	0,573	189,11
x_1*x_3	-56823	77373	-0,73	0,465	118,89
x_2*x_3	-159253	128425	-1,24	0,219	189,11
x_2*x_4	422947	123752	3,42	0,001	96,01
$x_1*x_2*x_3$	18752	33726	0,56	0,580	282,21
$x_1*x_2*x_4$	-80335	29463	-2,73	0,008	131,11
$x_2*x_3*x_4$	-51497	29463	-1,75	0,085	131,11
$x_1*x_2*x_3*x_4$	14666	7737	1,90	0,062	175,21

Tablo 4.19'daki Coefficients (Katsayılar) dikkate alınarak elde edilen meta-modelin Ortalama Toplam Çalışan Maliyetini (TÇM) tahmin eden regresyon modeli aşağıdaki eşitlikte gösterilmiştir.

$$\begin{aligned}
T\check{C}M = & 16361560 + 1307243 x_1 + 299884 x_2 + 437929 x_3 - 199111 x_4 - 72739 x_1 * x_2 \\
& - 56823 x_1 * x_3 - 159253 x_2 * x_3 + 422947 x_2 * x_4 + 18752 x_1 * x_2 * x_3 - 80335 x_1 * x_2 * x_4 \\
& - 51497 x_2 * x_3 * x_4 + 14666 x_1 * x_2 * x_3 * x_4
\end{aligned}$$

Meta-modelin Geerlilięi

Meta-modelin geerlilięini kontrol etmek iin kullanılan ARE deęerleri Tablo 4.20’de yer almaktadır.

$ARE(so, mo) = |(so-mo)/so|$ eřitlięidir (so=benzetim ıktısının deęeri, mo: meta-model ıktısının deęeri)

Tablo 4.20. Ortalama toplam alıřan maliyeti meta-modelinin geerlilięi

Regresyon Meta-Modelinin Geerlilięi						
x ₁	x ₂	x ₃	x ₄	mo	so	ARE
2	4	2	2	20.613.598,00	19.127.431,12	0,078
5	4	5	4	23.392.289,00	22.951.310,19	0,019
5	4	2	4	22.668.779,00	22.951.310,19	0,012
2	2	2	2	19.919.994,00	20.096.550,62	0,009
2	3	3	3	20.481.077,00	21.184.268,13	0,033
Ortalama ARE=						0,030

Meta-modelin ARE deęerlerinin ortalaması %3’tür. Kabul edilebilir bir hata payı olduęu iin meta-model geerlidir.

d. Ortalama Toplam Kâr Performans Ölütüne Göre Deney Tasarımı

Tablo 4.10.’daki faktöriyel deney tasarımındaki veriler MINITAB İstatistik programı ile analiz edilmiřtir. Meta-modelde x₁, x₂, x₃ ve x₄ olarak tanımlanan karar deęiřkenlerinin meta-modele olan etkilerinin ve kendi aralarındaki iliřkinin anlařılması iin öncelikle faktöriyel regresyon analizi yapılmaktadır. Ayrıntılı analiz sonuçları EK 11’de yer almakta olup etkileri görebilmek iin öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.21.’de verilmektedir.

Tablo 4.21. Ortalama toplam kârın x_1 , x_2 , x_3 ve x_4 karar değişkenlerine göre faktöriyel regresyon analizi

Factorial Regression: Karlilik versus x_1 ; x_2 ; x_3 ; x_4

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
13902247	95,61%	94,58%	93,14%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	15	2,69295E+17	1,79530E+16	92,89	0,000
Linear	4	1,29895E+17	3,24737E+16	168,02	0,000
x_1	1	5,52072E+16	5,52072E+16	285,64	0,000
x_2	1	4,59419E+13	4,59419E+13	0,24	0,628
x_3	1	5,56254E+16	5,56254E+16	287,81	0,000
x_4	1	1,90162E+16	1,90162E+16	98,39	0,000
2-Way Interactions	6	1,12078E+17	1,86797E+16	96,65	0,000
x_1*x_2	1	4,59419E+13	4,59419E+13	0,24	0,628
x_1*x_3	1	3,35097E+16	3,35097E+16	173,38	0,000
x_1*x_4	1	4,95490E+15	4,95490E+15	25,64	0,000
x_2*x_3	1	4,59419E+13	4,59419E+13	0,24	0,628
x_2*x_4	1	4,59419E+13	4,59419E+13	0,24	0,628
x_3*x_4	1	7,34756E+16	7,34756E+16	380,17	0,000
3-Way Interactions	4	2,72758E+16	6,81895E+15	35,28	0,000
$x_1*x_2*x_3$	1	4,59419E+13	4,59419E+13	0,24	0,628
$x_1*x_2*x_4$	1	4,59419E+13	4,59419E+13	0,24	0,628
$x_1*x_3*x_4$	1	2,71380E+16	2,71380E+16	140,41	0,000
$x_2*x_3*x_4$	1	4,59419E+13	4,59419E+13	0,24	0,628
4-Way Interactions	1	4,59419E+13	4,59419E+13	0,24	0,628
$x_1*x_2*x_3*x_4$	1	4,59419E+13	4,59419E+13	0,24	0,628
Error	64	1,23694E+16	1,93272E+14		
Total	79	2,81664E+17			

Tablo 4.21.'de yer alan değerleri analiz etmek için çıktıların P-value değerlerine bakılmaktadır. P-value değeri ≤ 0.05 olan karar değişkenlerinin ve etkileşimlerin istatistiksel olarak anlamlı olduğu kabul edilmektedir. Buna göre bu performans ölçütü için MINITAB programı yardımıyla yapılan faktöriyel tasarım analizinden elde edilen P-value değerlerinin; $Pvalue(x_1) = Pvalue(x_3) = Pvalue(x_4) = Pvalue(x_1*x_3) = Pvalue(x_1*x_4) = Pvalue(x_3*x_4) = Pvalue(x_1*x_3*x_4) = 0 \leq 0,05$ olması sebebiyle; tekli etkileşimlerden sadece x_1 , sadece x_3 ,

sadece x_4 ; ikili etkileşimlerden x_1*x_3 , x_1*x_4 , x_3*x_4 ; üçlü etkileşimlerden $x_1*x_3*x_4$ etkileşimlerinin istatistiksel olarak anlamlı olduğu, dördü etkileşimlerin istatistiksel olarak anlamlı olmadığı sonucu ortaya çıkmaktadır. Meta-model oluştururken meta-model denklemine istatistiksel olarak anlamlı olan etkileşimler katılmaktadır, fakat bu performans ölçütü için x_2 istatistiksel olarak anlamlı olmasa bile karar değişkeni olduğu için meta-model denkleminde yer almaktadır.

Meta-Model denkleminde yer alacak x_1 , x_2 , x_3 , x_4 , x_1*x_3 , x_1*x_4 , x_3*x_4 , $x_1*x_3*x_4$ etkileşimleri için Regresyon Analizi yapılır. Ayrıntılı analiz sonuçları EK 12’de yer almakta olup etkileri görebilmek için öncelikle incelememiz gereken Varyans Analizi tablosu Tablo 4.22.’de verilmektedir.

Tablo 4.22. Ortalama toplam kârın x_1 , x_2 , x_3 , x_4 , x_1*x_3 , x_1*x_4 , x_3*x_4 , $x_1*x_3*x_4$ etkileşimlerine göre regresyon analizi

Regression Analysis: Karlılık versus x_1 ; x_2 ; x_3 ; x_4 ; x_1*x_3 ; x_1*x_4 ; x_3*x_4 ; $x_1*x_3*x_4$

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
13369626	95,49%	94,99%	94,28%

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	472904759	30790392	15,36	0,000	
x_1	-116014353	7999738	-14,50	0,000	64,44
x_2	-757808	1494770	-0,51	0,614	1,00
x_3	-39596511	7999738	-4,95	0,000	64,44
x_4	-32508072	9632960	-3,37	0,001	41,53
x_1*x_3	33653565	2100834	16,02	0,000	118,89
x_1*x_4	33896955	2529739	13,40	0,000	99,53
x_3*x_4	8446409	2529739	3,34	0,001	99,53
$x_1*x_3*x_4$	-8185804	664342	-12,32	0,000	148,53

Tablo 4.22’deki Coefficients (Katsayılar) dikkate alınarak elde edilen meta-modelin Ortalama Toplam Karlılığı (TK) tahmin eden regresyon modeli aşağıdaki eşitlikte gösterilmiştir.

$$TK = 472904759 - 116014353 x_1 - 757808 x_2 - 39596511 x_3 - 32508072 x_4$$

$$+ 33653565 x_1*x_3 + 33896955 x_1*x_4 + 8446409 x_3*x_4 - 8185804 x_1*x_3*x_4$$

Meta-modelin Geçerliliği

Meta-modelin geçerliliğini kontrol etmek için kullanılan ARE değerleri Tablo 4.23'te yer almaktadır.

$ARE(so, mo) = |(so-mo)/so|$ eşitliğidir (so=benzetim çıktısının değeri, mo: meta-model çıktısının değeri)

Tablo 4.23. Ortalama toplam kâr meta-modelinin geçerliliği

Regresyon Meta-Modelinin Geçerliliği						
X ₁	X ₂	X ₃	X ₄	mo	so	ARE
5	4	5	4	431.412.924,00	424.519.431,77	0,016
2	2	2	2	333.652.555,00	315.806.460,47	0,057
5	4	2	4	435.190.314,00	424.519.431,77	0,025
2	2	5	2	369.233.218,00	346.117.211,63	0,067
2	4	5	2	367.717.602,00	371.528.349,87	0,010
Ortalama ARE=						0,035

Meta-modelin ARE değerlerinin ortalaması %3,5'tir. Kabul edilebilir bir hata payı olduğu için meta-model geçerlidir.

4.4.3. Karar Modeli ve Optimizasyon

İlgili performans ölçütleri baz alınarak, istatistiksel analizlerle oluşturulan regresyon meta-modeli eşitlikleri, amaç fonksiyonu olarak; meta-modelin kısıtları da karar modelinin kısıtları olarak matematiksel karar modelinde yer almaktadır. Her performans ölçütü için iyileştirme amacı belli olup bu amaçlar doğrultusunda oluşturulan matematiksel karar modelleri sırayla çözüme ulaştırılmıştır.

a. Ortalama Tamamlanan Toplam Proje Sayısı Performans Ölçütüne Göre Oluşturulan Karar Modeli ve Optimizasyonu

Ortalama tamamlanan toplam proje sayısının en büyüklenmesi istenmektedir. Bu nedenle aşağıdaki matematiksel model kurulur.

$$2 \leq x_1 \leq 5 \quad (4.1)$$

$$2 \leq x_2 \leq 4 \quad (4.2)$$

$$2 \leq x_3 \leq 5 \quad (4.3)$$

$$2 \leq x_4 \leq 4 \quad (4.4)$$

$$x_i \geq 0 \text{ ve tamsayı, } \forall i \quad (4.5)$$

kısıtları altında

ENB (En Büyük)

$$\begin{aligned} \text{TPS} = & 59,15 - 11,91 x_1 - 0,315 x_2 - 1,31 x_3 - 0,33 x_4 + 3,414 x_1 * x_3 + 4,024 x_1 * x_4 \\ & + 0,061 x_3 * x_4 - 0,836 x_1 * x_3 * x_4 \end{aligned} \quad (4.6)$$

Matematiksel model Excel Çözücü ile çözülerek $x_1 = 5$, $x_2 = 2$, $x_3 = 2$, $x_4 = 4$ ve $\text{TPS} = 76,7$ adet olarak bulunmaktadır. Mevcut durumda Ortalama Tamamlanan Toplam Proje Sayısı = 54,6 adet iken benzetim eniyilemesi ile sistem daha iyi bir duruma gelmiş ve Ortalama Tamamlanan Toplam Proje Sayısı = 76,7 olmuştur.

Meta-model karar modelinin Excel Çözücü ile çözülmesiyle elde edilen karar değişkeni değerleri, ARENA Benzetim Programı'nda oluşturulan benzetim modeline eklenerek benzetim modeli bu değerlerle yeniden çalıştırılmıştır. Bu kapsamda meta-model denkleminde ve benzetim modelinden elde edilen çıktılar Tablo 4.24.'te karşılaştırma tablosu olarak verilmiştir.

Tablo 4.24. Ortalama tamamlanan toplam proje sayısı karşılaştırma tablosu

Ortalama Tamamlanan Toplam Proje Sayısı Karşılaştırma Tablosu					
x_1	x_2	x_3	x_4	Meta-model Değeri (adet)	Benzetim Modeli Değeri (adet)
5	2	2	4	76,70	77,20

Tablo 4.24. incelendiğinde, TPS olarak tanımlanan Ortalama Tamamlanan Toplam Proje Sayısının, meta-modelin amaç fonksiyonun çözülmesiyle elde edilen değeri ile benzetim

modelinin 5 replikasyonla 10 yıl için yeniden çalıştırılmasıyla elde edilen değerlerin birbirine yakın değerler olduğu görülmektedir.

b. Projelerin Ortalama Toplam Tamamlama Süresi Performans Ölçütüne Göre Oluşturulan Karar Modeli ve Optimizasyonu

Projelerin ortalama toplam tamamlama süresinin en küçüklenmesi istenmektedir. Bu nedenle aşağıdaki matematiksel model kurulur.

$$2 \leq x_1 \leq 5 \quad (4.1)$$

$$2 \leq x_2 \leq 4 \quad (4.2)$$

$$2 \leq x_3 \leq 5 \quad (4.3)$$

$$2 \leq x_4 \leq 4 \quad (4.4)$$

$$x_i \geq 0 \text{ ve tamsayı, } \forall i \quad (4.5)$$

kısıtları altında

ENK (En Küçük)

$$\begin{aligned} \text{TTS} = & 127756 - 30767 x_1 - 273 x_2 - 15921 x_3 - 22215 x_4 + 6639 x_1 * x_3 + 8554 x_1 * x_4 \\ & + 4758 x_3 * x_4 - 1935 x_1 * x_3 * x_4 \end{aligned} \quad (4.7)$$

Matematiksel model Excel Çözücü ile çözülerek $x_1 = 5$, $x_2 = 4$, $x_3 = 2$ ve $x_4 = 2$ TTS = 28.819 saat olarak bulunur. Mevcut durumda Projelerin Ortalama Toplam Tamamlama Süresi = 55.672,08 saat iken benzetim eniyilemesi ile sistem daha iyi bir duruma gelmiş ve Projelerin Ortalama Toplam Tamamlama Süresi = 28.819 saat olmuştur.

Meta-model karar modelinin Excel Çözücü ile çözülmesiyle elde edilen karar değişkeni değerleri, ARENA Benzetim Programı'nda oluşturulan benzetim modeline eklenerek benzetim modeli bu değerlerle yeniden çalıştırılmıştır. Bu kapsamda meta-model denkleminde ve benzetim modelinden elde edilen çıktılar Tablo 4.25.'te karşılaştırma tablosu olarak verilmiştir.

Tablo 4.25. Projelerin ortalama toplam tamamlama süresi karşılaştırma tablosu

Projelerin Ortalama Toplam Tamamlanma Süresi Karşılaştırma Tablosu					
x_1	x_2	x_3	x_4	Meta-model Değeri (saat)	Benzetim Modeli Değeri (saat)
5	4	2	2	28.819	29.399,936

Tablo 4.25. incelendiğinde, TTS olarak tanımlanan Projelerin Ortalama Toplam Tamamlanma Süresinin, meta-modelin amaç fonksiyonun çözülmesiyle elde edilen değeri ile benzetim modelinin 5 replikasyonla 10 yıl için yeniden çalıştırılmasıyla elde edilen değerin birbirine yakın değerler olduğu görülmektedir.

c. Ortalama Toplam Çalışan Maliyeti Performans Ölçütüne Göre Oluşturulan Karar Modeli ve Optimizasyonu

Ortalama toplam çalışan maliyetinin en küçüklenmesi istenmektedir. Bu nedenle aşağıdaki matematiksel model kurulur.

$$2 \leq x_1 \leq 5 \quad (4.1)$$

$$2 \leq x_2 \leq 4 \quad (4.2)$$

$$2 \leq x_3 \leq 5 \quad (4.3)$$

$$2 \leq x_4 \leq 4 \quad (4.4)$$

$$x_i \geq 0 \text{ ve tamsayı}, \quad \forall i \quad (4.5)$$

kısıtları altında

ENK (En Küçük)

$$\begin{aligned} T\check{C}M = & 16361560 + 1307243 x_1 + 299884 x_2 + 437929 x_3 - 199111 x_4 - 72739 x_1 * x_2 \\ & - 56823 x_1 * x_3 - 159253 x_2 * x_3 + 422947 x_2 * x_4 + 18752 x_1 * x_2 * x_3 - 80335 x_1 * x_2 * x_4 - 51497 \\ & x_2 * x_3 * x_4 + 14666 x_1 * x_2 * x_3 * x_4 \end{aligned} \quad (4.8)$$

Matematiksel model Excel Çözücü ile çözülerek $x_1 = 2$, $x_2 = 4$, $x_3 = 5$, $x_4 = 2$ ve $T\check{C}M = 19.593.499,00$ TL olarak bulunur. Mevcut durumda Ortalama Toplam Çalışan Maliyeti =

20.458.262,08 TL iken benzetim eniyilemesi ile sistem daha iyi bir duruma gelmiş ve Ortalama Toplam Çalışan Maliyeti = 19.593.499,00 TL olmuştur.

