

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**PARAMETRİK TAHMİN MODELLERİN YAZILIM PROJELERİNE
UYGULANMASINA YÖNELİK BİR YAZILIM PAKETİNİN
GELİŞTİRİLMESİ**

Taner SONER

ELEKTRİK–ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2014**

Her hakkı saklıdır

TEZ ONAYI

Taner SONER tarafından hazırlanan “**Parametrik Tahmin Modellerinin Yazılım Projelerine Uygulanmasına Yönelik Bir Yazılım Paketinin Geliştirilmesi**” adlı tez çalışması **14/07/2014** tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç.Dr. A. Egemen YILMAZ

Jüri Üyeleri:

Başkan: Yrd.Doç.Dr. Ö. Özgür TANRIÖVER
Ankara Üniversitesi / Bilgisayar Mühendisliği Anabilim Dalı

Üye : Doç.Dr. A. Egemen YILMAZ
Ankara Üniversitesi / Elektrik-Elektronik Mühendisliği Anabilim Dalı

Üye : Yrd.Doç.Dr. Gökhan SOYSAL
Ankara Üniversitesi / Elektrik-Elektronik Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. İbrahim DEMİR
Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

14.07.2014

Taner SONER

ÖZET

Yüksek Lisans Tezi

PARAMETRİK TAHMİN MODELLERİN YAZILIM PROJELERİNE UYGULANMASINA YÖNELİK BİR YAZILIM PAKETİNİN GELİŞTİRİLMESİ

Taner SONER

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman : Doç. Dr. A. Egemen YILMAZ

Yazılım projelerinde, geliştirilecek yazılımın büyüklüğü ve harcanacak emeğin kestirimi projelerin doğru planlaması ve hesaplanması açısından çok önemlidir. Proje yöneticileri için tahmin modellerini kullanarak iş gücü ve zaman gibi planlamaların yapılabilme gereği açıktır. Efor tahmini yapılabilirse proje programını belirlemek kolaydır. Bu nedenle efor tahmini ile projelerde etkili bir iş gücü ve zamanlama planlaması yapılabilir. Yazılım projelerinde doğru maliyet tahmininde bulunmak, yazılımla ilgili tüm sektörler için kritik öneme sahiptir.

Dokümanlara aktarılmış birçok yazılım metriklerini tahmin etme metodu, aracı ve modeli mevcuttur. Bu tahmin teknikleri yardımıyla yazılım projelerinin efor, süre ve teslim hızı gibi büyüklükleri hesaplanabilir.

Bu çalışmada literatürde yer alan parametrik tahmin modelleri bir araya toplanmış, gerçek yazılım projesi metrikleri kullanılarak nümerik yöntemler yardımıyla yeni modeller teklif edilmiştir. Mevcut modellerle yeni modellerin performans kıyaslaması yapılmıştır. Bu kapsamda MATLAB GUI ortamında tüm modellerin bulunduğu yazılım paketi geliştirilmiştir. Söz konusu araç kutusu hazırlanarak, yazılım projeleriyle ilgilenen proje yöneticilerine proje başlangıcında yazılımcı iş gücü ve proje süresi planlamasında kolaylık sağlanması hedeflenmiştir.

Temmuz 2014 202 sayfa

Anahtar Kelimeler : Yazılım Efor Tahmini, Kod Satır Sayısı (Lines of Code - LOC), İşlev Puanı (Function Point - FP) Parametrik Tahmin Modelleri, Yazılım Paketi

ABSTRACT

M.Sc. Thesis

DEVELOPMENT OF A SOFTWARE PACKAGE FOR APPLICATION OF PARAMETRIC ESTIMATION MODELS TO SOFTWARE PROJECTS

Taner SONER

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical-Electronic Engineering

Supervisor : Assoc. Prof. A. Egemen YILMAZ

In software projects, estimating the size of software and the labor effort is very important for the projects to be planned and calculated correctly. It is clear that project managers need to make schedule and plan labor force by using estimation models. If effort estimation can be done, the project schedule will be determined easily. Therefore an effective workforce planning and scheduling can be made in the projects with the help of effort estimation. Making accurate cost estimation of software project is of crucial importance to all software industries.

There are many software metrics estimation methods, tools, and models that have been transferred to documents. Effort, time and speed of delivery of software projects can be calculated with the help of these effort estimation techniques.

In this study, parametric estimation models in the literature have been gathered together and with the help of numerical methods, new models have been proposed using real software project metrics. Performance benchmarking of the existing models and new model have been made. In this context, software package that includes all the models have been developed by using MATLAB GUI. The aim of preparing the mentioned toolbox is to help the project managers responsible for the software projects in the process of planning software developers, labor force and schedule at the beginning phase of the project.

July 2014 202 pages

Key Words : Software Effort Estimation, Lines of Code (LOC), Function Point (FP), Parametric Estimation Models, Software Package

ÖNSÖZ ve TEŞEKKÜR

Bu tez çalışması için öncelikle tez danışmanım Sayın Doç. Dr. A.Egemen YILMAZ'a (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı) çok teşekkür ederim. Kendisi tez konusu başta olmak üzere pek çok konudaengin bilgi, görüş ve tecrübeleriyle hep yol göstermiştir. Bilimsel çalışmalarının her aşamasında pratik çözümleriyle destek olmuş, yeri geldiğinde akademik danışman yeri geldiğinde arkadaş olarak manevi desteğini esirgememiştir.

Tez çalışmamda verdikleri destek ve yaptıkları fedakarlıklardan dolayı hayat arkadaşım ve sevgili eşim Esra SONER'e ve ailemizin tüm fertlerine minnettarlığımı ifade etmek isterim. Bu çalışmayı yoğun iş temposunda çoğu durumda da kendilerinden almak zorunda kaldığım zamanlarda gerçekleştirdim.

Ayrıca, her türlü vermiş olduğu destekten, özellikle çalışma yoğunluğunda göstermiş olduğu hoşgörüden dolayı KHO SAVBEN Teknoloji Yönetimi Anabilim Dalı Başkanı Yrd.Doç.Dr. Celal EVCI'ye, bilgi ve görüşlerini paylaşan KHO Elektronik Mühendisliği Bölüm Başkanı Dr.Fatih YAMAN'a teşekkürü borç bilirim.

Taner SONER

Ankara, Temmuz 2014

İÇİNDEKİLER

TEZ ONAY SAYFASI	
ETİK	i
ÖZET	ii
ABSTRACT	iii
ÖNSÖZ VE TEŞEKKÜR	iv
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	x
1. GİRİŞ	1
2. YAZILIM PROJE BÜYÜKLÜK KESTİRİM YÖNTEMLERİ	7
2.1 İşlevsel Büyüklük Kestirim Yöntemleri	7
2.2 İşlev Puanı (Function Points)	8
3. YAZILIM TAHMİN METODOLOJİSİ VE MODELLERİ	12
3.1 Yazılım Tahmin Modellerinin Tarihi Gelişimi	13
3.2 Parametrik Tahmin Modelleri	16
3.2.1 KLOC tabanlı parametrik tahmin modelleri	16
3.2.2 FP tabanlı parametrik tahmin modelleri	19
3.2.3 Yazılım projelerin metrikleri tahmininde denklem kullanımı	20
3.2.3.1 Yazılım büyüklüğü ve maksimum takım boyutundan tahmin edilen proje çalışma eforu	23
3.2.3.2 Yazılım büyüklüğü ve maksimum takım boyutundan tahmin edilen proje teslim oranı	24
3.2.3.3 Yazılım büyüklüğü ve maksimum takım boyutundan adam başına proje teslim hızı	25
3.2.3.4 Yazılım boyutu ve maksimum takım boyutundan tüm projenin teslim hızı	25
3.2.3.5 Sadece yazılım boyutundan projenin efor tahmini	26
3.2.3.6 Sadece yazılım boyutundan proje süresi tahmini	27
3.2.3.7 Sadece yazılım boyutundan bütün takımının proje teslim hızı	27
3.3 Parametrik Olmayan Tahmin Modelleri	28
3.4 Karşılaştırma	29
4. YÖNTEM	30
4.1 Araştırmanın Konusu ve Amacı	30
4.2 Araştırmanın Önemi	30
4.3 Araştırmanın Yöntemi	31
4.4 Kod Satır Sayısı Tahmin Modelleri Kıyaslanması ve Yeni Modellerin Önerilmesi	33
4.4.1 Mevcut KLOC tabanlı tahmin modellerinin kıyaslanması	33
4.4.2 Yeni KLOC tabanlı parametrik tahmin modellerinin önerilmesi	47
4.4.2.1 Efor = $C_1 + C_2 \times (KLOC)^{C_3}$ yeni parametrik tahmin modeli	47
4.4.2.2 İkinci derece polinom yeni parametrik tahmin modeli	48
4.4.2.3 Üçüncü derece polinom yeni parametrik tahmin modeli	49
4.4.2.4 Dördüncü derece polinom yeni parametrik tahmin modeli	50
4.4.2.5 Doğrulama ve yeni modellerle mevcut modellerin kıyaslaması	51

4.4.3 Mevcut FP tabanlı tahmin modellerinin kıyaslanması.....	56
4.4.4 Yeni FP tabanlı parametrik tahmin modellerinin önerilmesi.....	65
4.4.4.1 Efor = $C_1 + C_2 \times (FP)^{C_3}$ yeni parametrik tahmin modeli.....	66
4.4.4.2 İkinci derece polinom yeni parametrik tahmin modeli.....	67
4.4.4.3 Üçüncü derece polinom yeni parametrik tahmin modeli.....	68
4.4.4.4 Dördüncü derece polinom yeni parametrik tahmin modeli.....	69
4.4.4.5 Doğrulama ve yeni modellerle mevcut modellerin kıyaslaması.....	70
5. MATLAB’TE GRAFİKSEL KULLANICI ARABİRİMİ (GUI)	
KULLANIMI VE YAZILIM PAKETİNİN HAZIRLANMASI	75
5.1 Grafikselsel Kullanıcı Arabirimi (GUI) Çalışma Prensipleri.....	75
5.2 MATLAB’da GUI Hazırlama Teknikleri.....	76
5.3 MATLAB GUIDE Yardımıyla GUI Hazırlama.....	76
5.4 GUI Bileşenlerini Çalışma Ortamına Ekleme.....	78
5.5 GUI Bileşenlerinin Açıklaması.....	79
5.5.1 Push button.....	79
5.5.2 Toggle button.....	80
5.5.3 Radio button.....	80
5.5.4 Check box.....	80
5.5.5 Edit text.....	80
5.5.6 Static text.....	80
5.5.7 Slider.....	80
5.5.8 List box.....	80
5.5.9 Pop-Up menu.....	80
5.5.9 Axes.....	80
5.5.10 Panel.....	80
5.5.11 ActiveX component.....	80
5.6 GUI Çalışma Alanının Boyutlarını Değiştirmek.....	82
5.7 GUI Bileşenlerini Hizalamak.....	82
5.8 GUI Tasarımını Kaydetme ve Çalıştırma.....	83
5.9 GUI Çalışma Ortamının Programlanması.....	84
5.10 Yazılım Projeleri Parametrik Tahmin Modelleri Yazılım Paketinin MATLAB GUI Ortamında Hazırlanması.....	86
5.10.1 Yazılım projeleri parametrik modelleri araç kutusunun bileşenlerinin eklenmesi.....	87
5.10.1.1 Yazılım projeleri gelişim süreci.....	89
5.10.1.2 Yazılım proje takımının yapısı ve büyüklüğü.....	91
5.10.1.3 Yazılım proje takviminin PERT (Project Evaluation and Review Technique) kullanarak belirlenmesi.....	92
5.10.2 Yazılım projeleri parametrik modelleri araç kutusunun bileşenlerinin düzenlenmesi.....	96
5.10.3 Yazılım projeleri parametrik tahmin modelleri araç kutusu nesnelerin fonksiyonlarının düzenlenmesi.....	98
5.10.4 Yazılım projeleri parametrik tahmin modelleri yazılım paketi.....	98
5.10.5 Yazılım projeleri parametrik tahmin modelleri araç kutusu simülasyonu.....	99
6. SONUÇ	101
KAYNAKLAR	103

EKLER	106
EK 1 Efor = C₁ + C₂ x (KLOC)^{C₃} Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	107
EK 2 İkinci Derece Polinom Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	109
EK 3 Üçüncü Derece Polinom Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	111
EK 4 Dördüncü Derece Polinom Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	113
EK 5 Efor = C₁ + C₂ x (FP)^{C₃} Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	115
EK 6 İkinci Derece Polinom Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	117
EK 7 Üçüncü Derece Polinom Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	119
EK 8 Dördüncü Derece Polinom Yeni Parametrik Tahmin	
Modeli MATLAB Kodu	121
EK 9 Yazılım Projeleri Parametrik Tahmin Modelleri Yazılım	
Paketi MATLAB Kodu	123
ÖZGEÇMİŞ	202

ŞEKİLLER DİZİNİ

Şekil 1.1 Yazılım projelerinde efor tahmini akış diyagramı.....	5
Şekil 2.1 İşlev puanı dönüşüm süreci.....	8
Şekil 3.1 Yazılım efor tahmin algoritmik modelleri kullanarak KLOC değerlerine bağlı olarak efor değişimi.....	18
Şekil 4.1 Tez çalışması aşamaları akış şeması.....	32
Şekil 4.2 $C_1+C_2 \times (KLOC)^{C_3}$ yeni tahmin modelinin efor – KLOC grafiği.....	48
Şekil 4.3 İkinci derece polinom yeni tahmin modelinin efor – KLOC grafiği.....	49
Şekil 4.4 Üçüncü derece polinom yeni tahmin modelinin efor – KLOC grafiği.....	50
Şekil 4.5 Dördüncü derece polinom yeni tahmin modelinin Efor-KLOC grafiği.....	51
Şekil 4.6 $C_1+C_2 \times (KLOC)^{C_3}$ yeni tahmin modelinin Efor – FP grafiği.....	67
Şekil 4.7 İkinci derece polinom yeni tahmin modelinin Efor – FP grafiği.....	68
Şekil 4.8 Üçüncü derece polinom yeni tahmin modelinin Efor – FP grafiği.....	69
Şekil 4.9 Dördüncü derece polinom yeni tahmin modelinin Efor – FP grafiği.....	70
Şekil 5.1 GUIDE hızlı başlat penceresi.....	77
Şekil 5.2 GUI çalışma ortamı.....	78
Şekil 5.3 GUI çalışma ortamına bileşen ekleme.....	79
Şekil 5.4 GUI çalışma alanını boyut ayarlaması.....	82
Şekil 5.5 Hizalama araç kutusu.....	83
Şekil 5.6 GUI tasarımın kaydedilmesi.....	84
Şekil 5.7 GUI tasarımına ait M-File gösterimi.....	85
Şekil 5.8 Nesnelerin Callback alt programlarının açılması.....	86
Şekil 5.9 Araç kutusu bileşenlerinin eklenmesi.....	87
Şekil 5.10 Model panellerinin araç kutusundaki görünüşü.....	88
Şekil 5.11 Üç, dört, beş ve altı numaralı panellerinin araç kutusundaki görünüşü.....	89
Şekil 5.12 Yazılım proje takvimi PERT ağ yapısı.....	94
Şekil 5.13 Yazılım paketi iş gücü dağılım paneli.....	95
Şekil 5.14 Yazılım paketi iş gücü dağılımı yüzde giriş paneli.....	95
Şekil 5.15 Yazılım projeleri parametrik modelleri araç kutusu GUI çalışma sayfası....	97
Şekil 5.16 Yazılım projeleri parametrik modelleri araç kutusu.....	98
Şekil 5.17 KLOC tabanlı yazılım projeleri için uygulama yapılması.....	99
Şekil 5.18 FP tabanlı yazılım projeleri için uygulama yapılması.....	100

ÇİZELGELER DİZİNİ

Çizelge 1.1 Projelerin zamanlama planları açısından sonuçlanma oranları.....	3
Çizelge 2.1 İşlev puanı karmaşıklık tablosu.....	9
Çizelge 2.2 Genel sistem özellikleri.....	10
Çizelge 2.3 Bazı programlama dillerinin LOC/FP oranları.....	11
Çizelge 3.1 KLOC tabanlı parametrik efor tahmin modelleri.....	16
Çizelge 3.2 FP tabanlı parametrik efor tahmin modelleri	18
Çizelge 3.3 Örnek yazılım proje metriklerinin tahmini.....	20
Çizelge 3.4 Yazılım büyüklüğü ile efor tahmin örneği.....	20
Çizelge 3.5 Yazılım büyüklüğü ve takım boyutu ile efor tahmin örneği.....	21
Çizelge 3.6 Yazılım büyüklüğü ve maksimum takım boyutundan efor tahmin parametreleri.....	22
Çizelge 3.7 Yazılım büyüklüğü ve maksimum takım boyutundan proje teslim oranı parametreleri.....	23
Çizelge 3.8 Yazılım büyüklüğü ve maksimum takım boyutundan adam başına hız parametreleri.....	24
Çizelge 3.9 Yazılım büyüklüğü ve takım boyutundan proje hızı parametreleri.....	25
Çizelge 3.10 Yazılım büyüklüğünden efor tahmini parametreleri.....	25
Çizelge 3.11 Yazılım büyüklüğünden proje süresi tahmini parametreleri.....	26
Çizelge 3.12 Yazılım büyüklüğünden tüm takımın proje teslim hızı parametreleri.....	27
Çizelge 3.13 Modellerin karşılaştırma matrisi.....	28
Çizelge 4.1 KLOC tabanlı parametrik efor tahmin modelleri.....	33
Çizelge 4.2 Uzay havacılık (NASA) yazılım projeleri veri seti.....	33
Çizelge 4.3 ISBSG 96 adet yazılım projeleri veri seti	46
Çizelge 4.4 Parametrik modellerle hesaplanan eforların hataları	49
Çizelge 4.5 Yeni KLOC modellerle hesaplanan eforların hataları.....	50
Çizelge 4.6 FP tabanlı parametrik efor tahmin modelleri.....	50
Çizelge 4.7 ISBSG yazılım projeleri veri seti	51
Çizelge 4.8 ISBSG 96 adet yazılım projeleri veri seti	63

Çizelge 4.9 Parametrik modellerle hesaplanan eforların hataları	66
Çizelge 4.10 Yeni FP modellerle hesaplanan eforların hataları	66
Çizelge 5.1 Yazılım projesi iş gücü dağılımı için zamanlama tahminleri.....	83

1. GİRİŞ

Yazılım projelerinde doğru maliyet, zamanlama, planlama ve iş gücü tahmininde bulunma proje yöneticileri için kritik öneme sahiptir. Yazılım endüstrisi doğru efor tahminleri yapma konusunda yeteri kadar olgunlaşmamıştır. Son 30 yılda literatüre girmiş birçok yazılım metriklerini tahmin etme metodu, aracı ve modeli mevcuttur. Bu tahmin teknikleri yardımıyla yazılım projelerinin efor, zamanlama, teslim hızı ve maliyet tahmini gibi büyüklükleri hesaplanabilir. Bu modelleri kategorize edersek, uzman tahmini modeli, analogi tabanlı tahmin metotları, parametrik modeller, bulanık mantık tahmin şemaları, yapay sinir ağı tabanlı metotlar, karar ağacı yöntemi ve deneysel yöntemler olarak sıralanabilir (Basha ve Dhavachelvan 2010).

Bu çalışmada literatürde yer alan parametrik tahmin modelleri detaylı bir şekilde incelenmiş ve her bir modelin yazılım metrik tahmin denklemleri bir araya toplanmıştır. Ayrıca gerçekleştirilen yazılım projelerinin gerçek yazılım büyüklüğü ve işgücü değerleri kullanılarak yeni parametrik tahmin modelleri önerilmiştir. Mevcut ile önerilen yeni modeller arasında performans kıyaslaması yapılmıştır. Söz konusu kıyaslama sonucunda hata ölçütleri kullanılarak daha iyi tahminde bulunan modellerin önerilen yeni parametrik tahmin modelleri olduğu doğrulanmıştır.

Bu kapsamda yazılım proje başlangıcında proje yöneticilerine yardımcı olacak şekilde parametrik tahmin modelleri yazılım paketi hazırlanmıştır. Bu yazılım paketinin hazırlanmasındaki amaç, yazılım projeleriyle ilgilenen proje yöneticileri ve yazılımcılara proje başlangıcında yazılımcı iş gücü ve proje süresi planlamasında kolaylık sağlamaktır. Bu araç kutusunda literatürde mevcut parametrik tahmin modelleri ile önerilen yeni modeller bir araya getirilmiştir. Söz konusu araç kutusu marifetiyle proje yöneticileri literatürde mevcut parametrik tahmin yöntemleri efor kestirimlerini proje öncesinde görebilmektedir. Ayrıca gerçek yazılım proje verileriyle hata oranı azaltılmış daha iyi tahminlerde bulunan yeni modeller araç kutusuna ilave edilerek mevcut modeller arasında kıyaslama imkânı sağlanmıştır.

Yazılım projeleri doğru zamanlama ve maliyet planı yapılamaması konusunda öne çıkmış projelerdir. Yazılım proje yöneticileri için doğru yazılım büyüklüğü ve efor tahminleri yapmak önemli bir gerektir. Bu nedenle iyi bir efor tahmini ile yazılım projelerinde kolayca zamanlama ve maliyet planı yapılabilir. Eğer yanlış tahminlerle proje yönetim faaliyetleri planlanırsa yüksek maliyetlerle karşı karşıya kalınmaktadır. Yazılım projesine başlamadan önce yazılım büyüklüğü, iş gücü, proje süresi ve maliyet gibi yazılım proje metrikleri tahmin edilebilirse yaşanılacak sorunlar en aza inecektir.

ABD’de bulunan ve bir araştırma şirketi olan The Standish Group, 1993 yılından beri yazılım ve bilgi teknolojileri projelerinde meydana gelen başarısızlıkları incelemeye devam etmektedir. Bu kapsamda söz konu araştırma şirketi proje başarısızlıklarının içeriği ve üstesinden gelme yollarını tespit etmeye çalışmaktadır (Jones 1998).

2003 Araştırma şirketinin 1999 yılında yayımladıkları CHAOS Chronicles raporda tespit edilen hususların bazıları aşağıda sunulmaktadır:

- 23.000 uygulama incelemeye alınmıştır.
- Başarısız projelerin maliyeti 75 Milyar ABD \$’dır.
- Yazılım projelerinin sadece %26’sı başarılı bir şekilde tamamlanmıştır.
- Bütçesini veya planlı zamanını aşan (challenged) proje oranı %46’dır.
- İptal edilen proje oranı % 28’dir.
- Toplam maliyet aşımı 22 Milyar ABD \$’dır.

Yazılım projelerinde maliyet tahmini yapılması ilk olarak yazılım büyüklüğünün kestirilmesi, buna bağlı efor tahmin modelleri kullanımı ile iş gücünün hesaplanması ve son olarak proje zamanlama programının tahminin yapılması sürecini içerir. Standish Group’un hazırlığı raporda hem kamu hem de özel projelerden oluşmakta olan yazılım projelerin %90’ının bütçe ve zaman aşımı nedeniyle başarısız oldukları görülmüştür. Bunların %33’ü daha tamamlanmadan iptal edilmiştir. Yaklaşık üçte biri ilk tahmin edilen ve bütçelenen maliyeti %150 ile %200 arasında ve ortalama olarak %189 aştığı, yaklaşık üçte birinin de %200 ile %300 arasında ve ortalamada %222 aştığı gözlemlenmiştir.

Yazılım projelerindeki başarısızlığın temel nedenlerine bakıldığında şu şekilde bir araya toplanabilir. Teknik, yönetimsel ve sosyal olarak üç ana başlık altında sıralanabilir. Teknik nedenleri geniş bir perspektiften incelediğimizde, yazılım projelerinin başarısızlığına neden olan pek çok etken bulunmaktadır. Söz konusu etkenler arasında, yazılımın doğası gereği ve yazılımın büyüklüğü kestirimi ile direkt ilişkili karmaşıklık ön plana çıkmaktadır. Tüm yazılım üreten firmalarda, büyük ölçekli sistemlerin iptal edilme ya da gecikmiş şekilde tamamlanabilme olasılıkları küçük ölçekli uygulamalara oranla daha yüksektir. Çizelge 1.1’de, projelerin zamanlama planları açısından sonuçlanma oranlarını göstermektedir (Jones 1996).

Çizelge 1.1 Projelerin zamanlama planları açısından sonuçlanma oranları

İşlev Puanı	Erken	Zamanında	Gecikmiş	İptal Edilmiş
1 İşlev Puanı	14.68 %	83.16 %	1.92 %	0.25 %
10 İşlev Puanı	11.08 %	81.25 %	5.67 %	2.00 %
100 İşlev Puanı	6.06 %	74.77 %	11.83 %	7.33 %
1000 İşlev Puanı	1.24 %	60.76 %	17.67 %	20.33 %
10000 İşlev Puanı	0.14 %	28.03 %	23.83 %	48.00 %
100000 İşlev Puanı	0.00 %	13.67 %	21.33 %	65.00 %
Ortalama	5.53 %	56.94 %	13.71 %	23.82 %

Yazılım projelerinin başarısızlığında sıkça karşılaşılan teknik nedenler aşağıda sıralanmıştır:

- Yetkin olmayan yazılım teknolojilerinin kullanılması,
- Uygun olmayan yazılım araçlarının seçimi,
- İş süreçlerinin belirlenmemesi,
- Çözümlemenin (analysis) uygun bir şekilde gerçekleştirilmemesi,
- Geçmişe yönelik yazılım ölçüm verilerinin bulunmaması,
- Etkin bir mimarinin seçilmemesi,
- Etkin geliştirme yöntemlerinin kullanılmaması,
- Tasarım gözden geçirmelerinin (design review) yapılmaması,
- Kod denetimlerinin (code inspection) yapılmaması,

- Uygun olmayan, disiplinsiz test yöntemlerinin uygulanması,
- Belirtim (specification) ve tasarımın el ile yapılması,
- Düzenleşim (configuration) denetiminin uygulanmaması,
- Kullanıcı gereksinimlerinin belirsizliği veya %30'dan daha fazla değişmesi,
- Uygun programlama dillerinin kullanılmaması,
- Aşırı ve ölçülmemiş karmaşıklık düzeyi,
- Yeniden kullanılabilir bileşenlerin kullanılmaması,
- Uygun veritabanı planlamalarının ve tasarımının yapılmaması,
- Yeni teknolojileri kullanmaya geçişin birdenbire olması,
- Uzmanlaşmış personel ve deneyim eksikliği.

Yazılım proje yönetimindeki maliyet tahmini, projelerin doğru planlaması sırasında başlayan bir ihtiyaçtır. İhtiyaç duyulan proje için ne kadar sürede tamamlanacağını ve nasıl bir iş gücüne ihtiyaç duyulacağını tahmini yapılmadan yazılım projesinin işgücü ve zamanlama planlamasını yapmak zordur. Efor tahmini yapılabilirse proje programını belirlemek kolaylaşır. Bu nedenle yazılım projelerinde etkili bir planlama için iş gücü ve zamanlama tahmini yapmak gerekmektedir. Ancak doğru kestirim yapmanın zorluğu ortadadır. Proje başlangıcındaki eldeki veriler doğru tahmin yapmak için yeterli olmayabilir. Proje yöneticileri için tahmin modellerini kullanarak doğru işgücü ve zamanlama kestirimi yapılabilme gereği açıktır.

Yazılım projelerinde maliyet tahmini 1970'li yıllarda başlamış ve son on yıllarda birçok model ortaya konulmuştur. Doküman haline getirilmiş birçok yazılım tahmin etme metodolojisi, aracı ve modeli mevcuttur. Son 30 yılda birçok kantitatif yazılım maliyet modeli geliştirilmiştir. Bunlardan biri donanım maliyetinin belli bir yüzdesini alarak yazılım maliyetinin tahminini yapmaktır. Bu modellerin çoğunda çeşitli değişkenler giriş olarak kullanılır, çıkış olarak ise personel iş gücü tahmin edilir.

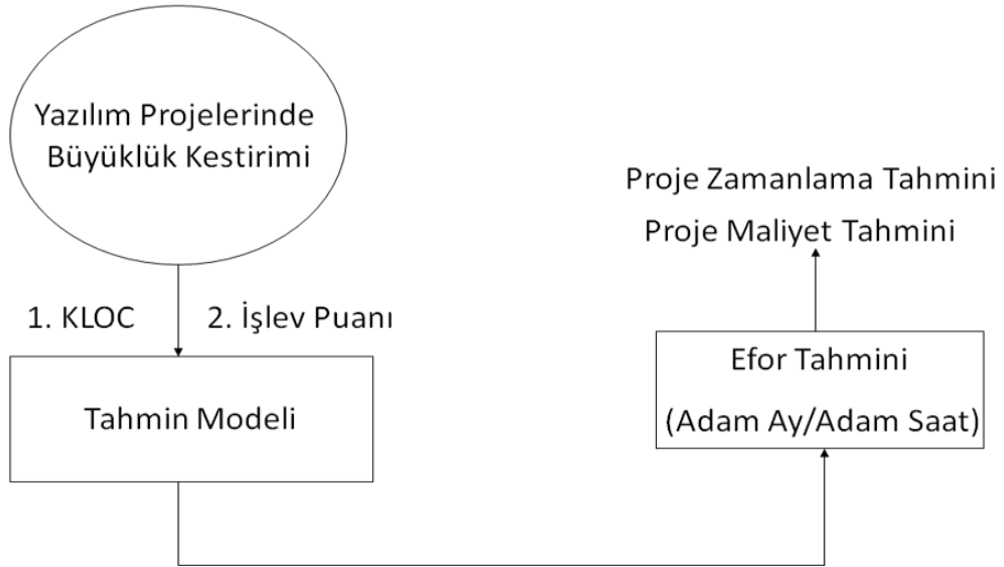
Tahmin modelleri; karmaşıklık, sistem boyutu, geliştirilme metodolojisi, proje karakteristiğinin tanımlama parametresi gibi büyüklükleri giriş olarak kabul eden analitik formüller kullanır.

Birçok model, projelerin yazılım tasarımı, geliştirme ve test etme aşamalarını ile ilgili tahminde bulunduğu için, modelin yazılım projesi ömür devrinin hangi safhaları için nasıl bir tahminde bulunacağı önemlidir. Bütün metodolojilerin ve modellerin önemli kısıtlamaları söz konusudur.

