# YAZILIM MÜHENDİSLİĞİ

## Yöntembilimi, Belgeleri ve Standartlarıyla

M. Erhan SARIDOĞAN (Ph.D)

**PAPATYA YAYINCILIK EĞİTİM**
İstanbul, Ankara, İzmir ve Adana

© **PAPATYA YAYINCILIK EĞİTİM**
Bilgisayar Sis. San. ve Tic. A.Ş.
Tel : (+90 212) 527 52 96 (+90 532) 311 31 10
Faks : (+90 212) 527 52 97
E-Posta : admin@papatyabilim.com.tr
Web : www.papatyabilim.com.tr

**Yazılım Mühendisliği** – M. Erhan SARIDOĞAN

3. Basım Ocak 2017

Editör : Toros Rifat ÇÖLKESEN (Ph. D)
Yayın Danışmanı : Cengiz UĞURKAYA (Post-Edu Institute)
Üretim : Necdet AVCI
Pazarlama : Mustafa DEMİR
Sayfa Düzenleme : Papatya & Kelebek Tasarım
Kapak Tasarım : Papatya & Kelebek Tasarım

Basım ve Ciltleme : Özkaracan Matbaacılık (Sertifika No: 12228)
Evren Mah. Gülbahar Cad. No:62 Güneşli/İstanbul

© Bu kitabın her türlü yayın hakkı Papatya Yayıncılık Eğitim A.Ş.'ye aittir. Yayınevinden yazılı izin alınmaksızın alıntı yapılamaz, kısmen veya tamamen hiçbir şekil ve teknikle ÇOĞALTILAMAZ, BASILAMAZ, YAYIMLANAMAZ. Kitabın, tamamı veya bir kısmının fotokopi makinesi, ofset gibi teknikle çoğaltılması, hem çoğaltan hem de bulunduranlar için yasadışı bir davranıştır.

Sarıdoğan, M. Erhan
Yazılım Mühendisliği / Sarıdoğan, M. Erhan – İstanbul: Papatya Yayıncılık Eğitim, 2017.
xxii, 568 s.; 24 cm.
Kaynakça ve dizin var.
ISBN 978-975-6797-57-0
1. Bilgisayar Sistemleri 2. Yazılım Modelleme 3. Benzetim 4. Proje Yönetimi 5. UML
I. Title

*Meslektaşlarıma yararlı olacağına inandığım bu kitabımı*
*değerli eşim Bruni ve sevgili kızım Eda'ya*
*atfediyorum.*

# Teşekkür

Bu kitabın ortaya çıkmasında beni yetiştirenlerin, bugünlere getirenlerin ve bu kadar bilgiyi öğrenmeme yardımda bulunanların payı büyüktür. Bu kişiler içerisinde annem ve babam başta olmak üzere, ilkokuldan başlayıp doktora eğitimime kadar bana yardımcı olan ve destek veren tüm öğretim üyelerine, ayrıca bana büyük destek veren eşime buradan bir kez daha teşekkür etmek isterim.

Ayrıca birlikte çalıştığım ve bana önemli miktarda bilgi sağlamış olan meslek arkadaşlarıma, İTÜ Bilgisayar ve Bilişim Fakültesi Bilgisayar Mühendisliği Bölümü öğretim üyelerinden sayın Prof. Dr. Nadia ERDOĞAN'a teşekkür ederim. Kitabın editörlüğünü üstlenen ve hazırlanmasında emeği geçen sayın Dr. Toros Rifat ÇÖLKESEN'e ve Papatya Bilim çalışanlarına da en içten teşekkürlerimi sunarım.

# İÇİNDEKİLER

**ÖNSÖZ** **xxi**

**Bölüm 1. Giriş** **23**

1.1. Bu Kitap Hakkında 24
1.2. Bu Kitabı Kimler Okumalı? 24
1.3. Kitapta Bulunan Konular 25

**Bölüm 2. Bilgisayar Sistemleri** **27**

2.1. Sistem Çeşitleri 27
2.2. Bilgi Sistemleri 28
2.2.1. Çevrimiçi Sistemler 28
2.2.2. Gerçek Zamanlı Sistemler 28
2.2.3. Karar Destek Sistemleri 30
2.2.4. Bilgi Tabanlı Sistemler 31
2.2.5. Veri Tabanı Yönetim Sistemleri 32
2.2.6. Kişisel Bilgisayarlar 33
2.2.7. Ofis Otomasyonu 34
2.2.8. Atölye Otomasyonu 34
2.2.9. İletişim Sistemleri 34
2.2.10. Endüstriyel Sistemler 35
2.2.11. Kontrol Sistemleri 36
Askeri, Trafik Kontrol, Robotik ve Gömülü Sistemler 36
2.3. Sistem Yapıları 39
2.3.1. Ana Sistem 39
2.3.2. Hedef Sistem 40
2.3.3. Test Sistemi 40
2.3.4. Geliştirme Sistemi Örneği 40
2.4. Sistem Bileşenleri 41
2.4.1. Donanım 41
2.4.2. Altyapı Yazılımı 43
İşletim Sistemi, Ara Katman Yazılımı ve Yardımcı Yazılımlar 43
2.4.3. Uygulama Yazılımları 45
Özel Sistem, İş Dünyası ve Mühendislik ve Bilimsel Yazılımları 43
Yapay Zeka, Veritabanı Yönetimi, İnternet ve AR-GE Yazılımları 45
2.5. Bilgi İşlemenin Temelleri 48