Meta-model karar modelinin Excel Çözücü ile çözülmesiyle elde edilen karar değişkeni değerleri, ARENA Benzetim Programı'nda oluşturulan benzetim modeline eklenerek benzetim modeli bu değerlerle yeniden çalıştırılmıştır. Bu kapsamda meta-model denkleminde ve benzetim modelinden elde edilen çıktılar Tablo 4.26.'da karşılaştırma tablosu olarak verilmiştir.

Tablo 4.26. Ortalama toplam çalışan maliyeti karşılaştırma tablosu

Ortalama Toplam Çalışan Maliyeti Karşılaştırma Tablosu					
x_1	x_2	x_3	x_4	Meta-model Değeri (TL)	Benzetim Modeli Değeri (TL)
2	4	5	2	19.593.499,00	23.002.549,85

Tablo 4.26. incelendiğinde, TÇM olarak tanımlanan Ortalama Toplam Çalışan Maliyetinin, meta-modelin amaç fonksiyonun çözülmesiyle elde edilen değeri ile benzetim modelinin 5 replikasyonla 10 yıl için yeniden çalıştırılmasıyla elde edilen değer birbirine yakın değerler olduğu görülmektedir.

d. Ortalama Toplam Kârlılık Performans Ölçütüne Göre Oluşturulan Meta- Karar Modeli ve Optimizasyonu

Ortalama Toplam Kârlılığın en büyüklenmesi istenmektedir. Bu nedenle aşağıdaki matematiksel model kurulur.

$$2 \leq x_1 \leq 5 \quad (4.1)$$

$$2 \leq x_2 \leq 4 \quad (4.2)$$

$$2 \leq x_3 \leq 5 \quad (4.3)$$

$$2 \leq x_4 \leq 4 \quad (4.4)$$

$$x_i \geq 0 \text{ ve tamsayı, } \forall i \quad (4.5)$$

kısıtları altında

ENB (En büyük)

$$\begin{aligned} \text{TK} = & 472904759 - 116014353 x_1 - 757808 x_2 - 39596511 x_3 - 32508072 x_4 \\ & + 33653565 x_1 * x_3 + 33896955 x_1 * x_4 + 8446409 x_3 * x_4 - 8185804 x_1 * x_3 * x_4 \end{aligned} \quad (4.9)$$

Matematiksel model Excel Çözücü ile çözülerek $x_1 = 5$, $x_2 = 2$, $x_3 = 2$, $x_4 = 4$ ve $\text{TK} = 436.705.930,00$ TL olarak bulunur. Mevcut durumda Ortalama Toplam Kârlılık = 341.372.420,829 TL iken benzetim eniyilemesi ile sistem daha iyi bir duruma gelmiş ve Ortalama Toplam Kârlılık = 436.705.930,00 TL olmuştur.

Meta-model karar modelinin Excel Çözücü ile çözülmesiyle elde edilen karar değişkeni değerleri, ARENA Benzetim Programı'nda oluşturulan benzetim modeline eklenerek benzetim modeli bu değerlerle yeniden çalıştırılmıştır. Bu kapsamda meta-model denkleminde ve benzetim modelinden elde edilen çıktılar Tablo 4.27.'de karşılaştırma tablosu olarak verilmiştir.

Tablo 4.27. Ortalama toplam kârlılık karşılaştırma tablosu

Ortalama Toplam Kârlılık Karşılaştırma Tablosu					
x_1	x_2	x_3	x_4	Meta-model Değeri (TL)	Benzetim Modeli Değeri (TL)
5	2	2	4	436.705.930,000	438.372.138,621

Tablo 4.27. incelendiğinde, TK olarak tanımlanan Ortalama Toplam Kârlılığının, meta-modelin amaç fonksiyonun çözülmesiyle elde edilen değeri ile benzetim modelinin 5 replikasyonla 10 yıl için yeniden çalıştırılmasıyla elde edilen değer birbirine yakın değerler olduğu görülmektedir.

4.4.4. Çok Ölçütlü Karar Modeli ve Optimizasyon

Bu çalışmada ortalama tamamlanan toplam proje sayısı, ortalama toplam proje tamamlanma süresi, ortalama toplam çalışan maliyeti, ortalama toplam kârlılık performans ölçütleri, dikkate alınan performans ölçütleri olarak belirlenmiştir. Bu kapsamda her performans ölçütünün ilgili amacına yönelik olarak optimum sonuçlara ve çalışan sayılarına ulaşmak için deney tasarımı, senaryo denemeleri yapılmış ve belirlenen karar değişkenleri baz alınarak çalışan sayıları kısıtları altındaki meta-model denklemleri çözülmüştür. Böylece her

performans ölçütü için optimum çalışan sayıları ve amaç fonksiyonu değerleri ayrı ayrı bulunmuştur. Çalışmanın bu kısmında ise tüm performans ölçütlerini aynı anda dikkate alan çok ölçütlü bir karar modeli geliştirilmiştir. Çok ölçütlü karar modeli geliştirirken Bileşenleri Ağırlıklandırma yaklaşımından yararlanılmıştır. Karar vericilerden hangi ölçütün daha önemli olduğu bilgisi alınarak ölçütler ağırlıklandırılmıştır.

Bileşenleri Ağırlıklandırma yaklaşımında; amaç fonksiyonu $F(x) = \text{ENB}\{f_1, f_2, \dots, f_k\}$ iken, w_i i'inci ölçütün ağırlığı ise $F(x) = w_1*f_1 + w_2*f_2 + \dots + w_k*f_k$ şeklinde yazılarak tek ölçüt haline dönüştürülür ve oluşturulan model çözülür. $F(x)$ aynı yönde fonksiyonlardan oluşur bu nedenle $\text{ENB}f_1 = -\text{ENK}f_1$ özelliğinden faydalanılır.

Performans ölçütleri aşağıdaki gibi f_1, f_2, f_3, f_4 fonksiyonlarıyla tanımlanmıştır:

$$f_1 = \text{TPS} = 59,15 - 11,91 x_1 - 0,315 x_2 - 1,31 x_3 - 0,33 x_4 + 3,414 x_1*x_3 + 4,024 x_1*x_4 \\ + 0,061 x_3*x_4 - 0,836 x_1*x_3*x_4$$

$$f_2 = \text{TTS} = 127756 - 30767 x_1 - 273 x_2 - 15921 x_3 - 22215 x_4 + 6639 x_1*x_3 + 8554 x_1*x_4 \\ + 4758 x_3*x_4 - 1935 x_1*x_3*x_4$$

$$f_3 = \text{TÇM} = 16361560 + 1307243 x_1 + 299884 x_2 + 437929 x_3 - 199111 x_4 - 72739 x_1*x_2 \\ - 56823 x_1*x_3 - 159253 x_2*x_3 + 422947 x_2*x_4 + 18752 x_1*x_2*x_3 - 80335 x_1*x_2*x_4 - \\ 51497 x_2*x_3*x_4 + 14666 x_1*x_2*x_3*x_4$$

$$f_4 = 472904759 - 116014353 x_1 - 757808 x_2 - 39596511 x_3 - 32508072 x_4 + 33653565 x_1*x_3 \\ + 33896955 x_1*x_4 + 8446409 x_3*x_4 - 8185804 x_1*x_3*x_4$$

Bu kapsamda çok ölçütlü karar modeli aşağıdaki şekilde tanımlanmaktadır:

$$2 \leq x_1 \leq 5 \quad (4.1)$$

$$2 \leq x_2 \leq 4 \quad (4.2)$$

$$2 \leq x_3 \leq 5 \quad (4.3)$$

$$2 \leq x_4 \leq 4 \quad (4.4)$$

$$x_i \geq 0 \text{ ve tamsayı, } \forall i \quad (4.5)$$

kısıtları altında

$$\text{ENB } f_1 = \text{TPS} = 59,15 - 11,91 x_1 - 0,315 x_2 - 1,31 x_3 - 0,33 x_4 + 3,414 x_1 * x_3 + 4,024 x_1 * x_4 \\ + 0,061 x_3 * x_4 - 0,836 x_1 * x_3 * x_4$$

$$\text{ENK } f_2 = \text{TTS} = 127756 - 30767 x_1 - 273 x_2 - 15921 x_3 - 22215 x_4 + 6639 x_1 * x_3 + 8554 x_1 * x_4 \\ + 4758 x_3 * x_4 - 1935 x_1 * x_3 * x_4$$

$$\text{ENK } f_3 = \text{TÇM} = 16361560 + 1307243 x_1 + 299884 x_2 + 437929 x_3 - 199111 x_4 - 72739 x_1 * x_2 \\ - 56823 x_1 * x_3 - 159253 x_2 * x_3 + 422947 x_2 * x_4 + 18752 x_1 * x_2 * x_3 - 80335 x_1 * x_2 * x_4 - \\ 51497 x_2 * x_3 * x_4 + 14666 x_1 * x_2 * x_3 * x_4$$

$$\text{ENB } f_4 = 472904759 - 116014353 x_1 - 757808 x_2 - 39596511 x_3 - 32508072 x_4 + 33653565 \\ x_1 * x_3 + 33896955 x_1 * x_4 + 8446409 x_3 * x_4 - 8185804 x_1 * x_3 * x_4$$

Tüm ölçütlerin fonksiyonlarını aynı anda enyilemek amacıyla tüm ölçütler ENB $F(x_1, x_2, x_3, x_4) = [f_1(x_1, x_2, x_3, x_4), f_2(x_1, x_2, x_3, x_4), f_3(x_1, x_2, x_3, x_4), f_4(x_1, x_2, x_3, x_4)]$ şeklinde tek ölçüte dönüştürülecek ve tek ölçüt için çözüm bulunacaktır. Bu kapsamda; tüm fonksiyonların aynı yönde olması için ENB $f_2 = - \text{ENK } f_2$ ve ENB $f_3 = - \text{ENK } f_3$ şeklinde belirlenmiştir. Karar vericilerle yapılan görüşmeler neticesinde performans ölçütlerini ifade eden f_1, f_2, f_3, f_4 fonksiyonları; $w_1 = 0.4, w_2 = 0.3, w_3 = 0.2, w_4 = 0.1$ şeklinde ağırlıklandırılmıştır.

Sonuç olarak karar modeli aşağıdaki gibi tanımlanmaktadır:

$$2 \leq x_1 \leq 5 \quad (4.1)$$

$$2 \leq x_2 \leq 4 \quad (4.2)$$

$$2 \leq x_3 \leq 5 \quad (4.3)$$

$$2 \leq x_4 \leq 4 \quad (4.4)$$

$$x_i \geq 0 \text{ ve tamsayı, } \forall i \quad (4.5)$$

kısıtları altında

$$\text{ENB } F(x_1, x_2, x_3, x_4) = 0,4 * [59,15 - 11,91 x_1 - 0,315 x_2 - 1,31 x_3 - 0,33 x_4 + 3,414 x_1 * x_3 + \\ 4,024 x_1 * x_4 + 0,061 x_3 * x_4 - 0,836 x_1 * x_3 * x_4] - 0,3 * [127756 - 30767 x_1 - 273 x_2 - 15921 x_3 -$$

$$\begin{aligned}
& 22215 x_4 + 6639 x_1 * x_3 + 8554 x_1 * x_4 + 4758 x_3 * x_4 - 1935 x_1 * x_3 * x_4] - 0,2 * [16361560 + \\
& 1307243 x_1 + 299884 x_2 + 437929 x_3 - 199111 x_4 - 72739 x_1 * x_2 - 56823 x_1 * x_3 - 159253 x_2 * x_3 \\
& + 422947 x_2 * x_4 + 18752 x_1 * x_2 * x_3 - 80335 x_1 * x_2 * x_4 - 51497 x_2 * x_3 * x_4 + 14666 x_1 * x_2 * x_3 * x_4] + \\
& 0,1 * [472904759 - 116014353 x_1 - 757808 x_2 - 39596511 x_3 - 32508072 x_4 + 33653565 x_1 * x_3 \\
& + 33896955 x_1 * x_4 + 8446409 x_3 * x_4 - 8185804 x_1 * x_3 * x_4] \quad (4.10)
\end{aligned}$$

Yukarıda tanımlanan karar modeli doğrusal olmayan karar modellerinin çözümünde kullanılan LİNGO programı ile çözülerek $x_1 = 5$, $x_2 = 2$, $x_3 = 2$, $x_4 = 4$ olarak bulunmuştur. LİNGO programının girdileri ve çıktıları EK 13'te yer almaktadır.

Karar değişkenlerinin değerlerinin ARENA Benzetim Programındaki benzetim modeline girilmesi ve modelin bu verilerle 5 replikasyonla 10 yıl için çalıştırılmasıyla Tablo 4.28.'deki sonuçlar elde edilmektedir.

Tablo 4.28. Çok ölçütlü karar modelinin optimum değerlerine göre benzetim modeli çıktıları

Karar Değişkenleri				Performans Ölçütleri			
x_1	x_2	x_3	x_4	TPS (adet)	TTS (saat)	TÇM (TL)	TK (TL)
5	2	2	4	77,200	48.840,253	22.990.616,965	438.372.138,621

Tablo 4.28. Bileşenleri ağırlıklandırma yaklaşımıyla çözülen çok ölçütlü karar modelinin karar değişkenlerinin alması önerilen değerlerini ve bu değerlerin benzetim modelindeki çıktıları göstermektedir. Bu çalışmada karar vericiler tarafından ölçütler önem sırasına göre; kârlılık, çalışan maliyeti, proje süresi ve proje sayısı olarak belirlenmiştir. Bu ölçütlerin 4.4.3. başlığındaki gibi ayrı ayrı ele alınması ve tümünün aynı anda ele alınmasıyla elde edilen benzetim modeli çıktıları Tablo 4.29.'da karşılaştırmalı olarak verilmektedir.

Tablo 4.29. Tek ölçütlü ve çok ölçütlü karar modellerinin benzetim çıktılarının karşılaştırılması

Performans Ölçütleri	Tek Ölçütlü Karar Modeli Çıktılarının Benzetim Modelinde Çalıştırılmasından Elde Edilen Ortalama Değerler	Çok Ölçütlü Karar Modeli Çıktılarının Benzetim Modelinde Çalıştırılmasından Elde Edilen Ortalama Değerler
TPS (adet)	77,200	77,200
TTS (saat)	29.399,936	48.840,253
TÇM (TL)	23.002.549,841	22.990.616,965
TK (TL)	438.372.138,621	438.372.138,621

Tablo 4.29. deęerlendirildięinde tek ölçütlü karar modelinin çıktılarıyla çok ölçütlü karar modelinin çıktıları arasında farklar olduęu görölmektedir. Bu kapsamda sonuçlar incelenerek farklı firmaların ya da farklı karar vericilerin bakış açılarına göre baz alınacak modeller deęiştcektir. Tüm ölçütleri göz önünde bulundurarak çalışan sayısını optimize etmek ya da ölçütler bazında ayrı ayrı eniyileme yapmak firmaların ve karar vericilerin ihtiyaçlarına ve tercihlerine göre şekillenecektir.

Çalışmanın bu bölümünde mevcut sistemi simüle eden benzetim modelinin çıktıları üzerinde durulmuş ve belirlenen performans ölçütlerinin amaçlarını eniyilemek üzere benzetim eniyilemesi yapılmıştır. Çıktı analizi, deney tasarımı, senaryo denemeleri ve meta-model oluşturma aşamaları her performans ölçütü için ayrıntılarıyla verilmiştir. Bu aşamaların sonucunda, her performans ölçütünün amaçlarına göre benzetim eniyilemesi yapılarak, sistemin eniyilemesi için Test ve Kurulum Ekiplerinin B ve C tecrübe seviyesinde olması gereken çalışan sayıları tespit edilmiştir. Her performans ölçütünün eniyilemesi ayrı ayrı yapılmış, en son aşamada ise tüm performans ölçütlerinin önem ağırlıklarına göre aynı anda göz önünde bulundurulduęu çok ölçütlü bir karar modeli geliştirilmiştir. Geliştirilen bu çok ölçütlü karar modelinin ölçütleri karar vericiler yardımıyla ağırlıklandırılmış ve model çözölerek çok ölçütlü karar modelinin optimum deęerini veren çalışan sayıları tespit edilmiştir. Tek ölçütlü karar modelleri ve çok ölçütlü karar modelinin çıktıları karşılaştırılarak hangi yöntemin kullanılacaęı karar vericilerin tercihinine bırakılmıştır.