Yazılım proje efor ve zamanlama tahminlerinde dört temel adım vardır. Bu adımlar müteakip maddelerde sıralandığı gibi özetlenebilir.

- Kod Satır Sayısı veya İşlev Puanı (Function Points) yöntemi yardımıyla yazılım projelerinde büyüklük tahmini.
- Adam-ay veya adam-saat olarak yazılım efor tahmini.
- Gün olarak proje zamanlama tahmini.
- Proje efor ve zamanlamasından proje maliyet tahmini.

Şekil 1.1’de Yazılım Projelerinde Efor Akış Diyagramı sunulmaktadır.



Şekil 1.1 Yazılım projelerinde efor tahmini akış diyagramı

Yazılım projeleri maliyet tahmini şekil 1.1’de sunulduğu gibi ilk adım olarak yazılım büyüklük kestiriminin gerçekleştirilmesi gerekmektedir. Yazılım büyüklük kestiriminde iki temel yöntemden söz edilebilir. Kod satır sayısı yazılımdaki kod satırlarını saymakla elde edilen geleneksel bir yöntemdir. İşlev Puan Analizi ise IBM’in satır sayısına

alternatif olarak yazılım büyüklük ölçümü için Allan Albrecht tarafından ortaya çıkarılmıştır. Eğer proje ile ilgili girdi çıktı gibi özellikler tahmin edilebiliyorsa, bunlar kullanılarak geliştirilecek sisteme ait bir İşlev Puanı hesabı yapılabilir ve sonuçlar Satır Sayısına (LOC) çevrilebilir.

Yazılım büyüklüğü kestiriminden sonraki adımda ise bir tahmin modeli kullanılarak yazılım projesi için adam ay veya adam saat cinsinden efor tahmini gerçekleştirilir. Son adımda efor tahmini sonucu bulunan çıktılar kullanılarak proje zamanlama ve maliyet tahmini gerçekleştirilir.

Bu tez çalışması ile literatürde mevcut olan parametrik tahmin modelleri kod satır sayılı ve işlev puanı başlığı altında iki ana grupta bir araya getirilmiştir. Denklem setleri kullanılarak gerçek yazılım proje metrikleri ile performans kıyaslaması yapılmış ve hata ölçütleri ile hata oranları tespit edilmiştir. İkinci kısımda ise gerçek yazılım projelerinin büyüklük ve iş gücü metrikleri kullanılarak nümerik yöntemler yardımıyla kod satır sayılı ve işlev puanlı yeni modeller önerilmiştir. Kullanılan veri setinin % 80 proje model önerilmek için kullanılırken diğer % 20'lik kısım doğrulama maksadıyla kullanılmıştır. Literatürdeki modellerle yeni modellerini performans kıyaslaması yine hata ölçütleri kullanılarak yapılmıştır.

Son bölümde ise MATLAB GUI ortamında mevcut modeller ile yeni önerilen modeller bir araya getirilerek bir yazılım paketi hazırlanmıştır. Yazılım paketi ile yazılım projelerinin iş gücü, proje süresi maliyeti gibi büyüklükleri kolaylıkla tahmin edilmektedir. Bu yazılım paketi ile yazılım proje yöneticilerine proje başlangıcında bir takım tahminler bulunarak yardımcı olmak hedeflenmiştir.

2. YAZILIM BÜYÜKLÜK KESTİRİM YÖNTEMLERİ

Yazılımın ölçülebilmesi, harcanılan zaman, emek, proje büyüklüğü ve kalite gibi faktörlerin belirlenmesine olanak sağlamaktadır. Organizasyonlar, bu verilere dayanarak ileride alacakları projeler için kestirim yapabilme imkânı bulabilmektedirler. Yazılım projelerinde kaliteyi arttırmak, her şeyden önce doğru ölçme yöntemlerine bağlıdır. Yazılım büyüklük kestiriminde kullanılan yöntemler; teknik büyüklük kestirim yöntemleri ve işlevsel büyüklük kestirim yöntemleri olarak sınıflandırılmıştır. Teknik büyüklük kestirim yönteminde en çok bilinen yöntem Satır Sayısı (Lines of Code - LOC) yöntemidir (Fenton 1994). Uygulamanın büyüklüğünü anlamak için bilgisayar programlarındaki kodların satırlarını sayma en geleneksel ve en yaygın şekilde kullanılan yazılım ölçümüdür. Kolaylığı ve doğrudan ölçülebilirliği açısından en fazla kullanılan yazılım ölçme yöntemi, satır sayısıdır. Ancak kullanılan bu yöntemin bazı dezavantajları vardır. Programlama dili farkı, deneyim farkı gibi nedenlerle LOC yöntemi projenin büyüklüğünü tahminleme de direkt olarak kullanılan bir yöntem değildir. Bu nedenle İşlevsel Büyüklük Ölçümü (Functional Size Measurement - FSM), yöntemleri kullanılmaktadır. Bu yöntemde yazılımın işlevselliğini temel alınmaktadır (Borandağ vd. 2013).

2.1 İşlev Puanı Tahmin Yöntemleri

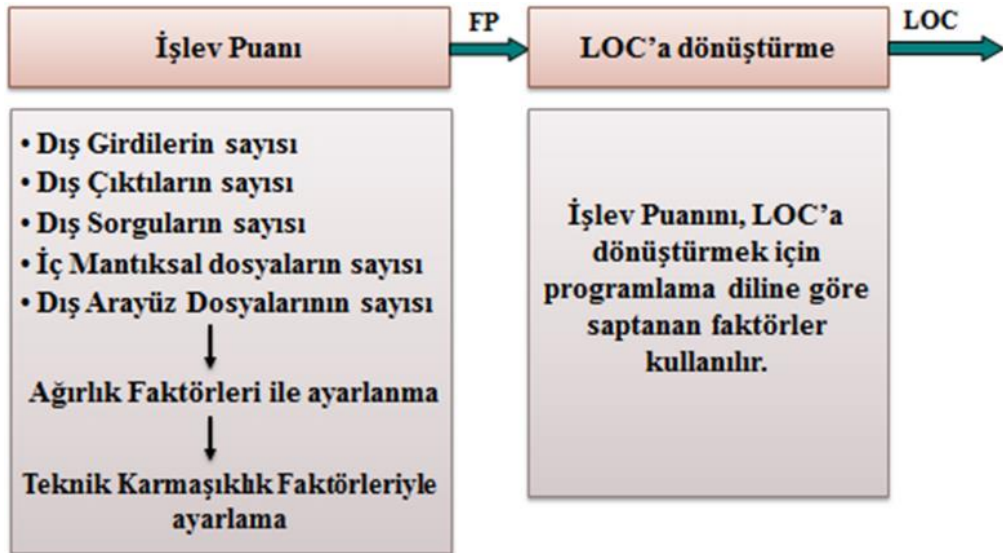
İşlev Puanı Yöntemi IBM tarafından 1979 yılında ilk olarak çalışılmıştır. Temel hedef geleneksel yöntem olan kod satır sayısını sayma büyüklük kestirim yöntemine alternatif bir yöntem keşfetmektir. İşlev Puanı Yöntemi Allan Albrecht tarafından ortaya çıkartılmıştır. 1983’de ise, Allan Albrecht ve John Gaffney tarafından Yönetim Bilgi Sistemlerinin büyüklüğünü ölçmek için FSM yöntemi geliştirilmiştir (Fetcke, Abran ve Dumke 2001). Daha sonra farklı kitleler tarafından orijinal FPA yöntemi üzerinde yapılan oynamalarla, aralarında ölçüm yöntemi farklı birçok FSM yöntemi geliştirilmiştir. Aşağıda bazı işlevsel büyüklük kestirim yöntemleri verilmiştir:

- İşlev Puanı (Function Points - FP),

- IFPUG İşlev Puanı Analizi (IFPUG Function Points Analysis - IFPUG FPA),
- Mark II İşlev Puanı (Mark II Function Points - MK II FP),
- Nesma İşlev Puanı (Nesma Function Points),
- Tam İşlev Puanı (Full Function Points - FFP),
- COSMIC Tam İşlev Puanı (COSMIC Full Function Points - COSMIC FFP),
- Nesne Puanı (Object Points),
- Nesne-Tabanlı İşlev Puanı (Object- Oriented Function Points - OO FP),
- Nesne-Tabanlı Yöntem İşlev Puanı (Object-Oriented Method Function Points - OOmFP)

2.2 İşlev Puanı (Function Points)

Bu yönteme göre verimlilik hesaplanan işlev puanına göre eforun adam-ay olarak belirlenir. Yazılım projesi ile ilgili girdi çıktı gibi parametreler tahmin edilebiliyorsa, bunlar kullanılarak geliştirilecek sisteme ait bir İşlev Puanı hesabı yapılabilir ve sonuçlar Satır Sayısına (LOC) çevrilebilir. Bu satır sayısından maliyet, emek ve süre tahmini yapılabilir. İşlev puanı dönüşüm süreci, şekil 2.1’de gösterilmiştir (Borandağ vd. 2013).



Şekil 2.1 İşlev puanı dönüşüm süreci

İşlev Puanı'nın hesaplanması ve Satır Sayısı'na dönüştürülmesi süreci beş adımdan oluşmaktadır.

Dış girdiler, uygulamanın dışından uygulamanın içine doğru olan süreçleri ve işlenebilir verileri gösterir. Veri genellikle uygulamaya içine eklenebilir, silinebilir veya güncellenebilir. Dış girdilere örnek olarak; kullanıcının bilgi girişi yaptığı veri giriş ekranları ve mantıksal dâhili dosyalar verilebilir.

Dış çıktılar, verinin uygulama sınırları içinden dışarı çıkmasına izin veren süreç veya işlemlerdir. Dış çıktılara örnek olarak; raporlar, doğrulama mesajları ve ekran çıktıları verilebilir.

Dış Sorgular: Kullanıcı isteği doğrultusunda alınan hızlı veri çıkışlarıdır. Dış sorgular dosyada saklanan veriyi değiştirmez veya güncellemez; sadece bilgiyi okumaktadır.

İç mantıksal dosyalar ise uygulama sınırları ile birlikte verilerin saklandığı mantıksal bir dosyadır. İç mantıksal dosyalara örnek olarak, dâhili kullanıcı verileri, saklanan veriler verilebilir. İşlev puanını hesaplanmasında takip edilen aşamaların ilkinde parametre sayıları tespit edilir. İkinci adımda karmaşıklık durumuna bu parametreler çizelge 2.1'deki ağırlık değerleri ile her biri ayrı ayrı çarpılıp toplanarak düzeltilmemişi işlev puanı hesaplanır.

Çizelge 2.1 İşlev puanı karmaşıklık tablosu

Bileşenler	Basit	Orta	Karmaşık
Dış Girdiler	3	5	6
Dış Çıktılar	4	6	7
Dış Sorgular	3	6	6
İç Mantıksal Dosya	7	15	15
Dış Arayüz Dosya	5	10	10

Her bir bileşenin zorluk derecesi basit, orta ve karmaşık gibi çizelge 2.1’de verilen rakamsal değerlere bağlı olarak ölçülebilmektedir. Bu ölçülen değerler toplanarak Düzeltilmemiş İşlev Puanı’ını oluşturmaktadır.

Üçüncü adımda ise Teknik Karmaşıklık Faktörünün (Technical Complexity Factor - TCF) hesaplanması için sistemin genel özelliklerinden faydalanılır. Çizelge 2.2’de verilen 14 genel sistem özelliği kullanılarak sistemin beklenen uygulama zorluğu için ilave bir Teknik Karmaşıklık Faktörü (TCF) hesaplanır.

Çizelge 2.2 Genel sistem özellikleri

Sıra.Nu.	Genel Sistem Özellikleri	Kısa Açıklama
1	Veri İletişimleri	Sistemin uygulaması ile bilgi değişimi veya transferinde yardımcı olmak için kaç tane iletişim aracı vardır?
2	Dağıtılan Veri / İşleme	Dağıtılan bilgi ve işleme fonksiyonları nasıl idare edilmektedir?
3	Performans	Hedefler, yanıtlama zamanı ve iş çıkarma performansı önemli midir?
4	Çok Kullanılan Konfigürasyon	Uygulamanın idare edileceği mevcut donanım platformu ne kadar yoğun kullanılmaktadır?
5	İşlem Oranı	İşlem oranı yüksek midir?
6	Çevrimiçi Veri Girişi	Hangi oranda bilgi çevrimiçi girilmektedir?
7	Son Kullanıcı Verimliliği	Uygulama son kullanıcı verimliliği için mi tasarlanmıştır?
8	Çevrimiçi Güncelleme	Kaç veri dosyası çevrimiçi güncellenmektedir?
9	Karmaşık İşlem Yapma	Dâhili işlem yapma karmaşık mıdır?
10	Yeniden Kullanılabilirlik	Uygulama yeniden kullanılabilir olması için mi tasarlanmıştır?
11	Dönüştürme / Kurulum Kolaylığı	Sistemde otomatik dönüşüm ve kurulum da dâhil edilmiş midir?
12	İşlevsel Kolaylık	Yedekleme, başlatma ve kurtarma gibi operasyonlar ne kadar otomatiktir?
13	Çoklu Saha Kullanımı	Uygulama çoklu örgüte sahip çoklu sahalar için özellikle mi tasarlanmış ve desteklenmiştir?
14	Değişimi Kolaylaştırma	Uygulama kullanıcı tarafından kullanım kolaylığı ve değişimi kolaylaştırmak için desteklenmiştir?

14 genel sistem özelliği için verilen her bir soruya 0 ile 5 arasında değerler verilir ve bu değerler toplanarak Etki Derecesi (Degree of Influence - DI) hesaplanır. Etki derecesi ilgili denklem eşitlik 3.2’de sunulmuştur.

0: hiç yok ya da etkisiz,

1: önemsiz etki,

2: az etkili,

3:orta düzeyde etkili

4: önemli düzeyde etkili,

5: güçlü etki

$$DI = \sum_{i=1}^{14} \text{Cevap}_i \quad (3.2)$$

$$TCF = 0,65 + 0,01 \times DI \quad (3.3)$$

Dördüncü adımda ise İşlev Puanı aşağıda verilen formül kullanılarak hesaplanır.

$$FP = UFP \times TCF \quad (3.4)$$

İşlev Puanı’nı, Satır Sayısına dönüştürmek için aşağıdaki formülden yararlanılır. Kullanılan programlama diline göre kod satır sayısı / işlev puanı dönüşüm oranları değişiklik arz etmektedir. Bu oranlar çizelge 2.3’te sunulmuştur.

$$LOC = FP \times \text{Prog. Dili LOC Katsayısı} \quad (3.5)$$

Çizelge 2.3 Bazı programlama dillerinin LOC/FP oranları

Programlama Dili	Kod Satır Sayısı / İşlev Puanı (LOC / FP)
C	128
C ++	53
COBOL	107
FORTRAN	105
DELPHİ 5	18
JAVA 2	46
VISUAL BASIC 6	24
SQL	13
Dördüncü Kuşak Diller	20
Nesne Tabanlı Diller	30

3. YAZILIM TAHMİN METODOLOJİSİ VE MODELLERİ

Doküman aktarılmış birçok yazılım tahmin etme metodolojisi, aracı ve modeli mevcuttur. Son 30 yılda çok sayıda kantitatif yazılım maliyet modeli geliştirilmiştir. Günümüzde yaygın olarak kullanılan yazılım maliyet tahmin yöntemleri dört grup altında toplamak mümkündür (Heemstra 1992). Bunlar;

- Uzmanlar görüşüyle yapılan maliyet tahmini,
- Benzetme yoluyla çıkarım yapılarak yapılan maliyet tahmini,
- Parametrik modellerle maliyet tahmini,
- Tümevarım – Tümdengelim maliyet tahmini.

Uzmanlar görüşü alınarak gerçekleştirilen maliyet tahmini yönteminde, maliyet tahmini gerçekleştirilecek yazılım projesinin uzmanın tecrübe alanının projeye ne ölçüde ilişkili olduğu çok önemlidir. Uzman kişi, uzmanlık alanının yazılım projesiyle ilişkili olduğu olduğu ölçüde doğru bir tahmin yapabilir. Uzman görüşüne dayalı tahminler daha çok subjektiftir. Uzmana göre değişiklik arz eder.

Benzetme yoluyla çıkarım yapılarak yapılan maliyet tahmini yönteminin doğru olarak uygulanabilmesi için önceden yapılan projelerin verilerinin ve karakteristik özelliklerinin bir yerde kaydedilmiş olması ve bu iş için kullanıma hazır olması gerekmektedir. Bu yöntem, küçük ve orta büyüklükteki projelerde tahmin edilen metrikler tatmin edici olabilmektedir. Ancak büyük ölçekli projelerde küçük ölçekli projelere göre daha kötü tahminler yapılabilmektedir.

Parametrik modellerle maliyet kestiriminde, proje zamanlama ve maliyet tahmini çok parametrelili fonksiyonlar yardımı ile elde edilen denklemler belirlenmektedir. Fonksiyonun değişkenleri proje için kritik olan yazılım büyüklüğü gibi maliyet parametrelerini ifade etmektedir. Tahmin yönteminin esası kestirim algoritması ve fonksiyonda kullanılan parametrelerdir. Fonksiyonlarda kullanılan parametrelerin ilk değerleri daha önceki projelerden elde edilen deneysel verilerden çıkarılmaktadır.

Tümevarım – Tümdengelim maliyet tahmininde geliştirilecek sistemin genel karakteristikleri göz önünde bulundurularak bir maliyet tahmininde bulunulur. Sonra toplam maliyet birleşenleri ortaya çıktıkça sistem birleşenlerine uygun bir şekilde paylaşılır. Tümdengelim maliyet kestiriminde ise geliştirilecek sistemin her bileşenin maliyeti o bileşeni geliştirecek yazılım geliştiricisi tarafından belirlenerek çıkarılır. Her bir bileşenin maliyeti toplanarak toplam maliyet bulunur (Adalier 2008).

3.1 Yazılım Tahmin Modellerinin Tarihi Gelişimi

Yazılım büyüklüğünün ölçülmesi kapsamında kullanılan geleneksel yöntem yazılımdaki kod satır sayısının sayılması yöntemidir. Kod satır sayısı yöntemi ilkel olmasında dolayı üzerine tartışılan bir metottur. Satır sayısına alternatif olarak IBM şirketince İşlev Puanı Yöntemi Allan Albrecht tarafından 1979 yılında yazılım büyüklük ölçümü için ortaya çıkarılmıştır. Yine 70'li yıllarda geliştirilen çevrimsellik karmaşıklığı (*Cyclomatic Complexity*) yazılımın karmaşıklığını ölçmeyi amaçlayan bir teoremin kullanımınıdır. Yazılımın tahmini modellerinde bir fonksiyon elde edilir. Yeni yazılım geliştirme projeleri için bu fonksiyon maharetiyle yazılım metrik tahminleri yapılabilirdi. Bu fonksiyonlar elde edilirken kod satır sayısı gibi temel parametreler temel alınmaktaydı. Bununla beraber fonksiyon parametreleri olarak etkileşimi, ekran sayısı, kullanılan dosya sayısı gibi parametreler de ilave edildi.

Parametrik modellerinde yazılım büyüklüğü olarak Kod Satır Sayısı ve İşlev Puanı temel alınarak yazılım metrikleri tahmini yapılır. Parametrik modellerin tarihi gelişimini aşağıda sunulmuştur.

Waltson ve Felix tarafından 1977 yılında IBM'de geliştirilen Waltson-Felix Modeli, efor ile kaynak kod satır sayısı arasında ilişki sağlar.

Bailey ve Basili (1981), verilen geliştirme ortamına en iyi uyum sağlayan efor tahmin denklemlerin geliştirilmesine izin veren bir meta-model olarak 1981 yılında

tanımlanmıştır. Oluşan tahmin modeli, projelerin arasındaki farklar ve onların çevresel faktörleri gibi veriler toplanmasına dayalı COCOMO ile benzerdir.

Halstead (1977), programlama yapısında derinlemesine analize gerek olmadan hata oranını öngören modeli 1977 yılında önermiştir. Bu model yazılımın kod uzunluğu ve hacim metrikleri önermek için kullanılır.

1977 yılında yayımlanan Doty Modeli kod satır sayısı için eforları tahmin etmede kullanılır. Bu model kullanıcı katılımlı, hafıza sınırlamalı ve müşteri odaklı değişiklik yeteneğine sahiplik gibi yazılım geliştirme ortamlarını çeşitli yönleriyle oluşturur.

Yazılım Maliyet Tahminleme çalışmalarında öncelikle yapılması gereken yazılım geliştirme çabasının tahmin edilmesini sağlamaktır. Bunu sağlamak için ürünün büyüklüğü, sistemin karmaşıklığı ve uygulama alanını göz önünde bulundurmak gerekir. (Fenton, N.E. ve Pfleeger S., 1998).

KLOC tabanlı parametrik tahmin modellerinde denklem seti incelendiğinde yapısı denklem 5.1'deki gibidir.

$$\text{Çaba} = A + B(\text{KLOC})^c \quad (3.1)$$

Yazılım kodun satır sayısının bir kuvvetin bir düzeltme çarpanı ve bir sabit eklenerek beraber kullanarak ihtiyaç duyulan yazılımcı eforunu hesapladığı düşünülmüştür. A, B ve C deneysel olarak elde edilen sabitlerdir. Büyüklük ise KLOC ve FP olarak yazılım boyutudur. Gerçekleşen bazı yazılım geliştirme projelerinden elde edilen verilerden eşitlik 5.1'de bulunan A, B ve C sabitleri elde edilmeye çalışılmıştır. Diğer bir ifadeyle elde edilen belli sayıda proje verileri kullanılarak, matematiksel olarak A, B ve C sabitleri bulunulmuştur.

Bu bölümde tanıtılan tahmin yöntemleri, satır sayısı temelinde olsa da, işlev puanına dayandırılması da mümkün olabilmektedir. Hesapların temelindeki önemli bir nokta,

gereken eforun ürünün büyüklüğünün bir kuvveti şeklinde hızla artmasıdır. Bunun sonucu olarak rasgele yaklaşımla yazılmış bir programın yazılması başta oldukça kolay olmasına rağmen yazılım büyüklüğü arttıkça iş gücü kestiriminde süreç zorlaşmaktadır. Birkaç bin satıra yakın büyüklükteki bir programı bir hafta süresinde yazmak mümkündür. Daha sonra program büyüdükçe hız yavaşlar. Bu program birkaç on bin satıra ulaştığında bir satır ilave etmenin bedeli birkaç günlük belki de bir kaç aylık çabadır. Dolayısıyla eklemenin oluşturacağı yan etkileri takip etmek zorlaşmıştır.

Temel olarak iki sebepten dolayı yazılım maliyet tahmini yapmak oldukça zordur. Birinci neden yazılımın soyut, elle dokunulamayan, fiziksel olarak alışılmış ürün tanımı dışında olmasıdır. İkinci neden ise yazılım geliştirme işi fiziksel bir işten ziyade entelektüel bir iş olmasıdır.

Londeix 1987'de makro tahminleme tekniklerinin ve mikro tahminleme tekniklerinin olduğunu belirtmiştir. Bu teknikler realisttik yazılım maliyet tahminlemesinden çok yönetsel amaçlar için kullanılmaktadır. Yukarıdan aşağıya ve aşağıdan yukarıya teknikleri uzman yargılarda, parametrik ve benzeşim tekniklerinde kullanılmaktadır. Uzman yargı tekniği bir uzman seçme ve bu uzmanın tecrübe, bilgi birikimi ile tahmin etmesi üzerine kuruludur. Parametrik ve benzeşim teknikleri önceki verileri bir veri tabanında kaydederek maliyeti tahmin etmede kullanılır. Bu çalışmada temelde parametrik benzeşim bazlı maliyet tahminleme tekniği detaylı olarak ele alınacaktır. Parametrik modellerde modelin doğruluğu ve tutarlılığı analistin bu parametreleri ortaya çıkarma yeterliliğiyle beraber artar. Ölçüt veri tabanının kullanım verimliliği ve doğruluğu ile beraber başarımlar artacaktır. Burada ölçütlerin bulunmasında deneysel ve gözlemsel parametrik model ve teorik parametrik model kullanılabilir. Deneysel parametrik modelde geçmişteki projelerden elde edilen verilerden elde edilen fonksiyonlar kullanılır. Parametrelerin ve sabitlerin bulunmasında istatistiksel analizler yapılır. Ünlü Rayleigh eğrisi bu şekilde ortaya çıkmıştır. Daha sonra Putnam 1978'de bu eğriyi geliştirmiş ve proje çaba ölçütlerini eklemiştir. Burada hiperbolik sekant fonksiyonu kullanmıştır. Parr 1980'de bir alternatif önermiştir.

Benzeşim bazlı tahminleme modellerinde aynı organizasyonda yapılan önceki projelere bakılır. Tahminlemede yapılmış projelerde benzer karmaşıklığa ve büyüklüğe bakılır. Gözlemler göstermektedir ki benzeşim bazlı modeller parametrik modellerden daha başarılı olmaktadır. Bu modellerde bir veya daha fazla uzman bulunmaktadır (Ayyıldız 2007).

Bu çalışmada KLOC ve FP tabanlı parametrik tahmin modelleri detaylı olarak incelenmiştir. Söz konusu modellerde efor kestirimi kod satır sayısı ve işlev puanı temel alınarak yapılmaktadır. Her bir parametrik efor tahmin modelinin denklemleri birbirinden farklıdır. Bu modeller, aynı metrikleri (örneğin yazılım kod satır sayısı) temel alsalar bile, farklı efor ve zamanlama (ve dolayısıyla geliştirme maliyeti) tahminlerinde bulunmaktadır. Çalışma kapsamında mevcut modellere ilave olarak gerçekleşen yazılım proje metrikleri kullanılarak yeni modeller önerilmiştir. Yazılım proje yöneticileri için kolaylık sağlamak adına literatürde mevcut ve önerilen modeller bir yazılım paketinde bir araya getirilmiştir.

3.2 Parametrik Tahmin Modelleri

Yazılım maliyet tahmin modelleri iki ana başlık altında toplanmaktadır. Bunlar parametrik tahmin modelleri ve parametrik olmayan tahmin modelleridir. Parametrik modelleri kendi aralarında iki farklı guba ayrılmaktadır. Bunlar kod satır sayısı tabanlı parametrik tahmin modelleri ve işlev puanı tabanlı parametrik tahmin modelleridir. Söz konusu modeller aşağıda sunulan bölümlerde detaylı ele alınmıştır.

3.2.1 KLOC tabanlı parametrik tahmin modelleri

Yazılım efor tahmininde kullanılan kod satır sayısı tabanlı tipik parametrik ana modeller aşağıda maddeler şeklinde sıralanabilir.

- Walston-Felix Modeli

- Bailey-Basili Modeli
- Doty Modeli
- Halstead Modeli
- Barry Boehm Modeli

Bu modeller, çeşitli kuruluşlarda tamamlanan çok sayıda yazılım projesi ve uygulamalarından elde edilmiştir (Kaur vd. 2008). Bu modellerin denklem setinde yazılım büyüklüğü olarak 1000 kod satır sayısı (KLOC) temel alınır. Modeller iş gücü tahmininde bulunur. Eforun birimi adam-aydır. KLOC tabanlı tahmin modellerini bir araya topladığımızda çizelge 3.1'deki tablo ortaya çıkmaktadır.

Çizelge 3.1 KLOC tabanlı parametrik efor tahmin modelleri

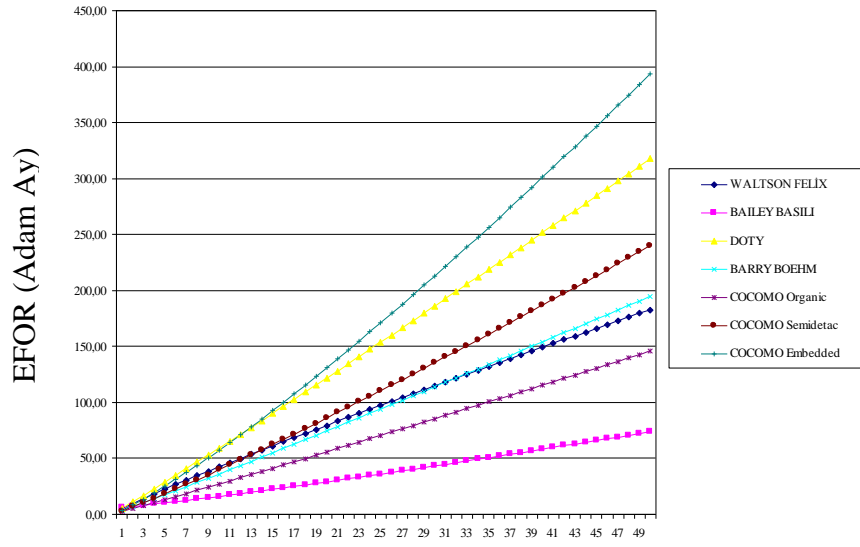
Sıra Nu	Parametrik Efor Tahmin Modelleri (Laird ve Brennan 2010)	
	Model İsmi	Denklem
1	Walston – Felix Modeli	$Efor=5.2 \times (KLOC)^{0.91}$
2	Bailey – Basili Modeli	$Efor =5.5 + 0.73 \times (KLOC)^{1.16}$
3	Halstead Modeli	$Efor= 5.2 \times (KLOC)^{1.5}$
4	Doty Modeli	$Efor=5.288 \times (KLOC)^{1.047}$
5	Barry Boehm Modeli	$Efor = 3.2 \times (KLOC)^{1.05}$
6	COCOMO Organic Modeli	$Efor= 2.4 \times KLOC^{1.05}$
7	COCOMO Semidetac Modeli	$Efor= 3.0 \times KLOC^{1.12}$
8	COCOMO Embedded Modeli	$Efor= 3.6 \times KLOC^{1.2}$

Walston ve Felix tarafından 1977 yılında IBM'de geliştirilen Waltson-Felix Modeli, efor ile kaynak kod satır sayısı arasında ilişki sağlar.