2.5.1. Bilgi İşleme Mantığı 49
Merkezi Bilgi İşleme ve Dağıtık Bilgi İşleme 49
2.5.2. Paralel Bilgi İşleme Özellikleri 51
Süreç Yönetimi, Koşutzamanlılık, İş Sıralama ve İletişim 52
2.6. Yazılım Mimarileri 53
2.6.1. Genel Yazılım Mimarisi 54
2.6.2. Kullanıcı-Sunucu Mimarileri 55
Kullanıcı, Sunucu, İşlem Yönetimi 55
2.6.3. İki Katmanlı Mimari 57
2.6.4. Üç Katmanlı Mimari 58
2.6.5. Dağıtık Mimariler 59
2.6.6. Yazılım Birimi Türleri 59
2.7. Özet 61
2.8. Sorular 61

**Bölüm 3. Bilgisayar Sistemi Mühendisliği 63**

3.1. Proje Yaşam Çevrimi 63
3.2. Sistem Maliyet Çözümlemesi 67
3.2.1.Edinme Şekli 67
3.2.2. Maliyet Unsurları 68
3.2.3. Edinme Maliyeti 69
Ortak, Hazır Sistem, Özel Geliştirilen Sistem Maliyetleri 69
3.2.4. Geliştirme Maliyetleri 71
3.3. Sistem Geliştirme Süreci 73
3.4. Proje Yönetimi 75
3.4.1. Yönetsel Planlar 75
3.4.2. Teknik Planlar 76
3.5. Sistem Mühendisliği 77
3.5.1. ISO/IEC 15288 78
3.5.2. Mühendislik Etkinlikleri 78
3.5.3. Donanım ve Donanım Mühendisliği 79
3.5.4. Yazılım ve Yazılım Mühendisliği 80
3.5.5. İnsan Mühendisliği 80
3.6. Sistem Çözümlemesi 83
3.6.1. Yapılabilirlik Araştırması 83
3.6.2. Kavramsal Tanımlama 85
3.6.3. Sistem İsterleri 85
Sistem Modelleme, Benzetim, İster Belirleme 85
İsterler Belirtimi, Test/Sınama ve Belgelendirme 87

3.7. Sistem Tasarımı 91
3.7.1. Tasarım Etmenleri 92
3.7.2. Tümleştirme Mimarileri 92
Merkezi Tümleştirme ve Dağıtık Tümleştirme Mimarisi 93
3.7.3. Altsistem Arayüzü 94
Arayüz Donanımı ve Arayüz Yazılımı 94
3.7.4. Sistem Bileşenleri 98
3.7.5. Sistem Mimari Tasarımı 99
3.7.6. Belgelendirme 100
3.8. Sistem Gerçekleştirimi 101
3.8.1. Donanım Geliştirme 101
3.8.2. Yazılım Geliştirme 101
Tanımlama, Geliştirme ve Bakım 101
3.8.3. Sistem Tümleştirme 102
3.8.4. Elle Denetim Yordamları 103
3.8.5. Belgelendirme 103
3.9. Sistem Kurulumu 103
3.9.1. Donanım Kurulumu 104
3.9.2. Yazılım Kurulumu 105
3.9.3. Kurulum Testleri 105
3.10. Sistem Testi 106
3.11. Sistem Teslimi 106
3.11.1. Eğitim 106
3.11.2. Destek 107
3.11.3. Garanti 107
3.12. Sistem Bakımı 107
3.12.1. Donanım Bakımı 108
Planlı Bakım, Onarım, İyileştirici Bakım ve Yazılım Bakımı 108
3.13. Belgelendirme 109
3.14. Yardımcı Araçlar 110
3.15. Özet 110
3.16. Sorular 111

**Bölüm 4. Yazılım Mühendisliği 113**

4.1. Yazılım 113
4.1.1. Tarihçe 114
4.1.2. Yazılım Karmaşıklığı 114
4.1.3. Yazılımın Algılanması 116
Yönetim, Müşteri ve Uygulayıcı 116