5. SONUÇ VE ÖNERİLER

Bu çalışmada, çok projeli kaynak kısıtlı yazılım geliştirme projelerinin, yazılım geliştirme yaşam döngüsü “gerçekleştirim” sürecinin benzetim modeli oluşturulmuş ve benzetim modelinin 4 farklı performans ölçütüne göre benzetim eniyilemesi yoluyla iyileştirilmesi yapılmıştır.

5.1. Sonuçlar

Bu araştırma kapsamında yapılan benzetim modelinin çalıştırılmasıyla ve benzetim eniyilemesi yapılmasıyla aşağıdaki sonuçlara ulaşılmıştır:

1. Ortalama Tamamlanan Toplam Proje Sayısının en büyüklenmesi için; Test Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 5, Test Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 2, Kurulum Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 2, Kurulum Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 4 ve Ortalama Tamamlanan Toplam Proje Sayısı = 76,698 adet olarak bulunmuştur. Mevcut sistemde, Ortalama Tamamlanan Toplam Proje Sayısı = 54,6 adet iken benzetim eniyilemesiyle önerilen sistemde bu sayının arttığı ve tamamlanan proje sayısının artması hedeflendiği için sistemin iyileştiği görülmüştür.
2. Projelerin Ortalama Toplam Tamamlama Süresinin en küçüklenmesi için; Test Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 5, Test Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 4, Kurulum Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 2 ve Kurulum Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 2 ve Projelerin Ortalama Toplam Tamamlama Süresi = 28.819 saat olarak bulunmuştur. Mevcut sistemde, Projelerin Ortalama Toplam Tamamlama Süresi = 55.672,08 saat iken benzetim eniyilemesiyle önerilen sistemde bu sürenin azaldığı ve projelerin tamamlanma süresinin azalması hedeflendiği için sistemin iyileştiği görülmüştür.
3. Ortalama Toplam Çalışan Maliyetinin en küçüklenmesi için; Test Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 2, Test Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 4, Kurulum Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 5, Kurulum Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 2 ve Ortalama Toplam Çalışan Maliyeti =

19.593.499,00 TL olarak bulunmuştur. Mevcut sistemde, Ortalama Toplam Çalışan Maliyeti = 20.458.262,08 TL iken benzetim eniyilemesiyle önerilen sistemde bu maliyetin azaldığı ve çalışan maliyetinin azalması hedeflendiği için sistemin iyileştiği görülmüştür.

4. Ortalama Toplam Kârlılığın en büyüklenmesi için; Test Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 5, Test Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 2, Kurulum Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 2, Kurulum Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 4 ve Ortalama Toplam Kârlılık = 436.705.930,00 TL olarak bulunmuştur. Mevcut durumda Ortalama Toplam Kârlılık = 341.372.420,829 TL iken benzetim eniyilemesiyle önerilen sistemde bu karlılığın arttığı ve toplam kârlılığın artması hedeflendiği için sistemin iyileştiği görülmüştür.
5. Bileşenleri ağırlıklandırma yöntemi ile tüm ölçütleri aynı anda göz önünde bulunduran çok ölçütlü karar modeli geliştirilmiş ve ölçütlerin ağırlıkları (önem sırası) karar vericiler tarafından; ortalama toplam kârlılığın en büyüklenmesi, ortalama toplam çalışan maliyetinin en küçüklenmesi, projelerin ortalama toplam tamamlama süresinin en küçüklenmesi, ortalama tamamlanan toplam proje sayısının en büyüklenmesi şeklinde belirlenmiştir. Çok ölçütlü karar modelinin çözülmesiyle; Test Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 5, Test Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 2, Kurulum Ekibi-B Tecrübe Seviyesindeki Çalışan Sayısı = 2, Kurulum Ekibi-C Tecrübe Seviyesindeki Çalışan Sayısı = 4 olması durumunda çalışan sayısının optimum olduğu sonucuna ulaşılmıştır. Bu kapsamda karar vericilerin karar vermelerine destek olmak amacıyla her ölçütün bu çalışan sayılarına göre aldıkları değerler ve her ölçütün tek ölçütlü karar modeli olarak çözülmesine göre aldıkları değerler karşılaştırılmıştır. Böylece, çok ölçütlü karar modelinin ya da tek ölçütlü karar modellerinin kullanılması yönündeki karar ve tercih firmaya ve karar vericilere bırakılmış olup modellerin karar vermede yol gösterici olduğu değerlendirilmiştir.

5.2. Öneriler

Bu çalışmada elde edilen bulgular göz önüne alınarak aşağıdaki önerilerde bulunulabilir:

1. Gelecekteki bir çalışmada bu benzetim modeli gerçek hayattaki bir çalışma için uygulanabilir.
2. Yazılım geliştirme projeleri geliştirilirken, bu çalışmadaki Kodlama, Test, Yeniden Çalışma, Entegrasyon, Doğrulama, Yeniden Çalışma ve Kurulum aşamaları dikkate alınabilir.
3. Mevcut sistemin öncelikle en az tecrübeli olan çalışanı seçmesi kısıtı değiştirilerek, öncelikle çalışma takvimi uygun olan çalışanların seçilmesi kısıtı benzetim modeline eklenebilir.
4. Benzetim modeline bazı ek kısıtlar eklenerek, gelen tüm projelerin tamamlanmasını hedefleyen bir çalışma yapılabilir.
5. Gelecekte yapılacak çalışmalarda, bu çalışmada yapılan, yazılım geliştirme yaşam döngüsünün Gerçekleştirim sürecini gösteren benzetim modeline, sistemde çalışan insan kaynaklarının öğrenme veya yorgunluk gibi insan etkilerini göz önünde bulunduran bir davranışsal modeli eklenebilir.
6. Bu çalışma yazılım geliştirme yaşam döngüsünün Gerçekleştirim sürecini ele almıştır. Gelecekte yapılacak çalışmalarda yazılım yaşam döngüsünün tüm süreçlerini iyileştirmeyi hedefleyen bir benzetim modeli kurulabilir.

KAYNAKLAR

- [1] T. D. Ku, “Türk Dil Kurumu-Yazılım Tanımı”, *Türk Dil Kurumu*, 2020.
<https://sozluk.gov.tr/>.
- [2] O. Qasimov, “Bilgi sistemlerinin proje yönetimi incelenmesi (proje yönetimi programının üzerinden)”, Azerbaycan Devlet İktisat Üniversitesi, 2017.
- [3] O. Hazir, U. Eryilmaz, ve M. Hafizoğlu, *Proje yönetimi: analitik yaklaşımlar*, sayı Eylül. Türkiye: PMI-TR, 2014.
- [4] PMI, *A guide to the project body of project management knowledge (PMBOK Guide)*, 2000. baskı. Newtown Square, Pennsylvania USA, 2000.
- [5] Ö. Uysal, “Proje yönetimi bilgi alanları ile özgün bir projenin geliştirilmesi : e-okuma takvimi”, *Bilecik Şeyh Edebali Üniversitesi Sos. Bilim. Enstitüsü Derg.*, c. 4, sayı June 2019, ss. 293–313, 2019, doi: 10.33905/bseusbed.477077.
- [6] C. Gencer ve A. Kayacan, “Yazılım proje yönetimi: şelale modeli ve çevik yöntemlerin karşılaştırılması”, *Bilişim Teknol. Derg.*, c. 10, sayı 3, ss. 335–352, 2017, doi: 10.17671/gazibtd.331054.
- [7] P. Savolainen ve J. J. Ahonen, “Knowledge lost: challenges in changing project manager between sales and implementation in software projects”, *Int. J. Proj. Manag.*, c. 33, sayı 1, ss. 92–102, 2015, doi: 10.1016/j.ijproman.2014.04.003.
- [8] Brooks, F., “No Silver Bullet: Essence and Accidents of Software Engineering”, *Computer (Long. Beach. Calif.)*, c. 20, sayı 4, ss. 10–19, 1987.
- [9] O. A. Erdem ve A. E. Younis, “Yazılım projelerinin geliştirme sürecinde yönetim”, *Bilişim Teknol. Derg.*, c. 7, sayı 1, ss. 1–9, 2014, doi: 10.12973/bid.2011.
- [10] O. P. Sanchez, M. A. Terlizzi, ve H. R. de O. C. de Moraes, “Cost and time project

- management success factors for information systems development projects”, *Int. J. Proj. Manag.*, c. 35, sayı 8, ss. 1608–1626, 2017, doi: 10.1016/j.ijproman.2017.09.007.
- [11] A. Mishra ve D. Dubey, “A comparative study of different software development life cycle models in different scenarios”, *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, c. 1, sayı 5, ss. 64–69, 2013, doi: ISSN: 2321-7782 (Online).
- [12] E. Ersoy, “Yaşam döngüsü yönetimi kavramının yazılım ürünlerine uygulanması”, İstanbul Teknik Üniversitesi, 2002.
- [13] S. Hun vd., “Optimization of software development life cycle quality for NPP safety software based on a risk-cost model”, *Ann. Nucl. Energy*, c. 135, s. 106961, 2020, doi: 10.1016/j.anucene.2019.106961.
- [14] T. Hanne ve S. Nickel, “A multiobjective evolutionary algorithm for scheduling and inspection planning in software development projects”, c. 167, ss. 663–678, 2005, doi: 10.1016/j.ejor.2004.07.014.
- [15] R. H. Rasch ve H. L. Tosi, “Factors affecting software developers’ performance: an integrated approach”, *MIS Q.*, c. 16, sayı 3, ss. 395–413, 1992.
- [16] S. Nidumolu, “The effect of coordination and uncertainty on software project performance : residual performance risk as an intervening variable”, *Inf. Syst. Res.*, c. 6, sayı 3, ss. 191–219, 1995.
- [17] R. D. Banker ve C. F. Kemerer, “Scale economies in new software development”, *IEEE Trans. Softw. Eng.*, c. 15, sayı 10, ss. 416–429, 1996.
- [18] P. D. Chatzoglou, N. Theriou, E. Demitriades, ve E. Macedonia, “Software project management and planning : the case of the Greek IT sector software project management and planning : the case of the Greek IT sector Prodromos D . Chatzoglou *

- Nikolaos G . Theriou Vasilios Aggelides”, *Int. J. Appl. Syst. Stud.*, c. 1, sayı 3, ss. 305–316, 2007, doi: 10.1504/IJASS.2007.017713.
- [19] K. G. Kouskouras ve A. C. Georgiou, “A discrete event simulation model in the case of managing a software project”, *Eur. J. Oper. Res.*, c. 181, sayı 1, ss. 374–389, 2007, doi: 10.1016/j.ejor.2006.05.031.
- [20] C. K. Chang, H. yi Jiang, Y. Di, D. Zhu, ve Y. Ge, “Time-line based model for software project scheduling with genetic algorithms”, *Inf. Softw. Technol.*, c. 50, sayı 11, ss. 1142–1154, 2008, doi: 10.1016/j.infsof.2008.03.002.
- [21] M. Di Penta, M. Harman, ve G. Antoniol, “The use of search-based optimization techniques to schedule and staff software projects: an approach and an empirical study”, *Softw. - Pract. Exp.*, c. 41, ss. 495–519, 2011, doi: 10.1002/spe.1001.
- [22] J. Duggan, J. Byrne, ve G. J. Lyons, “feature”, 2004.
- [23] N. Jin ve X. Yao, “Heuristic optimization for software project management with impacts of team efficiency”, *Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014*, ss. 3016–3023, 2014, doi: 10.1109/CEC.2014.6900527.
- [24] F. Luna, D. L. González-Álvarez, F. Chicano, ve M. A. Vega-Rodríguez, “The software project scheduling problem: A scalability analysis of multi-objective metaheuristics”, *Appl. Soft Comput. J.*, c. 15, ss. 136–148, 2014, doi: 10.1016/j.asoc.2013.10.015.
- [25] D. Rodríguez, M. Ruiz, J. C. Riquelme, ve R. Harrison, “Multiobjective simulation optimisation in software project management”, içinde *Genetic and Evolutionary Computation Conference, GECCO’11, July 12-16, 2011*, 2011, sayı January, ss. 1883–1890, doi: 10.1145/2001576.2001829.
- [26] X. Shen, L. L. Minku, R. Bahsoon, ve X. Yao, “Dynamic software project scheduling

- through a proactive-rescheduling method”, *IEEE Trans. Softw. Eng.*, c. 42, sayı 7, ss. 658–686, 2016, doi: 10.1109/TSE.2015.2512266.
- [27] X. N. Shen, L. L. Minku, N. Marturi, Y. N. Guo, ve Y. Han, “A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling”, *Inf. Sci. (Ny)*, c. 428, ss. 1–29, 2018, doi: 10.1016/j.ins.2017.10.041.
- [28] F. Wen ve C.-M. Lin, “Multistage Human Resource Allocation for Software Development by Multiobjective Genetic Algorithm”, *Open Appl. Math. J.*, c. 2, sayı 1, ss. 95–103, 2008, doi: 10.2174/1874114200802010095.
- [29] J. Xiao, X. T. Ao, ve Y. Tang, “Solving software project scheduling problems with ant colony optimization”, *Comput. Oper. Res.*, c. 40, sayı 1, ss. 33–46, 2013, doi: 10.1016/j.cor.2012.05.007.
- [30] E. Alba ve J. Francisco Chicano, “Software project management with GAs”, *Inf. Sci. (Ny)*, c. 177, sayı 11, ss. 2380–2401, 2007, doi: 10.1016/j.ins.2006.12.020.
- [31] A. C. Başar, “Dünyada ve Türkiye’de benzetim kullanım analizi: Türkiye için bir değerlendirme”, Başkent Üniversitesi, 2019.
- [32] M. A. Law ve W. D. Kelton, *Simulation Analysis*, 2. baskı, c. 1. 2, 1991.
- [33] B. Dengiz, “Sistem Benzetimi Dersi Ders Notları”. s. Başkent Üniversitesi, 2017.
- [34] M. Aksarayli ve A. Yildiz, “Process optimization with simulation modeling in a manufacturing system”, *Res. J. Appl. Sci. Eng. Technol.*, c. 3, sayı 4, ss. 318–329, 2011.
- [35] A. M. Law, “Statistical analysis of simulation output data: The practical state of the art”, içinde *Proceedings - Winter Simulation Conference*, 2015, c. 2015-Şubat, ss. 1810–1819, doi: 10.1109/WSC.2015.7408297.
- [36] B. Şimşek ve Y. T. İç, “Multi-response simulation optimization approach for the

- performance optimization of an alarm monitoring center”, *Saf. Sci.*, c. 66, ss. 61–74, 2014, doi: 10.1016/j.ssci.2014.02.001.
- [37] B. Dengiz ve O. Belgin, “Simulation optimization of a multi-stage multi-product paint shop line with response surface methodology”, *Simul. Trans. of the Soc. Model. Simul. Int.*, c. 90, sayı 3, ss. 265–274, 2014, doi: 10.1177/0037549713516508.
- [38] B. Dengiz ve K. S. Akbay, “Computer simulation of a PCB production line: Metamodeling approach”, *Int. J. Prod. Econ.*, c. 63, sayı 2, ss. 195–205, 2000, doi: 10.1016/S0925-5273(99)00013-4.
- [39] B. Dengiz, T. Bektas, ve A. E. Ultanir, “Simulation optimization based DSS application: A diamond tool production line in industry”, *Simul. Model. Pract. Theory*, c. 14, sayı 3, ss. 296–312, 2006, doi: 10.1016/j.simpat.2005.07.001.
- [40] M. I. Kellner, R. J. Madachy, ve D. M. Raffo, “Software process simulation modeling: Why? What? How?”, *J. Syst. Softw.*, c. 46, sayı 2, ss. 91–105, 1999, doi: 10.1016/S0164-1212(99)00003-5.
- [41] R. Agarwal, “A flexible model for multi-agent based simulation of software development process”, Auburn University, 2007.
- [42] I. Rus, H. Neu, ve J. Münch, “A Systematic Methodology for Developing Discrete Event Simulation Models of Software Development Processes”, “*A Syst. Methodol. Dev. Discret. event Simul. Model. Softw. Dev. Process. 4th Int. Work. Softw. Process Simul. Model. (ProSim 2003), Proceedings, Portland, Oregon, USA, May, 3-4, 2003, doi 10, 2003, [Çevrimiçi]. Available at: <http://arxiv.org/abs/1403.3559>.*
- [43] H. T. Tsai, H. Moskowitz, ve L. H. Lee, “Human resource selection for software development projects using Taguchi’s parameter design”, *Eur. J. Oper. Res.*, c. 151,

sayı 1, ss. 167–180, 2003, doi: 10.1016/S0377-2217(02)00600-8.