Bailey ve Basili (1981), verilen geliştirme ortamına en iyi uyum sağlayan efor tahmin denklemlerin geliştirilmesine izin veren bir meta-model olarak 1981 yılında tanımlanmıştır. Oluşan tahmin modeli, projelerin arasındaki farklar ve onların çevresel faktörleri gibi veriler toplanmasına dayalı COCOMO ile benzerdir.

M.H. Halstead (Halstead 1977), programlama yapısında derinlemesine analize gerek olmadan hata oranını öngören modeli 1977 yılında önermiştir. Bu model yazılımın kod uzunluğu ve hacim metrikleri önermek için kullanılır.

1977 yılında yayımlanan Doty Modeli kod satır sayısı için eforları tahmin etmede kullanılır. Bu model kullanıcı katılımlı, hafıza sınırlamalı ve müşteri odaklı değişiklik yeteneğine sahiplik gibi yazılım geliştirme ortamlarını çeşitli yönleriyle oluşturur.



KLOC

Şekil 3.1 Yazılım efor tahmin algoritmik modelleri kullanarak KLOC değerlerine bağlı olarak efor değişimi

Yazılım efor tahmininde kullanılan, farklı proje data setleriyle kalibre edilen birçok parametrik model formu mevcuttur. Bu modellerin 0-50 Bin Kod Satır Sayısı (Kilo Lines of Code-KLOC) arasındaki efor değerleri şekil.3.1’de gösterilmektedir. Bu grafik incelendiğinde parametrik modeller aynı yazılım büyüklüğü temel olsa bile efor tahminlerinde büyük farklılıklar dikkati çekmektedir. Bu yazılım tahmini ile ilgilenen yeni kişiler için şaşırtıcı bir sonuçtur, tabî ki sonuçların daha fazla tutarlı olması beklenir. Bu tutarsızlığın temel sebebi modelleri öneren kişilerin kısıtlı veriler kullanarak modelleri önermesidir. Data setlerinin farklı ve kısıtlı olması nedeniyle her

bir model farklı bir yazılım büyüklüğü-efor eğrisi sahip olmuştur. Bu farklılık data setlerinde bulunan projelerin karakteristiğinin farklı olmasında kaynaklanmıştır.

3.2.2 FP tabanlı parametrik tahmin modelleri

Yazılım efor tahmininde kullanılan işlev puanı tabanlı tipik parametrik ana modeller aşağıda maddeler şeklinde sıralanabilir.

- Albrecht-Gaffney Modeli
- Kemerer Modeli
- Matson-Barret-Meltichamp Modeli

Çizelge 3.2 FP tabanlı parametrik efor tahmin modelleri

Sıra Nu	Parametrik Efor Tahmin Modelleri (Laird ve Brennan 2010)	
	Model İsmi	Denklem
1	Albrecht–Gaffney Modeli	$Efor = 13,39 + 0,0545 \times FP$
2	Kemerer Modeli	$Efor = 60,62 + 7,728 \times (10^{-8}) \times FP^3$
3	Matson–Barret–Meltichamp Modeli	$Efor = 585,7 + 15,12 \times FP$

İşlev Puanı (FP) tabanlı parametrik tahmin modelleri literatür incelenerek çizelge 3.2’de bir araya getirilmiştir. Modeller denklemleriyle birlikte çizelgede sunulmuştur. Denklem setlerinde anlaşılacağı üzere yazılım büyüklüğü olarak aynı işlev puanı temel alınsa bile her bir modelin efor tahminlerinde büyük farklılıklar belirgin bir şekilde ortaya çıkmaktadır. Bu farklılıkların ortaya çıkmasındaki temel sebep modeller önerilirken kısıtlı veri kullanılmış ve verilerdeki proje metrikleri farklı karakteristik gösteren projelerden alınmıştır. Bu nedenle böyle büyük farklılıklar ortaya çıkmıştır.

3.2.3 Yazılım Projelerin Metrikleri Tahmininde Denklem Kullanımı (Peter 2010)

Yazılım projelerinde tahmin tekniklerinde regresyon denklemleri kullanılmaktadır. Bu denklemler, özellikle proje metriklerin tahmini hesaplamalarında yardımcı olur. Uygun denklemlere proje metrikleri eklenerek, efor ve zamanlama gibi büyüklükler hesaplanabilir. Bu tahmin tekniği, projelerin erken safhalarında, göstergesel proje tahminlerinin üretiminde yaygın olarak kullanılır. Bu metot, iş durum gereksinimi ve kotasyona dayalı bir tahmin üretiminde yeterince doğru değildir. Ancak, projeye ilgili kısa zaman ve az bilgiye sahip olduğumuz durumlarda, proje fikrinin mümkün olduğunun erken gösterilmesi için uygun bir metottur. Söz konusu durumda regresyon denklem tekniği ihtiyacı karşılar. Denklem tekniği, proje başlangıç tahmini üretiminde yaygın olarak kullanılan bir tahminlemedir.

Regresyon denklem seti ISBSG (International Software Benchmarking Standards Group) deposundaki verilerden üretilmiştir. Bu denklemleri aşağıdaki proje metrikleri hesaplanmasında kullanabiliriz. Proje metrikleri aşağıda sunulmaktadır:

- Proje Teslim Oranı (Project Delivery Rates) (Verimlilik (Productivity) Saat Fonksiyon Sayısı (hours per function point)
- Efor (adam saat- geliştirme takım için)
- Süre (geçen ay)
- Bir bütün olarak projenin (fonksiyon sayısı geçen ay) teslim hızı
- Kişi başına teslim hızı (gelişim takım elemanı geçen süre teslim edilen fonksiyon sayısı)

Aşağıdakiler bağımsız değişkenlerden sağlanan iki grup denklem:

- Boyuttan yararlanan denklemler (düzeltilmemiş fonksiyon sayısı) ve maksimum takım boyutu
- Sadece boyuttan yararlanan denklemler

Bu gruplar arasında, aşağıdaki maddeler tarafından sağlanan denklemler:

- Platform (anabilgisayar, orta, PC ve multiplatform)
- Dil tipi (3GL, 4 GL ve uygulama jeneratör)

- Gelişme Tipi (geliştirme ve yeni gelişme)
- Platform, dil tipi ve gelişme tipi kombinasyonu

Bu tahmin yöntemleri, proje büyüklüğünün hesaplanmasına veya bir tahminde bulunmasında yeterli değildir. Eğer takım boyutu bilinebiliyorsa veya tahmin edilebilirse bu iyidir. Ancak proje boyutunu bilmeden, sadece takım boyut tabanlı tahmin yapma girişimi hiçbir anlam ifade etmez. Bu bölümde, dil seviye temelli bir tahmini üretecek temel denklemlerin nasıl kullanıldığı gösterilmektedir. ISBSG denklemlerinin kullanımı kolaydır. Tablodan uygun denklemleri seçtikten sonra, tahmin üretmek için maksimum takım boyutu ve projenin hesaplanan fonksiyonel boyutları yerleştirilir. Örneğin, multi platform ortamda geliştirilen bir projenin tahminlerinin üretilmesini farzedelim. 260 işlev puanı büyüklüğünde ve maksimum takım büyüklüğü 4 olan bir proje olsun. Çizelge 3.3'deki söz konusu projenin denklem seti kullanılarak yazılım metrikleri elde edilmiştir.

Çizelge 3.3 Örnek yazılım proje metriklerinin tahmini

Proje Teslim Oranı (PDR)	$PDR = 38,97 \times \text{Boyut}^{-0,566} \times \text{Takım Boyutu}^{0,951}$	$= 38,97 \times 260^{-0,566} \times 4^{0,951}$	= 6,3 işlev puanı başına saat
Proje Eforu (PWE)	$PWE = 38,97 \times \text{Boyut}^{-0,434} \times \text{Takım Boyutu}^{0,951}$	$= 38,97 \times 260^{-0,434} \times 4^{0,951}$	= 1,627 saat
Proje Teslim Hızı Bütün Proje (SD)	$SD = 0,44 \times \text{Boyut}^{-0,852} \times \text{Takım Boyutu}^{-0,228}$	$= 38,97 \times 260^{-0,852} \times 4^{-0,228}$	= 37 ay başına işlev puanı
Proje Teslim Hızı - Yazılım Geliştirici Başına (SD)	$SD = 0,44 \times \text{Boyut}^{-0,852} \times \text{Takım Boyutu}^{-1,228}$	$= 38,97 \times 260^{-0,852} \times 4^{-1,228}$	= 9,2 geliştirici ay başına işlev puanı

Denklem kullanarak efor tahmini örneği çizelge 3.4'deki verilerden yararlanarak hesaplanmıştır. Örnekte, platformun farklı kombinasyonları için ISBSG verisinde türetilen efor denklemleri kullanılarak 500 işlev puanı için bir proje tahminindeki ballpark efor tahmini gösterilmiştir.

Çizelge 3.4 Yazılım büyüklüğü ile efor tahmin örneği

Platform ve Dil	Efor Denklemi	500 FP için Tahmini Efor
MF & 3GL	$Efor=51,34 \times FP^{0,730}$	4,794 adam saat
MF & 4GL	$Efor=18,39 \times FP^{0,838}$	3,360 adam saat
MR & 3GL	$Efor=51,34 \times FP^{0,742}$	4,343 adam saat
MR & 4GL	$Efor=51,34 \times FP^{0,830}$	5,053 adam saat
PC & 3GL	$Efor=51,34 \times FP^{0,889}$	3,469 adam saat
PC & 4GL	$Efor=51,34 \times FP^{0,710}$	2,561 adam saat
Multi & 3GL	$Efor=51,34 \times FP^{0,831}$	4,191 adam saat
Multi & 4GL	$Efor=51,34 \times FP^{0,867}$	3,470 adam saat

Proje danışmanları, proje yöneticileri ve yazılım uzmanları için projenin başlangıcından sonuna kadar ki ömür devrinin tüm safhalarındaki eforları ISBSG denklemleri yardımıyla önceden tahmin edilir.

Yazılım boyutu ve maksimum takım boyutu parametrelerinden yararlanılarak oluşturulan denklemlerin kullanıldığı sonuçlar devamında sunulmuştur.

Çizelge 3.5 Yazılım büyüklüğü ve takım boyutu ile efor tahmin örneği

Platform ve Dil	Yazılım Denklemi ve Maksimum Takım Boyutu Efor Denklemi	500 FP için Tahmini Efor (Maksimum Takım Boyut 5 için)
MR & 3GL	$Efor=42,94 \times FP^{0,395} \times MxTeam^{0,994}$	2,475 adam saat
MR & 4GL	$Efor=56,86 \times FP^{0,336} \times MxTeam^{0,967}$	2,175 adam saat
PC & 3GL	$Efor=28,66 \times FP^{0,501} \times MxTeam^{0,780}$	2,263 adam saat
Multi & 3GL	$Efor=36,44 \times FP^{0,509} \times MxTeam^{0,833}$	3,293 adam saat
Multi & 4GL	$Efor=9,35 \times FP^{0,718} \times MxTeam^{0,801}$	2,941 adam saat

3.2.3.1 Yazılım büyüklüğü ve maksimum takım boyutundan tahmin edilen proje çalışma eforu

Yazılım büyüklüğü ve maksimum takım boyutundan tahmin edilen proje çalışma eforu denklem 3.2 kullanılarak hesaplanmaktadır.

- Bağımlı (Y)= PWE (Normalize Proje Çalışma Eforu, Saat)
 - Bağımsız (X1)= Boyut (İşlev Puanı Yazılım Boyutu)
 - Bağımsız (X2)= Maksimum Takım Boyutu
 - Denklem: $PWE=C \times \text{Boyut}^{E1} \times \text{Maksimum Takım Boyutu}^{E2}$ (3.2)
- (C, E1 ve E2 parametreleri çizelge 3.6'dan alınacaktır.)

Çizelge 3.6 Yazılım büyüklüğü ve maksimum takım boyutundan efor tahmin parametreleri

Sınıf	C	E ₁	E ₂	N	R2	MRE
Tüm	57,39	0,442	0,710	584	0,57	0,37
Yeni Versiyon	79,12	0,384	0,692	381	0,53	0,37
Yeni Geliştirilen	37,48	0,504	0,759	203	0,61	0,37
MR	60,76	0,336	0,960	61	0,81	0,23
PC	23,67	0,570	0,678	83	0,53	0,42
Multi	34,49	0,490	0,876	125	0,67	0,30
3GL	51,74	0,474	0,693	367	0,55	0,38
4GL	32,90	0,532	0,692	141	0,73	0,30
Yeni & MR	35,09	0,403	1,080	16	0,86	0,24
Yeni & PC	11,06	0,723	0,634	50	0,54	0,42
Yeni & Multi	37,41	0,537	0,736	47	0,68	0,30
Yeni Ver. & MR	115,90	0,241	0,872	45	0,76	0,19
Yeni Ver. & Multi	38,97	0,434	0,951	78	0,64	0,31
Yeni & 3GL	39,40	0,511	0,762	127	0,63	0,38
Yeni & 4GL	6,53	0,833	0,585	43	0,70	0,37
Yeni Ver. & 3GL	70,85	0,413	0,673	240	0,48	0,39
Yeni Ver. & 4GL	64,10	0,395	0,728	98	0,78	0,27
MR & 3GL	42,94	0,395	0,994	27	0,67	0,28
MR & 4GL	56,86	0,336	0,967	30	0,87	0,18
PC & 3GL	28,66	0,501	0,780	65	0,56	0,42
Multi & 3GL	36,44	0,509	0,833	91	0,68	0,30
Multi & 4GL	9,35	0,718	0,801	32	0,82	0,23
Yeni Ver. & MR & 3GL	81,76	0,353	0,353	19	0,63	0,24
Yeni & PC & 3GL	14,24	0,656	0,724	37	0,57	0,42
Yeni & Multi & 3GL	72,34	0,470	0,666	30	0,65	0,43
Yeni & Multi & 4GL	6,72	0,772	0,839	16	0,82	0,16
Yeni Ver. & Multi & 3GL	25,63	0,538	0,909	61	0,66	0,30
Yeni Ver. & Multi & 4GL	13,98	0,628	0,829	16	0,81	0,21

3.2.3.2 Yazılım büyüklüğü ve maksimum takım boyutundan tahmin edilen proje teslim oranı

Yazılım büyüklüğü ve maksimum takım boyutundan tahmin edilen proje teslim oranı denklem 3.3 kullanılarak hesaplanmaktadır.

- Bağımlı (Y)= PDR (Proje Teslim Oranı, Saate Fonksiyon Sayısı)
 - Bağımsız (X1)= Boyut (İşlev Puanı Yazılım Boyutu)
 - Bağımsız (X2)= Maksimum Takım Boyutu
 - Denklem: $PDR=C \times \text{Yazılım Boyutu}^{E1} \times \text{Maksimum Takım Boyutu}^{E2}$ (3.3)
- (C, E1 ve E2 parametreleri çizelge 3.7'den alınacaktır.)

Çizelge 3.7 Yazılım büyüklüğü ve maksimum takım boyutundan proje teslim oranı parametreleri

Sınıf	C	E ₁	E ₂	N	R2	MRE
Tüm	57,39	-0,558	0,710	584	0,33	0,37
Yeni Versiyon	79,12	-0,616	0,692	381	0,33	0,37
Yeni Geliştirilen	37,48	-0,496	0,759	203	0,32	0,37
MR	60,76	-0,664	0,960	61	0,65	0,23
Multi	34,49	-0,510	0,875	125	0,46	0,30
3GL	51,74	-0,526	0,693	367	0,27	0,38
4GL	32,90	-0,468	0,692	141	0,42	0,30
Yeni & MR	35,09	-0,597	1,080	16	0,57	0,24
Yeni & Multi	37,41	-0,463	0,736	47	0,36	0,30
Yeni Ver. & MR	115,90	-0,759	0,872	45	0,73	0,19
Yeni Ver. & Multi	38,97	-0,566	0,951	78	0,50	0,31
Yeni & 3GL	39,40	-0,489	0,762	127	0,33	0,38
Yeni Ver & 4GL	64,10	-0,605	0,728	98	0,60	0,27
MR & 3GL	42,94	-0,605	0,994	27	0,61	0,28
MR & 4GL	56,86	-0,664	0,967	30	0,66	0,18
Multi & 3GL	36,44	-0,491	0,833	91	0,43	0,30
Multi & 4GL	9,35	-0,282	0,801	32	0,39	0,23
Yeni Ver. & MR & 3GL	81,76	-0,647	0,785	19	0,68	0,24
Yeni Ver. & MR & 4GL	162,70	-0,865	0,963	25	0,76	0,19
Yeni & Multi & 3GL	72,34	-0,530	0,666	30	0,38	0,43
Yeni & Multi & 4GL	6,72	-0,228	0,839	16	0,33	0,16
Yeni Ver.& Multi & 3GL	25,63	-0,462	0,909	61	0,46	0,30
Yeni Ver.& Multi & 4GL	13,98	-0,372	0,829	16	0,42	0,21

3.2.3.3 Yazılım büyüklüğü ve maksimum takım boyutundan adam başına proje teslim hızı

Yazılım büyüklüğü ve maksimum takım boyutundan adam başına proje teslim hızı denklem 3.4 kullanılarak hesaplanmaktadır.

- Bağımlı (Y)= Adam Başına Proje Teslim Hızı (Geçen Ay Karşılık İşlem Puanı)
 - Bağımsız (X1)= Boyut (Fonksiyon Sayısı Yazılım Boyutu)
 - Bağımsız (X2)= Maksimum Takım Boyutu
 - Denklem: Adam Başına Hız = $C \times \text{Boyut}^{E1} \times \text{Maksimum Takım Boyutu}^{E2}$ (3.4)
- (C, E1 ve E2 parametreleri çizelge 3.8'den alınacaktır.)

Çizelge 3.8 Yazılım büyüklüğü ve maksimum takım boyutundan adam başına hız parametreleri

Sınıf	C	E ₁	E ₂	N	R2	MRE
Tüm	0,778	0,696	-1,011	561	0,61	0,36
Yeni Versiyon	0,643	0,729	-1,005	369	0,62	0,36
Yeni & PC	3,373	0,465	-1,007	46	0,58	0,43
Yeni Ver. & Multi	0,436	0,851	-1,228	78	0,70	0,28
Yeni & 4GL	3,677	0,393	-0,872	42	0,44	0,39
Yeni Ver. & 3GL	0,667	0,724	-1,022	231	0,59	0,37
Multi & 3GL	0,479	0,793	-1,116	89	0,64	0,29
Yeni & PC & 3GL	4,464	0,431	-1,059	34	0,49	0,43

3.2.3.4 Yazılım boyutu ve maksimum takım boyutundan tüm projenin teslim hızı

Yazılım boyutu ve maksimum takım boyutundan tüm projenin teslim hızı denklem 3.5 kullanılarak hesaplanmaktadır.

- Bağımlı (Y)= Teslim Hızı (Geçen Süre-İnaktif Süre, Ay)
 - Bağımsız (X1)= Boyut (İşlev Puanı Yazılım Boyutu)
 - Bağımsız (X2)= Maksimum Takım Boyutu
 - Denklem: Proje Hızı = $C \times \text{Boyut}^{E1} \times \text{Maksimum Takım Boyutu}^{E2}$ (3.5)
- (C, E1 ve E2 parametreleri çizelge 3.9'dan alınacaktır.)

Çizelge 3.9 Yazılım büyüklüğü ve takım boyutundan proje hızı parametreleri

Sınıf	C	E ₁	E ₂	N	R2	MRE
Yeni Versiyon & Multi	0,44	0,852	-0,228	78	0,59	0,28

3.2.3.5 Sadece yazılım boyutundan projenin efor tahmini

Yazılım boyutundan projenin efor tahmini denklem 3.6 kullanılarak hesaplanmaktadır.

- Bağımlı (Y)= Teslim Hızı (Gelişmiş takım için normalize Proje çalışma eforu, Saat)
- Bağımsız (X)= Boyut (İşlev Puanı Yazılım Boyutu)
- Denklem: $PWE = C \times Boyut^{E_1}$ (3.6)

(C ve E₁ parametreleri çizelge 3.10'dan alınacaktır.)

Çizelge 3.10 Yazılım büyüklüğünden efor tahmini parametreleri

Sınıf	C	E ₁	N	R2	MRE
Tümü	33,37	0,770	1681	0,46	0,55
Yeni Versiyon	31,16	0,793	1147	0,43	0,55
Yeni Geliştirilen	23,25	0,814	534	0,45	0,55
MF	44,03	0,749	452	0,38	0,56
MR	35,43	0,783	128	0,60	0,43
PC	17,35	0,844	204	0,50	0,51
Multi	2,86	0,830	480	0,54	0,52
3GL	39,56	0,754	1105	0,42	0,57
4GL	20,06	0,832	359	0,58	0,46
Yeni & MF	34,49	0,809	86	0,33	0,58
Yeni & MR	19,08	0,883	49	0,59	0,42
Yeni & PC	13,83	0,884	126	0,50	0,53
Yeni & Multi	22,48	0,809	184	0,50	0,45
Yeni Ver. & MF	50,10	0,718	366	0,35	0,55
Yeni Ver. & MR	45,90	0,734	79	0,52	0,44
Yeni Ver. & PC	23,10	0,789	78	0,47	0,54
Yeni Ver. & Multi	16,35	0,912	296	0,56	0,52
Yeni & 3GL	29,16	0,790	351	0,44	0,55
Yeni & 4GL	5,28	1,032	112	0,60	0,49

3.2.3.6 Sadece yazılım boyutundan proje süresi tahmini

Yazılım boyutundan proje süresi tahmini denklem 3.7 kullanılarak hesaplanmaktadır.

- Bağımlı (Y)= Süre (Geçen Süre-İnaktif Süre, Ay)
 - Bağımsız (X)= Boyut (İşlev Puanı Yazılım Boyutu)
 - Denklem: $Süre = C \times Boyut^{E1}$ (3.7)
- (C ve E1 parametreleri çizelge 3.11'den alınacaktır.)

Çizelge 3.11 Yazılım büyüklüğünden proje süresi tahmini parametreleri

Sınıf	C	E ₁	N	R ₂	MRE
Yeni Geliştirilen	0,543	0,408	494	0,30	0,41
PC	0,507	0,418	191	0,33	0,39
Multi	0,589	0,394	394	0,28	0,44
4GL	0,507	0,429	304	0,36	0,37
Yeni & PC	0,297	0,505	115	0,42	0,42
Yeni & Multi	0,423	0,440	179	0,33	0,41
Yeni & 3GL	0,645	0,378	327	0,27	0,42
Yeni & 4GL	0,239	0,538	108	0,40	0,43
Yeni Ver. & 4GL	0,540	0,428	196	0,34	0,36
PC & 3GL	0,468	0,436	132	0,36	0,40
Multi & 4GL	0,201	0,599	98	0,46	0,40
Yeni & PC & 3GL	0,284	0,523	78	0,43	0,42
Yeni & PC & 4GL	0,324	0,459	29	0,41	0,32
Yeni & Multi & 3GL	0,558	0,392	128	0,33	0,37
Yeni & Multi & 4GL	0,107	0,679	48	0,38	0,52
Yeni Ver. & Multi & 4GL	0,174	0,656	50	0,63	0,22

3.2.3.7 Sadece yazılım boyutundan bütün takımının proje teslim hızı

Yazılım boyutundan bütün takımının proje teslim hızı denklem 3.8 kullanılarak hesaplanmaktadır.

- Bağımlı (Y)= Süre (İşlev Puanı Geçen Ay)
- Bağımsız (X)= Boyut (İşlev Puanı Yazılım Boyutu)
- Denklem: Proje Hızı = $C \times \text{Boyut}^{E_1}$ (3.8)

(C ve E1 parametreleri çizelge 3.12'den alınacaktır.)

Çizelge 3.12 Yazılım büyüklüğünden tüm takımın proje teslim hızı parametreleri

Sınıf	C	E ₁	N	R2	MRE
Tümü	1,183	0,661	1311	0,55	0,42
Yeni Geliştirilen	1,842	0,592	494	0,48	0,40
PC	1,972	0,582	191	0,49	0,42
Multi	1,698	0,606	394	0,48	0,42
4GL	1,974	0,571	304	0,50	0,40
Yeni & PC	3,370	0,495	115	0,41	0,39
Yeni & Multi	2,367	0,560	179	0,44	0,40
Yeni Ver. & Multi	1,536	0,615	215	0,46	0,44
Yeni & 3GL	1,550	0,622	327	0,50	0,42
Yeni & 4GL	4,181	0,462	108	0,33	0,43
Yeni Ver. & 4GL	1,853	0,572	196	0,48	0,36
MR & 3GL	0,464	0,846	70	0,74	0,34
PC & 3GL	2,138	0,564	132	0,49	0,43
Multi & 4GL	4,982	0,401	98	0,27	0,42
Yeni & PC & 3GL	3,520	0,477	78	0,39	0,42
Yeni & PC & 4GL	3,082	0,541	29	0,49	0,32
Yeni & Multi & 3GL	1,792	0,608	128	0,55	0,34
Yeni Ver. & Multi & 4GL	5,744	0,343	50	0,31	0,28

3.3 Parametrik Olmayan Tahmin Modelleri

Yazılım projeleri tahmin modellerinin iki ana başlık altında gruplandırmak mümkündür. Bunlar parametrik modeller ve parametrik olmayan modeller şeklinde sınıflandırılabilir.

Parametrik olmayan modeller kategorize edildiğinde; uzman görüşü modeli, karar ağacı yöntemi, bulanık mantık tahmin şemaları, yapay sinir ağ tabanlı metotlar, yukarıdan aşağı metodu şeklinde sıralanabilir.

Çokça kullanılan bu modeller değerlendirildiğinde hiç biri tüm projeler için uygun olmamaktadır. Bu kapsamda söz konusu yöntemlerin kombinasyonunu kullanmak daha iyi sonuç verebilmektedir. Uzman görüşü yöntemi ile benzeşim temelli yöntemleri birlikte kullanmak daha iyi sonuçlar vermektedir.

3.4 Karşılaştırma

Yaygın olarak kullanılan parametrik ve parametrik olmayan modellerin birbirlerine göre üstün ve zayıf özellikleri vardır. Bu durum çizelge 3.13’de sunulmuştur.

Çizelge 3.13 Modellerin karşılaştırma matrisi

	Yöntem	Güçlü Tarafları	Zayıf Tarafları
Parametrik Olmayan Metodlar	Uzman Görüşü	Konusunda çok iyi bir uzman görüşü alınırsa iyi tahminlerde bulunulur.	Uzmana göre değişkenlik arz eder.
		Çabuk sonuç verir.	Tutarsızlıklar olabilir. Eksik veriden dolayı yeniden tahminleme yapılmalıdır.
	Benzeşim	Gerçek projeler ve geçmiş tecrübe temelli gerçekleşir.	Benzer proje eksikliği yaşanabilir. Geçmiş veriler doğru olmayabilir.
		Yukarıdan Aşağı	Sistem seviyesinde odaklanma vardır.
	Aşağıdan yukarıya göre daha kolay ve hızlıdır.		Tutarlılık diğer yöntemlere göre azdır.
	Çok az detay gerektirir.		
	Aşağıdan Yukarı	Detaylı analiz gerektirir.	Maliyet faktörü önem kazanır.
		Proje izlemede diğerlerine göre avantaj sağlar.	Daha fazla çaba gerektirir. Başlangıçta tahmin yapmak zordur.
Parametrik Metodlar	Parametrik	Tekrarlanabilir sonuçlar verir. Objektiftir.	Geçmiş proje bilgilerinin kullanılması güncel duruma uygun olmayabilir.
		Tahminleme yönteminin anlaşılması kolaydır.	Genellemeler yapılması zordur.

4. YÖNTEM

4.1 Araştırmanın Konusu ve Amacı

Yazılım proje yöneticileri için doğru yazılım efor tahmini yapmak çok önemli bir konudur. Bu kapsamda iyi efor tahmini yardımıyla yazılım projelerinde etkili ve verimli zamanlama ve maliyet planı yapılabilir.

Gerçekleştirilen bu çalışma neticesinde; literatürde yer alan parametrik tahmin modelleri detaylı bir şekilde incelenmiş ve her bir parametrik tahmin modelinin yazılım metrik tahmin denklemleri bir araya toplanmıştır. Literatürde, genel olarak kabul görmüş bu modeller, aynı metrikleri (örneğin yazılım kod satır sayısı) temel alsalar bile, farklı efor ve proje süresi (ve dolayısıyla geliştirme maliyeti) tahminlerinde bulunmaktadır. Bu çalışma ile yazılım projelerinin gerçek kod satır sayısı ve işlev puanı değerleri kullanarak en uygun tahminlerde bulunan yeni modeller önerilmiştir. Literatürde mevcut modellerle yeni modellerin hata ölçütleriyle kıyaslaması yapılmıştır. Bu amaçla, geçmişte bitirilmiş olan yazılım projelerinin gerçek metrikleri ile oluşturulan modellerin performans karşılaştırması yapılmıştır. Tez kapsamında geliştirilen yazılım paketi ile farklı modeller uyarınca tahmin edilen yazılım maliyetlerinin kolaylıkla kıyaslaması yapılabilecektir. Hazırlanan yazılım paketine, yazılım projeleri için daha uygun tahminlerde bulunan yeni parametrik modeller eklenmiştir. Elde edilen sonuçlarla, söz konusu tahmin modellerinin geçerliliği ve etkinliği üzerinde yorumlar yapılmıştır.