4.2. Yazılım Özellikleri 118
4.2.1. Yazılım Sınıfları 119
4.2.2. Maliyet 120
4.2.3. Niteliksel Özellikler 120
4.2.4. Etkinlik 120
4.3. Yazılım Mühendisliğine Genel Bakış 121
4.3.1. Yazılım Mühendisi 121
4.3.2. Yazılım Geliştirme 122
Tanımlama, Geliştirme ve Bakım 122
4.3.3. Yaşam Çevrimi Verileri 124
4.4. Yazılım Mühendisliği Yöntembilimleri 126
4.4.1. Klasik Çevrim 126
4.4.2. "V" Modeli 127
4.4.3. Prototipleme/Örnekleme 128
4.4.4. Spiral Model 130
Evrimsel Geliştirme, Evrimsel Prototipleme (ilkörnekleme) 131
Artımlı Geliştirme, Araştırmaya Dayalı Geliştirme 132
4.4.5. Gelişigüzel Geliştirme 133
4.4.6. Yeni Teknikler 133
Özneye Yönelik Geliştirme, Bileşen Tabanlı Geliştirme 134
Özelliğe Yönelik Programlama, Uç Programlama 136
4.4.7. Tekniklerin Birleştirilmesi 138
4.5. Yazılım Mühendisliği Araçları 139
4.5.1. Yönetsel Araçlar 139
4.5.2. Çözümleme ve Tasarım Araçları 140
4.5.3. Gerçekleştirim Araçları 141
4.5.4. Bakım Araçları 142
4.5.5. Destekleyici Araçlar 142
4.5.6. Araç Seçimi ve Geliştirilmesi 143
4.6. Özet 144
4.7. Sorular 145

**Bölüm 5. Yazılım İsterleri Çözümlemesi 147**

5.1. İsterler Çözümlemesi Aşaması 148
5.1.1. Çözümleme Çalışmaları 148
5.1.2. İsterlerin Değişmesi 149
5.2. İsterlerin Belirlenmesi 150
5.2.1. İsterlerin Düzeyleri 150
5.2.2. İster Belirleme Yöntemleri 151

5.2.3. İsterlerin Tanımlanması 152
5.2.4. İsterlerin Sınıflandırılması 152
5.3. Çözümleme Yöntemleri 153
5.4. Yapısal Çözümleme 154
5.4.1. Veri Akış Diyagramı 155
5.4.2. Davranış Modellemesi 157
5.4.2.1. Süreç Etkinleştirme Tablosu 158
5.4.2.2. Durum Geçiş Diyagramı 158
5.4.2.3. Süreç Belirtimi 159
5.4.2.4. Karar Tabloları 159
5.4.2.5. Varlık İlişki Diyagramı 160
5.4.3. Veri Sözlüğü 161
5.5. Nesneye Yönelik Çözümleme 161
5.5.1. Nesne 162
Nesne Tanımlama, Nesne Sınıflandırma ve Nesne Yapıları 162
5.5.2. Modelleme 164
5.5.3. Prototipleme 165
5.5.4. Tasarım 165
5.6. Veri Yapılarına Yönelik Çözümleme 166
5.7. Prototipleme 166
5.8. Belgelendirme 167
5.8.1. Belge İçeriği 168
5.8.2. Belge Hazırlama Önerileri 168
5.9. Riskler 169
5.10. Özet 170
5.11. Sorular 171

**Bölüm 6. Yazılım Tasarımı 173**

6.1. Tasarım Aşaması 173
6.2. Yazılım Tasarım Süreci 176
6.2.1. Veri Tasarımı 177
6.2.2. Mimari Tasarımı 179
6.2.3. Yordamsal Tasarım 180
Yapısal Programlama Gösterimi ve Grafiksel Gösterim 180
6.2.4. Arayüz Tasarımı 183
6.2.4.1. Bileşen Arayüz Tasarımı 183
6.2.4.2. Kullanıcı Arayüz Yazılımı Tasarımı 184
6.2.4.3. Sistem-Altsistem Arayüz Yazılımı Tasarımı 184
6.3. Tasarım Yöntemleri 185

6.4. Veri Akışına Yönelik Tasarım 186
6.4.1. Akış Türleri 186
6.4.2. Tasarım Aşamaları 187
Dönüşüm Çözümlemesi, Ara İşlem Çözümlemesi 187
Modüler Tasarım, Tasarım Anlatımı 189
6.5. Nesneye Yönelik Tasarım 190
6.5.1. Temel Özellikler 191
6.5.2. Tipler ve Sınıflar 193
6.5.3. Nesne ve Sınıf İlişkisi 194
6.5.4. Nesneye Yönelik Tasarım Aşamaları 194
Nesneye Yönelik Olmayan ve Olan Diller İçin Tasarım 195
6.5.5. Nesneye Yönelimde Nitelik Ölçütleri 196
6.5.6. Nesneye Yönelik Tasarım İlkeleri 197
6.6. Veriye Yönelik Tasarım 198
6.7. Tasarım Kalıpları 198
6.7.1. Kalıp Tanımı ve Özellikleri 198
6.7.2. Karşı Kalıplar 200
6.7.3. Kalıp Örnekleri 200
6.8. Kullanıcı Arayüzü Tasarımı 201
6.8.1. İnsan-Bilgisayar Etkileşimi 201
6.8.2. İnsan-Bilgisayar Arayüz Tasarımı 203
6.8.3. Arayüz Yazılım Mimarisi 204
6.8.4. Arayüz Yazılım Birimleri 205
6.8.5. Arayüz Tasarım İlkeleri 205
Yüksek Nitelik, Kullanıcı Dostluğu, Güvenilirlik, Yardımlar 206
Hatalar ve Uyarılar, Yapısal Özellikler 209
6.8.6. Kullanıcı Arayüz Geliştirme Süreci 210
Çözümleme, Tasarım, Gerçekleştirim ve Test 211
6.9. Gerçek Zamanlı Sistem Tasarımı 212
6.9.1. Yapısal Özellikler 212
Mimari, Özkaynak Gereksinimleri, Gerçekleştirim Teknikleri 213
6.9.2. Dinamik Özellikler 214
Süreç Yönetimi, Zamansallık ve Aykırı Durum Denetimi 214
6.9.3. Niteliksel Özellikler 216
6.10. Veri Tabanı Uygulamaları Tasarımı 217
6.10.1. Mimari 217
6.10.2. Sınıflandırma 218
6.10.3. Sistem Yaşam Süreci 219
6.10.4. Tasarım 220