- [44] H. Neu, T. Hanne, J. Münch, S. Nickel, ve A. Wirsén, “Creating a code inspection model for simulation-based decision support”, içinde *Proceedings of the 4th International Workshop on Software Process Simulation and Modelling ProSim’03, Portland State University, May 3-4, 2003*, 2003, sayı June 2014, ss. 1–10, [Çevrimiçi]. Available at: <http://en.scientificcommons.org/20345164>.
- [45] R. Cherif, “Software process simulation modelling : a multi agent-based simulation approach”, Blekinge Institute of Technology, 2008.
- [46] B. Spasic ve B. S. S. Onggo, “Agent-based simulation of the software development process: a case study at AVL”, içinde *2012 Winter Simulation Conference, Proceedings, Berlin, Germany, Dec., 9-12, 2012*, doi: 10.1109/WSC.2012.6465117 [Online]. Available: <https://www.researchgate.net/publication/261500068>, 2012, ss. 1–12, [Çevrimiçi]. Available at: <http://ieeexplore.ieee.org.etechniccyt.idm.oclc.org/xpl/articleDetails.jsp?arnumber=6465229&tag=1>.
- [47] W. Song, H. Xi, D. Kang, ve J. Zhang, “An agent-based simulation system for multi-project scheduling under uncertainty”, *Simul. Model. Pract. Theory*, c. 86, sayı November 2017, ss. 187–203, 2018, doi: 10.1016/j.simpat.2018.05.009.
- [48] J. F. Gonçalves, J. J. M. Mendes, ve M. G. C. Resende, “A genetic algorithm for the resource constrained multi-project scheduling problem”, *Eur. J. Oper. Res.*, c. 189, sayı 3, ss. 1171–1190, 2008, doi: 10.1016/j.ejor.2006.06.074.
- [49] V. Yannibelli ve A. Amandi, “A knowledge-based evolutionary assistant to software development project scheduling”, *Expert Syst. Appl.*, c. 38, sayı 7, ss. 8403–8413, 2011,

doi: 10.1016/j.eswa.2011.01.035.

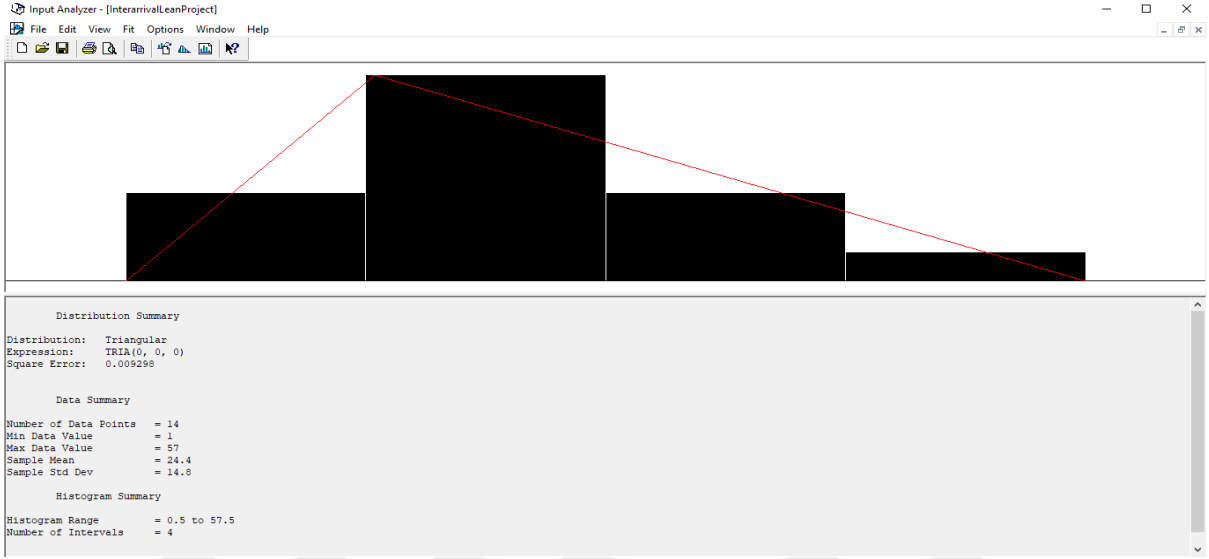
- [50] P. Akyıl Kurt, “Çok projeli kaynak kısıtlı proje çizelgeleme problemi: bir yazılım firmasında uygulama çalışması”, Başkent Üniversitesi, 2018.
- [51] A. Singh, “Managing the environmental problem of seawater intrusion in coastal aquifers through simulation-optimization modeling”, *Ecol. Indic.*, c. 48, ss. 498–504, 2015, doi: 10.1016/j.ecolind.2014.09.011.
- [52] M. Balaban, “Regresyon ve kriging meta-modelleri için kullanılan deney tasarımı yöntemleri”, *Düzce Üniversitesi Bilim ve Teknol. Derg.*, c. 7, sayı 3, ss. 1444–1455, 2019, doi: 10.29130/dubited.528940.
- [53] M. Balaban, “Kesirli universal kriging meta- modeli ve benzetim eniyilemesi”, Başkent Üniversitesi Fen Bilimleri Enstitüsü, 2018.
- [54] M. R. Pourhassan ve S. Raissi, “An integrated simulation-based optimization technique for multi-objective dynamic facility layout problem”, *J. Ind. Inf. Integr.*, c. 8, ss. 49–58, 2017, doi: 10.1016/j.jii.2017.06.001.
- [55] L. W. Friedman ve I. Pressman, “The meta-model in simulation analysis: can it be trusted?”, *J. Oper. Res. Soc.*, c. 39, sayı 10, ss. 939–948, 1988.
- [56] B. Dengiz, Y. T. İç, ve O. Belgin, “A meta-model based simulation optimization using hybrid simulation-analytical modeling to increase the productivity in automotive industry”, *Math. Comput. Simul.*, c. 120, ss. 120–128, 2015, doi: 10.1016/j.matcom.2015.07.005.
- [57] A. Aydemir Karadağ, B. Çakır, Y. T. İç, ve B. Dengiz, “Benzetim eniyilemesi ile bir işletmede sistem tasarımı”, içinde *YA/EM 2011 Yöneylem Araştırması ve Endüstri Mühendisliği 31. Ulusal Kongresi*, 2011, s. 25.

- [58] B. Dengiz, “Redesign of PCB production line with simulation and taguchi design”, içinde *Proceedings - Winter Simulation Conference*, 2009, sayı May, ss. 2197–2204, doi: 10.1109/WSC.2009.5429646.
- [59] J. P. C. Kleijnen ve R. G. Sargent, “A methodology for fitting and validating metamodels in simulation”, *Eur. J. Oper. Res.*, c. 120, sayı 1, ss. 14–29, 2000, doi: 10.1016/S0377-2217(98)00392-0.

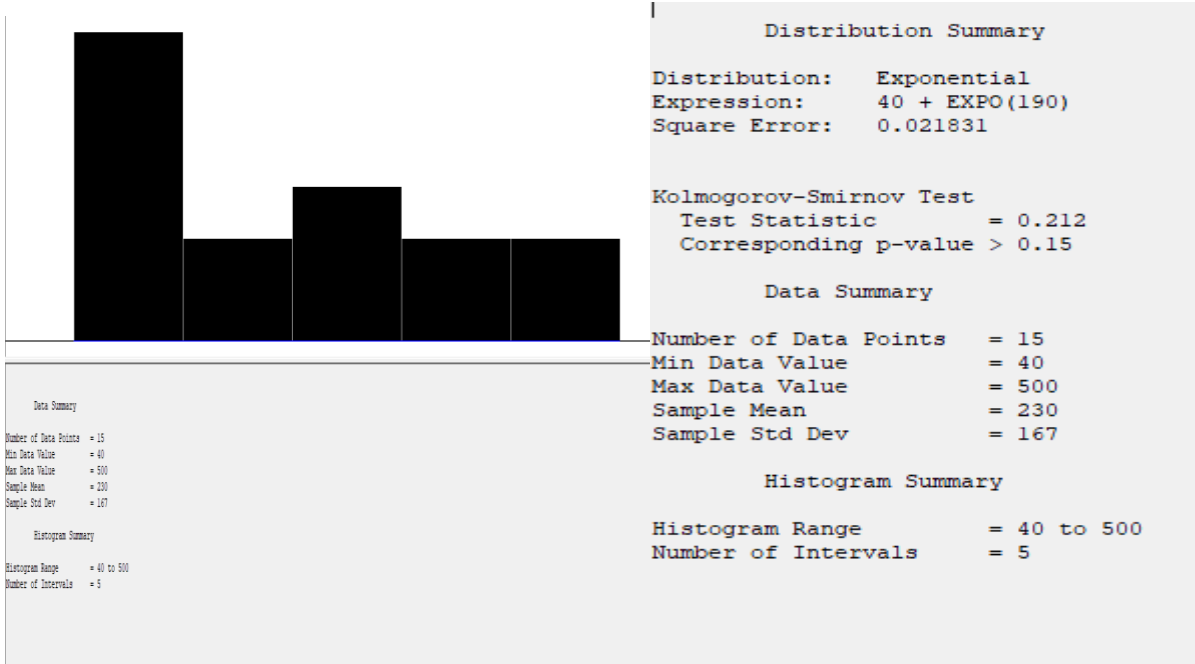


EKLER

EK 1: Yalın Proje Türü Geliş Zamanlarına Göre Girdi Analizi



EK 2: Yalın Proje Türü Proje Bedellerine Göre Girdi Analizi



EK 3: ARENA Benzetim Modeli Çıktısı

Category Overview

Values Across All Replications

Yuksek Lisans

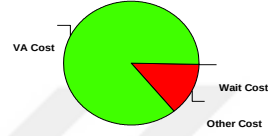
Replications: 5 Time Units: Hours

Key Performance Indicators

All Entities

Average

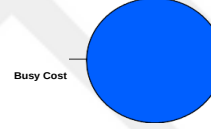
Non-Value Added Cost	0
Other Cost	2,864,550
Transfer Cost	0
Value Added Cost	17,584,041
Wait Cost	9,672
Total Cost	20,458,262



All Resources

Average

Busy Cost	20,458,262 *
Idle Cost	0
Usage Cost	0



Total Cost	20,458,262
------------	------------

* these costs are included in Entity Costs above.

System

Average

Total Cost	20,458,262
Number Out	55

Yuksek Lisans

Time Units: Hours

Replications: 5

Entity**Time**

VA Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	7941.36	184,50	7726.52	8078.73	7319.91	8489.91
proje_orta	6067.93	127,62	5911.06	6193.07	5428.98	6892.88
proje_yalin	3118.18	26,35	3088.96	3146.25	2625.75	3473.41

NVA Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	0.00	0,00	0.00	0.00	0.00	0.00
proje_orta	0.00	0,00	0.00	0.00	0.00	0.00
proje_yalin	0.00	0,00	0.00	0.00	0.00	0.00

Wait Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	14429.18	6.809,84	8491.36	22870.20	7763.47	23413.47
proje_orta	11247.27	1.883,62	8914.20	12602.82	1149.44	20511.44
proje_yalin	10605.32	2.648,94	7585.01	13404.21	416.26	24082.26

Transfer Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	0.00	0,00	0.00	0.00	0.00	0.00
proje_orta	0.00	0,00	0.00	0.00	0.00	0.00
proje_yalin	0.00	0,00	0.00	0.00	0.00	0.00

Other Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	1001.42	13,26	985.36	1013.87	891.10	1040.10
proje_orta	831.20	31,78	796.66	854.18	624.22	1064.22
proje_yalin	430.22	12,92	420.98	443.45	365.81	534.81

Total Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	23371.97	6.724,91	17351.53	31582.08	16754.63	31762.08
proje_orta	18146.40	1.841,99	15835.96	19326.89	7965.90	28119.89
proje_yalin	14153.71	2.625,07	11174.72	16933.84	3872.38	27883.84

Cost

VA Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	336122.78	9.594,29	327177.07	347072.60	312892.44	359100.44
proje_orta	176801.58	4.442,56	172522.92	180748.41	146324.91	207150.41
proje_yalin	85167.12	815,64	84663.26	86308.90	73947.80	102865.80

Yuksek Lisans

Replications: 5 Time Units: Hours

Entity

Cost

NVA Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	0.00	0,00	0.00	0.00	0.00	0.
proje_orta	0.00	0,00	0.00	0.00	0.00	0.
proje_yalin	0.00	0,00	0.00	0.00	0.00	0.
Wait Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	2809.53	124,67	2710.25	2978.50	2479.00	3071.
proje_orta	0.00	0,00	0.00	0.00	0.00	0.
proje_yalin	0.00	0,00	0.00	0.00	0.00	0.
Other Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	60318.99	5.589,71	54016.18	63873.67	32970.72	65556.
proje_orta	32639.81	4.106,76	28145.35	35732.81	10611.78	67037.
proje_yalin	18060.22	1.913,64	16580.43	19999.64	10242.75	33680.
Transfer Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	0.00	0,00	0.00	0.00	0.00	0.
proje_orta	0.00	0,00	0.00	0.00	0.00	0.
proje_yalin	0.00	0,00	0.00	0.00	0.00	0.
Total Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
proje_kompleks	399251.31	10.152,54	393248.13	413414.38	370080.23	424358.
proje_orta	209441.38	6.379,75	200668.27	214275.34	181896.43	263950.
proje_yalin	103227.33	2.320,42	101392.41	105731.14	88921.37	125135.

Other

Yuksek Lisans

Replications: 5 Time Units: Hours

Entity

Other

Number In			Minimum	Maximum
	Average	Half Width	Average	Average
proje_kompleks	31.4000	8,07	23.0000	40.0000
proje_orta	83.2000	8,48	76.0000	91.0000
proje_yalin	150.80	4,34	146.00	155.00



Number Out				
	Average	Half Width	Average	Average
proje_kompleks	3.4000	1,11	2.0000	4.0000
proje_orta	15.2000	2,96	13.0000	19.0000
proje_yalin	36.0000	6,02	31.0000	44.0000

WIP			Minimum	Maximum	Minimum	Maxim
	Average	Half Width	Average	Average	Value	Va
proje_kompleks	14.9699	4,94	10.1375	19.9119	0.00	39.00
proje_orta	37.5103	1,99	34.7386	38.7660	0.00	75.00
proje_yalin	59.6519	6,75	50.6606	65.4238	0.00	119.

Yuksek Lisans

Replications: 5 Time Units: Hours

Queue

Time

Waiting Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Dogrulama Süreci Kompleks.Queue	8516.91	2.495,43	5807.96	10899.66	2217.12	16479.
Dogrulama Süreci Orta.Queue	8440.10	1.227,28	6781.05	9202.11	516.63	15631.
Dogrulama Süreci Yalin.Queue	7162.22	2.012,54	4725.05	9171.93	71.6668	16506.
Entegrasyon Süreci Kompleks.Queue	476.87	166,68	315.68	639.97	0.00	1856.
Entegrasyon Süreci Orta.Queue	141.57	57,16	83.3531	204.85	0.00	756.
Entegrasyon Süreci Yalin.Queue	303.71	87,80	211.72	402.99	0.00	1461.
Kodlama Süreci Kompleks.Queue	1.0106	2,59	0.02002925	4.7462	0.00	140.
Kodlama Süreci Orta.Queue	0.06358480	0,02	0.03878619	0.08183742	0.00	0.99
Kodlama Süreci Yalin.Queue	4.2807	2,21	2.5255	6.6286	0.00	262.
Kompleks_Kodlama Yapani Yeniden Calisma1 icin Tutma.Queue	1058.96	492,48	564.14	1622.63	0.00	3080.
Kompleks_Kodlama Yapani Yeniden Calisma2 icin Tutma.Queue	1249.26	484,31	633.51	1623.92	0.00	2929.
Kurulum Süreci Kompleks.Queue	4225.01	4.840,07	1614.72	11045.84	155.04	12796.
Kurulum Süreci Orta.Queue	173.20	75,31	105.79	257.39	0.00	846.
Kurulum Süreci Yalin.Queue	399.81	97,77	321.29	519.86	0.00	1287.
Orta_Kodlama Yapani Yeniden Calisma1 icin Tutma.Queue	502.97	141,41	314.37	618.09	0.00	3139.
Orta_Kodlama Yapani Yeniden Calisma2 icin Tutma.Queue	704.36	254,05	521.58	1040.58	0.00	3122.
Test Süreci Kompleks.Queue	6603.20	1.630,80	4751.96	8302.49	0.00	15755.
Test Süreci Orta.Queue	6859.36	987,34	5785.36	7997.55	0.00	16182.
Test Süreci Yalin.Queue	6792.52	1.461,56	4908.77	8113.75	0.00	16702.
Yalin_Kodlama Yapani Yeniden Calisma1 icin Tutma.Queue	605.73	114,07	505.42	721.96	0.00	3560.
Yalin_Kodlama Yapani Yeniden Calisma2 icin Tutma.Queue	726.77	138,12	561.75	874.74	0.00	3136.