4.2 Araştırmanın Önemi

Son 30 yılda birçok yazılım maliyet modeli literatüre girmiştir. Bu modellerin büyük bir kısmını da parametrik efor tahmin modelleri oluşturmaktadır. Bu modellere özellikle yazılım projelerine başlamadan önce bir yol haritası çizilmesi adına ihtiyaç duyulmaktadır. ABD’de bulunan ve bir araştırma şirketi olan The Standish Group, 23000 yazılım uygulamalarını inceleyip bir rapor hazırlamıştır. Bu kapsamda yazılım projelerinin sadece %26’sı başarılı bir şekilde tamamlanmıştır. Bütçesini veya planlı

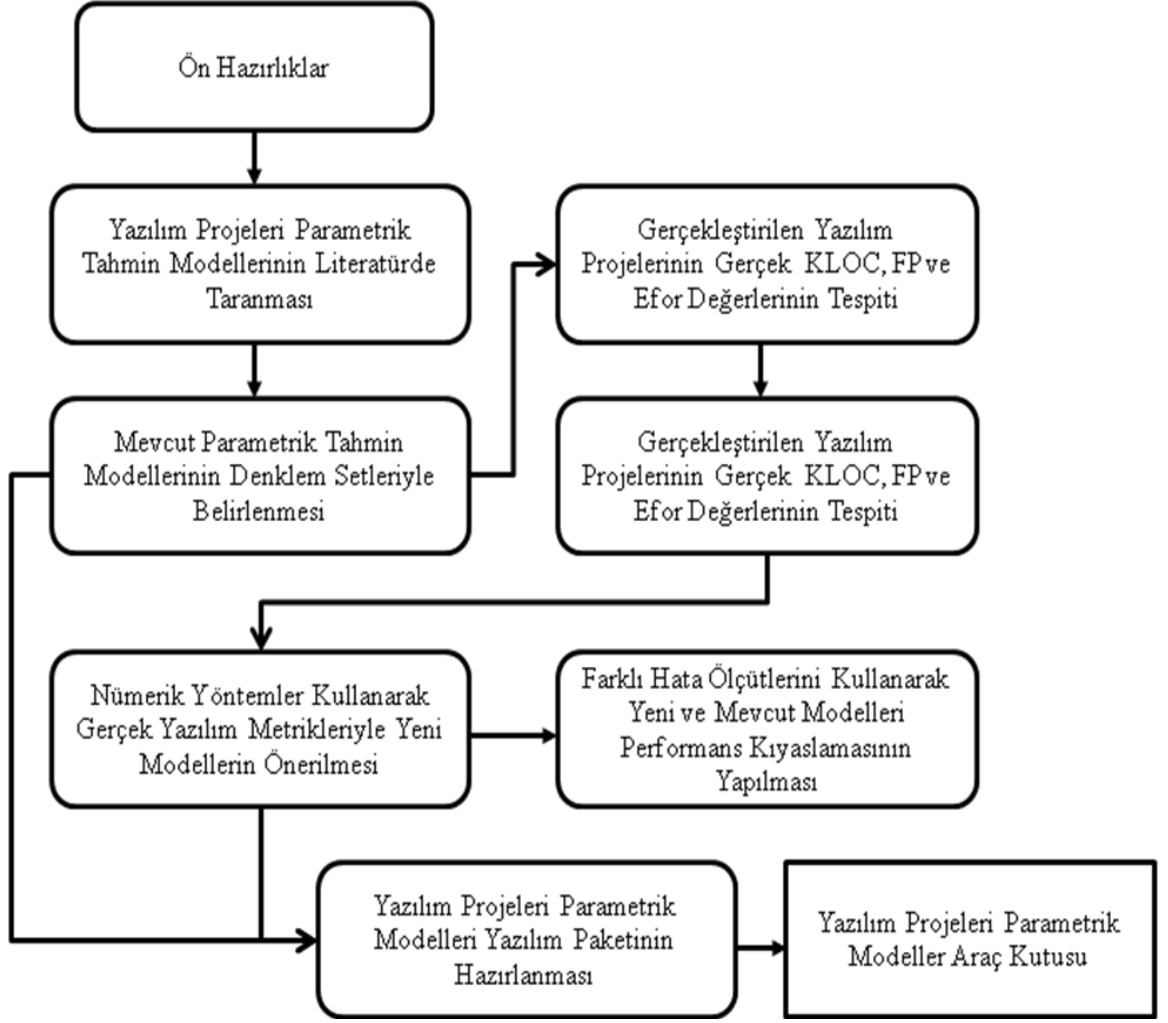
zamanını aşan (challenged) proje oranı %46'dır. Ayrıca iptal edilen proje oranı % 28'dir (Jones 1998).

Bu veriler ışığında etkili bir yazılım maliyet modelin hazırlanması kritik öneme sahiptir. Tez çalışması kapsamında, özellikle yazılım proje yöneticilerinin kullanılması maksadıyla yeni modeller önerilmiş, literatürdeki mevcut modellerle bu önerilen modeller bir araç kutusunda toplanmıştır. Buradaki amaç, yazılım proje yöneticileri için tüm modellerin tahminlerini rahatlıkla karşılaştıracağı yazılım paketi imkânına sahip olmalarıdır. Böylelikle araç kutusu çıktıları ile tüm modellerin kıyaslaması rahatlıkla yapılabilmektedir.

4.3 Araştırmanın Yöntemi

Tez araştırmasına detaylı bir literatür taraması ile başlanmıştır. İlgili kitap ve yayınlardan tespit edilen parametrik tahmin modelleri denklem setleriyle bir araya toplanmıştır. Diğer taraftan tamamlanan gerçek yazılım proje verileri araştırılmıştır. Yazılım metrikleri olarak tamamlanan her bir yazılım projesinin gerçek kod satır sayısı, işlev puanı ve eforları listelenmiştir. Bu gerçek veri setleri kullanarak bir takım nümerik yöntemlerle mevcut modellerde bulunan katsayıların en iyi tahim yapacak şekilde seçimi gerçekleştirilmiştir. Bir sonraki bölümde kat sayı seçiminde kullanılan yöntemlerden detaylı bir şekilde bahsedilmiştir. Yeni önerilen modellerle mevcut modellerin tahmin ettiği efor ve proje süresi büyüklükleri hata ölçütleri kullanarak hangi modellerin en uygun tahminde bulunduğu tespit edilmeye çalışılmıştır. Önerilen yeni modellerin daha iyi tahminlerde bulunduğu hata ölçütleri kullanılarak doğrulanmıştır.

Sonraki aşamada yazılım projeleri parametrik tahmin modelleri için yazılım paketi geliştirilmiştir. Bu yazılım paketi kapsamında, literatürdeki tüm modeller ve önerilen yeni modelleri bulunduğu araç kutusu tasarlanmıştır. Yazılım projelerine başlamadan önce is gücü ve proje süresi gibi kritik proje yönetim değerlerini kolaylıkla hesaplanması amaçlanmıştır. Tez çalışması sürecinde planlanan ve takip edilen aşamalar şekil 4.1'de yer alan akış şemasında sunulmuştur.



Şekil 4.1 Tez çalışması aşamaları akış şeması

4.4 Kod Satır Sayısı Tahmin Modelleri Kıyaslanması ve Yeni Modellerin Önerilmesi

Bu bölümde tez çalışması kapsamında kod satır sayısı tabanlı mevcut parametrik modellerinin tahminleri ile gerçek yazılım proje veri seti karşılaştırılmıştır. Literatürde mevcut her bir modelin denklem setleri kullanılarak efor tahminlerinde bulunulmuştur. Bu hesaplanan değerlerle gerçek yazılım projeleri efor değerleri kıyaslanmıştır.

Bu kıyaslamayı gerçekleştirebilmek adına hata ölçütleri kullanılmıştır. Literatürde mevcut her bir modelin tahmin performansları kıyaslanmıştır.

4.4.1 Mevcut KLOC tabanlı tahmin modellerinin kıyaslanması

Literatürde mevcut dokuz ayrı KLOC tabanlı efor tahmin modeli çizelge 4.1’de sunulmuştur. Bu modellerle literatürde yerini almış gelen kabul görmüş kod satır sayılı parametrik tahmin modelleridir.

Çizelge 4.1’de sunulan denklem setlerin yapısı üç ayrı değişken ve bir sabit değerden oluşmaktadır. Sabit değer KLOC ile gösterilen kod satır sayısıdır. Değişken olarak ise KLOC sabitinin bir çarpanı, yine KLOC sabitinin üstel değişkeni mevcuttur. Üçüncü değişken ise bu ifadeyle toplanan değişkendir.

Çizelge 4.1 KLOC Tabanlı Parametrik efor tahmin modelleri

Sıra Nu	Parametrik Efor Tahmin Modelleri (Laird ve Brennan 2010)	
	Model İsmi	Denklem
1	Walston – Felix Modeli	$Efor=5.2 \times (KLOC)^{0.91}$
2	Bailey – Basili Modeli	$Efor = 5.5 + 0.73 \times (KLOC)^{1.16}$
3	Halstead Modeli	$Efor= 5.2 \times (KLOC)^{1.5}$
4	Doty Modeli	$Efor=5.288 \times (KLOC)^{1.047}$
5	Barry Boehm Modeli	$Efor = 3.2 \times (KLOC)^{1.05}$
6	COCOMO Organic Modeli	$Efor= 2.4 \times KLOC^{1.05}$
7	COCOMO Semidetac Modeli	$Efor= 3.0 \times KLOC^{1.12}$
8	COCOMO Embedded Modeli	$Efor= 3.6 \times KLOC^{1.2}$

Çizelge 4.2’de 384 farklı ISBSG veri tabanında bulunan yazılım projelerinin verisi bulunmaktadır. Her bir yazılım projesinin uygulamaya dayalı gerçek eforları veri setinde mevcuttur. Yazılım projelerinin büyüklüğü kod satır sayısı olarak alınmıştır. Gerçek eforun birimi ise adam-aydır.

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009)

Proje Numarası	LOC (Kod Satır Sayısı)	Gerçek Efor (Adam-Ay)
1	96	0,028
2	100	1,335
3	147	0,023
4	150	0,398
5	150	0,028
6	179	0,455
7	215	182,455
8	260	0,085
9	309	0,051
10	352	0,881
11	400	0,352
12	400	0,665
13	410	0,824
14	458	1,977
15	520	2,176
16	559	1,125
17	600	1,381
18	659	2,761
19	693	0,341
20	770	0,449
21	790	1,778
22	800	47,080
23	800	2,727
24	820	1,523
25	973	1,210
26	1000	1,068
27	1039	2,159
28	1110	4,602

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

29	1188	1,091
30	1201	1,477
31	1240	0,813
32	1250	2,614
33	1255	11,909
34	1287	11,114
35	1296	2,165
36	1379	1,705
37	1400	2,841
38	1421	5,420
39	1425	1,773
40	1440	3,784
41	1453	67,097
42	1516	3,330
43	1562	13,477
44	1600	1,261
45	1618	5,813
46	1632	14,955
47	1686	11,523
48	1728	3,699
49	1836	2,682
50	1880	5,335
51	1946	16,222
52	1958	2,688
53	2000	3,409
54	2000	5,051
55	2160	27,364
56	2200	7,460
57	2261	2,818
58	2325	12,330

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

59	2354	2,602
60	2480	6,136
61	2494	1,830
62	2550	1,165
63	2700	6,784
64	2725	1,364
65	2755	7,494
66	2761	3,892
67	2922	8,767
68	2926	2,693
69	2967	2,273
70	3000	2,506
71	3000	6,716
72	3100	2,670
73	3146	1,642
74	3189	9,659
75	3200	5,852
76	3243	1,699
77	3250	3,943
78	3265	2,386
79	3300	54,648
80	3386	3,392
81	3400	3,898
82	3400	2,489
83	3400	6,358
84	3441	5,841
85	3449	3,619
86	3500	2,420
87	3662	2,159
88	3722	7,636

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

89	3821	2,705
90	4000	9,341
91	4045	9,153
92	4137	26,239
93	4200	18,784
94	4200	3,733
95	4332	3,784
96	4431	1,994
97	4520	4,284
98	4630	6,909
99	4720	11,051
100	5000	2,307
101	5049	24,648
102	5182	20,222
103	5190	39,960
104	5500	2,506
105	5835	12,261
106	5890	5,057
107	5951	6,972
108	6000	6,415
109	6000	3,619
110	6000	6,051
111	6099	15,790
112	6267	7,091
113	6328	8,733
114	6381	7,722
115	6400	5,511
116	6500	4,920
117	6773	5,420
118	6800	14,955

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

119	6800	27,989
120	6866	10,818
121	6982	31,159
122	7000	4,977
123	7000	6,688
124	7013	13,517
125	7063	4,034
126	7083	1,210
127	7685	20,409
128	7800	27,131
129	7917	1,250
130	8000	13,608
131	8100	12,307
132	8191	6,631
133	8192	5,466
134	8200	1,909
135	8290	14,517
136	8434	21,489
137	8436	28,977
138	8695	7,034
139	8756	11,761
140	8796	3,972
141	9000	23,182
142	9000	16,051
143	9000	7,920
144	9129	32,597
145	9144	6,108
146	9200	9,148
147	9444	7,386
148	10073	10,881

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

149	10100	6,989
150	10400	18,125
151	10704	42,068
152	10760	156,955
153	10799	19,767
154	10980	17,909
155	11057	30,551
156	11100	6,165
157	11300	8,523
158	11417	43,750
159	11474	19,318
160	11491	18,977
161	11500	13,540
162	11587	18,375
163	11653	1,943
164	11661	2,318
165	12281	17,636
166	12428	11,710
167	12596	1,568
168	12667	6,295
169	12800	61,199
170	13000	6,364
171	13000	2,540
172	13155	24,784
173	13450	15,676
174	13450	13,864
175	13591	4,131
176	13900	10,716
177	13990	3,813
178	14460	8,955

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

179	14500	17,000
180	14512	17,966
181	14518	49,148
182	14570	60,864
183	14600	23,784
184	15000	29,432
185	15054	16,750
186	15181	18,398
187	16000	34,665
188	16000	17,813
189	16000	3,739
190	16012	24,619
191	16200	39,114
192	16200	20,966
193	17000	7,642
194	17000	13,295
195	17000	12,273
196	17111	27,273
197	17400	34,227
198	17624	2,386
199	18000	3,591
200	18000	16,023
201	18236	16,688
202	18533	14,250
203	18700	29,489
204	18800	10,801
205	18980	1516,739
206	19051	31,455
207	19085	9,375
208	19100	23,636

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

209	19332	19,114
210	19477	3,847
211	19500	12,216
212	20050	39,443
213	20087	40,324
214	20443	89,795
215	20543	31,875
216	20569	2,528
217	20571	11,545
218	21000	12,801
219	21390	473,455
220	21880	14,273
221	22000	4,398
222	22788	75,920
223	22882	7,648
224	23300	15,239
225	23332	77,148
226	23951	21,182
227	23968	4,233
228	24000	17,830
229	24000	18,091
230	24600	3,261
231	24700	22,415
232	25072	15,511
233	25632	7,091
234	25856	11,642
235	26000	11,375
236	26153	2,511
237	26200	18,216
238	26800	23,068

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

239	26950	32,216
240	28000	17,830
241	28307	8,284
242	28318	51,568
243	28430	37,119
244	29000	21,716
245	29000	22,193
246	29908	28,591
247	30000	10,983
248	30000	19,091
249	30044	104,545
250	30722	32,239
251	30800	13,568
252	31480	17,131
253	31800	20,932
254	32000	5,824
255	32379	11,670
256	32548	102,580
257	33500	13,886
258	34000	9,409
259	35000	6,375
260	35080	14,028
261	35089	51,136
262	35721	25,420
263	36700	28,733
264	36982	86,932
265	37000	47,455
266	37000	132,591
267	37543	63,364
268	38073	27,938

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

269	38224	21,898
270	38730	10,369
271	39000	13,943
272	40000	1,761
273	42000	7,080
274	42234	52,636
275	42500	30,807
276	42500	61,807
277	43000	9,034
278	43445	82,915
279	44000	14,278
280	45983	23,557
281	46343	65,341
282	46378	29,489
283	46500	13,636
284	46655	29,511
285	47000	72,977
286	47130	73,125
287	47250	345,670
288	47454	24,398
289	47583	42,614
290	47878	40,830
291	48586	27,636
292	49000	5,807
293	49200	90,000
294	49600	23,233
295	50000	33,313
296	50000	23,568
297	51000	22,011
298	51120	42,563

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

299	51228	48,097
300	51737	32,477
301	53754	155,864
302	55100	71,489
303	57464	25,364
304	57725	22,085
305	58000	21,795
306	59000	2,580
307	59000	51,256
308	59174	48,580
309	59397	8,188
310	60000	1,176
311	60000	17,659
312	62002	26,131
313	63100	34,375
314	64500	18,483
315	64521	68,523
316	68950	40,045
317	70000	24,108
318	72000	13,063
319	72072	32,545
320	77770	32,852
321	81095	158,170
322	81117	16,330
323	81800	53,165
324	82800	141,006
325	82902	708,239
326	83000	241,898
327	83698	15,295
328	84000	69,205

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

329	86413	80,739
330	87850	21,426
331	89000	168,182
332	91500	91,068
333	99000	31,091
334	101998	414,784
335	107000	94,898
336	108000	140,136
337	111600	340,909
338	115770	49,756
339	116826	328,472
340	122921	39,443
341	126900	14,091
342	128000	14,386
343	130000	76,222
344	132000	134,830
345	138000	73,977
346	139028	93,932
347	139480	38,591
348	141000	22,091
349	141727	43,040
350	146000	71,080
351	163756	87,278
352	167000	43,466
353	172550	78,807
354	173000	29,148
355	174269	24,108
356	177135	102,284
357	181000	108,352
358	184720	73,892

Çizelge 4.2 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

359	185311	103,188
360	190000	76,222
361	198000	24,318
362	264000	50,659
363	298779	269,847
364	300000	163,409
365	305834	312,483
366	334800	179,727
367	339015	96,364
368	412409	90,750
369	426000	246,023
370	436649	188,295
371	437004	11,080
372	511681	114,182
373	545010	69,710
374	549000	44,523
375	716164	168,239
376	791000	336,716
377	996676	14,795
378	1041687	13,636
379	1118000	51,136
380	1298600	229,898
381	1325780	10,494
382	1366000	408,784
383	1774849	592,318
384	2339728	658,045

ISBSG veri tabanında bulunan 384 farklı yazılım projelerinin verisi yazılım projeleri veri seti kullanılarak yeni KLOC tabanlı parametrik modeller önerilmiştir.

Şahin ve Gürbüz'e (2014) göre örneklem uzayını 100.000 (yüz bin)'den fazla olması durumunda kullanılacak veri setinin sayısı en az 384 olması gerekmektedir. Bu nedenle yeni modeller önerilirken 384 gerçek yazılım proje metriği kullanılmıştır.

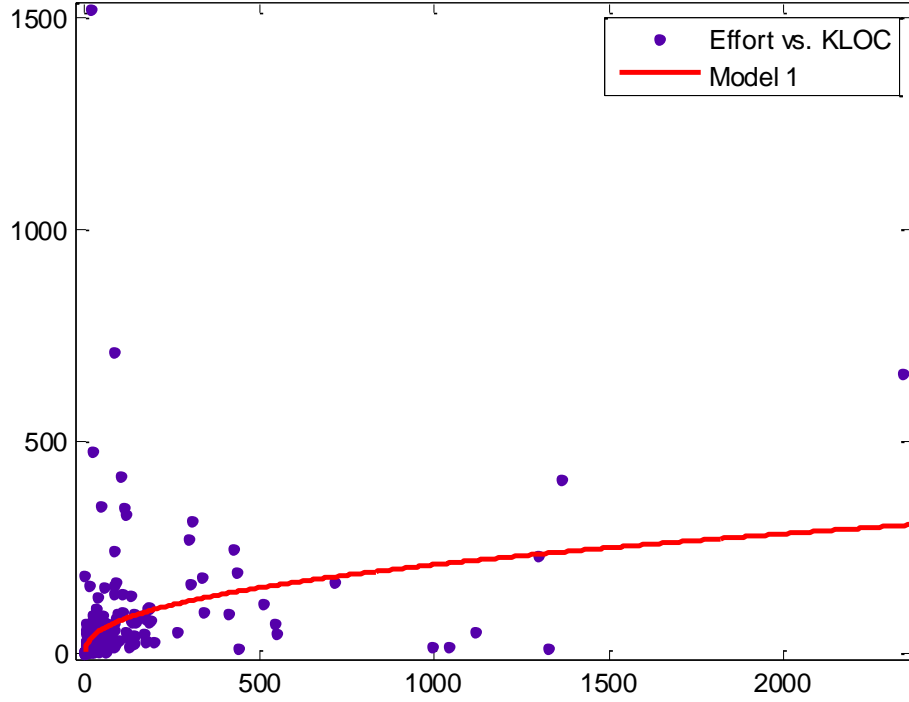
4.4.2 Yeni KLOC tabanlı parametrik tahmin modellerinin önerilmesi

Bu bölümde ise MATLAB Toolboxes uygulaması kullanılarak, 384 proje veri setinden yararlanılarak, yeni modeller önerilmiştir. Yeni modellerin tahminleri ile gerçek yazılım proje efor değerleri karşılaştırılmış, hata ölçütü kullanılarak mevcut modellerle yeni modellerin tahminleri arasında performans kıyaslaması yapılmıştır.

4.4.2.1 Efor = C₁ + C₂ x (KLOC)^{C₃} yeni parametrik tahmin modeli

Literatürde mevcut parametrik modellerin efor tahmini için denklem yapısı $C_1 + C_2 x (KLOC)^{C_3}$ formatı şeklindedir. Burada ortalama karekök hata (RMSE) değerini minimize edecek şekilde C_1, C_2 ve C_3 katsayıları belirlenmek hedeflenmiştir. Bu katsayıları belirlemek maksadıyla MATLAB Toolboxes uygulaması kullanılmıştır. Eğri uydurma araç kutusunda $f(x) = a*(x^b)+c$ denklem tipi seçilmiştir. Uygulamaya veri seti olarak, gerçekleştirilen yazılım projelerinin gerçek efor ve KLOC değerleri girilmiştir. Gerçekleştirilen fit etme sonucunda $C_1 = -3.729$, $C_2 = 11.03$ ve $C_3 = 0.4277$ olarak bulunmuştur. Yeni modelin Efor – KLOC değişimi şekil 4.2'deki grafikte kırmızı hat şeklinde sunulmuştur. Mavi noktalarla gösterilen büyüklükler ise yazılım projelerinin gerçek efor ve kod satır sayısı değerleridir.

Yeni Model denklem seti $Efor = -3.729 + 11.03 x (KLOC)^{0.4277}$ şeklindedir. Elde edilen değerlerin analizi kapsamında ortalama karekök hata ölçütü kullanılarak yeni parametrik efor tahmin modeli sonuçları ile gerçek efor değerleri arasındaki farklar değerlendirilmiştir.



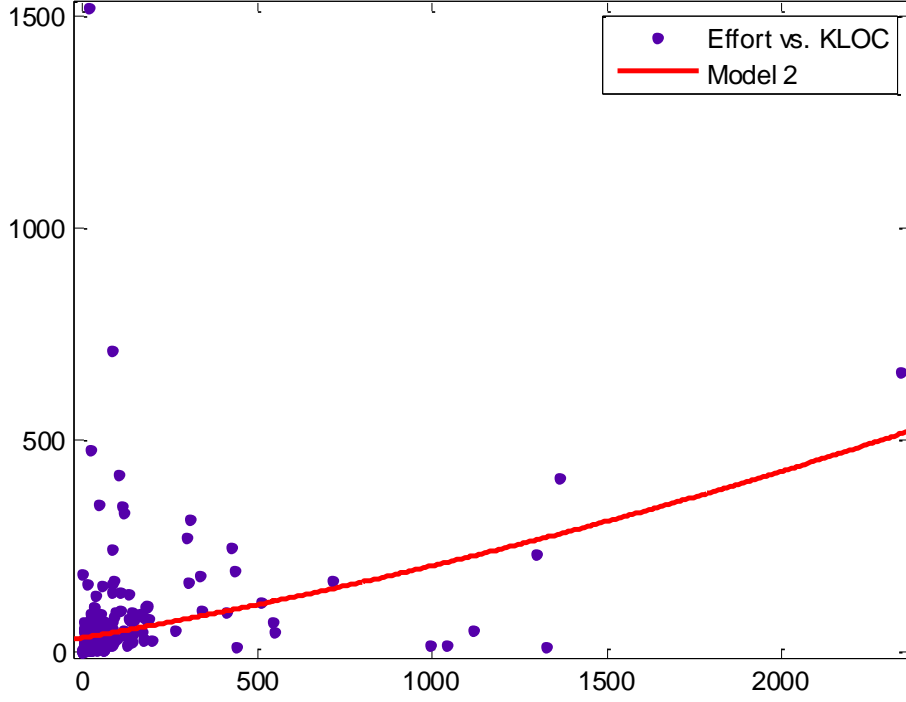
Şekil 4.2 $C_1+C_2 \times (\text{KLOC})^{C_3}$ Yeni Tahmin Modelinin Efor – KLOC Grafiği

4.4.2.2 İkinci derece polinom yeni parametrik tahmin modeli

Bu bölümde ikinci derece polinom şeklinde bir model önerilmesi hedeflenmiştir. Bu modeli denklem yapısı $C_1(\text{KLOC})^2 + C_2(\text{KLOC}) + C_3$ formatı şeklindedir. Burada ortalama karekök hata (RMSE) değerini minimize edecek şekilde C_1, C_2 ve C_3 katsayıları belirlenmek hedeflenmiştir. Bu katsayıları belirlemek amacıyla MATLAB Toolboxes uygulaması kullanılmıştır. Eğri uydurma araç kutusunda quadratic polinom denklem tipi seçilmiştir.

Uygulamaya veri seti olarak, gerçekleştirilen yazılım projelerinin gerçek efor ve KLOC değerleri girilmiştir. Gerçekleştirilen fit etme sonucunda $C_1 = 0.00002645$, $C_2 = 0.1437$ ve $C_3 = 31.69$ olarak bulunmuştur. Yeni modelin Efor – KLOC değişimi şekil 4.3’deki grafikte kırmızı hat şeklinde sunulmuştur. Mavi noktalarla gösterilen büyüklükler ise yazılım projelerinin gerçek efor ve kod satır sayısı değerleridir.

Yeni Model denklem seti $Efor = 0.00002645(KLOC)^2 + 0.1437(KLOC) + 31.69$ şeklindedir.



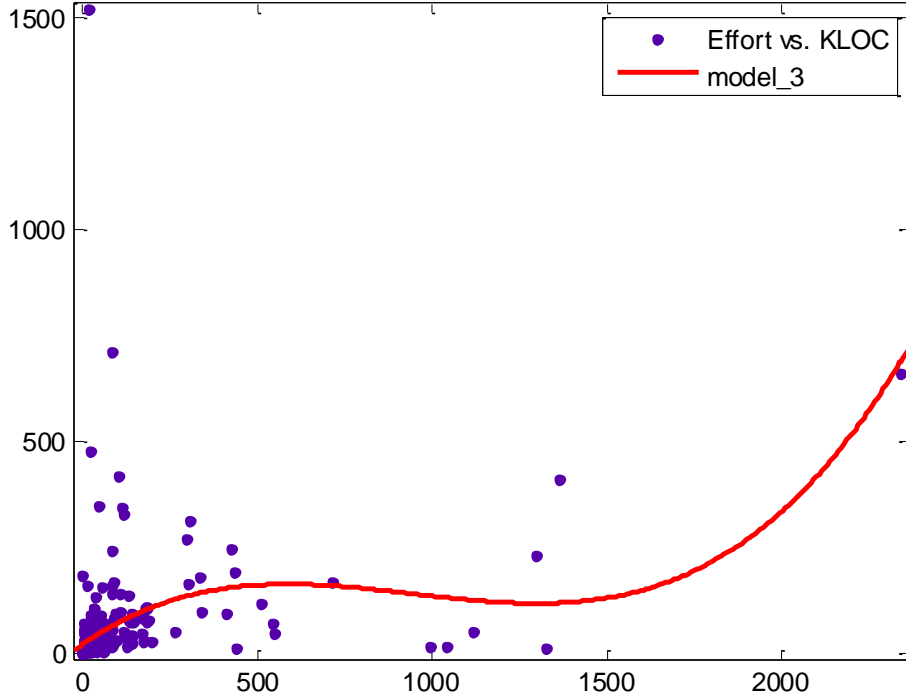
Şekil 4.3 İkinci derece polinom yeni tahmin modelinin Efor – KLOC grafiği

4.4.2.3 Üçüncü derece polinom yeni parametrik tahmin modeli

Bu bölümde üçüncü derece polinom şeklinde bir model önerilmesi hedeflenmiştir. Bu modeli denklem yapısı $C_1(KLOC)^3 + C_2(KLOC)^2 + C_3(KLOC) + C_4$ formatı şeklindedir. Burada ortalama karekök hata (RMSE) değerini minimize edecek şekilde C_1, C_2, C_3 ve C_4 katsayıları belirlenmek hedeflenmiştir. Bu katsayıları belirlemek amacıyla MATLAB Toolboxes uygulaması kullanılmıştır. Eğri uydurma araç kutusunda cubic polinom denklem tipi seçilmiştir. Uygulamaya veri seti olarak gerçekleştirilen yazılım projelerinin gerçek efor ve KLOC değerleri girilmiştir. Gerçekleştirilen fit etme sonucunda $C_1 = 2.493 (10)^{-7}$, $C_2 = 7.052 (10)^{-4}$, $C_3 = 0.5698$ ve $C_4 = 19.98$ olarak bulunmuştur. Yeni modelin Efor – KLOC değişimi şekil 4.4'deki

grafikte kırmızı hat şeklinde sunulmuştur. Mavi noktalarla gösterilen büyüklükler ise yazılım projelerinin gerçek efor ve kod satır sayısı değerleridir.

Yeni Model denklem seti $Efor = 2.493 (10)^{-7}(KLOC)^3 + 7.052 (10)^{-4}(KLOC)^2 + 0.5698 (KLOC) + 19.98$ şeklindedir.