6.11. İnternet Uygulamaları Tasarımı 221
6.11.1. Amaç 221
6.11.2. Arayüz 222
6.11.3. Güvenlik 222
6.11.4. Tasarım Önerileri 222
6.12. Tasarımda Genel Kurallar 224
6.12.1. Modüler Tasarım 224
Modüler Tasarım Ölçütleri, Modül Türleri 224
İşlevsel Bağımsızlık, Modüler Tasarım İlkeleri 226
6.12.2. Tekrar Kullanım 227
6.12.3. Denetim ve İşlevin Ayrılması 228
6.12.4. Genel Kurallar 228
6.13. Belgelendirme 229
6.14. Riskler 230
6.15. Özet 231
6.16. Sorular 231

**Bölüm 7. Yazılım Gerçekleştirimi 233**

7.1. Programlama Dilleri 234
7.1.1. Dillerin Tarihçesi 234
7.1.2. Dillerin Gelişimi 235
Birinci Nesil Diller, İkinci Nesil Diller, Üçüncü Nesil Diller 235
Dördüncü Nesil Diller ve Beşinci Nesil Diller 236
7.1.3. Dillerin Özellikleri 239
Nesneye Yönelik ve Gerçek Zamanlı Dillerin Özellikleri 241
7.1.4. Dil Seçimi 243
7.1.5. Dillerin Uygulama Alanları 243
7.1.6. Yeni Diller 245
7.2. Kod Çevrim İşlemi 245
7.2.1. Derleyiciler 245
7.2.2. Yorumlayıcılar 246
7.2.3. Geliştirme Ortamı 246
7.2.4. Yürütme Ortamı 248
CORBA, DDS, HLA ve Ticari Ara Katmanlar 249
7.3. Kodlama Biçimleri 252
7.3.1. Kodlama Dili 253
7.3.2. Kod Belgelemesi 253
7.3.3. Veri Bildirimi 257
7.3.4. Deyim Yapıları 257

7.4. Etkinlik 258
7.4.1. Kod Etkinliği 28
7.4.2. Bellek Etkinliği 259
7.4.3. Giriş/Çıkış Etkinliği 259
7.4.4. Atık Toplama 260
7.4.5. Aykırı Durumların Kotarılması 262
7.5. Temel İlkeler 264
7.5.1. Kodlamada Niteliksel Özellikler 264
7.5.2. Modül Oluşturma 267
7.5.3. Kod Yazımı 268
7.6. Belgelendirme 269
7.7. Riskler 270
7.8. Özet 270
7.9. Sorular 271

**Bölüm 8. Yazılım Testi 273**

8.1. Yazılım Testi/Sınamasının Temelleri 274
8.1.1. Testin Amaçları 275
8.1.1.1. Deneme Testleri 275
8.1.1.2. Kabul Testleri 275
8.1.2. Testin Yapılışı 276
8.1.3. Test Yöntemleri 277
Saydam Kutu Testi, Kara Kutu Testi 278
8.1.4. Özel Sistemlerin Testleri 279
8.1.5. Otomatik Test Araçları 281
8.2. Test Stratejileri 283
8.2.1. Birim Testi 284
Birim Gerçekleştiriminde Hatalar 285
Birim Testi Yöntemleri, Birim Testinin Yapılışı 285
8.2.2. Tümleştirme Testi 287
Yukarıdan Aşağı ve Aşağıdan Yukarıya Tümleştirme 288
8.2.3. Yeterlilik Testleri 289
Doğrulama, Geçerleme ve Rasgele Testler 289
8.2.4. Sistem Testi 294
Yükleme Testi, Germe Testi ve Geri Kazanma Testi 294
Emniyet Testi, Güvenlik Testi, Başarım Testi 296
8.2.5. Kabul Testleri 298
Üretim Hattı Testleri, Kullanım Hattı Testleri 298
Deneme Testleri, Alfa ve Beta Testleri ve Kabul Kıstasları 299