Cost

Yuksek Lisans

Replications: 5 Time Units: Hours

Queue

Cost

Waiting Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Dogrulama Süreci	0.00	0,00	0.00	0.00	0.00	0.
Kompleks.Queue						
Dogrulama Süreci Orta.Queue	0.00	0,00	0.00	0.00	0.00	0.
Dogrulama Süreci Yalin.Queue	0.00	0,00	0.00	0.00	0.00	0.
Entegrasyon Süreci	0.00	0,00	0.00	0.00	0.00	0.
Kompleks.Queue						
Entegrasyon Süreci Orta.Queue	0.00	0,00	0.00	0.00	0.00	0.
Entegrasyon Süreci Yalin.Queue	0.00	0,00	0.00	0.00	0.00	0.
Kodlama Süreci	0.00	0,00	0.00	0.00	0.00	0.
Kompleks.Queue						
Kodlama Süreci Orta.Queue	0.00	0,00	0.00	0.00	0.00	0.
Kodlama Süreci Yalin.Queue	0.00	0,00	0.00	0.00	0.00	0.
Kompleks_Kodlama Yapani	0.00	0,00	0.00	0.00	0.00	0.
Yeniden Calisma1 icin						
Tutma.Queue						
Kompleks_Kodlama Yapani	0.00	0,00	0.00	0.00	0.00	0.
Yeniden Calisma2 icin						
Tutma.Queue						
Kurulum Süreci	0.00	0,00	0.00	0.00	0.00	0.
Kompleks.Queue						
Kurulum Süreci Orta.Queue	0.00	0,00	0.00	0.00	0.00	0.
Kurulum Süreci Yalin.Queue	0.00	0,00	0.00	0.00	0.00	0.
Orta_Kodlama Yapani Yeniden	0.00	0,00	0.00	0.00	0.00	0.
Calisma1 icin Tutma.Queue						
Orta_Kodlama Yapani Yeniden	0.00	0,00	0.00	0.00	0.00	0.
Calisma2 icin Tutma.Queue						
Test Süreci Kompleks.Queue	0.00	0,00	0.00	0.00	0.00	0.
Test Süreci Orta.Queue	0.00	0,00	0.00	0.00	0.00	0.
Test Süreci Yalin.Queue	0.00	0,00	0.00	0.00	0.00	0.
Yalin_Kodlama Yapani Yeniden	0.00	0,00	0.00	0.00	0.00	0.
Calisma1 icin Tutma.Queue						
Yalin_Kodlama Yapani Yeniden	0.00	0,00	0.00	0.00	0.00	0.
Calisma2 icin Tutma.Queue						

Other

Yuksek Lisans

Replications: 5 Time Units: Hours

Queue

Other

Number Waiting	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Dogrulama Süreci Kompleks.Queue	2.7558	1,41	1.5664	4.0755	0.00	8.00
Dogrulama Süreci Orta.Queue	9.6119	1,49	7.8351	10.8066	0.00	26.00
Dogrulama Süreci Yalin.Queue	15.5411	2,51	11.9532	16.8178	0.00	38.00
Entegrasyon Süreci Kompleks.Queue	0.1785	0,07	0.1057	0.2533	0.00	3.00
Entegrasyon Süreci Orta.Queue	0.1672	0,06	0.0964	0.2335	0.00	3.00
Entegrasyon Süreci Yalin.Queue	0.7021	0,25	0.4318	0.9669	0.00	6.00
Kodlama Süreci Kompleks.Queue	0.00120372	0,00	0.00002134	0.00577923	0.00	1.00
Kodlama Süreci Orta.Queue	0.00016140	0,00	0.00008973	0.00021674	0.00	1.00
Kodlama Süreci Yalin.Queue	0.01974857	0,01	0.01122449	0.03127645	0.00	2.00
Kompleks_Kodlama Yapani Yeniden Calisma1 icin Tutma.Queue	0.3969	0,19	0.2233	0.6421	0.00	2.00
Kompleks_Kodlama Yapani Yeniden Calisma2 icin Tutma.Queue	0.1781	0,09	0.1157	0.2966	0.00	3.00
Kurulum Süreci Kompleks.Queue	0.6792	0,66	0.3682	1.6232	0.00	5.00
Kurulum Süreci Orta.Queue	0.08139811	0,04	0.04830398	0.1204	0.00	4.00
Kurulum Süreci Yalin.Queue	0.4527	0,15	0.3130	0.5795	0.00	5.00
Orta_Kodlama Yapani Yeniden Calisma1 icin Tutma.Queue	0.6608	0,20	0.4582	0.9007	0.00	4.00
Orta_Kodlama Yapani Yeniden Calisma2 icin Tutma.Queue	0.3471	0,11	0.2540	0.4638	0.00	4.00
Test Süreci Kompleks.Queue	7.0208	2,14	4.5857	8.6692	0.00	25.00
Test Süreci Orta.Queue	17.8760	0,84	16.6840	18.3249	0.00	46.00
Test Süreci Yalin.Queue	31.6179	5,38	24.8378	36.8121	0.00	70.00
Yalin_Kodlama Yapani Yeniden Calisma1 icin Tutma.Queue	1.4778	0,31	1.1981	1.7854	0.00	6.00
Yalin_Kodlama Yapani Yeniden Calisma2 icin Tutma.Queue	0.8611	0,30	0.6179	1.2581	0.00	5.00

Yuksek Lisans

Replications: 5 Time Units: Hours

Resource

Usage

Instantaneous Utilization	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Deployer_B	0.8523	0,02	0.8411	0.8735	0.00	1.00
Deployer_C	0.8595	0,03	0.8252	0.8891	0.00	1.00
Deployer_D	0.8407	0,02	0.8246	0.8583	0.00	1.00
Deployer_F	0.8030	0,03	0.7717	0.8218	0.00	1.00
Developer_A	0.8943	0,01	0.8820	0.9084	0.00	1.00
Developer_B	0.7920	0,03	0.7682	0.8331	0.00	1.00
Developer_C	0.4959	0,07	0.4265	0.5588	0.00	1.00
Developer_D	0.4562	0,11	0.3056	0.5209	0.00	1.00
Developer_E	0.1548	0,09	0.07437410	0.2663	0.00	1.00
Developer_F	0.08330445	0,05	0.04914970	0.1450	0.00	1.00
Tester_B	0.9726	0,00	0.9675	0.9757	0.00	1.00
Tester_C	0.9611	0,01	0.9560	0.9688	0.00	1.00
Tester_D	0.9535	0,00	0.9500	0.9582	0.00	1.00
Tester_E	0.8414	0,01	0.8338	0.8462	0.00	1.00
Number Busy	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Deployer_B	1.6147	0,03	1.5927	1.6513	0.00	2.00
Deployer_C	1.6273	0,06	1.5631	1.6820	0.00	2.00
Deployer_D	1.5918	0,04	1.5611	1.6264	0.00	2.00
Deployer_F	0.8030	0,03	0.7717	0.8218	0.00	1.00
Developer_A	1.6955	0,02	1.6741	1.7206	0.00	2.00
Developer_B	2.2429	0,10	2.1715	2.3681	0.00	3.00
Developer_C	1.8285	0,25	1.5775	2.0581	0.00	4.00
Developer_D	1.2838	0,32	0.8586	1.4612	0.00	3.00
Developer_E	0.3016	0,17	0.1418	0.5159	0.00	2.00
Developer_F	0.08330445	0,05	0.04914970	0.1450	0.00	1.00
Tester_B	1.8374	0,01	1.8280	1.8433	0.00	2.00
Tester_C	1.8160	0,02	1.8061	1.8303	0.00	2.00
Tester_D	2.7546	0,01	2.7443	2.7681	0.00	3.00
Tester_E	0.8414	0,01	0.8338	0.8462	0.00	1.00

Yuksek Lisans

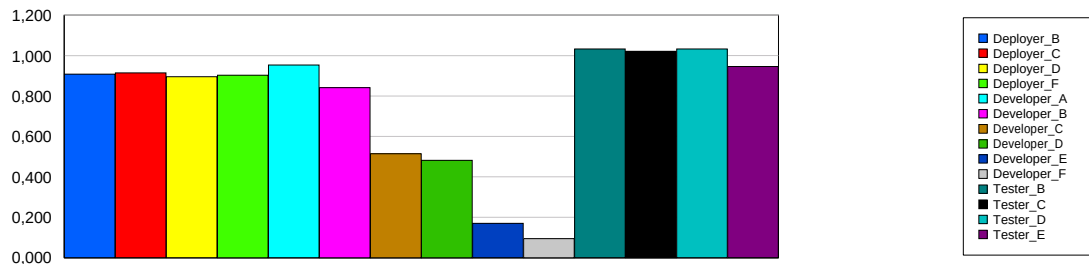
Replications: 5 Time Units: Hours

Resource

Usage

Number Scheduled	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Deployer_B	1.7778	0,00	1.7778	1.7778	0.00	2.00
Deployer_C	1.7778	0,00	1.7778	1.7778	0.00	2.00
Deployer_D	1.7778	0,00	1.7778	1.7778	0.00	2.00
Deployer_F	0.8889	0,00	0.8889	0.8889	0.00	1.00
Developer_A	1.7778	0,00	1.7778	1.7778	0.00	2.00
Developer_B	2.6667	0,00	2.6667	2.6667	0.00	3.00
Developer_C	3.5556	0,00	3.5556	3.5556	0.00	4.00
Developer_D	2.6667	0,00	2.6667	2.6667	0.00	3.00
Developer_E	1.7778	0,00	1.7778	1.7778	0.00	2.00
Developer_F	0.8889	0,00	0.8889	0.8889	0.00	1.00
Tester_B	1.7778	0,00	1.7778	1.7778	0.00	2.00
Tester_C	1.7778	0,00	1.7778	1.7778	0.00	2.00
Tester_D	2.6667	0,00	2.6667	2.6667	0.00	3.00
Tester_E	0.8889	0,00	0.8889	0.8889	0.00	1.00

Scheduled Utilization	Average	Half Width	Minimum Average	Maximum Average
Deployer_B	0.9083	0,02	0.8959	0.9288
Deployer_C	0.9154	0,03	0.8792	0.9461
Deployer_D	0.8954	0,02	0.8781	0.9149
Deployer_F	0.9034	0,03	0.8681	0.9245
Developer_A	0.9537	0,01	0.9417	0.9678
Developer_B	0.8411	0,04	0.8143	0.8880
Developer_C	0.5143	0,07	0.4437	0.5788
Developer_D	0.4814	0,12	0.3220	0.5480
Developer_E	0.1697	0,10	0.07974963	0.2902
Developer_F	0.0937	0,05	0.05529342	0.1631
Tester_B	1.0335	0,00	1.0282	1.0369
Tester_C	1.0215	0,01	1.0159	1.0296
Tester_D	1.0330	0,00	1.0291	1.0380
Tester_E	0.9466	0,01	0.9380	0.9520



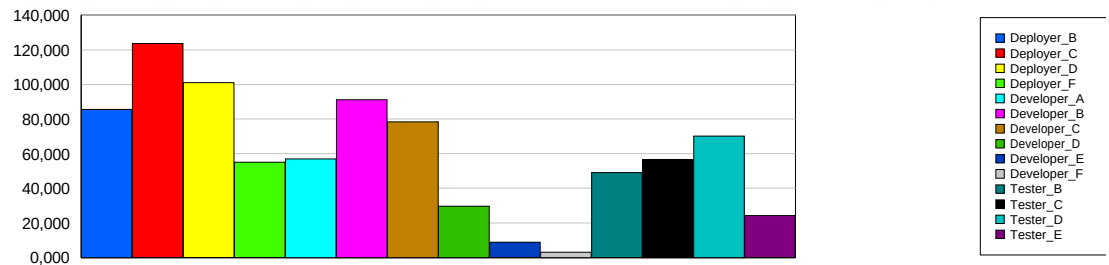
Yuksek Lisans

Replications: 5 Time Units: Hours

Resource

Usage

Total Number Seized			Minimum	Maximum
	Average	Half Width	Average	Average
Deployer_B	85.6000	4,93	80.0000	91.0000
Deployer_C	123.60	8,26	112.00	128.00
Deployer_D	101.00	7,19	91.0000	105.00
Deployer_F	55.0000	6,63	48.0000	63.0000
Developer_A	57.0000	4,73	52.0000	62.0000
Developer_B	91.2000	3,44	88.0000	95.0000
Developer_C	78.4000	11,43	66.0000	90.0000
Developer_D	29.6000	7,27	20.0000	34.0000
Developer_E	8.8000	4,06	4.0000	13.0000
Developer_F	3.0000	1,52	2.0000	5.0000
Tester_B	49.0000	0,88	48.0000	50.0000
Tester_C	56.6000	3,35	54.0000	61.0000
Tester_D	70.2000	6,17	65.0000	76.0000
Tester_E	24.2000	3,09	21.0000	26.0000



Cost

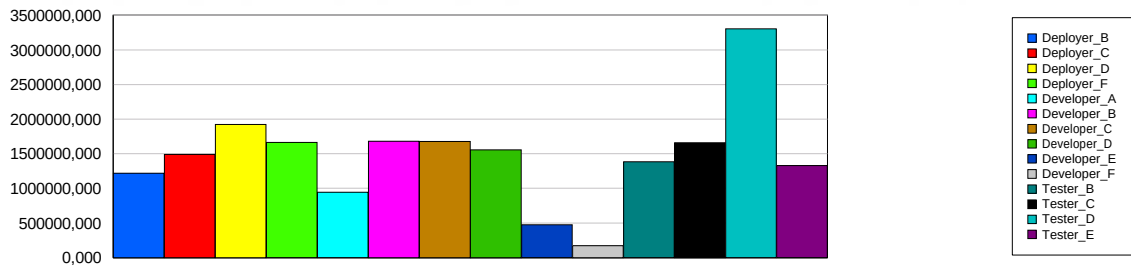
Yuksek Lisans

Replications: 5 Time Units: Hours

Resource

Cost

Busy Cost			Minimum	Maximum
	Average	Half Width	Average	Average
Deployer_B	1216030.20	23.223,56	1193866.45	1244124.54
Deployer_C	1489171.70	55.582,83	1424241.32	1546886.86
Deployer_D	1921704.41	48.806,13	1870288.72	1976532.56
Deployer_F	1661689.19	52.032,21	1596787.94	1700470.07
Developer_A	943020.37	12.806,56	930335.07	956440.22
Developer_B	1680851.17	70.203,89	1636111.46	1763676.56
Developer_C	1676157.01	225.162,86	1451003.76	1892910.88
Developer_D	1555624.51	382.364,33	1043467.29	1752413.74
Developer_E	473505.17	276.694,83	223554.16	813434.59
Developer_F	172352.33	93.705,04	101465.77	300112.64
Tester_B	1381272.30	13.430,29	1365618.70	1392338.53
Tester_C	1656877.73	19.086,60	1636036.60	1679400.90
Tester_D	3303426.00	29.977,93	3261103.78	3318553.51
Tester_E	1326579.99	10.686,95	1314485.64	1334152.31



Idle Cost				
	Average	Half Width	Average	Average
Deployer_B	0.00	0,00	0.00	0.00
Deployer_C	0.00	0,00	0.00	0.00
Deployer_D	0.00	0,00	0.00	0.00
Deployer_F	0.00	0,00	0.00	0.00
Developer_A	0.00	0,00	0.00	0.00
Developer_B	0.00	0,00	0.00	0.00
Developer_C	0.00	0,00	0.00	0.00
Developer_D	0.00	0,00	0.00	0.00
Developer_E	0.00	0,00	0.00	0.00
Developer_F	0.00	0,00	0.00	0.00
Tester_B	0.00	0,00	0.00	0.00
Tester_C	0.00	0,00	0.00	0.00
Tester_D	0.00	0,00	0.00	0.00
Tester_E	0.00	0,00	0.00	0.00

Yuksek Lisans

Replications: 5 Time Units: Hours

Resource

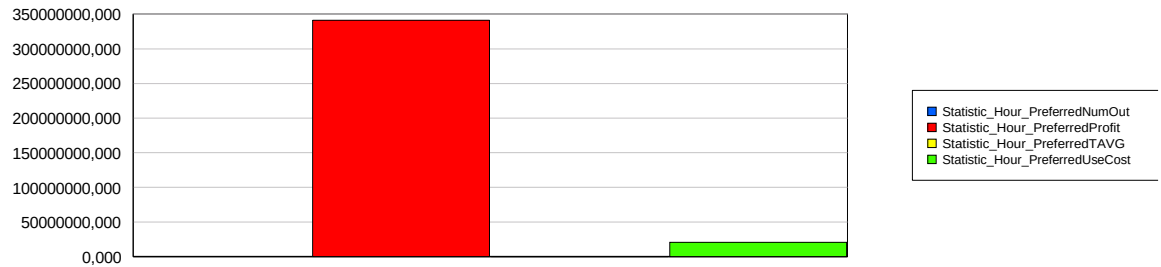
Cost

Usage Cost			Minimum	Maximum
	Average	Half Width	Average	Average
Deployer_B	0.00	0,00	0.00	0.00
Deployer_C	0.00	0,00	0.00	0.00
Deployer_D	0.00	0,00	0.00	0.00
Deployer_F	0.00	0,00	0.00	0.00
Developer_A	0.00	0,00	0.00	0.00
Developer_B	0.00	0,00	0.00	0.00
Developer_C	0.00	0,00	0.00	0.00
Developer_D	0.00	0,00	0.00	0.00
Developer_E	0.00	0,00	0.00	0.00
Developer_F	0.00	0,00	0.00	0.00
Tester_B	0.00	0,00	0.00	0.00
Tester_C	0.00	0,00	0.00	0.00
Tester_D	0.00	0,00	0.00	0.00
Tester_E	0.00	0,00	0.00	0.00

User Specified

Output

Output			Minimum	Maximum
	Average	Half Width	Average	Average
Statistic_Hour_PreferredNumOut	54.6000	6,06	49.0000	60.0000
Statistic_Hour_PreferredProfit	341372421	36.775.575,61	315806460	384074530
Statistic_Hour_PreferredTAVG	55672.08	10.315,12	44362.21	67785.96
Statistic_Hour_PreferredUseCost	20458262	386.612,75	20070429	20884538



EK 4: ARENA Process Analyzer Sonuçları

1 Replikasyon için Senaryo Analizleri

Process Analyzer - [Senaryo_Analizleri.pan]

File Edit View Insert Tools Run Help

Project Items

Scenarios

Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario 5

Scenario 6

Scenario 7

Scenario 8

Scenario 9

Scenario 10

Scenario 11

Scenario 12

Scenario 13

Scenario 14

Scenario 15

Scenario 16

Controls

Number of ...