Şekil 4.4 Üçüncü derece polinom yeni tahmin modelinin Efor – KLOC grafiği

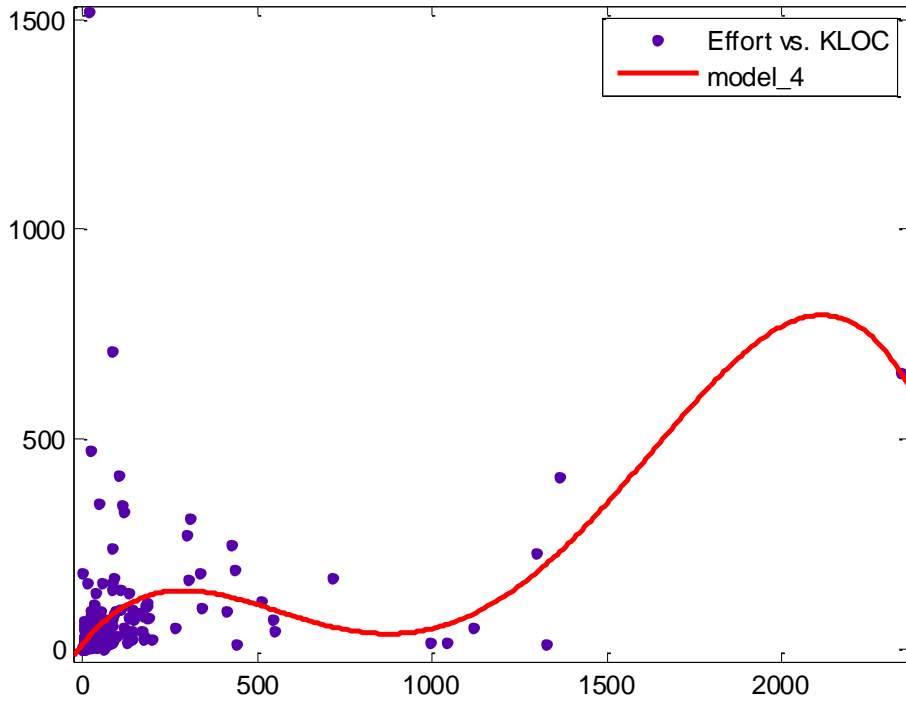
4.4.2.4 Dördüncü derece polinom yeni parametrik tahmin modeli

Bu bölümde dördüncü derece polinom şeklinde bir model önerilmesi hedeflenmiştir. Bu modeli denklem yapısı $C_1(KLOC)^4 + C_2(KLOC)^3 + C_3(KLOC)^2 + C_4(KLOC) + C_5$ formatı şeklindedir. Burada ortalama karekök hata (RMSE) değerini minimize edecek şekilde C_1, C_2, C_3, C_4 ve C_5 katsayıları belirlenmek amaçlanmıştır. Bu katsayıları belirlemek amacıyla MATLAB Toolboxes uygulaması kullanılmıştır. Eğri uydurma araç kutusunda dördüncü derece polinom denklem tipi seçilmiştir. Uygulamaya veri seti olarak, gerçekleştirilen yazılım projelerinin gerçek efor ve KLOC değerleri girilmiştir.

Gerçekleştirilen fit etme sonucunda $C_1 = -4.928 (10)^{-10}$, $C_2 = 2.154 (10)^{-6}$, $C_3 = -0.00267$, $C_4 = 1.047$ ve $C_5 = 10.94$ olarak bulunmuştur.

Yeni modelin Efor – KLOC değişimi şekil 4.5'deki grafikte kırmızı hat şeklinde sunulmuştur. Mavi noktalarla gösterilen büyüklükler ise yazılım projelerinin gerçek efor ve kod satır sayısı değerleridir.

Yeni Model denklem seti $Efor = -4.928 (10)^{-10} (KLOC)^4 + 2.154 (10)^{-6} (KLOC)^3 - 0.00267 (KLOC)^2 + 1.047 (KLOC) + 10.94$ şeklindedir.



Şekil 4.5 Dördüncü derece polinom yeni tahmin modelinin Efor – KLOC grafiği

4.4.2.5 Doğrulama ve yeni modellerle mevcut modellerin kıyaslaması

Yeni modeller 384 (Tüm veri setinin % 80'i) adet yazılım projesi versiyonu kullanılarak önerilmişti ve önerilen modellerin doğrulaması amacıyla 96 adet (Tüm veri setinin % 20'si) yazılım proje verileri kullanılarak ilk olarak literatürde mevcut parametrik

modellerin ortalama karekök hata oranları her bir model için ayrı hesaplanmıştır. Doğrulama için kullanılan veri seti 480 proje arasından rastgele seçilerek kullanılmıştır. Bu 96 adet proje bilgi çizelge 4.2’de sunulmuştur.

Çizelge 4.3 ISBSG 96 adet yazılım projeleri veri seti (Data Release 11, 2009)

Proje Numarası	LOC (Kod Satır Sayısı)	Gerçek Efor (Adam-Ay)
1	113	1,784
2	779	0,761
3	25	1,244
4	582	2,199
5	14	2,813
6	442	3,517
7	203	15,330
8	277	2,631
9	44	5,864
10	702	1,489
11	687	2,460
12	28	4,813
13	77	2,341
14	149	6,705
15	493	12,500
16	217	3,670
17	359	4,818
18	193	1,250
19	205	3,784
20	171	6,114
21	387	5,710
22	263	10,580
23	42	4,063
24	39	31,284
25	181	5,568
26	51	7,159
27	262	12,784
28	77	23,915
29	233	14,602
30	205	3,545
31	874	4,744
32	1153	11,381
33	1305	7,102
34	34	20,097
35	749	9,773
36	242	54,000
37	71	7,875

Çizelge 4.3 ISBSG 96 adet yazılım projeleri veri seti (Data Release 11, 2009) (devam)

38	89	17,278
39	8	15,864
40	539	149,045
41	177	30,199
42	73	12,705
43	161	53,068
44	711	4,176
45	46	46,455
46	72	9,142
47	2470	12,636
48	593	7,386
49	733	17,091
50	82	10,273
51	106	15,739
52	42	39,778
53	143	16,847
54	306	5,932
55	356	12,477
56	142	105,795
57	599	12,545
58	298	7,398
59	59	301,670
60	331	16,909
61	193	11,563
62	35	25,188
63	775	38,909
64	319	6,040
65	180	54,568
66	180	21,352
67	457	9,528
68	14	44,602
69		151,801
70	287	21,994
71	194	30,170
72	250	47,102
73	216	93,068
74	143	22,386
75	838	16,483
76	421	37,955
77	14	24,369
78	2190	129,722
79	61	102,097
80	179	102,097
81	35	70,699
82	66	595,920

Çizelge 4.3 ISBSG 96 adet yazılım projeleri veri seti (Data Release 11, 2009) (devam)

83	29	15,017
84	465	9,756
85	103	206,040
86	488	35,386
87	234	45,239
88	2734	60,625
89	19	114,545
90	121	201,222
91	90	305,193
92	176	762,563
93	700	127,227
94	94	336,716
95	174	51,136
96	1475	592,318

Modellerin kıyaslanmasında, iki değişik hata ölçütü kullanılarak parametrik efor tahmin yöntemleri sonuçları ile gerçek efor değerleri arasındaki farklar değerlendirilmektedir. Hata modelleri, Ortalama Karekök Hata (Root Mean Square Error - RMSE), ve Ortalama Yüzde Hata (Mean Magnitude of Relative Error - MMRE)'dir. Ortalama Kare Hata bir model tarafından tahmin edilen değerler arasındaki farklılıkları ölçmek için sıkça kullanılır. Ortalama Karekök Hata aşağıda sunulan denklemlerle gösterilmektedir (Sheta, Rine ve Ayesh 2008).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (E_{tahmin} - E_{gerçek})^2} \quad (4.1)$$

Literatürde Ortalama Yüzde Hata modeli temel performans ölçütü olarak kabul edilir. Ortalama Yüzde Hata bağıl hataların mutlak değerinin yüzdesidir (Benediktsson, Dalcher, Reed ve Woodman 2003). Şu şekilde yazılmaktadır:

$$MMRE = \frac{1}{N} \sum_{i=1}^N \frac{|E_{tahmin} - E_{gerçek}|}{E_{gerçek}} \quad (4.2)$$

Denklem (4.1) ve denklem (4.2) kullanılarak parametrik modellerin tahmin ettiği eforlar ile gerçek eforlar arasındaki hatalar hesaplanmıştır. Her bir model için bulunan hata değerleri çizelge 4.4’de sunulmaktadır.

Çizelge 4.4 Parametrik modellerle hesaplanan eforların hataları

Hata Ölçütü	Kullanılan Efor Tahmin Modelleri ve Hataları					
	Walston Felix Modeli	COCOMO Organic	Bailey Basili Modeli	Barry Boehm Modeli	Doty Modeli	Halstead Modeli
RMSE	593,24	757,27	491,64	1031,80	1712,94	46626,61

Hesaplanan hata değerlerini incelediğimizde Bailey Basili parametrik efor tahmin modelinin diğer efor tahmin modellerine göre yazılım projeleri veri seti için en iyi sonuçlar verdiği tespit edilmiştir. Diğer iyi sonuç veren parametrik modeller ise Waltson-Felix ve COCOMO Organic olarak sıralanmaktadır. Yeni önerilen KLOC tabanlı parametrik modellerin performansı doğrulama veri seti kullanılarak ortaya konulmuştur. Bu kapsamda önerilen dört yeni model için ortalama karekök hata oranları hesaplanmış ve çizelge 4.5’de sunulmuştur.

Çizelge 4.5 Yeni KLOC modellerle hesaplanan eforların hataları

Hata Ölçütü	Yeni KLOC Efor Tahmin Modelleri ve Hataları			
	1. KLOC Modeli	2. KLOC Modeli	3. KLOC Modeli	4. KLOC Modeli
RMSE	106,18	110,64	107,66	99,89

Hesaplanan hata oranlarına bakıldığında, yeni önerilen modellerin daha iyi tahminlerde bulunduğu görülmektedir. Bunun temel sebebi literatürdeki mevcut parametrik modellerin kısıtlı veri setleri ile önerilmiş olmasıdır. Bu veri setleri kullanılarak model önerilerinde bulunulmuştur. Bu nedenle o veri setlerine uygun eğri denklemleri elde edilmiştir. Tez çalışmasında önerilen modeller için 384 farklı proje veri seti kullanılmıştır. Veri setinde bulunan proje sayısının yeterli olması nedeniyle önerilen modellerin daha iyi tahminlerde bulunduğu değerlendirilmektedir.

4.4.3 Mevcut FP tabanlı tahmin modellerinin kıyaslanması

İşlev puanı (FP) tabanlı Parametrik Tahmin Modelleri kaynaklar incelenerek denklem setleriyle birlikte bir araya getirilmiştir. Literatürde mevcut FP tabanlı efor tahmin modelleri çizelge 4.6’de sunulmuştur.

Çizelge 4.6 FP tabanlı parametrik efor tahmin modelleri

Sıra Nu	Parametrik Efor Tahmin Modelleri (Laird ve Brennan 2010)	
	Model İsmi	Denklem
1	Albrecht–Gaffney Modeli	$Effort = 13.39 + 0.0545 \times FP$
2	Kemerer Modeli	$Effort = 60.62 + 7.728 \times (10^{-8}) \times FP^3$
3	Matson–Barret–Meltichamp Modeli	$Effort = 585.7 + 15.12 \times FP$

Çizelge 4.7’de ISBSG veri tabanında bulunan 384 farklı gerçekleştirilen yazılım proje bilgi sunulmaktadır. Her bir yazılım projesinin uygulamaya dayalı gerçek eforları veri setinde mevcuttur. Yazılım projelerinin büyüklüğü işlev puanı olarak alınmıştır. Gerçek eforun birimi ise adam-aydır.

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009)

Proje Numarası	FP (İşlev Puanı)	Gerçek Efor (Adam-Ay)
1	254	10,511
2	443	4,523
3	74	6,250
4	3	0,159
5	478	125,000
6	620	103,182
7	297	42,324
8	183	2,614
9	179	1,540
10	832	119,176
11	183	4,483
12	198	14,545
13	135	41,420
14	568	33,369
15	849	10,648
16	108	7,250
17	460	9,472
18	1587	42,557
19	66	2,881
20	56	5,733
21	2533	14,347
22	400	0,852
23	22	0,761
24	8	2,250
25	204	44,210
26	83	4,858
27	90	16,358
28	250	16,614
29	174	10,659
30	152	13,574
31	4	0,619
32	48	3,682
33	82	72,170
34	140	5,682
35	135	2,835
36	826	14,222
37	309	13,295
38	338	3,670
39	260	23,580
40	66	16,563
41	90	16,358
42	261	12,358
43	152	3,795

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

44	16	3,045
45	78	1,494
46	252	18,398
47	186	1,659
48	292	17,011
49	166	5,375
50	83	7,574
51	459	22,023
52	694	36,006
53	1053	4,301
54	969	85,864
55	79	9,131
56	673	34,125
57	97	6,580
58	7	0,403
59	1282	47,892
60	270	9,426
61	338	10,727
62	68	14,318
63	127	3,330
64	53	1,438
65	191	4,188
66	13	1,949
67	2780	4,551
68	303	20,284
69	3625	67,807
70	287	11,136
71	111	7,369
72	154	13,682
73	212	4,295
74	3	0,369
75	116	7,063
76	52	19,159
77	41	4,972
78	240	10,352
79	859	64,614
80	44	7,324
81	1306	417,619
82	14	0,949
83	189	7,119
84	158	13,455
85	188	7,693
86	465	57,955
87	230	5,284
88	304	30,551

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

89	132	8,438
90	811	6,114
91	5	3,148
92	29	5,295
93	149	7,034
94	24	10,000
95	1790	86,034
96	700	20,330
97	355	13,722
98	82	0,864
99	7	0,313
100	67	9,682
101	342	12,216
102	233	10,324
103	76	9,545
104	199	15,000
105	447	9,545
106	320	7,540
107	455	4,744
108	267	48,216
109	74	4,568
110	7555	651,409
111	324	33,335
112	152	14,040
113	57	1,875
114	203	2,830
115	702	205,506
116	255	0,977
117	137	10,494
118	214	9,460
119	121	15,278
120	218	28,392
121	88	5,028
122	65	3,773
123	249	12,523
124	220	2,324
125	58	0,915
126	55	4,205
127	484	16,432
128	60	9,131
129	185	55,222
130	147	6,648
131	137	10,881
132	84	5,341
133	588	5,210

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

134	176	19,023
135	21	1,659
136	270	25,665
137	11	0,227
138	391	3,841
139	242	16,324
140	130	1,250
141	57	18,278
142	140	3,966
143	20	0,210
144	2486	26,045
145	766	23,636
146	40	1,483
147	190	6,653
148	598	16,818
149	9296	14,841
150	128	4,989
151	31	3,284
152	1250	47,102
153	69	0,790
154	389	5,051
155	470	90,000
156	483	24,705
157	164	30,000
158	12	10,097
159	320	29,489
160	109	0,955
161	245	20,068
162	67	3,193
163	106	20,909
164	1384	2,324
165	21	1,523
166	154	10,989
167	4272	100,000
168	36	15,119
169	94	3,523
170	355	5,165
171	359	20,148
172	37	0,847
173	3156	51,670
174	354	35,000
175	8	0,494
176	364	24,659
177	730	17,659
178	12	0,108

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

179	538	7,642
180	183	47,642
181	307	9,068
182	159	11,795
183	303	35,778
184	294	7,091
185	348	30,534
186	200	10,994
187	474	9,341
188	282	11,011
189	215	11,057
190	124	15,909
191	142	0,989
192	163	53,597
193	63	1,307
194	345	6,415
195	4	0,136
196	170	11,210
197	230	5,466
198	395	33,523
199	271	10,000
200	28	2,182
201	189	8,920
202	228	16,347
203	75	11,028
204	78	1,909
205	4	0,227
206	1130	89,483
207	455	60,625
208	39	0,818
209	199	21,284
210	58	1,580
211	397	26,523
212	68	18,455
213	407	8,364
214	135	27,778
215	94	3,523
216	270	19,489
217	14	0,318
218	107	2,648
219	51	2,188
220	243	10,080
221	100	3,136
222	321	2,386
223	379	0,199

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

224	93	42,000
225	55	5,841
226	116	7,955
227	94	21,017
228	56	1,682
229	349	3,290
230	117	17,653
231	477	58,756
232	99	1,989
233	242	8,352
234	71	1,398
235	556	9,540
236	14	1,699
237	3403	96,273
238	156	4,250
239	332	49,841
240	106	1,489
241	244	1,085
242	79	16,483
243	57	2,148
244	29	1,563
245	122	20,239
246	28	5,381
247	357	9,920
248	22	11,176
249	276	3,210
250	1094	80,494
251	98	16,898
252	99	1,449
253	6	0,205
254	43	3,256
255	227	27,205
256	1570	39,614
257	441	2,580
258	27	2,551
259	250	32,068
260	101	2,307
261	114	13,483
262	45	1,028
263	99	3,170
264	1069	1,705
265	142	73,875
266	99	6,818
267	2189	71,551
268	414	2,614

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

269	23	8,767
270	246	13,926
271	103	6,869
272	192	10,142
273	76	5,920
274	324	31,244
275	320	53,165
276	22	2,602
277	454	29,227
278	103	44,364
279	1044	22,097
280	94	16,898
281	25	2,034
282	116	9,222
283	60	2,676
284	83	12,955
285	57	3,563
286	76	4,063
287	118	3,290
288	101	2,722
289	748	67,642
290	165	6,000
291	273	2,818
292	112	3,364
293	7	0,534
294	495	8,739
295	288	2,619
296	118	25,483
297	8	0,540
298	397	30,597
299	38	61,034
300	109	12,568
301	424	39,455
302	344	31,938
303	107	6,051
304	42	1,432
305	171	22,273
306	1925	79,966
307	400	3,375
308	73	6,881
309	62	11,250
310	44	8,841
311	470	11,182
312	158	5,114
313	179	3,153

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

314	100	2,682
315	979	28,392
316	27	1,324
317	305	65,455
318	279	2,188
319	113	3,722
320	157	6,352
321	4484	173,761
322	202	10,489
323	736	73,693
324	198	6,165
325	352	8,523
326	479	6,722
327	386	16,011
328	142	4,426
329	9	1,358
330	564	8,983
331	19	0,597
332	309	9,375
333	65	15,239
334	95	5,932
335	2592	9,750
336	228	5,153
337	220	6,273
338	153	27,239
339	11	1,136
340	258	8,909
341	308	27,182
342	6	1,023
343	1052	222,182
344	1023	8,216
345	1449	18,551
346	203	7,170
347	344	71,011
348	368	25,074
349	168	5,153
350	107	5,540
351	30	4,858
352	229	1,932
353	496	8,790
354	255	20,091
355	17	0,307
356	162	9,966
357	782	13,313
358	103	54,392

Çizelge 4.7 ISBSG yazılım projeleri veri seti (Data Release 11, 2009) (devam)

359	156	1,290
360	146	13,318
361	43	0,852
362	338	4,949
363	11	0,619
364	185	16,290
365	72	11,705
366	184	11,080
367	696	34,222
368	546	46,102
369	949	21,392
370	173	40,284
371	2406	66,477
372	183	8,602
373	302	3,858
374	699	45,795
375	30	4,670
376	124	9,347
377	34	5,750
378	116	8,591
379	703	13,267
380	1281	33,716
381	143	53,602
382	279	7,415
383	201	8,125
384	155	40,625

ISBSG veri tabanında bulunan 384 farklı yazılım projelerinin verisi yazılım projeleri veri seti kullanılarak yeni FP tabanlı parametrik modeller önerilmiştir. Şahin ve Gürbüz'e göre (2014) örneklem uzayını 100.000 (yüz bin)'den fazla olması durumunda kullanılacak veri setinin sayısı en az 384 olması gerekmektedir. Bu nedenle yeni modeller önerilirken 384 gerçek yazılım proje metriği kullanılmıştır.

4.4.4 Yeni FP tabanlı parametrik tahmin modellerinin önerilmesi

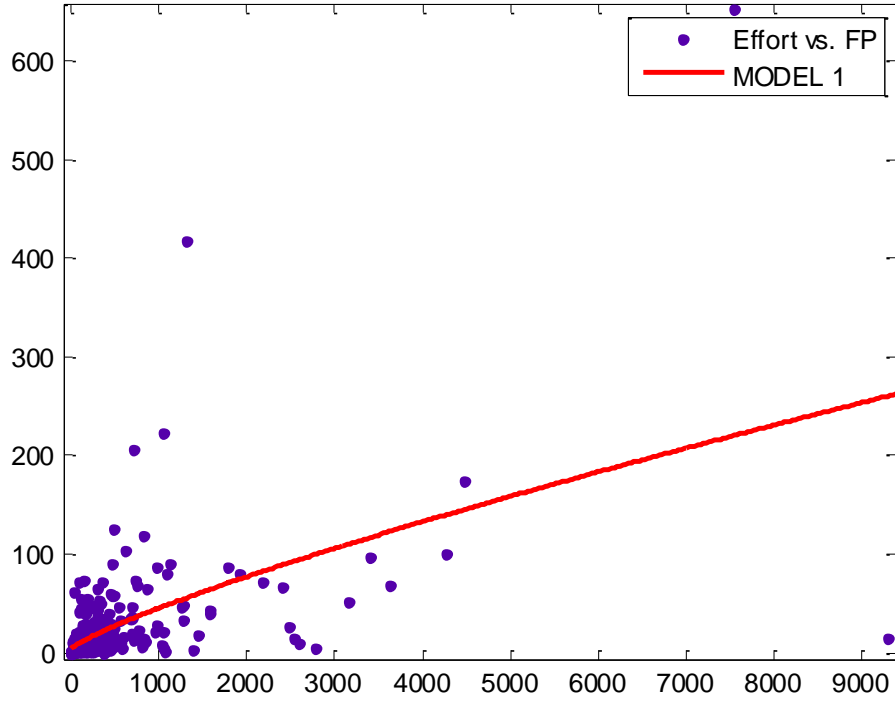
Bu bölümde MATLAB Toolboxes uygulaması kullanılarak, 384 proje veri setinden yararlanılarak, yeni modeller önerilmiştir. Yeni modellerin tahminleri ile gerçek yazılım

proje efor deęerleri karřılařtırılmıř, hata ölçütü kullanılarak mevcut modellerle yeni modellerin tahminleri arasında performans kıyaslaması yapılmıřtır.

4.4.4.1 Efor = C₁ + C₂ x (FP)^{C₃} yeni parametrik tahmin modeli

Literatürde mevcut parametrik modellerin efor tahmini için denklem yapısı C₁+C₂x(FP)^{C₃} formatı şeklindedir. Burada ortalama karekök hata (RMSE) deęerini minimize edecek şekilde C₁,C₂ ve C₃ katsayıları belirlenmek hedeflenmiřtir. Bu katsayıları belirlemek maksadıyla MATLAB Toolboxes uygulaması kullanılmıřtır. Eğri uydurma araç kutusunda f(x) = a*(x^b)+c denklem tipi seçilmiřtir.

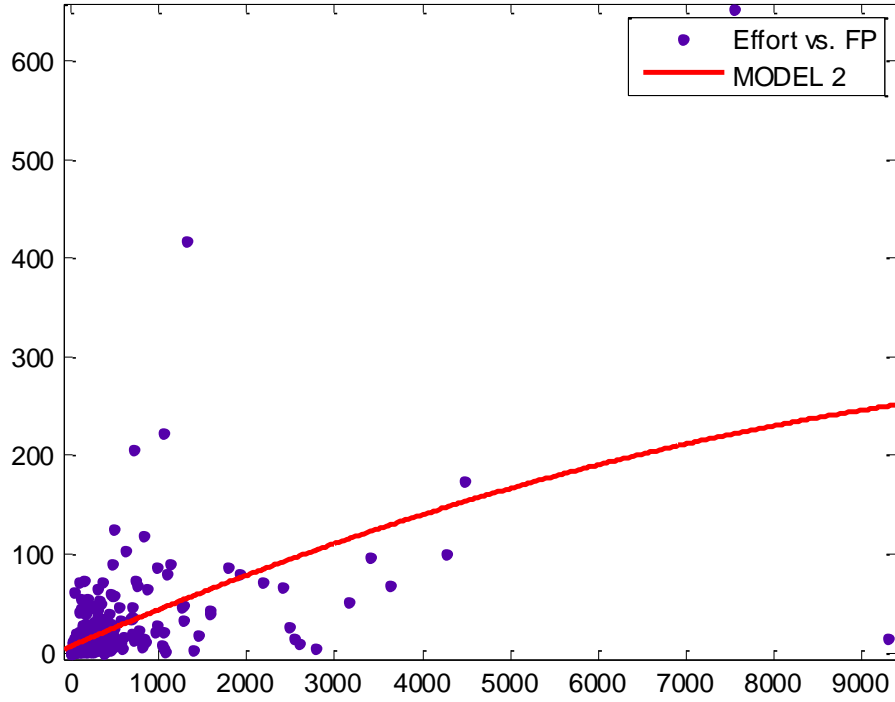
Uygulamaya veri seti olarak, gerekleřtirilen yazılım projelerinin gerek efor ve FP deęerleri girilmiřtir. Gerekleřtirilen fit etme sonucunda C₁ = 3.587, C₂ = 0.1539 ve C₃ = 3.587 olarak bulunmuřtur. Yeni modelin Efor – FP deęiřimi řekil 4.6'deki grafikte kırmızı hat şeklinde sunulmuřtur. Mavi noktalarla gősterilen büyüklükler ise yazılım projelerinin gerek efor ve iřlev puanı deęerleridir. Yeni Model denklem seti Efor = 3.587 + 0.1529 x (FP)^{0.8126} şeklindedir.



Şekil 4.6 $C_1+C_2 \times (KLOC)^{C_3}$ yeni tahmin modelinin Efor – FP grafiği

4.4.4.2 İkinci derece polinom yeni parametrik tahmin modeli

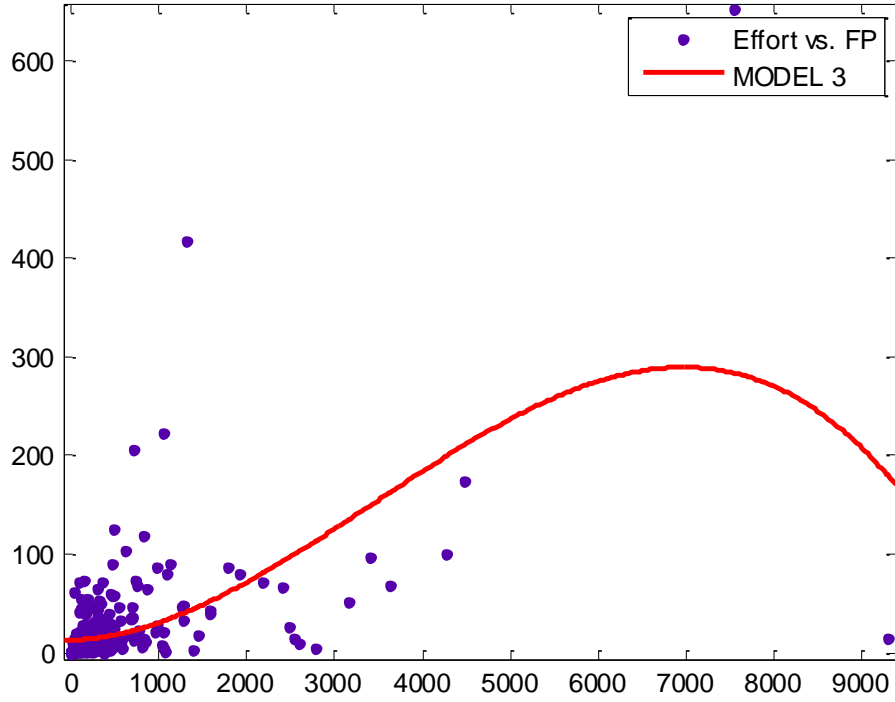
Bu bölümde ikinci derece polinom şeklinde bir model önerilmesi hedeflenmiştir. Bu modeli denklem yapısı $C_1(FP)^2 + C_2(FP) + C_3$ formatı şeklindedir. Burada ortalama karekök hata (RMSE) değerini minimize edecek şekilde C_1, C_2 ve C_3 katsayıları belirlenmek hedeflenmiştir. Bu katsayıları belirlemek amacıyla MATLAB Toolboxes uygulaması kullanılmıştır. Eğri uydurma araç kutusunda quadratic polinom denklem tipi seçilmiştir. Uygulamaya veri seti olarak, gerçekleştirilen yazılım projelerinin gerçek efor ve FP değerleri girilmiştir. Gerçekleştirilen fit etme sonucunda $C_1 = -1.361(10)^{-3}$, $C_2 = 0.03884$ ve $C_3 = 6.589$ olarak bulunmuştur. Yeni modelin Efor – FP değişimi şekil 4.7'deki grafikte kırmızı hat şeklinde sunulmuştur. Mavi noktalarla gösterilen büyüklükler ise yazılım projelerinin gerçek efor ve işlev puanı değerleridir. Yeni Model denklem seti $Efor = -1.361(10)^{-3} (FP)^2 + 0.03884 (FP) + 6.589$ şeklindedir.