8.3. Test Yönetimi 301
8.4. Hata Ayıklama 302
8.4.1. Hata Ayıklama Süreci 302
8.4.2. Yaygın Hataların Özellikleri 304
8.4.3. Hata Ayıklama Yöntemleri 304
8.5. Belgelendirme 305
8.6. Riskler 306
8.7. Özet 307
8.8. Sorular 308

**Bölüm 9. Yazılım Bakımı 309**

9.1. Yazılım Bakımının Temelleri 309
9.1.1. Bakım Türleri 309
9.1.1.1. Düzeltici Bakım 310
9.1.1.2. Uyarlayıcı Bakım 310
9.1.1.3. İyileştirici Bakım 310
9.1.1.4. Önleyici Bakım 311
9.1.2. Örgütlenme 311
9.1.3. Bakım Aşamaları 312
9.1.4. Raporlama 313
9.2. Bakım Kolaylığı 314
9.2.1. Denetim Etmenleri 314
9.2.2. Bakımın Niteliği 315
9.2.3. Niceliksel Ölçümler 315
9.2.4. Bakım Sorunları 316
9.2.5. Geliştirici İçin Kurallar 317
9.3. Bakımın Yan Etkileri 317
9.3.1. Kodlamanın Etkilenmesi 317
9.3.2. Verilerin Etkilenmesi 318
9.3.3. Belgelendirmenin Etkilenmesi 319
9.3.4. Başarımın Etkilenmesi 319
9.4. Belgelendirilmemiş Yazılımların Bakımı 319
9.4.1. Aynı Kod Üzerinde Bakım 320
9.4.2. Kod Taşıma 320
9.4.3. Tersine Mühendislik 321
9.4.4. Yeniden Yapılanma 322
9.5. Riskler 324
9.6. Özet 324
9.7. Sorular 324

**Bölüm 10. Yazılım Nitelik Güvencesi** **327**
10.1. Yazılım Niteliği 327
10.1.1. Nitelik Etmenleri 328
Kullanıma, Taşınmaya ve Yönelik Özellikler 328
10.1.2. Nitelik Metrikleri 330
10.1.3. Nitelik Güvence Gereksinimi 331
10.1.4. Nitelik Güvence Etkinlikleri 332
10.1.5. Nitelik Güvence Yönetimi 334
10.1.6. Toplam Nitelik Yönetimi 335
10.1.7. Örgütlenme ve Nitelik 335
10.2. Süreç Modelleri 336
10.3. Nitelik Sistem Standartları 337
10.3.1. CMM 338
10.3.2. CMMI 340
10.3.3. Trillium 341
10.3.4. TickIT 342
10.3.5. SPICE 343
10.3.6. ISO 9000 345
10.3.7. AQAP-150 / 160 345
10.3.8. Karşılaştırma ve Öneriler 347
10.4. Gözden Geçirmeler 348
10.4.1. Eşdüzey Gözden Geçirme 348
10.4.2. Resmi Teknik Gözden Geçirmeler 349
10.4.3. Birleşik Gözden Geçirme 350
10.4.4. İnceleme Kuralları 351
10.4.5. Yazılım Ürün Değerlendirmesi 353
10.4.6. İnceleme Kontrol Listeleri 353
10.4.6.1. Sistem Mühendisliği 354
10.4.6.2. Yazılım Proje Planlaması 354
10.4.6.3. Yazılım İsterleri Çözümlemesi 354
10.4.6.4. Yazılım Tasarımı 356
10.4.6.5. Kodlama 357
10.4.6.6. Yazılım Tümleştirme ve Test 357
10.4.6.7. Yazılım Bakımı 360
10.5. Resmi Nitelik Güvence Yöntemleri 360
10.5.1. Doğruluğun Kanıtlanması 360
10.5.2. İstatistiksel Yaklaşım 361
10.5.3. Temiz Oda Süreci 361
10.5.4. Yardımcı Araç Desteği 362

10.6. Yazılım Güvenilirliği 362
10.6.1. Güvenilirlik ve Yararlanılabilirlik Ölçümleri 363
10.6.2. Yazılım Emniyeti 364
10.7. Sistem Aktarımı 365
10.7.1. Donanım Aktarımı 366
10.7.2. Yazılım Aktarımı 367
10.7.2.1. Yazılım Sözleşme Türleri 367
10.7.2.2. Lisanslar 367
10.7.2.3. Aktarılacak Öğeler 368
10.7.3. Belgelendirme Aktarımı 369
10.7.4. Bütünleştirici Süreçler 370
10.7.5. Garanti Koşulları 370
10.7.6. Eğitim 371
10.8. Yazılım Niteliği Ölçüm Unsurları 371
10.8.1. Yazılım Ürün Metrikleri 372
10.8.2. Metrik Türleri 373
10.8.3. Metrik Tanımları 374
Süreç Metrikleri, Boyut Metrikleri, Zamanlama Metrikleri 374
Maliyet ve Kaynak Metrikleri, Ürün Nitelik Metrikleri 375
Bakım ve Okunabilirlik Metrikleri, Üretkenlik Metrikleri 376
10.8.4. Yazılım Ölçme Zorlukları 376
10.8.5. Doğrulama ve Değerlendirme 377
10.8.6. Başarım ve Etkinlik 378
10.9. Özet 378
10.10. Sorular 379