Number of ...

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

Scenario Properties					Controls					Responses				
S	Name	Program File	Reps	Number of Test_B	Number of Test_C	Number of Dep_B	Number of Dep_C	Num Reps	Statistic_Hour_PreferredNumOut	Statistic_Hour_PreferredTAVG	Statistic_Hour_PreferredUseCost	TotalProjectRevenue	Total Profit	
1	Mevcut Sistem	1 : Mevcut Sistem-Saatlik-PreferredOrder.Backup.p	5	2.0000	2.0000	2.0000	2.0000	5	54.600	55672.076	20458262.103	361509547.912	341372420.829	
2	Scenario 1	1 : Mevcut Si	1	2.0000	2.0000	2.0000	2.0000	1	60.000	44362.206	20070428.665	335492020.982	315806460.466	
3	Scenario 2	1 : Mevcut Si	1	2.0000	2.0000	2.0000	4.0000	1	70.000	42817.644	19603451.866	404035371.791	384442476.385	
4	Scenario 3	1 : Mevcut Si	1	2.0000	2.0000	5.0000	2.0000	1	70.000	46924.739	19127431.122	390097899.499	371528349.867	
5	Scenario 4	1 : Mevcut Si	1	2.0000	2.0000	5.0000	4.0000	1	70.000	42993.312	20096550.623	369543586.301	350022990.907	
6	Scenario 5	1 : Mevcut Si	1	2.0000	4.0000	2.0000	2.0000	1	60.000	44362.206	20070428.665	335492020.982	315806460.466	
7	Scenario 6	1 : Mevcut Si	1	2.0000	4.0000	2.0000	4.0000	1	70.000	42817.644	19603451.866	404035371.791	384442476.385	
8	Scenario 7	1 : Mevcut Si	1	2.0000	4.0000	5.0000	2.0000	1	70.000	46924.739	19127431.122	390097899.499	371528349.867	
9	Scenario 8	1 : Mevcut Si	1	2.0000	4.0000	5.0000	4.0000	1	70.000	42993.312	20096550.623	369543586.301	350022990.907	
10	Scenario 9	1 : Mevcut Si	1	5.0000	2.0000	2.0000	2.0000	1	58.000	27265.067	22386074.013	342638300.026	320649763.745	
11	Scenario 10	1 : Mevcut Si	1	5.0000	2.0000	2.0000	4.0000	1	82.000	49884.043	21890927.936	431023996.760	409525919.003	
12	Scenario 11	1 : Mevcut Si	1	5.0000	2.0000	5.0000	2.0000	1	79.000	47959.070	22407893.976	501022896.687	478720495.966	
13	Scenario 12	1 : Mevcut Si	1	5.0000	2.0000	5.0000	4.0000	1	80.000	47559.415	22951310.185	461599275.175	439027578.528	
14	Scenario 13	1 : Mevcut Si	1	5.0000	4.0000	2.0000	2.0000	1	58.000	27265.067	22386074.013	342638300.026	320649763.745	
15	Scenario 14	1 : Mevcut Si	1	5.0000	4.0000	2.0000	4.0000	1	82.000	49884.043	21890927.936	431023996.760	409525919.003	
16	Scenario 15	1 : Mevcut Si	1	5.0000	4.0000	5.0000	2.0000	1	79.000	47959.070	22407893.976	501022896.687	478720495.966	
17	Scenario 16	1 : Mevcut Si	1	5.0000	4.0000	5.0000	4.0000	1	80.000	47559.415	22951310.185	461599275.175	439027578.528	

2 Replikasyon için Senaryo Analizleri

Process Analyzer - [Senaryo_Analizleri.pan]

File Edit View Insert Tools Run Help

</

3 Replikasyon için Senaryo Analizleri

Process Analyzer - [Senaryo_Analizleri.pan]

File Edit View Insert Tools Run Help

Project Items

Scenarios

Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario 5

Scenario 6

Scenario 7

Scenario 8

Scenario 9

Scenario 10

Scenario 11

Scenario 12

Scenario 13

Scenario 14

Scenario 15

Scenario 16

Controls

Number of Test_B

Number of Test_C

Number of Dep_B

Number of Dep_C

Num Reps

Scenario Properties					Controls					Responses				
S	Name	Program File	Reps	Number of Test_B	Number of Test_C	Number of Dep_B	Number of Dep_C	Num Reps	Statistic_Hour_PreferredNumOut	Statistic_Hour_PreferredTAVG	Statistic_Hour_PreferredUseCost	TotalProjectRevenue	Total Profit	
1	Mevcut Sistem	1 : Mevcut Sistem-Saatlik-PreferredOrder.Backup.p	5	2.0000	2.0000	2.0000	2.0000	5	54.600	55672.076	20458262.103	361509547.912	341372420.829	
2	Scenario 1	1 : Mevcut Sistem-S	3	2.0000	2.0000	2.0000	2.0000	3	56.000	55646.308	20426805.575	361529319.713	341285956.819	
3	Scenario 2	1 : Mevcut Sistem-S	3	2.0000	2.0000	2.0000	4.0000	3	59.000	50739.075	20190502.653	388244432.170	368291689.829	
4	Scenario 3	1 : Mevcut Sistem-S	3	2.0000	2.0000	5.0000	2.0000	3	60.667	51522.546	19866555.871	379932897.145	360490336.973	
5	Scenario 4	1 : Mevcut Sistem-S	3	2.0000	2.0000	5.0000	4.0000	3	60.333	51229.470	20337110.924	378003706.233	357878044.501	
6	Scenario 5	1 : Mevcut Sistem-S	3	2.0000	4.0000	2.0000	2.0000	3	56.000	55646.308	20426805.575	361529319.713	341285956.819	
7	Scenario 6	1 : Mevcut Sistem-S	3	2.0000	4.0000	2.0000	4.0000	3	66.333	44037.076	22672370.124	373621652.747	351424604.955	
8	Scenario 7	1 : Mevcut Sistem-S	3	2.0000	4.0000	5.0000	2.0000	3	60.667	51522.546	19866555.871	379932897.145	360490336.973	
9	Scenario 8	1 : Mevcut Sistem-S	3	2.0000	4.0000	5.0000	4.0000	3	60.333	51229.470	20337110.924	378003706.233	357878044.501	
10	Scenario 9	1 : Mevcut Sistem-S	3	5.0000	2.0000	2.0000	2.0000	3	55.000	29409.961	22630023.842	311343993.351	288941180.348	
11	Scenario 10	1 : Mevcut Sistem-S	3	5.0000	2.0000	2.0000	4.0000	3	77.000	51118.594	22738533.114	460165427.658	437789106.972	
12	Scenario 11	1 : Mevcut Sistem-S	3	5.0000	2.0000	5.0000	2.0000	3	77.000	54549.366	23054997.936	522484531.538	499605775.040	
13	Scenario 12	1 : Mevcut Sistem-S	3	5.0000	2.0000	5.0000	4.0000	3	75.333	42506.931	23432336.869	459722065.650	436698126.127	
14	Scenario 13	1 : Mevcut Sistem-S	3	5.0000	4.0000	2.0000	2.0000	3	55.000	29409.961	22630023.842	311343993.351	288941180.348	
15	Scenario 14	1 : Mevcut Sistem-S	3	5.0000	4.0000	2.0000	4.0000	3	77.000	51118.594	22738533.114	460165427.658	437789106.972	
16	Scenario 15	1 : Mevcut Sistem-S	3	5.0000	4.0000	5.0000	2.0000	3	77.000	54549.366	23054997.936	522484531.538	499605775.040	
17	Scenario 16	1 : Mevcut Sistem-S	3	5.0000	4.0000	5.0000	4.0000	3	75.333	42506.931	23432336.869	459722065.650	436698126.127	

4 Replikasyon için Senaryo Analizleri

Process Analyzer - [Senaryo_Analizleri.pan]

File Edit View Insert Tools Run Help

Project Items

Scenarios

Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario 5

Scenario 6

Scenario 7

Scenario 8

Scenario 9

Scenario 10

Scenario 11

Scenario 12

Scenario 13

Scenario 14

Scenario 15

Scenario 16

Controls

Number of ...

Number of ...

Scenario Properties				Controls					Responses				
S	Name	Program File	Reps	Number of Test_B	Number of Test_C	Number of Dep_B	Number of Dep_C	Num Reps	Statistic_Hour_PreferredNumOut	Statistic_Hour_PreferredTAVG	Statistic_Hour_PreferredUseCost	TotalProjectRevenue	Total Profit
1	Mevcut Sistem	1 : Mecut Sistem-Saatlik-PreferredOrder Backup.p	5	2.0000	2.0000	2.0000	2.0000	5	54.600	55672.076	20458262.103	361509547.912	341372420.829
2	Scenario 1	1 : Mecut Sistem-S	4	2.0000	2.0000	2.0000	2.0000	4	56.000	55513.739	20478919.196	366444656.363	346117211.632
3	Scenario 2	1 : Mecut Sistem-S	4	2.0000	2.0000	2.0000	4.0000	4	62.250	50825.704	19966656.431	402260154.626	382505633.526
4	Scenario 3	1 : Mecut Sistem-S	4	2.0000	2.0000	5.0000	2.0000	4	61.500	51222.684	19912638.651	386426220.722	366849114.417
5	Scenario 4	1 : Mecut Sistem-S	4	2.0000	2.0000	5.0000	4.0000	4	60.250	51225.728	20311474.426	381463668.693	361328470.311
6	Scenario 5	1 : Mecut Sistem-S	4	2.0000	4.0000	2.0000	2.0000	4	56.000	55513.739	20478919.196	366444656.363	346117211.632
7	Scenario 6	1 : Mecut Sistem-S	4	2.0000	4.0000	2.0000	4.0000	4	64.250	40392.616	23011938.513	365832676.823	343390790.041
8	Scenario 7	1 : Mecut Sistem-S	4	2.0000	4.0000	5.0000	2.0000	4	61.500	51222.684	19912638.651	386426220.722	366849114.417
9	Scenario 8	1 : Mecut Sistem-S	4	2.0000	4.0000	5.0000	4.0000	4	60.250	51225.728	20311474.426	381463668.693	361328470.311
10	Scenario 9	1 : Mecut Sistem-S	4	5.0000	2.0000	2.0000	2.0000	4	54.250	29459.573	22547274.661	300485206.195	278135639.658
11	Scenario 10	1 : Mecut Sistem-S	4	5.0000	2.0000	2.0000	4.0000	4	75.750	50995.012	22822711.934	458563455.429	436203783.167
12	Scenario 11	1 : Mecut Sistem-S	4	5.0000	2.0000	5.0000	2.0000	4	77.500	52328.254	23035010.245	496331842.420	473452063.703
13	Scenario 12	1 : Mecut Sistem-S	4	5.0000	2.0000	5.0000	4.0000	4	74.750	43540.758	23352231.352	444728131.073	421682244.843
14	Scenario 13	1 : Mecut Sistem-S	4	5.0000	4.0000	2.0000	2.0000	4	54.250	29459.573	22547274.661	300485206.195	278135639.658
15	Scenario 14	1 : Mecut Sistem-S	4	5.0000	4.0000	2.0000	4.0000	4	75.750	50995.012	22822711.934	458563455.429	436203783.167
16	Scenario 15	1 : Mecut Sistem-S	4	5.0000	4.0000	5.0000	2.0000	4	77.500	52328.254	23035010.245	496331842.420	473452063.703
17	Scenario 16	1 : Mecut Sistem-S	4	5.0000	4.0000	5.0000	4.0000	4	74.750	43540.758	23352231.352	444728131.073	421682244.843

5 Replikasyon için Senaryo Analizleri

Process Analyzer - [Senaryo_Analizleri.pan]

File Edit View Insert Tools Run Help

Project Items

Scenarios

Mevcut S...

Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario 5

Scenario 6

Scenario 7

Scenario 8

Scenario 9

Scenario 10

Scenario 11

Scenario 12

Scenario 13

Scenario 14

Scenario 15

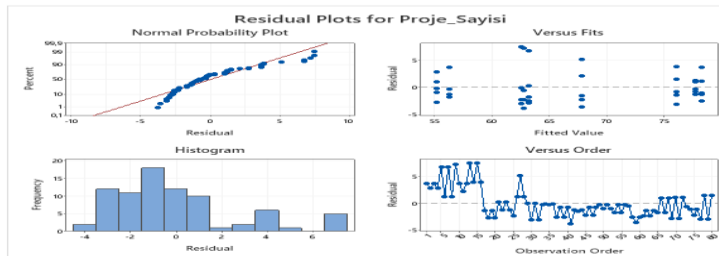
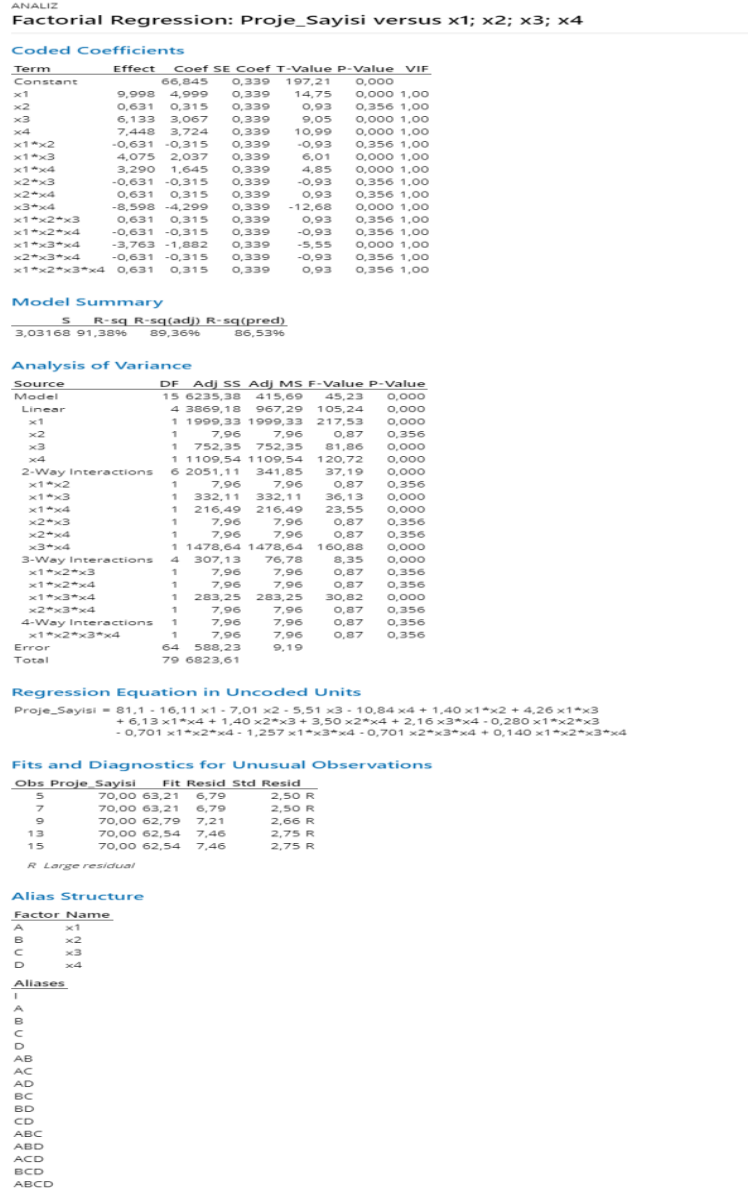
Scenario 16

Controls

Number of ...