Şekil 4.7 İkinci derece polinom yeni tahmin modelinin Efor – FP grafiği

4.4.4.3 Üçüncü derece polinom yeni parametrik tahmin modeli

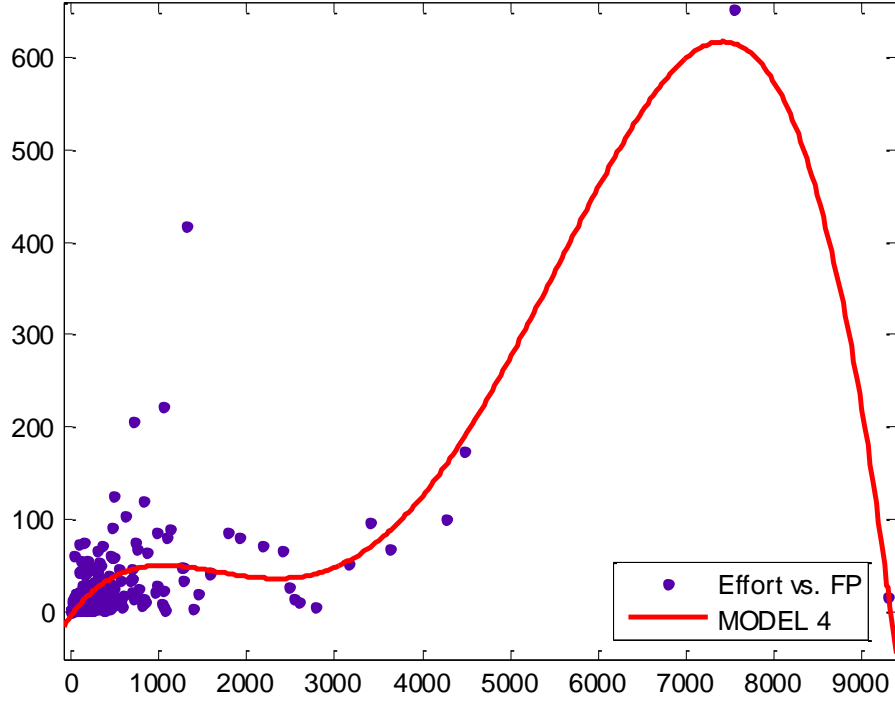
Bu bölümde üçüncü derece polinom şeklinde bir model önerilmesi hedeflenmiştir. Bu modeli denklem yapısı $C_1(FP)^3 + C_2(FP)^2 + C_3(FP) + C_4$ formatı şeklindedir. Burada ortalama karekök hata (RMSE) değerini minimize edecek şekilde C_1, C_2, C_3 ve C_4 katsayıları belirlenmek hedeflenmiştir. Bu katsayıları belirlemek amacıyla MATLAB Toolboxes uygulaması kullanılmıştır. Eğri uydurma araç kutusunda cubic polinom denklem tipi seçilmiştir. Uygulamaya veri seti olarak, gerçekleştirilen yazılım projelerinin gerçek efor ve FP değerleri girilmiştir. Gerçekleştirilen fit etme sonucunda $C_1 = -1.563 (10)^{-9}$, $C_2 = 1.611 (10)^{-5}$, $C_3 = 3.357 (10)^{-3}$ ve $C_4 = 12.92$ olarak bulunmuştur. Yeni modelin Efor – FP değişimi şekil 4.8’deki grafikte kırmızı hat şeklinde sunulmuştur. Mavi noktalarla gösterilen büyüklükler ise yazılım projelerinin gerçek efor ve işlev puanı değerleridir. Yeni Model denklem seti $Efor = -1.563 (10)^{-9} (FP)^3 + 1.611 (10)^{-5} (FP)^2 + 3.357 (10)^{-3} (FP) + 12.92$ şeklindedir.



Şekil 4.8 Üçüncü derece polinom yeni tahmin modelinin Efor – FP grafiği

4.4.4.4 Dördüncü derece polinom yeni parametrik tahmin modeli

Bu bölümde dördüncü derece polinom şeklinde bir model önerilmesi hedeflenmiştir. Bu modeli denklem yapısı $C_1(FP)^4 + C_2(FP)^3 + C_3(FP)^2 + C_4(FP) + C_5$ formatı şeklindedir. Burada ortalama karekök hata (RMSE) değerini minimize edecek şekilde C_1, C_2, C_3, C_4 ve C_5 katsayıları belirlenmek amaçlanmıştır. Bu katsayıları belirlemek amacıyla MATLAB Toolboxes uygulaması kullanılmıştır. Eğri uydurma araç kutusunda dördüncü derece polinom denklem tipi seçilmiştir. Uygulamaya veri seti olarak, gerçekleştirilen yazılım projelerinin gerçek efor ve FP değerleri girilmiştir. Gerçekleştirilen fit etme sonucunda $C_1 = -1.734 (10)^{-12}$, $C_2 = 2.5 (10)^{-8}$, $C_3 = -9.577(10)^{-5}$, $C_4 = 0.1264$ ve $C_5 = -4.212$ olarak bulunmuştur. Yeni modelin Efor – FP değişimi Şekil 4.9'deki grafikte kırmızı hat şeklinde sunulmuştur. Mavi noktalarla gösterilen büyüklükler ise yazılım projelerinin gerçek efor ve kod satır sayısı değerleridir. Yeni Model denklem seti $Efor = -1.734 (10)^{-12} (FP)^4 + 2.5 (10)^{-8} (FP)^3 + -9.577(10)^{-5} (FP)^2 + 0.1264 (FP) - 4.212$ şeklindedir.



Şekil 4.9 Dördüncü derece polinom yeni tahmin modelinin Efor – FP grafiği

4.4.4.5 Doğrulama ve yeni modellerle mevcut modellerin kıyaslaması

Yeni modeller 384 (Tüm veri setinin % 80'i) adet yazılım projesi versiyonu kullanılarak önerilmişti ve önerilen modellerin doğrulaması amacıyla 96 adet (Tüm veri setinin % 20'si) yazılım proje verileri kullanılarak ilk olarak literatürde mevcut parametrik modellerin ortalama karekök hata oranları her bir model için ayrı hesaplanmıştır. Doğrulama için kullanılan veri seti 480 proje arasından rastgele seçilerek kullanılmıştır. Bu 96 adet proje bilgisi Çizelge 4.9'de sunulmuştur.

Çizelge 4.8 ISBSG 96 adet yazılım projeleri veri seti (Data Release 11, 2009)

Proje Numarası	FP (Kod Satır Sayısı)	Gerçek Efor (Adam-Ay)
1	220	3,386364
2	927	142,2727
3	1300	0,670455
4	1345	14,95455
5	1350	0,460227
6	1600	21,43182
7	1618	2,386364
8	2000	113,6364
9	2248	2,056818
10	2800	27,30114
11	3694	10,15909
12	4200	3,897727
13	4869	1,153409
14	4898	6,477273
15	4910	5,102273
16	5200	36,69886
17	5396	48,75
18	6250	9,761364
19	7456	12,07955
20	8310	15,78977
21	8404	7,017045
22	9000	5,176136
23	9087	14,18182
24	9167	1,119318
25	9400	1,647727
26	10000	2,369318
27	11000	23,09091
28	11967	2,477273
29	11993	30,04545
30	12000	18,63636
31	12060	89,88068
32	12443	93,93182
33	12908	151,8011
34	14017	4,8125
35	14290	26,26705
36	14800	7,982955
37	16000	25,81818
38	16751	0,744318
39	18322	1,113636
40	18749	12,72727
41	19800	14,83523
42	20830	10,21591
43	21500	6,511364

Çizelge 4.8 ISBSG 96 adet yazılım projeleri veri seti (Data Release 11, 2009) (devam)

44	21900	107,0455
45	22488	1,954545
46	23229	2,005682
47	23901	2,795455
48	24000	7,818182
49	24543	25,81818
50	25123	2,329545
51	25345	19,90341
52	26142	4,170455
53	27580	10,92045
54	28000	11,82955
55	30374	11,30114
56	31800	17,55682
57	32000	51,46591
58	34086	13,6875
59	34567	1,988636
60	35000	1,863636
61	35230	9,755682
62	35760	2,789773
63	36178	102,5795
64	41380	12,01705
65	45100	27,13068
66	47136	3,977273
67	47450	35,64773
68	48000	0,289773
69	48600	73,89205
70	48817	57,71023
71	51900	22,35227
72	59303	4,602273
73	63000	13,76136
74	64700	6,471591
75	64805	15,21591
76	66000	40,82955
77	67750	0,238636
78	74402	82,5
79	78910	5,227273
80	78910	2,693182
81	79560	10,38636
82	80050	2,664773
83	80660	1,079545
84	81000	5,045455
85	89030	16,21023
86	107000	39,92614
87	110000	6,653409
88	116000	11,07955

Çizelge 4.8 ISBSG 96 adet yazılım projeleri veri seti (Data Release 11, 2009) (devam)

89	132000	0,772727
90	134216	22,58523
91	229900	6,221591
92	251135	6,897727
93	512630	51,13636
94	791000	14,23864
95	1118000	24,0625
96	1774849	84,03977

Verilerin analizinde, iki değişik hata ölçütü kullanılarak işlev puanı tabanlı parametrik efor tahmin yöntemleri sonuçları ile gerçek efor değerleri arasındaki farklar değerlendirilmiştir.

Denklem (4.3) ve denklem (4.4) kullanılarak parametrik modellerin tahmin ettiği eforlar ile gerçek eforlar arasındaki hatalar hesaplanmıştır. Her bir işlev puanı tabanlı model için bulunan hata değerleri çizelge 4.10'de sunulmaktadır.

Çizelge 4.9 Parametrik modellerle hesaplanan eforların hataları

Hata Ölçütü	Kullanılan Efor Tahmin Modelleri ve Hataları		
	Albrecht - Gaffney	Kemerer	Matson – Barret - Meltichamp
RMSE	32,80	231,18	9388,35

Bu kapsamda MMRE ve RMSE olmak üzere iki farklı hata ölçütü kullanılmış, parametrik tahmin modellerinin hata değerleri bulunmuştur. Hesaplanan hata değerlerini incelediğimizde Albrecht Gaffney parametrik efor tahmin modelinin diğer efor tahmin modellerine göre daha iyi sonuçlar verdiği tespit edilmiştir.

Yeni önerilen FP tabanlı parametrik modellerin performansı doğrulama veri seti kullanılarak ortaya konulmuştur. Bu kapsamda önerilen dört yeni model için ortalama karekök hata oranları hesaplanmış ve çizelge 4.10’de sunulmuştur.

Çizelge 4.10 Yeni FP modellerle hesaplanan eforların hataları

Hata Ölçütü	Yeni KLOC Efor Tahmin Modelleri ve Hataları			
	1. KLOC Modeli	2. KLOC Modeli	3. KLOC Modeli	4. KLOC Modeli
RMSE	28,10	28,53	31,22	26,39

Hesaplanan hata oranlarına bakıldığında, yeni önerilen modellerin daha iyi tahminlerde bulunduğu görülmektedir. Bunun temel sebebi literatürdeki mevcut parametrik modellerin kısıtlı veri setleri ile önerilmiş olmasıdır. Bu veri setleri kullanılarak model önerilerinde bulunulmuştur. Bu nedenle o veri setlerine uygun eğri denklemleri elde edilmiştir. Tez çalışmasında önerilen modeller için 384 farklı proje veri seti kullanılmıştır. Veri setinde bulunan proje sayısının yeterli olması nedeniyle önerilen modellerin daha iyi tahminlerde bulunduğu değerlendirilmektedir.

5. MATLAB'TE GRAFİKSEL KULLANICI ARABİRİMİ (GUI) KULLANIMI VE YAZILIM PAKETİNİN HAZIRLANMASI

Çalışmanın bu bölümünde literatür bulunan mevcut parametrik tahmin modelleri ve tez kapsamında elde edilen yeni modelleri bir araya getirilerek, MATLAB ortamında bir araç kutusu oluşturulmuştur. Bu yazılım paketinin hazırlanma amacı, yazılım projeleriyle ilgilenen proje yöneticileri ve yazılımcılara kolaylık sağlamak ve parametrik tahmin yöntemleri arasında kıyaslama imkânını sağlamaktır.

Bu kapsamda MATLAB Grafikselle Kullanıcı Arabirimi (GUI)'nde araç kutusu nasıl hazırlanır, hazırlama teknikleri nelerdir, hazırlanan yazılım paketi üzerinde ne gibi değişikliklerin yapılacağı konusunda detaylı bilgiler verilmiştir. Ayrıca parametrik tahmin modelleri için hazırlanan araç kutusunun tanıtım son bölümde sunulmuştur.

5.1 Grafikselle Kullanıcı Arabirimi (GUI) Çalışma Prensipleri

MATLAB GUI kullanıcıyla etkileşimli bir şekilde çalışan, içerdiği nesnelerin kullanılması amacıyla alt programlar koşturulmasını sağlayan grafikselle bir program arayüzüdür. Söz konusu bileşen nesneler içinde radyo butonlar, yazı kutuları, araç çubukları, liste kutuları vb. bulunmaktadır. Bunun haricinde GUI, MATLAB'in sağladığı hesaplama imkânlarını kullanarak grafik çizimi, veri alımı gibi birçok fonksiyonel işlem yapabilme kabiliyetine sahiptir.

GUI'de bulunan her bir nesne, tanımlanan program dosyasında callback diye isimlendirilen alt programlara sahiptir. Bu kapsamda her nesne ile ilgili olaylarda GUI o nesnenin fonksiyonuna ait callback alt programlarını çalıştırır. Böylelikle GUI hem bir arayüz olarak çalışırken hem de bir program çağrılarını icra ettiren bir mekanizma olarak çalışır.

Yukarıda açıkladığımız programlama olay tabanlı programlama diye isimlendirilir. Bu programlamada MATLAB GUI tarafından her bir olaya ait alt programlar birbirinde bağımsız çalıştırılır.

5.2 MATLAB’da GUI Hazırlama Teknikleri

MATLAB GUI hazırlamak için iki farklı yol bulunmaktadır. Bunlardan biri MATLAB GUIDE yardımıyla diğeri ise M-File programlama yöntemiyle hazırlamaktır. Özellikle GUI tasarımına yeni başlayan, hızlı bir şekilde arayüzler tasarlamak isteyenler MATLAB GUIDE yardımıyla kolayca araç kutuları yapabilmektedirler.

Bu yöntem, bileşen nesnelere sürükleyip bırak ilkesine dayanır. Nesnelere için açılan pencerelerde bileşenlerin özelliklerinin değiştirilmesiyle arayüzler hazırlanmaktadır. Bu yöntemle hazırlanan arayüzler, daha sonrasında karşılaşılabilecek düzenlemelere ve değişikliklere izin verir.

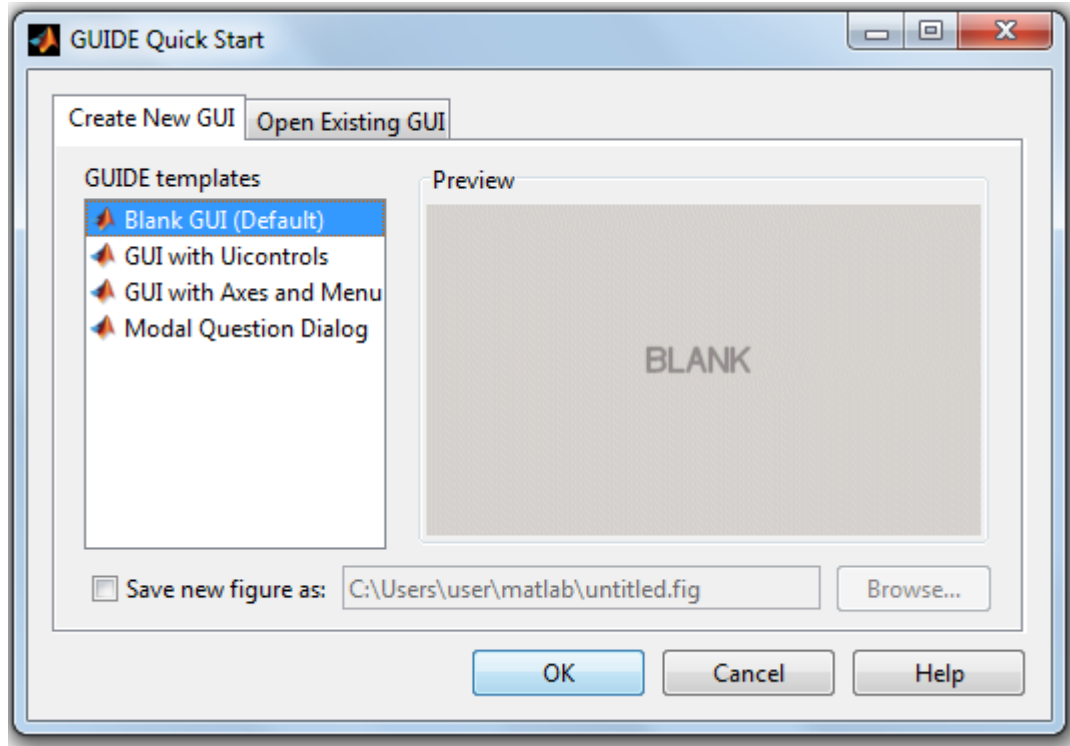
M-File yönteminde ise hazırlanacak arayüzdeki bileşenlerin alt programları tamamı ilgili kod satır yazılarak oluşturulur. Bu yöntemde yazılımcı arayüz alt programlarının tamamına hâkimdir. Ancak uzman programlama yeteneği ve bilgisine sahip olan yazılımcıların kullanacağı bir yöntemdir. Burada programcı her türlü değişikliği kod yazarak yapabildiği için yazılımcılar açısından esneklik sağlayan GUI hazırlama tekniğidir.

5.3 MATLAB GUIDE Yardımıyla GUI Hazırlama

MATLAB GUIDE, grafiksel kullanıcı arabirimi için kullanım kolaylığı sağlayan bir ortamdır. GUIDE ortamında tıkla-sürükleyip bırak tekniği ile grafiksel kullanıcı arabiriminde bulunan butonlar, metin kutuları, grafikler şekillendirilen panellere kolaylıklar ilave edilebilir. Ayrıca, eklenen nesnelere üzerinde birçok görsel ayarlama da

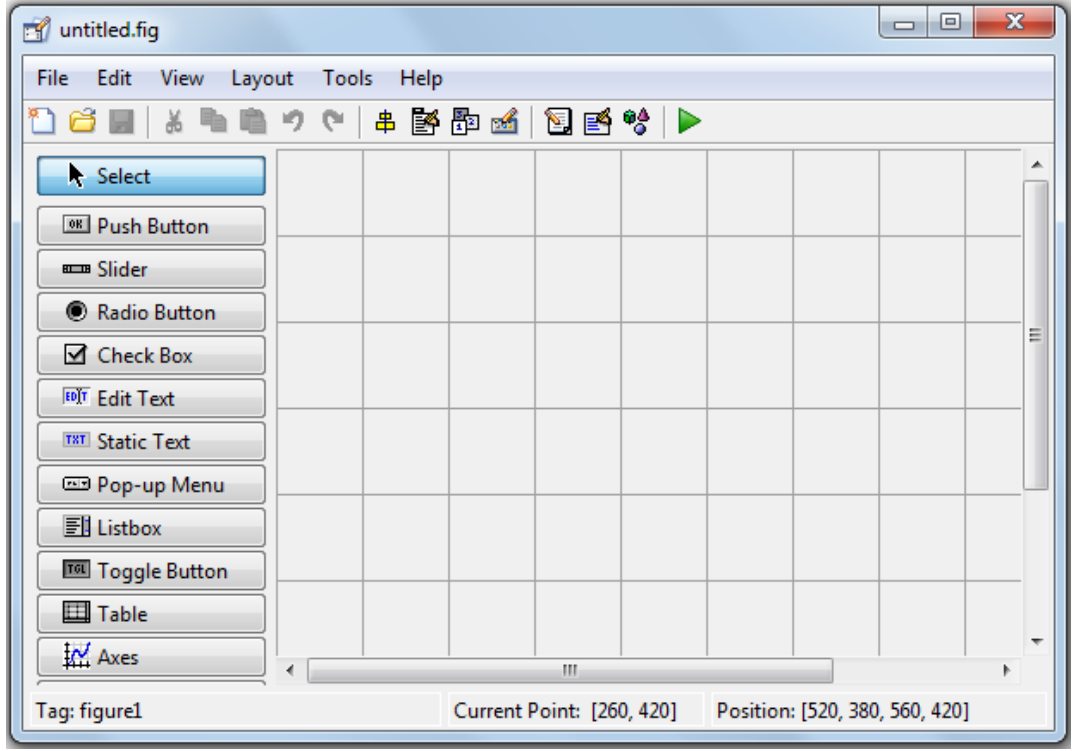
yapılabilmektedir. Ortamın söz konusu özelliği tasarımcılar için sunduğu imkânlardan biridir.

MATLAB GUIDE aracını çalıştırmak için komut satırına guide komutu girilir. Diğer yol ise Start düğmesine tıklanarak MATLAB/GUIDE seçeneği açılır. Şekil 5.1'deki pencere açılır.



Şekil 5.1 GUIDE hızlı başlat penceresi

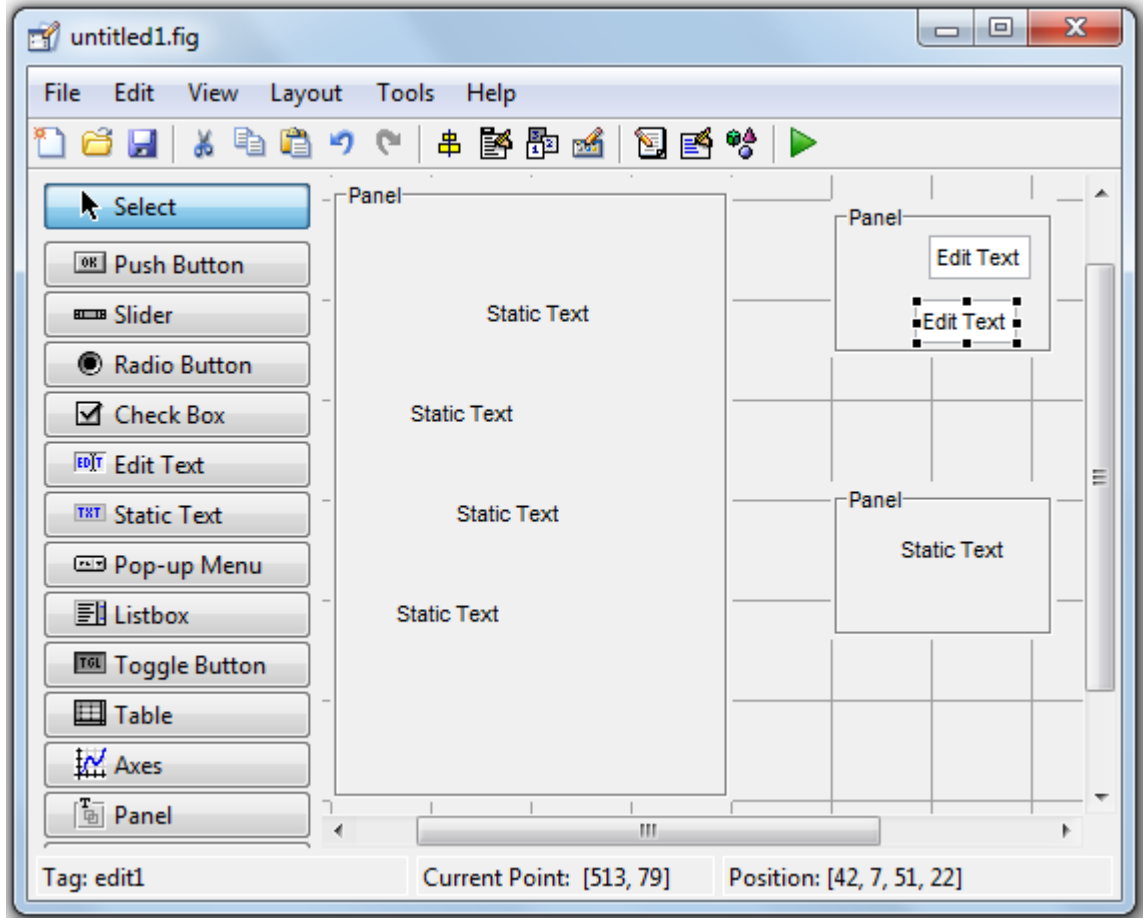
Yeni GUI tasarımı yapılmak isteniyorsa Blank GUI seçeneği tıklanır. Bu adımdan sonra açılan pencere onaylanarak GUI çalışma ortamına ulaşırız. Var olan bir tasarım açılmak isteniyorsa Open Existing GUI alternatifi seçilir.



Şekil 5.2 GUI çalışma ortamı

5.4 GUI Bileşenlerini Çalışma Ortamına Ekleme

MATLAB GUI'de araç kutusu hazırlamak için sol bölümde bulunan bileşenlerin çalışma ortamında eklenmesi gerekmektedir. Bunun için sol sütun üzerindeki istenilen bileşene tıklanır, daha sonra çalışma ortamında istenilen yere tıklanarak ihtiyaç duyulan nesne eklenmiş olur. Diğer bir teknik, istenilen bileşenin üzerine gelinip farelin sol tuşuna basılıp sürüklenerek çalışma ortamına ilave edilebilir. Yukarıda bahsi geçen konuyla ilgili durum şekil 5.3'te sunulmuştur.



Şekil 5.3 GUI çalışma ortamına bileşen ekleme

5.5 GUI Bileşenlerinin Açıklaması

MATLAB GUI'de araç kutusu hazırlamak birçok bileşen mevcuttur. Bu bileşenleri tanımak adına fonksiyonlarından sırasıyla aşağıda bahsedilmiştir. Özelliklerinin yanı sıra nasıl programlanacağına da değinilmiştir.

5.5.1 Push button

Baş çalıştır özelliğın olan bileşendir. Çalışma ortamına eklenen butonun üzerine tıklanmasıyla kullanılır. Söz konusu butonun yapacağı komut ise ilgili callback yazılan koddur.

5.5.2 Toggle button

İki farklı pozisyonda çalışan bir bileşendir. Söz konusu bileşen basılı iken farklı bir alt programı çağırırken, basılı durumda değil iken başka bir işlemin gerçekleştirilmesinde kullanılır.

5.5.3 Radio button

İki fazla durumun olduğu ve bu durumlardan sadece bir tanesinin seçileceği hallerde kullanılan bir bileşendir.

5.5.4 Check box

Kullanıcı tarafına seçim imkânı sağlayan bir bileşendir. Birden fazla seçeneğin işaretlenmesi gereken durumlarda bu nesne kullanılır.

5.5.5 Edit text

Kullanıcı tarafından bilgi girilmesi gereken durumda sıklıkla kullanılan bir bileşendir. Söz konusu bileşene dışarıdan bir değer veya bilgi girişi yapılır.

5.5.6 Static text

Hazırlanan araç kutusu içinde bulunan bileşenlerin açıklanması adına kullanıcıya bilgi veren metin kutusudur. Ayrıca bir sonuç veya değer göstermek amacıyla da kullanılabilir.

5.5.7 Slider

Kullanıcı tarafından veri giriři olduđu durumlarda, girilen deęerin kaydırılarak alınmasına imkân veren bileřendir.

5.5.8 List box

Kullanıcıya bilgi verme kapsamında kullanılacađı gibi mevcut bir liste kutusundan bir deęerin listeden seçmek amacıyla kullanılabilir.

5.5.9 Pop-Up menu

Kullanıcıdan girilmesi istenen bilgilerin açılan bir listeden seçilmesi süretiyle elde edilmesini sađlayan bileřendir.

5.5.9 Axes

Hazırlanan araç kutusu ile ilgili grafik çizimlerinin kullanıcıya sunulması sađlayan bileřendir.

5.5.10 Panel

Tasarımcının yazılım paketinde kullanacađı bileřenlerin GUI alıřma ortamında gruplandırılması ve bir arada düzenli bir řekilde gösterilmesi maksadıyla kullanılan bileřendir. Böylelikle yazılım paketlerinin kullanıcı dostu bir hal alması sađlanır.

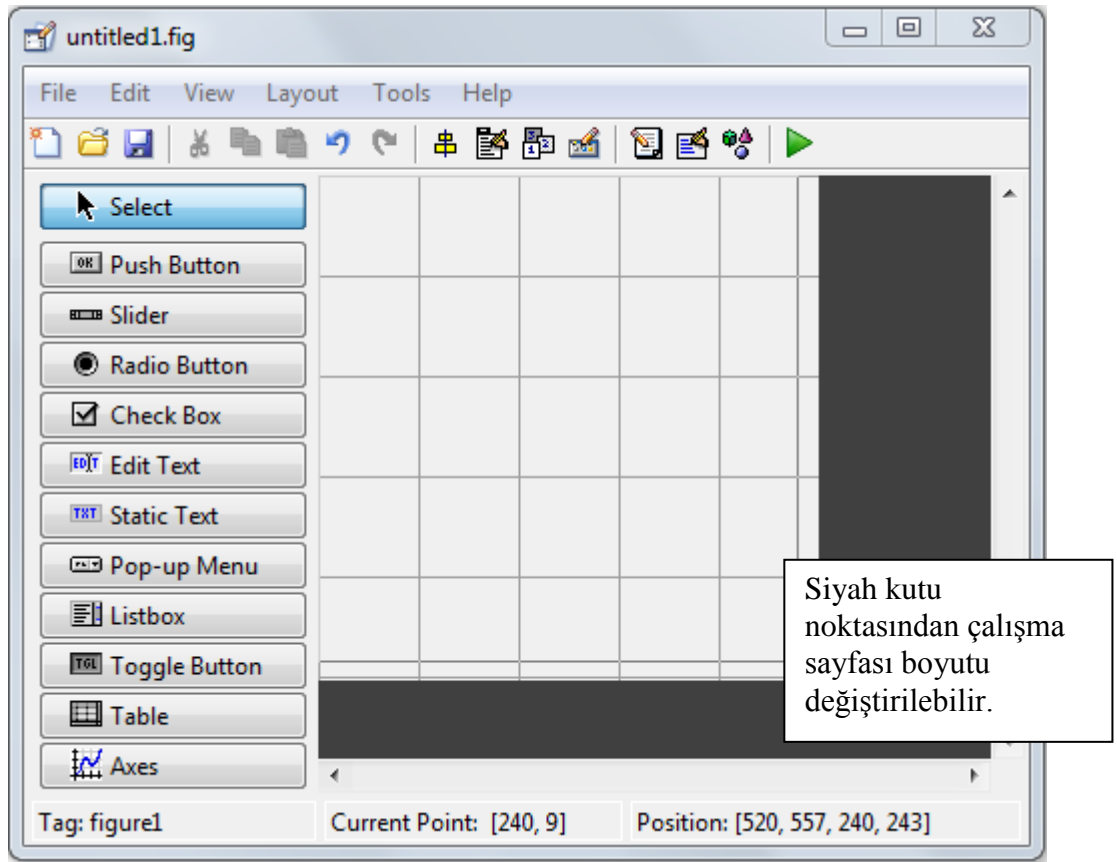
5.5.11 ActiveX compenent

Tasarımların hazırlanmasında sadece yukarıda bahsi geen nesnelere kullanılmaz. Yazılımcılara ayrıca ActiveX adı verilen ve deęiřik alternatifleri olan bileřenlerin

kullanılmasına da imkân verilir. Böylelikle hazırlanacak tasarımlar için GUI arayüzünün kullanımını daha esnek hal almış olur.

5.6 GUI Çalışma Alanının Boyutlarını Değiştirmek

GUI çalışma ortamının sağ alt tarafında bulunan siyah kutucuk üzerine imleç getirilir. Siyah kutucuk üzerindeki imleç sol tuşa basılı tutularak çalışma sayfası istenilen boyuta getirilebilir. Söz konusu ayarlama işlemi ile ilgili gösterim şekil 5.4'te sunulmuştur.