**Bölüm 11. Düzenleşim Yönetimi 381**

11.1. Sistem Düzenleşim Yönetimi 381
11.2. Yazılım Düzenleşim Yönetimi 382
11.2.1. Düzenleşim Yönetim Planı 382
11.2.2. Düzenleşim Yönetim Sistemi 383
11.3. Düzenleşim Yönetimi Süreci 383
11.3.1. Tanımlama 384
11.3.2. Sürüm Denetimi 386
11.3.3. Düzenleşim Denetimi 386
11.3.4. Değişiklik Denetimi 387
Değişiklik Denetimi Düzeyleri, Yetkilisi ve Süreci 387
11.3.5. Denetim 390
11.3.6. Durum Değerlendirmesi ve Raporlama 391

11.4. Düzenleşim Denetimi İle Geliştirme 392
11.4.1. Sabitleme 392
11.4.2. Sabitleme İle Geliştirme 393
11.4.3. Günlük Birleştirme 394
11.5. Ürün Çoğaltma 394
11.5.1. Çoğaltma Türleri 395
11.5.2. Kurulum Dosyaları 395
11.5.3. Belgelendirme 396
11.5.4. Ürün Teslimi 396
11.6. Özet 397
11.7. Sorular 397

**Bölüm 12. Yazılım Geliştirme Yöntembilimleri 399**

12.1. Yazılım Geliştirme Standartlarının Geçmişi 400
12.1.1. Askeri Sistemlerin Öncülüğü 400
12.1.2. Tarihçe 401
12.1.3. Belgelendirme 403
12.2. IEEE/EIA 12207 404
12.2.1. Standardın Kullanımı 405
12.2.2. Süreçler 405
Temel Süreçler, Destekleyici Süreçler ve Örgütsel Süreçler 409
12.2.3. Uyarlama 410
12.2.4. Kılavuzluk 410
12.2.5. Destekleyici Standartlar 410
12.2.6. Belgeler 411
12.2.7. Karşılaştırma 412
12.3. Yazılım Geliştirme Süreci 413
12.3.1. Yazılım Proje Planlaması 416
Planlar, Personel ve Uygulama 416
12.3.2. Yazılım Geliştirme Ortamının Kurulması 418
Yazılım Geliştirme Ortamı, Yazılım Test Ortamı ve Kullanım Ortamı 418
12.3.3. Yazılım İsterleri Çözümlemesi 419
Sistem Yazılım İsterleri ve Yazılım Öğe Çözümlemesi 420
Test Hazırlığı 420
12.3.4. Yazılım Tasarımı 420
Sistem Yazılım Tasarımı, Öğe Tasarımı 422
Ayrıntılı Tasarım, İzlenebilirlik 422
12.3.5. Yazılım Gerçekleştirimi ve Birim Testi 423
Yazılım Gerçekleştirimi ve Birim Testi 423

12.3.6. Yazılım Tümleştirme ve Test 424
12.3.7. Yeterlilik Testi 424
12.3.8. Sistem Tümleştirme ve Test 425
12.3.9. Sistem Yeterlilik Testi 425
12.3.10. Yazılımı Kullanıma Hazırlama 425
Yürütülebilir Yazılım, Yazılım Kullanıcı Kılavuzları 425
12.3.11. Yazılımı Aktarmaya Hazırlama 426
12.3.12. Yazılım Bakımı 426
12.3.13. Genel Belge Listesi 426
12.4. Model-Güdümlü Mimari 428
12.4.1. MDA Kullanılışı 428
12.4.2. MDA Araçları 429
12.5. Özet 430
12.6. Sorular 430

**Bölüm 13. Proje Yönetimi 433**

13.1. Proje ve Program 434
13.1.1. Proje Başlatılması 434
13.1.2. Proje Yönetimi 435
13.1.3. Proje Hedefleri 436
13.1.4. Proje Yönetim Süreçleri 436
13.2. Proje Tümleştirme Yönetimi 438
13.2.1. Proje Planı 438
13.2.2. İş Dağılım Ağacı 438
13.2.3. Değişiklik Denetimi 440
13.3. Proje Kapsam Yönetimi 440
13.4. Proje Zaman Yönetimi 441
13.4.1. Ağ Diyagramı 441
13.4.2. Öncül Çizelgeleme 443
13.4.3. PERT 443
13.4.4. Zaman Planlaması 444
İnsan Etmeni, Proje Takibi ve Denetimi ve Kişisel Zaman Yönetimi 444
13.5. Proje Maliyet Yönetimi 446
13.5.1. Kaynak Planlaması 447
Donanım ve Yazılım Kaynakları, İnsan Kaynakları ve Planlama 447
13.5.2. Maliyet Kestirimi 449
13.5.3. Maliyet Kalemleri 450
13.5.4. Maliyet Bütçelemesi ve Denetimi 451