Scenario Properties				Controls					Responses				
S	Name	Program File	Reps	Number of Test_B	Number of Test_C	Number of Dep_B	Number of Dep_C	Num Reps	Statistic_Hour_PreferedNumOut	Statistic_Hour_PreferedTAVG	Statistic_Hour_PreferedUseCost	TotalProjectRevenue	Total Profit
1	Mevcut Sistem	1 : Mecut Sistem-Saatlik-PreferredOrder Backup.p	5	2.0000	2.0000	2.0000	2.0000	5	54.600	55672.076	20458262.103	361509547.912	341372420.829
2	Scenario 1	1 : Mecut Sistem-S	5	2.0000	2.0000	2.0000	2.0000	5	54.600	55672.076	20458262.103	361509547.912	341372420.829
3	Scenario 2	1 : Mecut Sistem-S	5	2.0000	2.0000	2.0000	4.0000	5	62.200	51368.952	19966237.753	407779906.302	388028458.710
4	Scenario 3	1 : Mecut Sistem-S	5	2.0000	2.0000	5.0000	2.0000	5	60.400	51529.366	19901581.531	377516058.843	358023363.813
5	Scenario 4	1 : Mecut Sistem-S	5	2.0000	2.0000	5.0000	4.0000	5	59.600	51324.641	20388083.972	379556782.133	359316582.615
6	Scenario 5	1 : Mecut Sistem-S	5	2.0000	4.0000	2.0000	2.0000	5	54.600	55672.076	20458262.103	361509547.912	341372420.829
7	Scenario 6	1 : Mecut Sistem-S	5	2.0000	4.0000	2.0000	4.0000	5	65.600	43925.666	23104361.356	378228991.714	355809144.479
8	Scenario 7	1 : Mecut Sistem-S	5	2.0000	4.0000	5.0000	2.0000	5	60.400	51529.366	19901581.531	377516058.843	358023363.813
9	Scenario 8	1 : Mecut Sistem-S	5	2.0000	4.0000	5.0000	4.0000	5	59.600	51324.641	20388083.972	379556782.133	359316582.615
10	Scenario 9	1 : Mecut Sistem-S	5	5.0000	2.0000	2.0000	2.0000	5	56.200	29816.069	22532359.876	309582524.109	287257201.170
11	Scenario 10	1 : Mecut Sistem-S	5	5.0000	2.0000	2.0000	4.0000	5	77.200	48840.253	22990616.965	460934208.387	438372138.621
12	Scenario 11	1 : Mecut Sistem-S	5	5.0000	2.0000	5.0000	2.0000	5	78.800	49597.875	22924151.693	487311442.908	464523965.415
13	Scenario 12	1 : Mecut Sistem-S	5	5.0000	2.0000	5.0000	4.0000	5	77.600	43405.581	23425571.160	462060293.488	438926304.673
14	Scenario 13	1 : Mecut Sistem-S	5	5.0000	4.0000	2.0000	2.0000	5	56.200	29816.069	22532359.876	309582524.109	287257201.170
15	Scenario 14	1 : Mecut Sistem-S	5	5.0000	4.0000	2.0000	4.0000	5	77.200	48840.253	22990616.965	460934208.387	438372138.621
16	Scenario 15	1 : Mecut Sistem-S	5	5.0000	4.0000	5.0000	2.0000	5	78.800	49597.875	22924151.693	487311442.908	464523965.415
17	Scenario 16	1 : Mecut Sistem-S	5	5.0000	4.0000	5.0000	4.0000	5	77.600	43405.581	23425571.160	462060293.488	438926304.673

EK 5: Ortalama Tamamlanan Toplam Proje Sayısı Performans Ölçütüne Göre Oluşturulan Meta-Model Faktöriyel Regresyon Analizi MİNİTAB Sonucu



EK 6: Ortalama Tamamlanan Toplam Proje Sayısı Performans Ölçütüne Göre

Oluşturulan Meta-Model Regresyon Analizi MİNİTAB Sonucu

ANALİZ

Regression Analysis: Proje_Sayisi versus x1; x2; x3; x4; x1*x3; x1*x4; x3*x4; x1*x3*x4

Regression Equation

Proje_Sayisi = 59,15 - 11,91 x1 + 0,315 x2 - 1,31 x3 - 0,33 x4 + 3,414 x1*x3 + 4,024 x1*x4
+ 0,061 x3*x4 - 0,836 x1*x3*x4

Coefficients

Term	Coef	SE	Coef	T-Value	P-Value	VIF
Constant	59,15	6,94	8,53	0,000		
x1	-11,91	1,80	-6,61	0,000	64,44	
x2	0,315	0,337	0,94	0,352	1,00	
x3	-1,31	1,80	-0,73	0,470	64,44	
x4	-0,33	2,17	-0,15	0,880	41,53	
x1*x3	3,414	0,473	7,22	0,000	118,89	
x1*x4	4,024	0,570	7,06	0,000	99,53	
x3*x4	0,061	0,570	0,11	0,915	99,53	
x1*x3*x4	-0,836	0,150	-5,59	0,000	148,53	

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
3,01158	90,56%	89,50%	88,02%

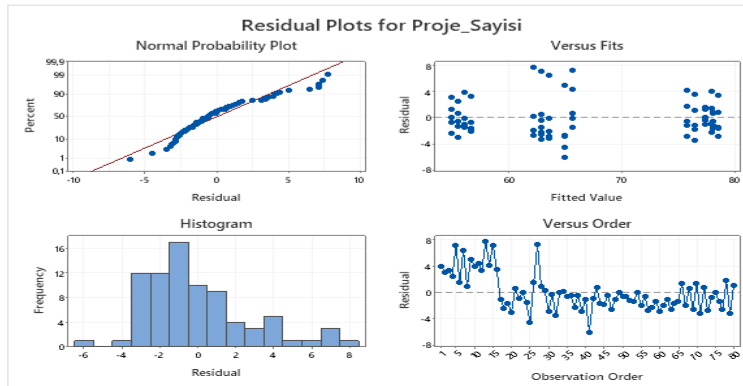
Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	8	6179,67	772,458	85,17	0,000
x1	1	396,05	396,048	43,67	0,000
x2	1	7,96	7,959	0,88	0,352
x3	1	4,78	4,776	0,53	0,470
x4	1	0,21	0,206	0,02	0,880
x1*x3	1	472,16	472,163	52,06	0,000
x1*x4	1	452,21	452,209	49,86	0,000
x3*x4	1	0,10	0,104	0,01	0,915
x1*x3*x4	1	283,25	283,252	31,23	0,000
Error	71	643,94	9,070		
Lack-of-Fit	7	55,71	7,959	0,87	0,538
Pure Error	64	588,23	9,191		
Total	79	6823,61			

Fits and Diagnostics for Unusual Observations

Obs	Proje_Sayisi	Fit	Resid	Std Resid
5	70,00	62,90	7,10	2,50 R
7	70,00	63,53	6,47	2,28 R
13	70,00	62,22	7,78	2,74 R
15	70,00	62,85	7,15	2,52 R
27	73,00	65,63	7,37	2,60 R
41	59,00	65,00	-6,00	-2,11 R

R Large residual



EK 7: Projelerin Ortalama Toplam Tamamlama Süresi Performans Ölçütüne Göre Oluşturulan Meta-Model Faktöriyel Regresyon Analizi MİNİTAB Sonucu

ANALİZ

Factorial Regression: Proje_Suresi versus x1; x2; x3; x4

Coded Coefficients

Term	Effect	Coef	SE Coef	T-Value	P-Value	VIF
Constant		46872	348	134,82	0,000	
x1		-6562	3281	348	-9,44	0,000 1,00
x2		-546	273	348	-0,78	0,435 1,00
x3		3813	1907	348	5,48	0,000 1,00
x4		1344	672	348	1,93	0,058 1,00
x1*x2		546	273	348	0,78	0,435 1,00
x1*x3		3751	1876	348	5,40	0,000 1,00
x1*x4		5343	2672	348	7,68	0,000 1,00
x2*x3		546	273	348	0,78	0,435 1,00
x2*x4		-546	273	348	-0,78	0,435 1,00
x3*x4		-6045	3022	348	-8,69	0,000 1,00
x1*x2*x3		-546	273	348	-0,78	0,435 1,00
x1*x2*x4		546	273	348	0,78	0,435 1,00
x1*x3*x4		-8708	4354	348	-12,52	0,000 1,00
x2*x3*x4		546	273	348	0,78	0,435 1,00
x1*x2*x3*x4		-546	273	348	-0,78	0,435 1,00

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
3109,49	87,51%	84,58%	80,48%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	15	4335689265	289045951	29,89	0,000
Linear	4	1194179515	298544879	30,88	0,000
x1	1	861291999	861291999	89,08	0,000
x2	1	5956714	5956714	0,62	0,435
x3	1	290801454	290801454	30,08	0,000
x4	1	36129348	36129348	3,74	0,058
2-Way Interactions	6	1601037274	266839546	27,60	0,000
x1*x2	1	5956714	5956714	0,62	0,435
x1*x3	1	281471335	281471335	29,11	0,000
x1*x4	1	570958692	570958692	59,05	0,000
x2*x3	1	5956714	5956714	0,62	0,435
x2*x4	1	5956714	5956714	0,62	0,435
x3*x4	1	730737104	730737104	75,58	0,000
3-Way Interactions	4	1534515762	383628940	39,68	0,000
x1*x2*x3	1	5956714	5956714	0,62	0,435
x1*x2*x4	1	5956714	5956714	0,62	0,435
x1*x3*x4	1	1516645618	1516645618	156,86	0,000
x2*x3*x4	1	5956714	5956714	0,62	0,435
4-Way Interactions	1	5956714	5956714	0,62	0,435
x1*x2*x3*x4	1	5956714	5956714	0,62	0,435
Error	64	618809699			
Total	79	4954498962			

Regression Equation in Uncoded Units

Proje_Suresi = 108746 - 27129 x1 + 6064 x2 - 12283 x3 - 13119 x4 - 1213 x1*x2 + 5911 x1*x3 + 6735 x1*x4 - 1213 x2*x3 - 3032 x2*x4 + 2939 x3*x4 + 243 x1*x2*x3 + 606 x1*x2*x4 - 1571 x1*x3*x4 + 606 x2*x3*x4 - 121 x1*x2*x3*x4

Fits and Diagnostics for Unusual Observations

Obs	Proje_Suresi	Fit	Resid	Std Resid
1	44362	53454	-9091	-3,27 R
3	44362	53454	-9091	-3,27 R
9	42818	48974	-6156	-2,21 R
13	42993	49516	-6523	-2,35 R
15	42993	49516	-6523	-2,35 R
27	51868	44608	7260	2,61 R

R Large residual

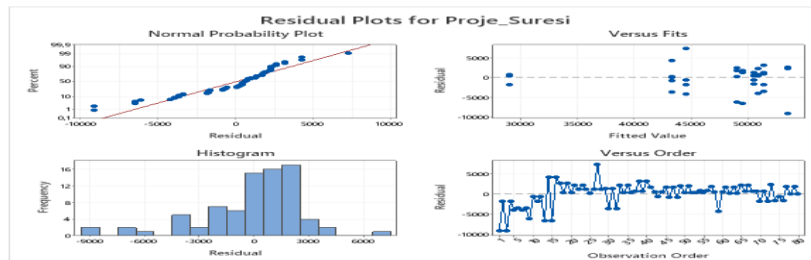
Alias Structure

Factor Name

A x1
B x2
C x3
D x4

Aliases

I
A
B
C
D
AB
AC
AD
BC
BD
CD
ABC
ABD
ACD
BCD
ABCD



EK 8: Projelerin Ortalama Toplam Tamamlama Süresi Performans Ölçütüne Göre Oluşturulan Meta-Model Regresyon Analizi MİNİTAB Sonucu

ANALİZ

Regression Analysis: Proje_Suresi versus x1; x2; x3; x4; x1*x3; x1*x4; x3*x4; x1*x3*x4

Regression Equation

Proje_Suresi = 127756 - 30767 x1 - 273 x2 - 15921 x3 - 22215 x4 + 6639 x1*x3 + 8554 x1*x4 + 4758 x3*x4 - 1935 x1*x3*x4

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	127756	7024	18,19	0,000	
x1	-30767	1825	-16,86	0,000	64,44
x2	-273	341	-0,80	0,426	1,00
x3	-15921	1825	-8,72	0,000	64,44
x4	-22215	2198	-10,11	0,000	41,53
x1*x3	6639	479	13,85	0,000	118,89
x1*x4	8554	577	14,82	0,000	99,53
x3*x4	4758	577	8,24	0,000	99,53
x1*x3*x4	-1935	152	-12,77	0,000	148,53

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
3050,07	86,67%	85,17%	83,07%

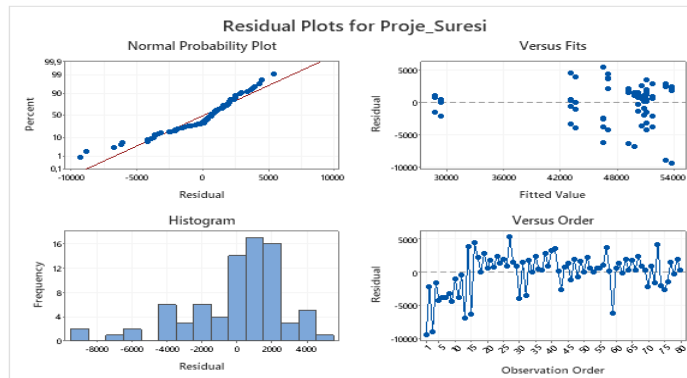
Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	8	4293992263	536749033	57,70	0,000
x1	1	2644037749	2644037749	284,22	0,000
x2	1	5956714	5956714	0,64	0,426
x3	1	708014459	708014459	76,11	0,000
x4	1	950637860	950637860	102,19	0,000
x1*x3	1	1785150149	1785150149	191,89	0,000
x1*x4	1	2043754921	2043754921	219,69	0,000
x3*x4	1	632362677	632362677	67,97	0,000
x1*x3*x4	1	1516645618	1516645618	163,03	0,000
Error	71	660506699	9302911		
Lack-of-Fit	7	41697001	5956714	0,62	0,741
Pure Error	64	618809698	9668902		
Total	79	4954498962			

Fits and Diagnostics for Unusual Observations

Obs	Proje_Suresi	Fit	Resid	Std Resid
1	44362	53727	-9364	-3,26 R
3	44362	53181	-8819	-3,07 R
13	42993	49789	-6796	-2,37 R
15	42993	49244	-6250	-2,18 R
59	40393	46518	-6126	-2,13 R

R Large residual



EK 9: Ortalama Toplam Çalışan Maliyeti Performans Ölçütüne Göre Oluşturulan Meta-Model Faktöriyel Regresyon Analizi MİNİTAB Sonucu

ANALİZ

Factorial Regression: Calisan_Maliyeti versus x1; x2; x3; x4

Coded Coefficients

Term	Effect	Coef	SE Coef	T-Value	P-Value	VIF
Constant		21599801	49589	435,57	0,000	
x1		2477975	1238987	49589	24,98	0,000 1,00
x2		282371	141185	49589	2,85	0,006 1,00
x3		-129967	-64983	49589	-1,31	0,195 1,00
x4		448889	224444	49589	4,53	0,000 1,00
x1*x2		-282371	-141185	49589	-2,85	0,006 1,00
x1*x3		591408	295704	49589	5,96	0,000 1,00
x1*x4		-195938	-97969	49589	-1,98	0,053 1,00
x2*x3		-282371	-141185	49589	-2,85	0,006 1,00
x2*x4		282371	141185	49589	2,85	0,006 1,00
x3*x4		92444	46222	49589	0,93	0,355 1,00
x1*x2*x3		282371	141185	49589	2,85	0,006 1,00
x1*x2*x4		-282371	-141185	49589	-2,85	0,006 1,00
x1*x3*x4		125859	62930	49589	1,27	0,209 1,00
x2*x3*x4		-282371	-141185	49589	-2,85	0,006 1,00
x1*x2*x3*x4		282371	141185	49589	2,85	0,006 1,00

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
443541	92,17%	90,33%	87,76%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	15	1,48183E+14	9,87888E+12	50,22	0,000
Linear	4	1,28770E+14	3,21924E+13	163,64	0,000
x1	1	1,22807E+14	1,22807E+14	624,25	0,000
x2	1	1,59466E+12	1,59466E+12	8,11	0,006
x3	1	3,37827E+11	3,37827E+11	1,72	0,195
x4	1	4,03003E+12	4,03003E+12	20,49	0,000
2-Way Interactions	6	1,27180E+13	2,11967E+12	10,77	0,000
x1*x2	1	1,59466E+12	1,59466E+12	8,11	0,006
x1*x3	1	6,99528E+12	6,99528E+12	35,56	0,000
x1*x4	1	7,67833E+11	7,67833E+11	3,90	0,053
x2*x3	1	1,59466E+12	1,59466E+12	8,11	0,006
x2*x4	1	1,59466E+12	1,59466E+12	8,11	0,006
x3*x4	1	1,70919E+11	1,70919E+11	0,87	0,355
3-Way Interactions	4	5,10080E+12	1,27520E+12	6,48	0,000
x1*x2*x3	1	1,59466E+12	1,59466E+12	8,11	0,006
x1*x2*x4	1	1,59466E+12	1,59466E+12	8,11	0,006
x1*x3*x4	1	3,16811E+11	3,16811E+11	1,61	0,209
x2*x3*x4	1	1,59466E+12	1,59466E+12	8,11	0,006
4-Way Interactions	1	1,59466E+12	1,59466E+12	8,11	0,006
x1*x2*x3*x4	1	1,59466E+12	1,59466E+12	8,11	0,006
Error	64	1,25906E+13	1,96728E+11		
Total	79	1,60774E+14			