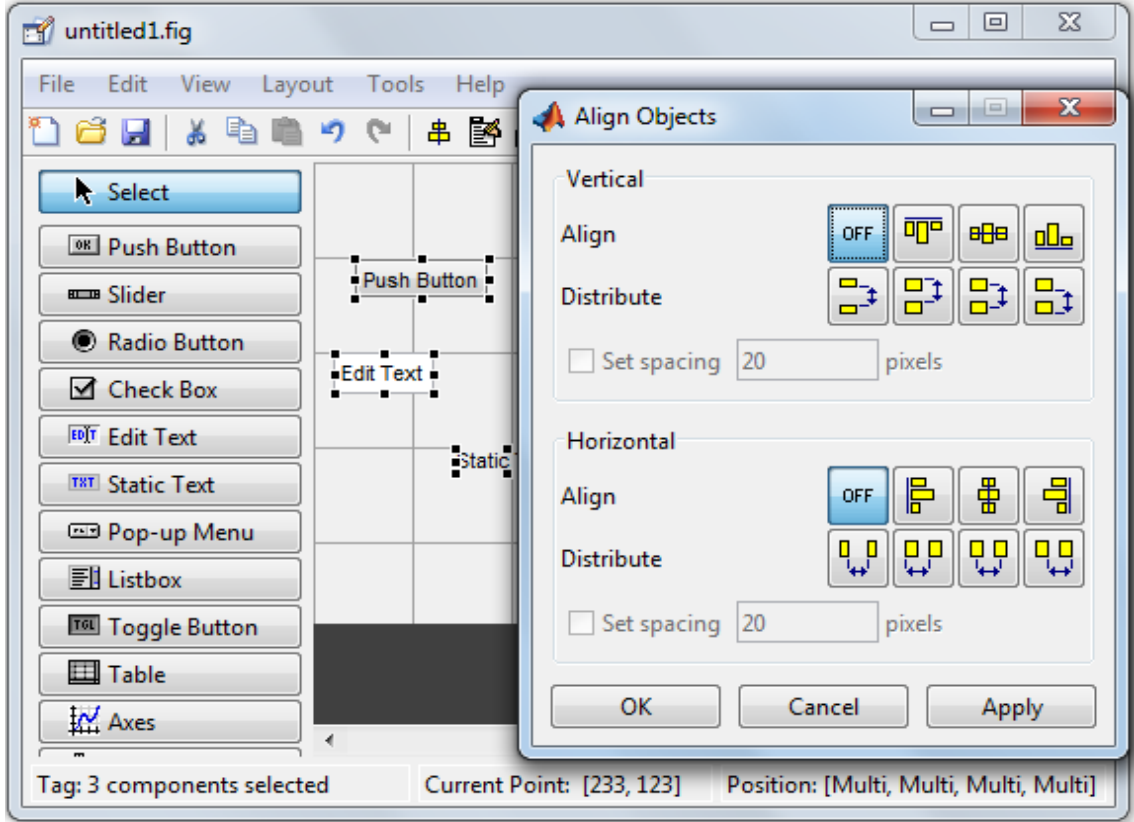


Şekil 5.4 GUI çalışma alanını boyut ayarlaması

5.7 GUI Bileşenlerini Hizalamak

GUI çalışma ortamına eklenen bileşenleri hizalamak için öncelikle nesnelere seçilir. Birden fazla bileşen seçilmek isteniyorsa, sağ tıklanıp oluşan kesikli çizgili kutucuğa bütün hizalanmak istenen nesnelere girilmesi sağlanır. Böylece kutucuk içinde kalan nesnelere

seçilmiş olur. Ayrıca, bileşenleri Ctrl tuşunu basılı tutarak farenin sol tuşu ile birlikte teker teker seçme imkânı da vardır. Hizalanacak nesnelere seçildikten sonra Tools oradan Align Objects tıklanarak hizalama araç kutusu açılır. Şekil 5.5'te hizalama araç kutusu görülmektedir.



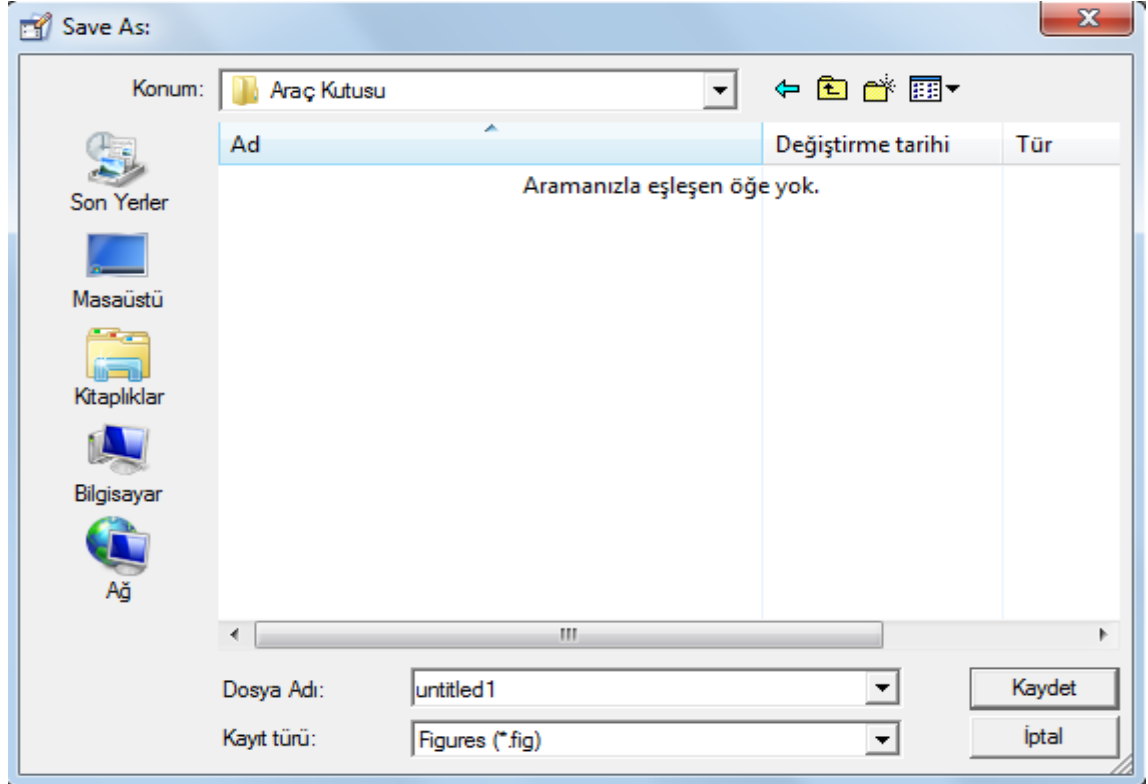
Şekil 5.5 Hizalama araç kutusu

Açılan hizalama araç kutusu kullanılarak, nesnelere yatay ve dikey hizalama kutucuğu seçildikten sonra OK tuşuna basılarak işlem tamamlanır. Hizalama maksadıyla yapılan işlemler ilgili Ctrl+Z kısayoluyla yapılan işlemler geri alınabilir.

5.8 GUI tasarımını kaydetme ve çalıştırma

GUI'de hazırlanan bir tasarımı çalıştırmak için Tools daha sonra Run komutu çalıştırılır. Bu komutla birlikte tasarımın çalıştırılabilmesi için yeni bir pencere açılır. Run edilebilmesi için tasarımın kaydedilmesi gerektiği hakkında uyarı verir.

Açılan pencere onaylanarak tasarımın hangi isimle kaydedileceği uyarısı alınır. Bu uyarı penceresine tasarımın dosya ismi girilerek kaydedilir. Bu kaydedilme işlemi şekil 5.6'da sunulmaktadır.

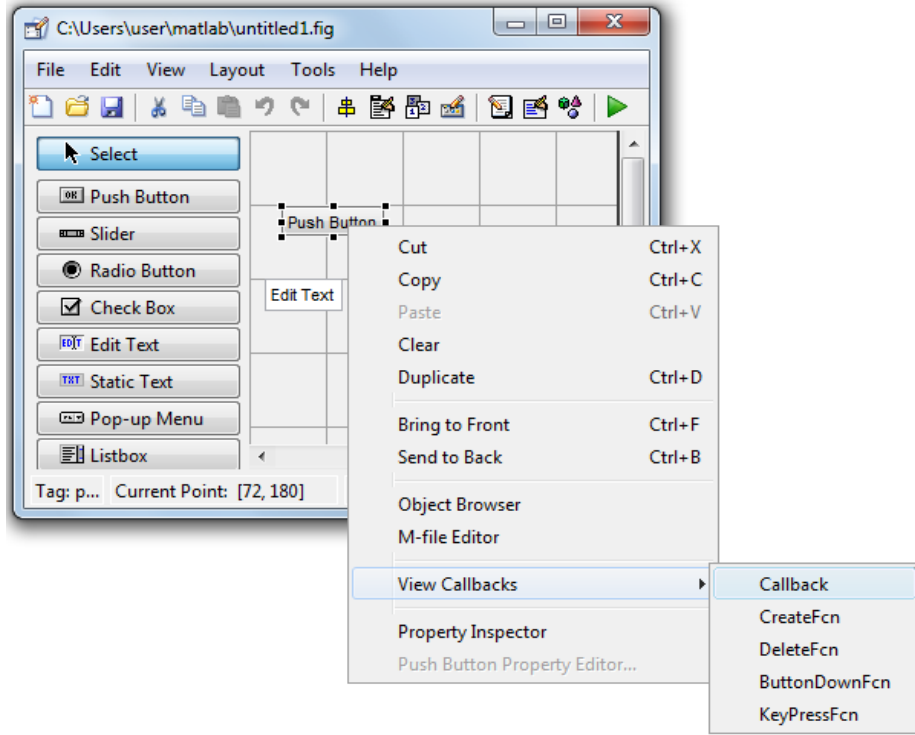


Şekil 5.6 GUI tasarımın kaydedilmesi

Yukarıdaki pencereyi onaylayarak kapatılması durumunda karşımıza Change the MATLAB Directory uyarısı çıkması halinde bu pencereyi Ok tuşuna basarak onaylayabiliriz. Bu onaylama işlemiyle çalıştırılacak dosyanın yolunu bulacak şekilde bir ayar değişikliği yapılmış olur.

5.9 GUI Çalışma Ortamının Programlanması

Bir GUI çalışma ortamının programlanması ne anlama gelmektedir. Hazırlanan GUI tasarımı kaydedildiğinde MATLAB kod satırlarından oluşan .m uzantılı bir dosya oluşur.



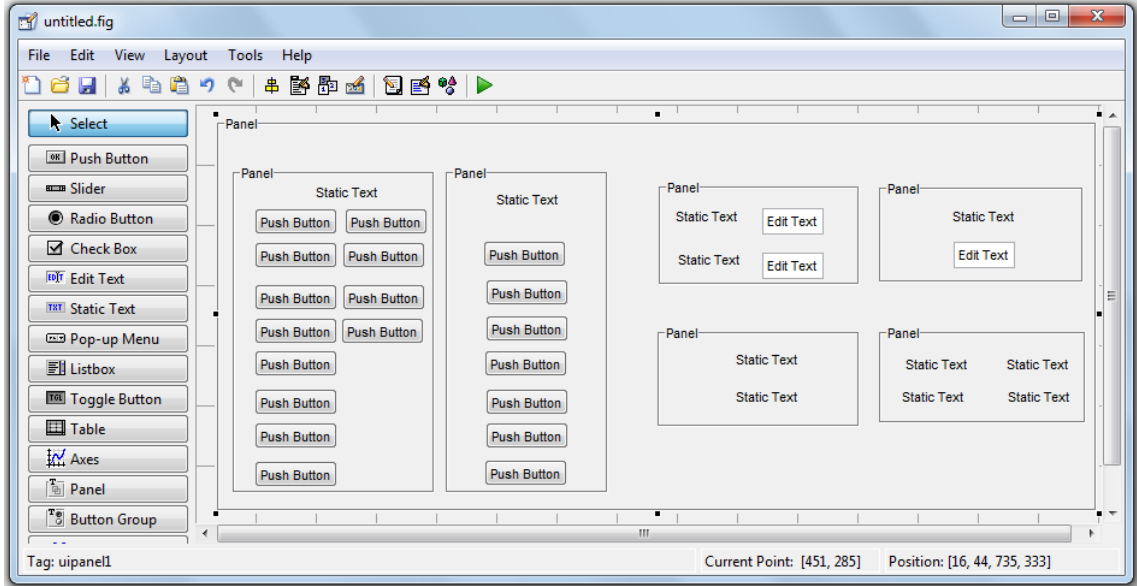
Şekil 5.8 Nesnelerin Callback alt programlarının açılması

5.10 Yazılım Projeleri Parametrik Tahmin Modelleri Yazılım Paketinin MATLAB GUI Ortamında Hazırlanması

Çalışmanın bu bölümünde literatür taraması neticesinde tespit edilen, mevcut parametrik tahmin modelleri denklem setleriyle bir araya getirilmiştir. Yine tez çalışması kapsamında elde edilen yeni modeller denklem setleriyle bir araya toplanmıştır. MATLAB GUI ortamında hazırlanan yazılım paketi yardımıyla bir araç kutusu oluşturulmuştur. Bu yazılım paketinin hazırlanmasındaki amaç, yazılım projeleriyle ilgilenen proje yöneticileri ve yazılımcılara proje başlangıcında yazılımcı iş gücü planlamasında kolaylık sağlamaktır. Bu araç kutusu marifetiyle proje yöneticileri literatürde mevcut parametrik tahmin yöntemleri efor kestirimlerini proje öncesinde görebilmektedir. Ayrıca gerçek yazılım proje verileriyle hata oranı azaltılmış daha iyi tahminlerde bulunan yeni modeller araç kutusuna ilave edilerek mevcut modeller arasında kıyaslama imkânı sağlanmıştır.

5.10.1 Yazılım projeleri parametrik modelleri araç kutusunun bileşenlerinin eklenmesi

Araç kutusu tasarımında bir panel üzerinde on ayrı panel bulunmaktadır. Sol tarafta bulunan bileşenler sütunundan on bir panel GUI çalışma ortamına eklenmiştir. Şekil 5.9’da araç kutusu için gerekli bileşenlerin eklendiği durum gösterilmektedir.

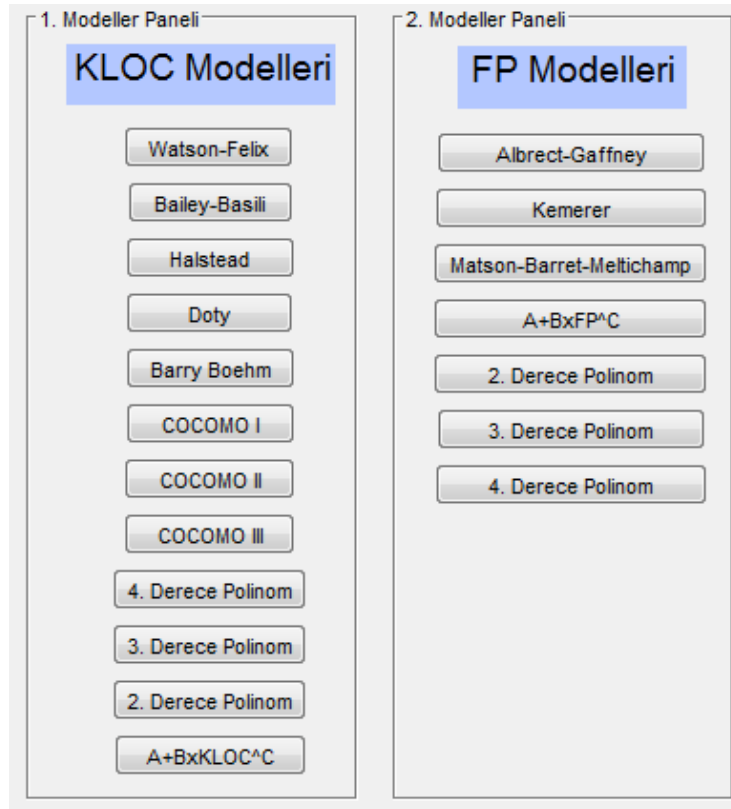


Şekil 5.9 Araç kutusu bileşenlerinin eklenmesi

Araç kutusunda mevcut panellerden birinci ve ikincisi parametrik tahmin modelleri bir araya getirmek amacıyla kullanılmıştır. İlk panel kod satır sayısı tabanlı modeller için tahsis edilmiştir. İkinci panel ise işlev puanı tabanlı modeller için kullanılmıştır.

Kod satır sayısı tabanlı modelleri bir araya getirmek maksadıyla literatürden sekiz, tez kapsamında önerilen dört yeni model olmak üzere toplamda 12 KLOC tabanlı parametrik model aynı panel içine yerleştirilmiştir. Panel içindeki modeller için birer push buton bulunmaktadır. Bu butona basılmak suretiyle her bir modelin denklem setleri oluşturulmakta ve diğer panellerden yapılan giriş verileri kullanılarak gerekli hesaplamalar yapılmaktadır.

Modeller için tahsis edilen ikinci panel işlev puanı tabanlı modelleri için kullanılmıştır. Bu panelde bulunan FP tabanlı parametrik modellerin üç tanesi literatür taramasından, diğer dördü ise tez çalışmalarını sonucu önerilen elde edilmiştir. Araç kutusunun model panelleri kullanılarak, literatürde mevcut parametrik modeller ve önerilen yeni paneller için yazılım proje metrikleri rahatlıkla tahmin edilebilmektedir. Şekil 5.10’da model panellerinin araç kutusundaki görünüşü sunulmuştur.

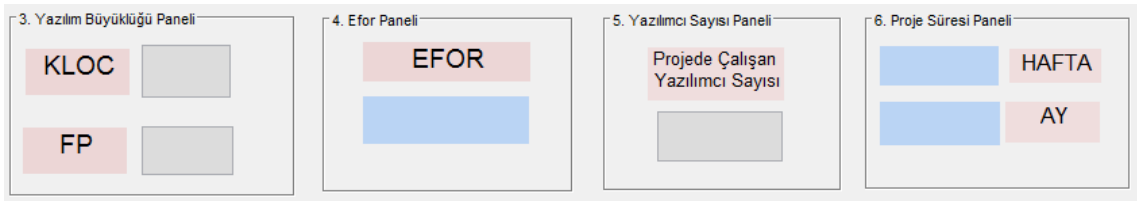


Şekil 5.10 Model panellerinin araç kutusundaki görünüşü

Üçüncü panel kullanıcı tarafından yazılım büyüklüğünün girilmesi amacıyla tasarlanmıştır. Panel içinde iki static text, iki edit text kullanılmıştır. Birinci statik metin kutusu KLOC büyüklüğünü etiketlemek için, edit text kutusunun soluna yerleştirilmiştir. İkinci statik metin kutusu yine benzer şekilde FP yazılım büyüklüğünü etiketlemek amacıyla edit text kutusunun soluna yerleştirilmiştir. Bu paneldeki edit text kutuları kullanıcı tarafından girilen yazılım büyüklük metriklerini tutmaktadır. Yazılım paketine veri girişi bu metin kutuları yardımıyla gerçekleştirilmektedir.

Dördüncü panel her bir model için hesaplanan efor değerlerinin kullanıcıya yansıtılması amacıyla kullanılmıştır. Bu panelde iki statik metin kutusu vardır. Biri efor etiketi olarak diğeri ise parametrik denklem setleri yardımıyla elde edilen verinin kullanılması kapsamında kullanılmıştır.

Beşinci panel projede çalışan yazılımcı sayısı bilgisinin kullanıcı tarafından girilmesi kapsamında hazırlanmıştır. Bu panelde bir edit text ve bir static text kullanılmıştır. Edit text kutusu yazılımcı sayısı bilgisi saklamak ve gerektiğinde kullanmak için panele yerleştirilmiştir.



Şekil 5.11 Üç, dört, beş ve altı numaralı panellerinin araç kutusundaki görünüşü

Altıncı panel ise yine araç kutusunu kullanacak yazılım proje yöneticileri için hazırlanmıştır. Bu panelde dört statik metin kutusu kullanılmıştır. İşlev puan tabanlı modellerde efor tahmininin birimi adam hafta, kod satır sayısı tabanlı modellerde ise adam aydır. Bu nedenle bu panelde proje süreleri hafta ve ay olarak zaman dilimine ayrılmıştır. İki metin kutusu hesaplanan zaman değerlerini hafta veya ay olarak göstermektedir. Diğer iki metin kutusu ise bu değerleri etiketlemek için kullanılmıştır.

5.10.1.1 Yazılım projeleri gelişim süreci

Yazılım ile ilgili yapılan tanımlara bakıldığında, bilinenin aksine yazılım sadece kod yazıp, veri tabanı ile ilişkilendirilmesi değildir. Yazılım geliştirme süreci, kod yazmaktan çok daha fazlasıdır. Bu kapsamda yazılım geliştirme, hem yazılım hazırlandığı hem de kullanım süresi boyunca geçirdiği tüm aşamaları içeren bir süreçtir (Kartın 2008).

Yazılımın fonksiyonları ile ilgili gereksinimler devamlı bir güncellemenin paralelinde gelişip değiştiği için, yazılım geliştirme aşamaları da sürekli bir döngü şeklinde ele alınmaktadır. Böylelikle döngü içerisinde herhangi bir aşamada önceki aşamalara dönüş yapma imkânı mevcuttur. Literatürde yazılım projesi geliştirme sürecinde birçok farklı modelden bahsetmek mümkündür. Ancak yazılım mühendisliğinde kullanılan diğer modellere de dayanak olan “Çağlayan Modeli (Waterfall Model)” yazılım yaşam döngüsünü analiz, tasarım, kodlama, test ve bakım olarak beş ayrı aşamada incelemektedir.

Analiz aşamasında geliştirilecek yazılımın ne maksatla kullanılacağı belirlenir. Sorun bu aşamada tanımlanır. Yazılım projesinde kod yazılmasına geçilmeden önce yazılımın doğru şekilde tanımlanması gerekmektedir. Gereksinimin ne olduğu bu aşama da açıkça ortaya konur. Bu aşamada personel, donanım ve sistem gereksinimleri belirlenir, sistem fizibilite çalışması yapılır, kullanıcı gereksinimleri analiz edilir. Bu adımlar kapsamında proje planı oluşturulması da bu aşama hazırlanır.

Tasarım aşamasında ise analiz sonucunda belirlenen gereksinimlere çözüm olabilecek yazılımın temel yapısının oluşturulur. Bu aşamada yazılım bileşenleri arasındaki arayüzler, mimari tasarım, veri tasarımı, kullanıcı ara yüzü tasarımı süreçlerini kapsamaktadır.

Kodlama aşamasında, tasarım süreci ele edilen bilgiler ışığında yazılım gerçekleştirilir. Bu süreç, yazılım geliştirilmesi ve kullanıcıya ulaştırılması boyunca devam eder. Yazılım geliştirme ortamı, programlama dili, veri tabanı yönetim sistemi, yazılım geliştirme araçları seçimi kodlama aşamasında gerçekleştirilir.

Test aşaması, yazılım kodunu hazırlanmasına müteakip gerçekleştirilen kontrol ve doğrulama adımları içerir. Hazırlanan uygulama yazılımı analiz aşamasında tespit edilen gereksinimleri karşılayıp karşılamadığı bu aşamada test edilir. Yazılım geliştirilmesinde ilk testler genelde programcı tarafından yapılır. Bununla birlikte, asıl hata ayıklama ve geribildirim hizmeti test ekipleri tarafından gerçekleştirilir.

Test aşamasında sonra süreçte gerçekleştirilen adım bakım aşamasıdır. Yazılım tesliminde sonra hata giderme ve yeni eklentileri kapsayacak şekilde güncellemeler yapılır. Yazılım kullanıma başlamasında sonra yazılımın desteklemesi sürecini kapsar. Yazılımın eksiklerinin giderilmesi, iyileştirilmesi gibi alt başlıkları içeren son aşamadır.

5.10.1.2 Yazılım proje takımının yapısı ve büyüklüğü

Uluslararası Yazılım Kıyaslama Standart Grup (ISBSG International Software Benchmarking Standart Group) verileri doğrultusunda, yazılım projelerinde görev alan personel incelenmiştir. Bu konuda önemli olan parametreler, proje takımının yapısı ve büyüklüğü ile proje ekibinde bulunan personelin faaliyetlerdeki kişisel rolleridir. Söz konusu parametreler birçok faktöre bağlıdır. Bu faktörler, proje büyüklüğü, proje tipi, proje risk faktörü, organizasyon yapısı ve proje geliştirme süreci şeklinde sıralanabilir. Yazılım projelerinde rol alan personel sayısı projeden projeye farklılık göstermektedir. Bu kişilerin proje üstlendikleri roller ve rollere göre ücretlerde farklılık göstermektedir.

Bir yazılım projesi için toplam maliyeti tahmin edebilmek için ilk olarak proje ekip yapısını ve bu ekte bulunan personelin proje geliştirme sürecinde ne kadar rol üstleneceklerini bilmek gerekir.

Yazılım projelerinde harcanan iş gücü hesaplanmasında proje tamamlanma oranından büyük ölçüde faydalanılır. Doğru proje maliyet tahmini yapılabilmesi için proje ekibinde çalışanların hangi rolde ne kadar emek harcadığı çok önemlidir.

Uluslararası Yazılım Kıyaslama Standart Grup (ISBSG) verileri incelendiğinde yeni geliştirilen çeşitli yazılım projelerinde rol alan ekip çalışanlarının emek yüzdeleri yapılan yazılımlara göre farklılıklar arz etmektedir. Bu oranlar farklılık arz ettiği için yazılım paketinde kullanılacak emek yüzdelerini tespit etmek maksadıyla bir sonraki bölümde Proje Değerlendirme ve Tekrar Gözden Geçirme Tekniği (Project Evaluation and Review Technique PERT) yaklaşımı kullanılmıştır. PERT yaklaşımında üç tahmin zaman vardır. Bunlar iyimser, kötümser ve en büyük olasılıklı proje tamamlama zamanlarıdır. Her bir yazılım proje aşaması için efor oranlarının en büyük olasılıklı

proje tamamlama zamanı olarak alınmıştır. PERT yaklaşımıyla proje aşamaları için iş gücü tahminleri örnek bir uygulamayla tespit edilmiştir.

5.10.1.3 Yazılım proje takviminin PERT (Project Evaluation and Review Technique) kullanarak belirlenmesi

Proje Değerlendirme ve Gözden Geçirme Tekniği (PERT - Project Evaluation and Review Technique) ABD hükümetinin Polaris Balistik Füze Projesi ihtiyaçlarını karşılamak maksadıyla 1957 yılında geliştirilmiştir. PERT zaman tahminleri için bir olasılık yaklaşımı kullanır. PERT, karakteristiği belirsiz olan projeler için geçerliliği olan bir tekniktir. Örneğin detaylı bir yazılım projesinin geliştirilme sürecinde bulunan faaliyetleri belirlemek için kullanılabilir. Yazılım projelerinin ortak özelliği olan belirsizlikler PERT ile hesaba katılır. PERT, yazılım projelerindeki belirsizlikleri dikkate alırken ağ yapı kullanarak proje takvimini hazırlar.

PERT'te zaman tahmininde kullanılmak maksadıyla proje teslim süresi için üç ayrı zaman tanımlanmıştır. Bunlar; iyimser zaman, kötümser zaman ve en büyük olasılıklı zaman şeklindedir. İyimser zaman, her şeyin en iyi şekilde devam ettiği projenin tamamlandığı en kısa zaman dilimidir. Kötümser zaman ise, proje boyunca herşey kötü devam etmiş, hatalar üst üste gelmiştir. Hal böyle olunca ki proje tamamlanma zamanı kötümser zamandır. En büyük olasılık zaman ise proje boyunca herşey normal devam ettiği, projenin tamamlandığı normal zaman dilimidir. Bu üç tahmin edilen zaman diliminden proje tamamlanması için geçen süre tahmin edilir. Türetilen bu süreyi hesaplamak için aşağıda sunulan denklemden yararlanılır.

$$T_e = (T_o + 4T_m + T_p) / 6 \quad (5.1)$$

T_e Tahmini zaman,

T_o İyimser zaman,

T_p Kötümser zaman,

T_m ise projenin tamamlanması için harcanan en büyük olasılıklı zamandır.

Yukarıda sunulan denkleme göre her bir zaman parametresinin ağırlıkları mevcuttur. Yazılım projesi takvimi için kullanılan ağırlıklar, iyimser için 1, en büyük olasılıklı zaman için 4 ve kötümser için yine 1 alınmıştır. Çizelge 5.1’de yazılım projeleriyle ilgili görevler için iyimser, kötümser ve en büyük olasılıklı tahminler sunulmuştur.

Çizelge 5.1 Yazılım projesi iş gücü dağılımı için zamanlama tahminleri

Görev	İyimser Zaman Tahmini (Adam Gün)	En Büyük Olasılıklı Zaman Tahmini (Adam Gün)	Kötümser Zaman Tahmini (Adam Gün)
Gereksinim Analizi	5	7	9
Prototip	1	2	3
Mimari ve Proje Planlama	1	2	3
Tasarım Aşaması	11	14	17
Kodlama Aşaması	17	21	25
Konfigürasyon Yönetimi	1	3	5
Kullanıcı Dokümantasyonu	8	10	12
Birim Testleri	2	3	4
Fonksiyon Testleri	3	5	7
Entegrasyon Testleri	1	5	9
Sistem Testleri	5	6	7
Alan Testleri	2	3	4
Kabul Testleri	3	4	5
Kurulum ve Kullanıcı Eğitimi	1	2	3
Proje Yönetimi	11	13	15

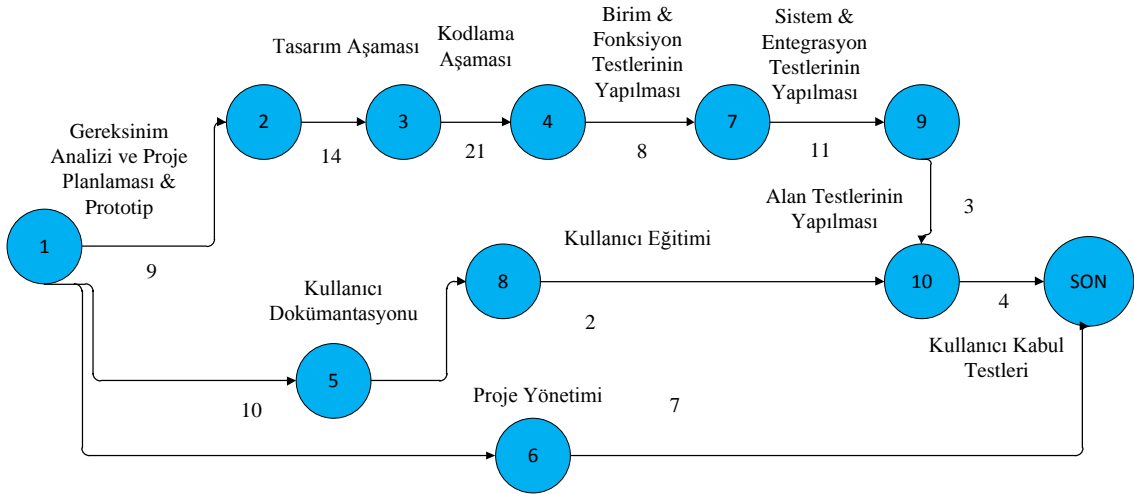
Her bir görev için üç ayrı zaman parametresi denklem seti kullanılarak tahmini zamanlar hesaplanabilir.