13.6. Proje Nitelik Yönetimi 451
13.6.1. Nitelik Planlama 452
13.6.2. Nitelik Güvence ve Denetim 452
13.7. Proje İnsan Kaynakları Yönetimi 452
13.7.1. Örgütsel Planlama 452
13.7.2. Ekip Oluşturma 453
13.8. Proje İletişim Yönetimi 453
13.8.1. Bilgi Dağıtımı 453
13.8.2. Kapanış 454
13.9. Proje Risk Yönetimi 454
13.9.1. Yönetim Planlaması 454
13.9.2. Risk Yönetimi Etkinlikleri 455
Risklerin Tanımlanması, Risklerin Çözümlenmesi 455
Risklerin Planlanması, Risklerin İzlenmesi 456
13.9.3. Risk Yönetiminin Uygulanması 457
13.9.4. Yazılım Risk Yönetimi 458
13.10. Proje Edinme Yönetimi 460
13.10.1. Planlama 460
13.10.2. İhale ve Sözleşme 460
13.10.3. Yazılım Edinme Seçenekleri 460
13.11. Proje Örgütlenmesi 462
13.11.1. Yazılım Geliştirme Proje Yönetimi 466
13.11.1.1. Gerçekleştirme Yöntemi 466
13.11.1.2. Kestirim 466
13.11.2. Örgütsel Planlama 468
Yazılım Geliştirme Yöneticisi, Yazılım Düzenleşim Yönetimi 468
Nitelik Güvence Yönetimi, Test ve Tümleştirme Yönetimi 470
Yazılım Geliştirme Grupları ve Geliştirme Ekipleri 470
13.11.3. Yazılım Geliştirme Planı 471
13.12. Özet 471
13.13. Sorular 472

**Bölüm 14. Yazılım Geliştirme İçin Pratik Öneriler 473**

14.1. Yönetsel Öneriler 473
14.1.1. Proje Yönetimi 474
14.1.2. İnsan Kaynakları Planlaması 474
14.1.3. Maliyet Kestirimi ve Planlama 476
14.1.4. Metrik Kullanımı 477
14.1.5. Kazanılmış Değerlerin İzlenmesi 477

14.1.6. Nitelik Hedeflerinin İzlenmesi 478
14.1.7. Disiplinin Sağlanması 479
14.2. Süreçlere Yönelik Öneriler 480
14.2.1. İsterlerin Yönetimi ve İzlenmesi 480
14.2.2. Sistem Tabanlı Yazılım Tasarımı 481
14.2.3. Beraber Çalışabilirlik 481
14.2.4. Arayüzlerin Tanımlanması ve Denetimi 482
14.2.5. Yaşam Çevrimi Düzenleşim Yönetimi 482
14.3. Geliştirmeye Yönelik Öneriler 484
14.3.1. Tasarım 484
14.3.2. Tekrar Kullanım 485
14.3.3. İsterlerin ve Tasarımın Denetlenmesi 486
14.3.4. Gerçekleştirim 487
14.3.5. Sürekli Test 488
14.3.6. Sık Derleme ve Test 489
14.3.7. Hata Ayıklama 490
14.3.8. Ölçme Süreci Uygulaması 491
14.4. Niteliği Artırmak İçin Pratik Öneriler 491
14.4.1. İşlevsel Nitelik 491
14.4.2. Güvenilirlik 492
14.4.3. Bakım Kolaylığı 492
14.4.4. Kullanışlı Sistem Geliştirme 493
14.5. Toplantı Kuralları 494
14.5.1. Toplantı Verimi 494
14.5.2. Toplantı Belgeleri 496
14.6. Özet 496

**Ek A. Yazılım Mühendisliği Belgeleri 497**

A.1. Standart Yapı 497
Kapak, Listeler, Belge Başı 497
Belge Ortası, Belge Sonu ve Ekler 498
A.2. Proje Öneri Belgesi 499
A.3. Proje Tanımlama Belgesi 500
A.4. Proje Yönetim Planı 501
A.5. İşletim Kavramı Tanımlaması 504
A.6. Yapılabilirlik Raporu 505
A.7. Sistem Mühendisliği Yönetim Planı 506
A.8. Sistem Altsistem Belirtimi 508
A.9. Sistem Altsistem Tasarım Tanımlaması 510