Regression Equation in Uncoded Units

Calisan_Maliyeti = 27819346 - 1026854 x1 - 3137452 x2 - 2184550 x3 - 4018374 x4 + 627490 x1*x2 + 424012 x1*x3 + 778032 x1*x4 + 627490 x2*x3 + 1568726 x2*x4 + 874160 x3*x4 - 125493 x1*x2*x3 - 313745 x1*x2*x4 - 160278 x1*x3*x4 - 313745 x2*x3*x4 + 62749 x1*x2*x3*x4

Fits and Diagnostics for Unusual Observations

Obs	Calisan_Maliyeti	Fit	Resid	Std Resid
11	19603452	22190104	-2586652	-6,52 R
59	23011939	22190104	821834	2,07 R
75	23104361	22190104	914257	2,30 R

R Large residual

Alias Structure

Factor Name

A x1

B x2

C x3

D x4

Aliases

I

A

B

C

D

AB

AC

AD

BC

BD

CD

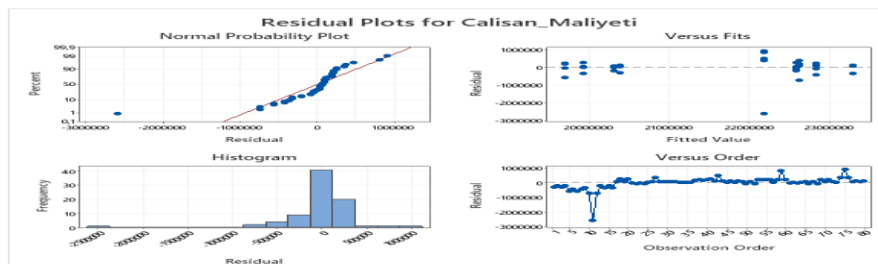
ABC

ABD

ACD

BCD

ABCD



EK 10: Ortalama Toplam Çalışan Maliyeti Performans Ölçütüne Göre Oluşturulan Meta-Model Regresyon Analizi MİNİTAB Sonucu

ANALİZ

Regression Analysis: Calisan_Maliyeti versus x1; x2; x3; x4; x1*x2; x1*x3; x2*x3; x2*x4; x1*x2*x3; x1*x2*x4; x2*x3*x4; x1*x2*x3*x4

Regression Equation

Calisan_Maliyeti = 16361560 + 1307243 x1 + 299884 x2 + 437929 x3 - 199111 x4 - 72739 x1*x2
- 56823 x1*x3 - 159253 x2*x3 + 422947 x2*x4 + 18752 x1*x2*x3
- 80335 x1*x2*x4 - 51497 x2*x3*x4 + 14666 x1*x2*x3*x4

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	16361560	1237516	13,22	0,000	
x1	1307243	294628	4,44	0,000	64,44
x2	299884	513516	0,58	0,561	87,01
x3	437929	294628	1,49	0,142	64,44
x4	-199111	174089	-1,14	0,257	10,00
x1*x2	-72739	128425	-0,57	0,573	189,11
x1*x3	-56823	77373	-0,73	0,465	118,89
x2*x3	-159253	128425	-1,24	0,219	189,11
x2*x4	422947	123752	3,42	0,001	96,01
x1*x2*x3	18752	33726	0,56	0,580	282,21
x1*x2*x4	-80335	29463	-2,73	0,008	131,11
x2*x3*x4	-51497	29463	-1,75	0,085	131,11
x1*x2*x3*x4	14666	7737	1,90	0,062	175,21

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
492399	89,90%	88,09%	85,27%

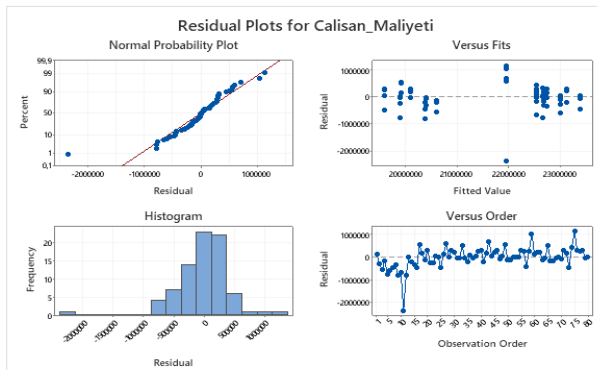
Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	12	1,44529E+14	1,20441E+13	49,68	0,000
x1	1	4,77309E+12	4,77309E+12	19,69	0,000
x2	1	82686524297	82686524297	0,34	0,561
x3	1	5,35666E+11	5,35666E+11	2,21	0,142
x4	1	3,17163E+11	3,17163E+11	1,31	0,257
x1*x2	1	77779479508	77779479508	0,32	0,573
x1*x3	1	1,30769E+11	1,30769E+11	0,54	0,465
x2*x3	1	3,72830E+11	3,72830E+11	1,54	0,219
x2*x4	1	2,83208E+12	2,83208E+12	11,68	0,001
x1*x2*x3	1	74958027375	74958027375	0,31	0,580
x1*x2*x4	1	1,80261E+12	1,80261E+12	7,43	0,008
x2*x3*x4	1	7,40722E+11	7,40722E+11	3,06	0,085
x1*x2*x3*x4	1	8,71064E+11	8,71064E+11	3,59	0,062
Error	67	1,62446E+13	2,42457E+11		
Lack-of-Fit	3	3,65400E+12	1,21800E+12	6,19	0,001
Pure Error	64	1,25906E+13	1,96728E+11		
Total	79	1,60774E+14			

Fits and Diagnostics for Unusual Observations

Obs	Calisan_Maliyeti	Fit	Resid	Std Resid
11	19603452	21958906	-2355454	-5,30 R
59	23011939	21958906	1053033	2,37 R
75	23104361	21958906	1145455	2,58 R

R Large residual



EK 11: Ortalama Toplam Kâr Performans Ölçütüne Göre Oluşturulan Meta-Model

Faktöriyel Regresyon Analizi MİNİTAB Sonucu

ANALİZ

Factorial Regression: Karlılık versus x1; x2; x3; x4

Coded Coefficients

Term	Effect	Coef	SE Coef	T-Value	P-Value	VIF
Constant		384502291	1554318	247,38	0,000	
x1		52539144	26269572	1554318	16,90	0,000 1,00
x2		-1515617	-757808	1554318	-0,49	0,628 1,00
x3		52737760	26368880	1554318	16,96	0,000 1,00
x4		30835206	15417603	1554318	9,92	0,000 1,00
x1*x2		1515617	757808	1554318	0,49	0,628 1,00
x1*x3		40932691	20466345	1554318	13,17	0,000 1,00
x1*x4		15739923	7869961	1554318	5,06	0,000 1,00
x2*x3		1515617	757808	1554318	0,49	0,628 1,00
x2*x4		-1515617	-757808	1554318	-0,49	0,628 1,00
x3*x4		-60611713	-30305857	1554318	-19,50	0,000 1,00
x1*x2*x3		-1515617	-757808	1554318	-0,49	0,628 1,00
x1*x2*x4		1515617	757808	1554318	0,49	0,628 1,00
x1*x3*x4		-36836117	-18418059	1554318	-11,85	0,000 1,00
x2*x3*x4		1515617	757808	1554318	0,49	0,628 1,00
x1*x2*x3*x4		-1515617	-757808	1554318	-0,49	0,628 1,00

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
13902247	95,61%	94,58%	93,14%

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	15	2,69205E+17	1,79530E+16	92,89	0,000
Linear	4	1,29895E+17	3,24737E+16	168,02	0,000
x1	1	5,52072E+16	5,52072E+16	285,64	0,000
x2	1	4,59419E+13	4,59419E+13	0,24	0,628
x3	1	5,56254E+16	5,56254E+16	287,81	0,000
x4	1	1,90162E+16	1,90162E+16	98,39	0,000
2-Way Interactions	6	1,12078E+17	1,86797E+16	96,65	0,000
x1*x2	1	4,59419E+13	4,59419E+13	0,24	0,628
x1*x3	1	3,35097E+16	3,35097E+16	173,38	0,000
x1*x4	1	4,95490E+15	4,95490E+15	25,64	0,000
x2*x3	1	4,59419E+13	4,59419E+13	0,24	0,628
x2*x4	1	4,59419E+13	4,59419E+13	0,24	0,628
x3*x4	1	7,34756E+16	7,34756E+16	380,17	0,000
3-Way Interactions	4	2,72758E+16	6,81895E+15	35,28	0,000
x1*x2*x3	1	4,59419E+13	4,59419E+13	0,24	0,628
x1*x2*x4	1	4,59419E+13	4,59419E+13	0,24	0,628
x1*x3*x4	1	2,71380E+16	2,71380E+16	140,41	0,000
x2*x3*x4	1	4,59419E+13	4,59419E+13	0,24	0,628
4-Way Interactions	1	4,59419E+13	4,59419E+13	0,24	0,628
x1*x2*x3*x4	1	4,59419E+13	4,59419E+13	0,24	0,628
Error	64	1,23694E+16	1,93272E+14		
Total	79	2,81664E+17			

Regression Equation in Uncoded Units

Karlılık = 420110767 - 105910239 x1 + 16840189 x2 - 29492398 x3 - 7247789 x4 - 3368038 x1*x2 + 31632742 x1*x3 + 28844898 x1*x4 - 3368038 x2*x3 - 8420094 x2*x4 + 3394352 x3*x4 + 673608 x1*x2*x3 + 1684019 x1*x2*x4 - 7175393 x1*x3*x4 + 1684019 x2*x3*x4 - 336804 x1*x2*x3*x4

Fits and Diagnostics for Unusual Observations

Obs	Karlılık	Fit	Resid	Std Resid
2	320649764	291925158	28724605	2,31 R
4	320649764	291925158	28724605	2,31 R
10	409525919	435948118	-26422199	-2,12 R
12	409525919	435948118	-26422199	-2,12 R
27	393648765	365703156	27945609	2,25 R

R Large residual

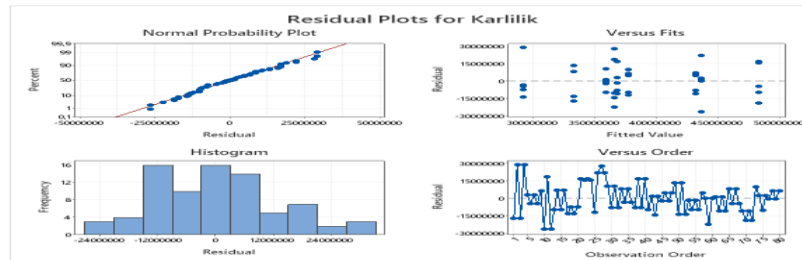
Alias Structure

Factor Name

A x1
B x2
C x3
D x4

Aliases

I
A
B
C
D
AB
AC
AD
BC
BD
CD
ABC
ABD
ACD
BCD
ABCD



EK 12: Ortalama Toplam Kâr Performans Ölçütüne Göre Oluşturulan Meta-Model

Regresyon Analizi MİNİTAB Sonucu

ANALİZ

Regression Analysis: Karlilik versus x1; x2; x3; x4; x1*x3; x1*x4; x3*x4; x1*x3*x4

Regression Equation

Karlilik = 472904759 - 116014353 x1 - 757808 x2 - 39596511 x3 - 32508072 x4 + 33653565 x1*x3 + 33896955 x1*x4 + 8446409 x3*x4 - 8185804 x1*x3*x4

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	472904759	30790392	15,36	0,000	
x1	-116014353	7999738	-14,50	0,000	64,44
x2	-757808	1494770	-0,51	0,614	1,00
x3	-39596511	7999738	-4,95	0,000	64,44
x4	-32508072	9632960	-3,37	0,001	41,53
x1*x3	33653565	2100834	16,02	0,000	118,89
x1*x4	33896955	2529739	13,40	0,000	99,53
x3*x4	8446409	2529739	3,34	0,001	99,53
x1*x3*x4	-8185804	664342	-12,32	0,000	148,53

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
13369626	95,49%	94,99%	94,28%

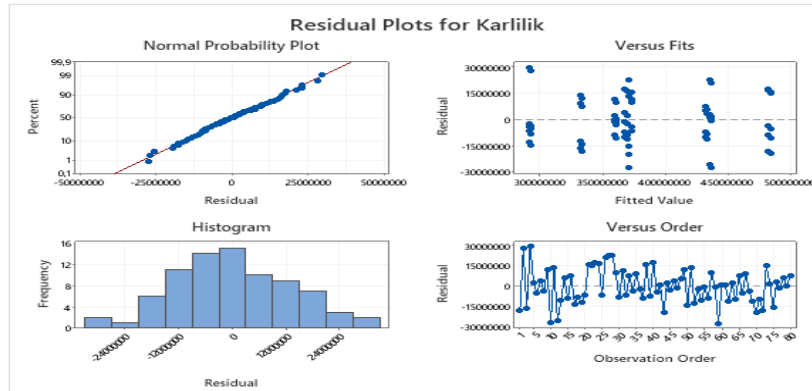
Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	8	2,68973E+17	3,36216E+16	188,10	0,000
x1	1	3,75933E+16	3,75933E+16	210,32	0,000
x2	1	4,59419E+13	4,59419E+13	0,26	0,614
x3	1	4,37926E+15	4,37926E+15	24,50	0,000
x4	1	2,03564E+15	2,03564E+15	11,39	0,001
x1*x3	1	4,58688E+16	4,58688E+16	256,61	0,000
x1*x4	1	3,20929E+16	3,20929E+16	179,54	0,000
x3*x4	1	1,99265E+15	1,99265E+15	11,15	0,001
x1*x3*x4	1	2,71380E+16	2,71380E+16	151,82	0,000
Error	71	1,26910E+16	1,78747E+14		
Lack-of-Fit	7	3,21593E+14	4,59419E+13	0,24	0,974
Pure Error	64	1,23694E+16	1,93272E+14		
Total	79	2,81664E+17			

Fits and Diagnostics for Unusual Observations

Obs	Karlilik	Fit	Resid	Std Resid
2	320649764	292682967	27966797	2,22 R
4	320649764	291167350	29482414	2,34 R
10	409525919	436705927	-27180008	-2,16 R
12	409525919	435190310	-25664391	-2,04 R
59	343390790	371007816	-27617026	-2,19 R

R Large residual



EK 13: Çok Ölçütlü Karar Modeli LINGO çözümü

Lingo 19.0 - [Lingo Model (Text Only) - EK-X_Lingo_Ağırlıklandırma_Model]

File Edit Solver Window Help

```

max=0.4*(472904759-116014353*x1-757808*x2-39596511*x3-32508072*x4+33653565*x1*x3+33896955*x1*x4+8446409*x3*x4-8185804*x1*x3*x4)-0.3*(16361560+1307243*x1+299884*x2+437829
2 <= x1;
x1 <= 5;
2 <= x2;
x2 <= 4;
2 <= x3;
x3 <= 5;
2 <= x4;
x4 <= 4;
@gin(x1);
@gin(x2);
@gin(x3);
@gin(x4);
end

```

Lingo 19.0 - [EK-X_Lingo_Ağırlıklandırma]

File Edit Solver Window Help

```

Local optimal solution found.
Objective value:                0.1679106E+09
Objective bound:                0.1679106E+09
Infeasibilities:                0.000000
Extended solver steps:         0
Total solver iterations:        6
Elapsed runtime seconds:       0.06

Model Class:                    PINLP

Total variables:                4
Nonlinear variables:            4
Integer variables:              4

Total constraints:              9
Nonlinear constraints:          1

Total nonzeros:                12
Nonlinear nonzeros:            4

```

Variable	Value	Reduced Cost
X1	5.000000	0.000000
X2	2.000000	0.000000
X3	2.000000	0.000000
X4	4.000000	0.000000

Row	Slack or Surplus	Dual Price
1	0.1679106E+09	1.000000
2	3.000000	0.000000
3	0.000000	8342886.
4	0.000000	322555.1
5	2.000000	0.000000
6	0.000000	562421.0
7	3.000000	0.000000
8	2.000000	0.000000
9	0.000000	0.2882322E+08



FERDA GÜVEN

YÜKSEK LİSANS TEZİ

2021