Gereksinim Analizi : $T_e = [5 + 4x(7) + 9] / 6 = 7$ adam gün

Prototip: $T_e = [1 + 4x(2) + 3] / 6 = 2$ adam gün

Mimari ve Proje Planlama: $T_e = [1 + 4x(2) + 3] / 6 = 2$ adam gün

Tasarım Aşaması: $Te = [11 + 4x(14) + 17] / 6 = 14$ adam gün
Kodlama Aşaması: $Te = [17 + 4x(21) + 25] / 6 = 21$ adam gün
Konfigürasyon Yönetimi: $Te = [1 + 4x(3) + 5] / 6 = 3$ adam gün
Kullanıcı Dokümantasyonu: $Te = [8 + 4x(10) + 12] / 6 = 10$ adam gün
Birim Testlerinin Yapılması: $Te = [2 + 4x(3) + 4] / 6 = 3$ adam gün
Fonksiyon Testleri: $Te = [3 + 4x(5) + 7] / 6 = 5$ adam gün
Entegrasyon Testleri: $Te = [1 + 4x(5) + 9] / 6 = 5$ adam gün
Sistem Testlerinin Yapılması: $Te = [5 + 4x(6) + 7] / 6 = 6$ adam gün
Alan Testlerinin Yapılması: $Te = [2 + 4x(3) + 4] / 6 = 3$ adam gün
Kabul Testlerinin Yapılması: $Te = [3 + 4x(4) + 5] / 6 = 4$ adam gün
Kurulum ve Kullanıcı Eğitimi: $Te = [1 + 4x(2) + 3] / 6 = 2$ adam gün
Proje Yönetimi: $Te = [11 + 4x(13) + 15] / 6 = 13$ adam gün



Şekil 5.12 Yazılım proje takvimi PERT ağ yapısı

Yazılım projelerinin gereksinim analizinden kabul testlerine kadar olan her aşamasını içerecek şekilde PERT tekniği tespit edilen en büyük olasılıklı değerleri, yazılım paketinin proje iş gücü dağıtım panelinin yüzde default değerleri olarak kullanılmıştır. Her bir modelin denklem seti yardımıyla hesaplanan efor değerleri bu yüzde default değerler oranında söz konusu paneldeki ilgili boşluklara atanır. Birim adam ay olarak ayarlanmıştır. Şekil 5.13 yazılım paketinin iş gücü dağıtım paneli sunulmuştur.

7. Proje İş Gücü Dağılımı Paneli

	ADAM AY		ADAM AY		ADAM AY
Gereksinim Analizi		Birim Testlerinin Yapılması		Konfigürasyon Yönetimi	
Prototip		Fonksiyon Testlerinin Yapılması		Kabul Testlerinin Yapılması	
Mimari ve ve Proje Planlama		Entegrasyon Testlerinin Yapılması		Kullanıcı Dokümantasyonu	
Tasarım Aşaması		Sistem Testlerinin Yapılması		Kurulum ve Kullanıcı Eğitimi	
Kodlama Aşaması		Alan Testlerinin Yapılması		Proje Yönetimi	

Şekil 5.13 Yazılım paketi iş gücü dağılım paneli

Ayrıca araç kutusuna, proje iş gücü dağılım yüzdelerinin manuel olarak da girilebilmesi için bir panel ilave edilmiştir. Kullanıcı bu paneli kullanarak iş gücü dağılımını istenilen oranlarda girebilmektedir. Panel şekil 5.14’te sunulmuştur.

8. Proje İş Gücü Dağılımı Yüzde Giriş Paneli

	YÜZDE		YÜZDE		YÜZDE
Gereksinim Analizi		Birim Testlerinin Yapılması		Konfigürasyon Yönetimi	
Prototip		Fonksiyon Testlerinin Yapılması		Kabul Testlerinin Yapılması	
Mimari ve ve Proje Planlama		Entegrasyon Testlerinin Yapılması		Kullanıcı Dokümantasyonu	
Tasarım Aşaması		Sistem Testlerinin Yapılması		Kurulum ve Kullanıcı Eğitimi	
Kodlama Aşaması		Alan Testlerinin Yapılması		Proje Yönetimi	

Şekil 5.14 Yazılım paketi iş gücü dağılımı yüzde giriş paneli

Yine araç kutusunda bulunan bütçe paneli yardımıyla proje maliyeti herbir model için hesaplanabilmektedir. Proje çalışanlarının varsayılan saat ücretleri İngiltere hükümeti tarafından Vasıflı Çalışanlar için Uygulama Kodları ve Standart Meslek Sınıflandırma Kodları dokümanından alınmıştır. Ayrıca çalışanların saat ücretleri panel on yardımıyla ellede girilebilmektedir. Proje kapsamında çalışan proje yöneticisi, program analisti,

program geliştiricisi, yazılım tasarımcısı ve test edenler için saat ücretleri ayrı ayrı girilebilmektedir.

5.10.2 Yazılım projeleri parametrik modelleri araç kutusunun bileşenlerinin düzenlenmesi

Araç kutusu bileşenlerin GUI çalışma ortamına eklendikten sonra bir takım düzenlemeler yapmak gerekmektedir. Bu kapsamda altı panel için yeni adlandırma yapılmıştır. Şekil 5.15’de sunulduğu üzere, birinci panel KLOC tabanlı modeller için tahsis edilmiştir. Bu kapsamda literatürden Waltson-Felix, Bailey-Basili, Halstead, Doty, Barry Boehm ve COCOMO I-II-III parametrik modelleri ile tez çalışmaları sonucunda elde edilen süper metrik, ikinci dereceden polinom, üçüncü dereceden polinom ve dördüncü dereceden polinom yazılım projeleri tahmin modelleri söz konusu panele eklenmiştir.

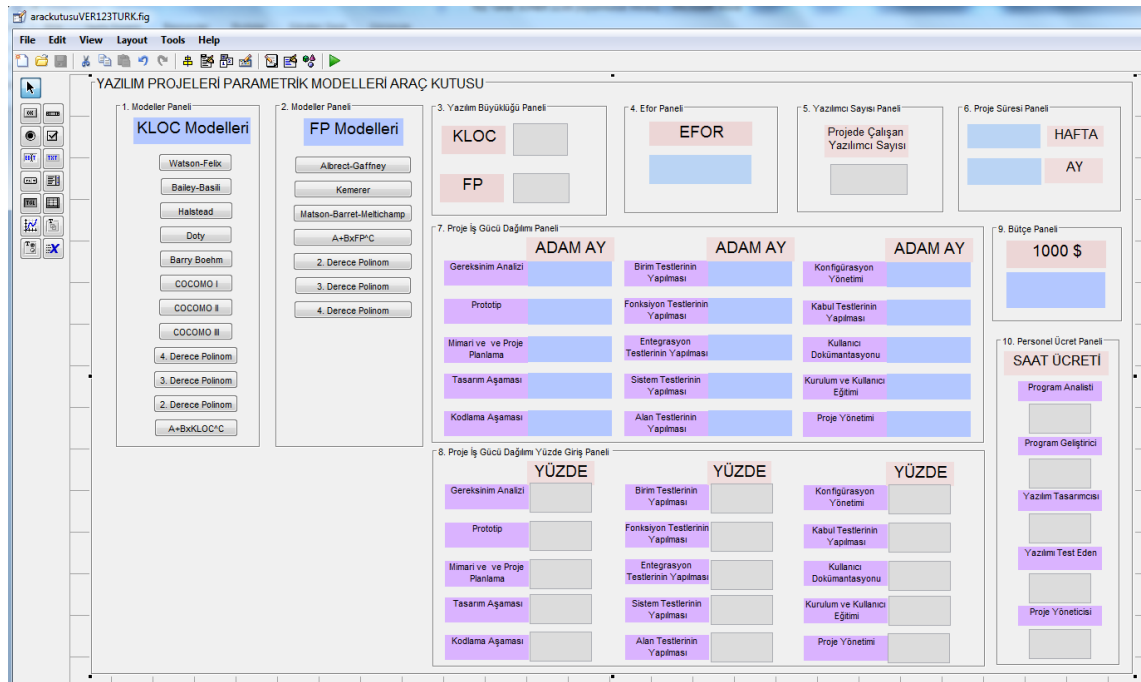
İkinci panel FP tabanlı tahmin modeller için tahsis edilmiş, literatürden tespit edilen Albrect-Gaffney, Kemerer ve Matson-Barret-Metchamp modelleri bu panel ilave edilmiştir. Ayrıca yeni önerilen modeller olarak süper metrik modeli, ikinci dereceden polinom, üçüncü dereceden polinom ve dördüncü dereceden polinom yazılım projeleri tahmin modelleri söz konusu panele eklenmiştir.

Üçüncü panel yazılım büyüklüğü paneli olarak tahsis edilmiştir. Bu panel kullanıcılar tarafında yazılım metrik bilgilerinin araç kutusuna girilmesi amacıyla kullanılmaktadır. Modellerin denklem setlerinde kullanılmak üzere iki farklı yazılım büyüklüğü metriği girişine imkân verecek şekilde tasarlanmıştır. Bu metrikler işlev puanı ve kod satır sayısıdır.

Dördüncü panel, işlev puanı ve kod satır sayısı tabanlı tahmin modellerinin tahmin ettiği efor değerini KLOC modelleri için adam-ay FP modelleri için adam-hafta olarak şekilde kullanıcılara sunmak amacıyla araç kutusuna eklenmiştir.

Beşinci panel, yazılım projesinde çalışan yazılımcı bilgisinin girilmesi amacıyla araç kutusuna eklenmiştir. Kullanıcı proje kapsamında çalışan yazılımcı sayısını bu panel yardımıyla girebilmektedir.

Altıncı panel ise, işlev puanı ve kod satır sayısı tabanlı tahmin modellerinin tahmin ettiği efor değerini kullanarak proje süresinin ay ve haftalık zaman dilimleri haline kullanıcıya sunmak amacıyla tasarıma ilave edilmiştir. Burada araç kutusunu kullanırken kullanım kolaylığı sağlamak hedeflenmiştir.



Şekil 5.15 Yazılım projeleri parametrik modelleri araç kutusu GUI çalışma sayfası

Yedinci ve sekizinci paneller proje iş gücü dağılımını belirlemek amacıyla yazılım paketine ilave edilen panellerdir. Bu paneller yardımıyla yazılım proje yöneticisi iş gücü dağılımına göre ne kadar emek gerektiğini kolaylıkla tespit edebilmektedir.

Son iki panel ise maliyet tahminin yapıldığı panellerdir. Yazılım projesinde çalışanların saat ücretleri girilerek, maliyet tahmin yapılabilir. Saat ücretler girilmeden de default değerlere göre proje bütçesi tahmin edilebilir.

5.10.3 Yazılım projeleri parametrik tahmin modelleri araç kutusu nesnelerin fonksiyonlarının düzenlenmesi

Yazılım projeleri araç kutusu için eklenen her bir nesnenin önceki bölümden bahsetmiştik. Her bir nesnenin araç kutusunda yapacağı işe göre fonksiyonlarına bir takım kod eklemelerinin yapılması gerekmektedir. Bölümde hangi nesne için nasıl bir kod eklemesi oldu, bu konu hakkında bilgi verilecektir. Buradaki mantık kullanıcı tarafından panel 3'e yazılım büyüklük bilgisi girilmektedir. Bu bilgilerin her bir tahmin modelin denklem setine yerleştirip, işlem yaptırılması ve elde edilen efor tahmininin panel 5'e yazdırılmasıdır. Daha sonra ki adımda ise kullanıcı tarafından girilen projede çalışan yazılımcı sayısı bilgisinin tahmin edilen eforla ilişkilendirilip panel 6'ya hafta veya ay olarak yazdırılmaktadır. Bu kapsamda nesnelerin fonksiyonlarında yapılan kod düzenlemelerinin hepsi bir sonraki bölüme eklenen MATLAB kodunda sunulmuştur.

5.10.4 Yazılım projeleri parametrik tahmin modelleri yazılım paketi

MATLAB GUI kullanılarak hazırlanan yazılım projeleri için parametrik tahmin modelleri yazılım paketinin son hali şekil 5.16'de sunulmuştur. Yazılım paketi kapsamında oluşturulan on ayrı panelin kullanıma hazır hali şekilde mevcuttur.

YAZILIM PROJELERİ PARAMETRIK MODELLERİ ARAÇ KUTUSU

1. Modeller Paneli

2. Modeller Paneli

3. Yazılım Büyüklüğü Paneli

4. Efor Paneli

5. Yazılımcı Sayısı Paneli

6. Proje Süresi Paneli

7. Proje İş Gücü Dağılımı Paneli

8. Proje İş Gücü Dağılımı Yüzde Giriş Paneli

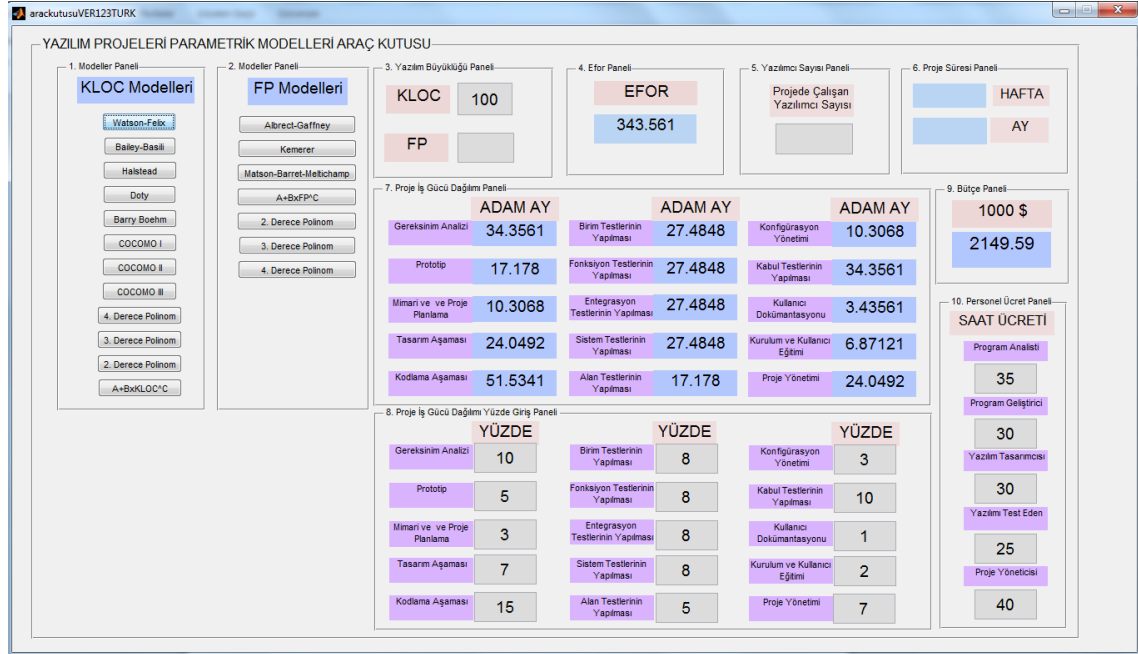
9. Bütçe Paneli

10. Personel Ücret Paneli

Şekil 5.16 Yazılım projeleri parametrik modelleri araç kutusu

5.10.5 Yazılım projeleri parametrik tahmin modelleri araç kutusu simülasyonu

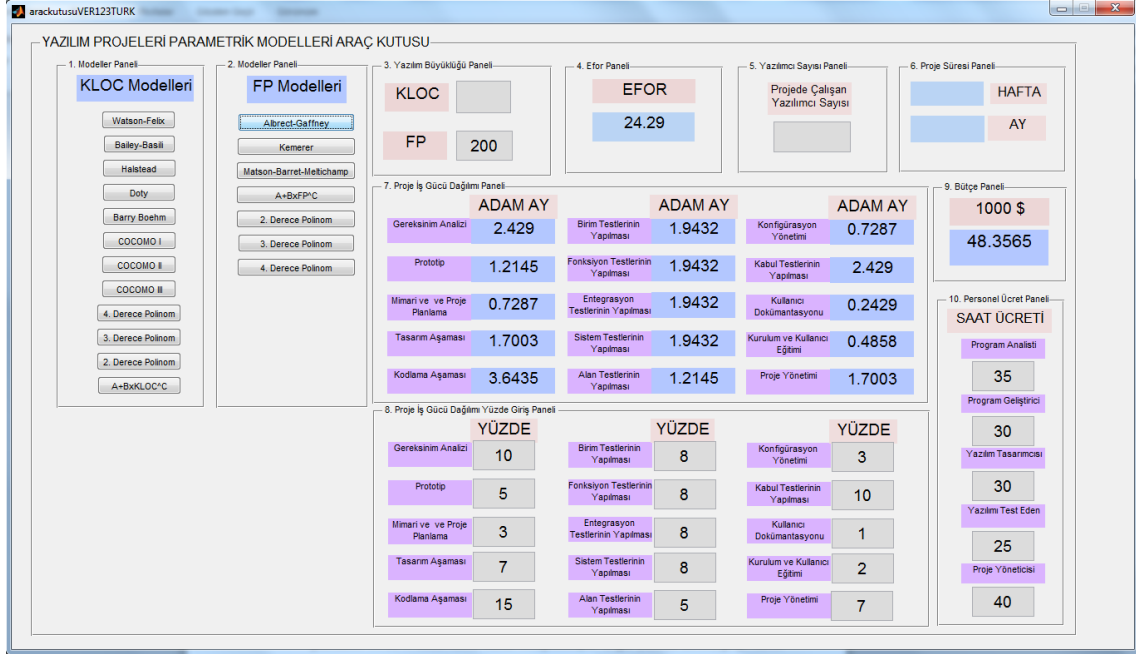
Yazılım projeleri parametrik tahmin modelleri için hazırlanan yazılım paketinde gerekli kod ilaveleri yapıldıktan sonra kaydedilmiştir. Araç kutusu aktif etmek için run figure komutuna farenin sol tuşuyla tıklanır. KLOC tabanlı bir tahmin yapılmak isteniyorsa, yazılım büyüklüğü paneline hazırlanacak yazılımın kod satır sayısı girilmelidir. Ayrıca projede çalışan yazılımcı sayısı bilgisi döndürücü panele girilebilir. KLOC tabanlı yazılım projeleri için örnek uygulama Şekil 5.16’de sunulmuştur. Bu uygulamada kullanıcı tarafından yazılım büyüklüğü olarak 100 KLOC değeri girilmiştir. KLOC modellerinde Watson-Felix model tuşuna sol tıklanarak yazılım iş gücü değeri efor panelinde 343.561 adam-ay olarak hesaplanmıştır. Diğer yandan, proje iş gücü dağılımları elle sekizinci panele girilerek, efor girilen değerler oranında yedinci panel aktarılmıştır. Ayrıca proje kapsamında çalışanların saat ücretleri onuncu panele girilerek, yazılım projesinin maliyeti tahmin edilmiştir. Bu uygulama her bir model ayrı ayrı tekrar edilebilir.



Şekil 5.17 KLOC tabanlı yazılım projeleri için uygulama yapılması

FP tabanlı yazılım projeleri için örnek uygulama şekil 5.17’de sunulmuştur. Bu uygulamada kullanıcı tarafından yazılım büyüklüğü olarak 200 FP değeri girilmiştir. FP

modellerinde Albrecht-Gaffney modeli tuşuna sol tıklanarak yazılım iş gücü değeri efor panelinde 24.29 adam-hafta olarak hesaplanmıştır.



Şekil 5.18 FP Tabanlı yazılım projeleri için uygulama yapılması

Diğer yandan, proje iş gücü dağılımları elle sekizinci panele girilerek, efor girilen değerler oranında yedinci panel aktarılmıştır. Ayrıca proje kapsamında çalışanların saat ücretleri onuncu panele girilerek, yazılım projesinin maliyeti tahmin edilmiştir. Bu uygulama her bir model ayrı ayrı tekrar edilebilir.

6. SONUÇ

Bu çalışma, üç ana kısımdan oluşmaktadır. Birinci bölüm, literatürde mevcut kod satır sayısı ve işlev puanı tabanlı parametrik tahmin modelleri detaylı olarak incelenmesi ve yazılım projeleri metriklerini kolaylıkla hesaplayabilmek amacıyla tahmin modellerinin denklemleri bir araya getirilmesi aşamalarından oluşmaktadır.

İkinci kısımda gerçekleştirilen yazılım projeleri veri setleri kullanarak nümerik yöntemlerle yeni modeller önerilmiştir. Önerilen yeni modellerle elde edilen denklem setlerine gerçek yazılım büyüklük metrikleri kullanarak iş gücü ve proje süresi hesaplamaları yapılmıştır. İkinci kısım kapsamında hata ölçütleri kullanılarak literatürde mevcut parametrik tahmin modelleri ile önerilen modellerin performans kıyaslaması yapılmıştır.

Üçüncü kısımda ise MATLAB GUI ortamında hazırlanan yazılım paketi yardımıyla bir yazılım projeleri parametrik tahmin modelleri araç kutusu oluşturulmuştur. Mevcut ve önerilen tahmin modelleri bu araç kutusunda kullanıma sunulmuştur.

Son 30 yılda, genel olarak kabul görmüş olan birçok yazılım maliyet modeli mevcuttur. Kod satır sayısı tabanlı mevcut parametrik modeller; Bailey-Basili, Waltson-Felix, Halstead, Doty, Barry Boehm şeklinde sıralanabilir. Yine işlev puanı temelli parametrik modelleri; Albrecht-Gaffney, Kemerer, Matson-Barret-Meltichamp olarak gruplandırılabilir. Ancak bu modeller, aynı metrikleri (örneğin yazılım kod satır sayısı) temel alsalar bile, farklı efor ve proje süresi (ve dolayısıyla geliştirme maliyeti) tahminlerinde bulunmaktadır.

Bu çalışma kapsamında modeller uyarınca tahmin edilen yazılım maliyetlerinin kolaylıkla kıyaslama yapılabilmesi amaçlanmıştır. Mevcut KLOC ve FP tabanlı parametrik modellerin yanı sıra, yeni modeller önerilmesi amacıyla NASA'da gerçekleştirilen yazılım projeleri başta olmak üzere yazılım veri setlerindeki gerçek yazılım büyüklüğü ve işgücü değerleri kullanılarak nümerik yöntemlerin temel alındığı

MATLAB araç kutusu uygulaması kullanılmış ve yeni parametrik modeller denklem setleriyle önerilmiştir. Ayrıca yeni parametrik modellerin tahminleri ile gerçek yazılım proje efor değerleri karşılaştırılmış, hata ölçütü kullanılarak literatürde mevcut modellerde daha iyi tahminde bulunduğu doğrulanmıştır.

Tez çalışması kapsamında yazılım proje başlangıcında proje yöneticilerine yardımcı olacak şekilde parametrik tahmin modelleri yazılım paketi hazırlanmıştır. Bu yazılım paketinin hazırlanmasındaki amaç, yazılım projeleriyle ilgilenen proje yöneticilerine ve ihtiyaç sahibi makama proje başlangıcında yazılımcı iş gücü ve proje süresi planlamasında kolaylık sağlamaktır. Bu araç kutusunda, literatürde mevcut parametrik tahmin modelleri ile önerilen yeni modeller bir araya getirilmiştir. Söz konusu araç kutusu marifetiyle proje yöneticileri literatürde mevcut parametrik tahmin yöntemleri iş gücü kestirimlerini proje öncesinde görebilmektedir. Ayrıca gerçek yazılım proje verileriyle hata oranı azaltılmış daha iyi tahminlerde bulunan yeni modeller araç kutusuna ilave edilerek mevcut modeller arasında kıyaslama imkânı sağlanmıştır.

KAYNAKLAR

- Anonim. 1977. Doty Associates, Inc. Software Cost Estimates Study, vol. 1, s. 77-220.
- Anonim. 2013. Web Sitesi: http://en.wikipedia.org/wiki/Waterfall_model Erişim tarihi: 14.12.2013
- Adalier, O. 2008. Yapay Zeka Yöntemleri İle Yazılım Projelerinde Maliyet Kestirimi. Doktora tezi (basılmamış). Ege Üniversitesi, İzmir.
- Ahn, Y., Suh, J., Kim, S., and Kim, H. 2003 The Software Maintenance Project Effort Estimation Model Based on Function Points, Journal of Software Maintenance and Evolution: Research and Practice, pp.71-85.
- Albrecht, A.J. 1979. Function-Point Method (Measuring Applications Development Productivity), IBM Application Development Joint Share and Guide Symposium, Monterey.
- Ayyıldız, M. 2007. Yazılım Projeleri Ölçüm Sonuçları Veri Tabanının Oluşturulması ve Yeni Yazılım Projelerinin Maliyet Tahmininde Kullanılması. Doktora tezi (basılmamış). Yıldız Teknik Üniversitesi, İstanbul.
- Bailey, J.W. and Basili, V.R. 1981. A meta Model for Software Development Resource Expenditure, Proceedings of the International Conference on Software Engineering, pp. 107–115.
- Basha, S. and Dhavachelvan, P. 2010. Analysis of Empirical Software Effort Estimation Models. International Journal of Computer Science and Information Security, vol. 7, no. 3, pp. 68-72.
- Benediktsson, O., Dalcher, D., Reed, K., and Woodman, M. 2003. COCOMO based effort estimation for iterative and incremental software development, Software Quality Journal, vol. 11, pp. 265–281.
- Boehm, B., Bradford, C., Horowitz, E., Madachy, R., Shelby, R. And Westland, C. 1995. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0, Annals of Software Engineering Special Volume, Amsterdam, The Netherlands.
- Borandağ, E., Yücalar, F. ve Şahinaslan, Ö. 2013 Yazılım Projelerinde Büyüklük Tahmini [http:// http://ab.org.tr/ab13/bildiri/77.pdf](http://ab.org.tr/ab13/bildiri/77.pdf), Erişim Tarihi: 13.07.2014
- Fenton, N.E. 1994. Software Measurement: A Necessary Scientific Basis”, IEEE Transactions on Software Engineering, vol.20 no.3, pp. 199-206
- Fenton, N.E. and Pfleenger, S. 1998, S.L., Software Mertics: A Rigorous and Practical Approach, International Thomson Computer Press.

- Fetcke, T., Abran, A. and Dumke, R. 2001. A Generalized Representation for Selected Functional Size Measurement Methods, 11th International Workshop on Software Measurement, Montreal, Canada.
- Gürbüz, S. ve Şahin, F. 2014. Sosyal Bilimlerde Araştırma Yöntemleri, Seçkin Yayınları.
- Halstead, M.H. 1977. Elements of Software Science, Elsevier, New York.
- Heemstra, F.J. 1992. Software cost estimation, Information and Software Technology, S.627-639.
- Jones, C. 1996. Patterns of Software System Failure and Success, International Thomson Computer Press, Boston.
- Jones, C. 1998. Project Management Tools and Software Failures and Successes. CrossTalk, <http://www.stsc.hill.af.mil./crosstalk/1998/07/tools.pdf> pp. 13-17.
- Kartın, E. 2008 Güvenli Yazılım Geliştirme, Beykent Üniversitesi Computer Based Business Application CEN 404 Seminer Notları, İstanbul.
- Kaur, J., Singh, S., and Kahlon, K.S. 2008. Comparative Analysis of the Software Effort Models, Proceeding of World Academy of Science, Engineering and Technology, vol. 36, pp. 485-487.
- Kemerer, C.F. 1987. An Empirical Validation of Software Cost Estimation Models. Communications of the ACM 30.
- Laird, L.M. and Brennan, M.C. 2010. Software Measurement and Estimation: A Practical Approach, pp. 103.
- Linda, M.L. and Carol, B. 2006. Software Measurement and Estimation: A Practical Approach pp. 79-112.
- Paar, F.N. 1980. Parr Model: An Alternative to Rayleigh Norden Curve Model for Software Development Effort. IEEE Transactions on Software Engineering, SE-6, No:3.
- Peter, R. 2010. Practical Software Project Estimation Book pp. 243-252.
- Savaş, K. 2007. Kontrol Sistemleri için MATLAB’te GUI Uygulamaları Tasarımı. Lisans Tezi (basılmamış). Marmara Üniversitesi, İstanbul.
- Sheta, A., Rine, D. and Ayesh, A. 2008. Development of Software Effort and Schedule Estimation Models Using Soft Computing Techniques, IEEE Congress on Evolutionary Computation, pp. 1283-1289.

Singh, B.K., and Misra, A.K. 2012. Software Effort Estimation by Genetic Algorithm Tuned Parameters of Modified Constructive Cost Model for NASA Software Projects, *International Journal of Computer Applications*, vol.59, no. 9, pp. 22-26.

Symons, C.R. 1988. Function Point Analysis: Difficulties and Improvements, *IEEE Transactions on Software Engineering*, vol. SE-14, no. 1, s. 2-11.

Walston, C.E., and Felix, C.P. 1977. A Method of Programming Measurement and Estimation, *IBM Systems Journal*, vol. 16, no. 1, pp. 54-73.