A.10. Sistem Tümleştirme Planı 510
A.11. Sistem Test Planı 511
A.12. Sistem Aktarım Planı 512
A.13. Yazılım Kurulum Planı 512
A.14. Teknik Anlaşma 513
A.15. Yazılım Geliştirme Planı 514
A.16. Yazılım İsterleri Belirtimi 518
A.17. Yazılım Tasarım Tanımlaması 519
A.18. Arayüz İsterleri Belirtimi 520
A.19. Arayüz Tasarım Tanımlaması 521
A.20. Veri Tabanı Tasarım Tanımlaması 521
A.21. Yazılım Test Planı 521
A.22. Yazılım Test Tanımlaması 522
A.23. Yazılım Test Raporu 523
A.24. Yazılım Nitelik Güvence Planı 523
A.25. Yazılım Düzenleşim Yönetim Planı 524

**Ek B. İş Kırınım Yapısı 527**

**Ek C. Birleşik Modellere Dili UML 531**

C.1. UML Diyagramları 532
Modelleme Elemanları, Nesneler Arası İlişkiler ve Diyagramlar 533
C.2. Kullanım Senaryolar 535
C.2.1. Aktörlerin ve Kullanım Senaryolarının Bulunması 536
C.2.2. Kullanım Senaryolarının Detaylandırılması 536
C.2.3. Kullanım Senaryosu Modelinin Yapılandırılması 537
C.3. UML İle Tasarım 538
C.3.1. Yapısal Tasarım 538
C.3.2. Davranış Tasarımı 540
C.3.3. Ayrıntılı Yapısal Tasarım 541
C.3.4. Ayrıntılı Davranış Tasarımı 542
C.3.5. Ayrıntılı Sistem Mimarisi 543
C.4. Gerçekleştirim 544
C.5. Belgelendirme 545

**Ek D. Yazılım Mühendisliği Tanımları 547**

**Kaynakça 553**

**Dizin 559**

# ÖNSÖZ

Yazılımın stratejik önemi ve yazılım üretiminin karmaşık doğası yazılım mühendisliği disiplinini günümüzün en aktif araştırma ve uygulama alanlarından biri haline getirmiştir. Bu sürekli gelişen kavram ve pratiğin ulusal yazılım sektörü tarafından özümsenmesi ve kullanılması, ulusal yazılım sektörünün rekabet gücünü arttırması ve yenilikçi üretim yapabilmesi için yaşamsal önem taşımaktadır. Yazılım mühendisliğini geniş bir şekilde kapsamak, yazılımın bir program olmadığını vurgulamak, yazılımın bir sistem oluşturma disiplini ile nasıl geliştirilmesi gerektiğini anlatmak üzere bu kitabı hazırladım.

Kitabımda belirli bir yöntembilim, derleyici, yazılım paketi ya da bilgisayar ortamına bağlı kalmaksızın en genel şekilde yazılım mühendisliği temellerini açıklamaya çalıştım. Bunun için de sürekli olarak bir sistemin bütününü ele aldım. Her ne kadar bu konu üzerinde yabancı dilde çok çeşitli kaynaklar bulunsa da, yazılım mühendisliği konularına değinen, kapsamlı ve bütünüyle Türkçe olan ayrı bir eser ortaya koymak istedim.

Bazı terimlerin Türkçe karşılıkları henüz tam olarak oturmamış olduğu için anlaşılabilirliği bozmamak ve farklı kaynaklardan takibini kolaylaştırmak için Türkçeleri yanında İngilizce karşılıklarını parantez içinde vermeyi uygun buldum. Umarım terim açıklamalarında ve programlarda kullandığım Türkçe karşılıklar bir anlam karmaşası yaratmaz.

Kitabın baştan sona okunması ile yazılım mühendisliğinin ne denli karmaşık bir iş olduğu hakkında çok iyi bir fikir edinileceği kanısındayım. Özellikle yazılım geliştirme süreçleri alanında ortaya konan fikirlerin dikkate alınması ile yöneticiler, tasarımcılar ve kodlayıcılar büyük projelere daha cesurca atılabileceklerdir. Amatör olarak programcılıkla uğraşanlar da bir kez disiplinli yazılım geliştirmeye alıştıktan sonra her fırsatta bu ilkelere uymaktan ve uygulatmaktan vazgeçmeyeceklerdir.

Dr. M. Erhan SARIDOĞAN

# BÖLÜM

# 1 Giriş

Dünyamızdaki her alana bilgisayarların girmesiyle günlük yaşantımızdaki önemi de artık iyice anlaşılmıştır. Bilgisayarın yalnızca basit birkaç hesaplama yaptırmak için kullanılan bir makine olmadığı, iyi kullanıldığı takdirde birçok bakımdan insanlara hizmet ettiği bilinmektedir. Bunun için makineye insanların ne istediklerini iyi anlatabilmeleri gerekmektedir.

İnsanların bilgisayarla iletişim kurmaları amacıyla bilgisayarların programlanması çeşitli programlama dilleri ile yapılmaktadır. Bilgisayarlar geliştikçe, buna bağlı olarak da insanların istekleri ve beklentileri arttıkça, ortaya iletişimin sağlanmasının yanında başka sorunlar çıkmaktadır. İsteklerin en iyi şekilde tanımlanabilmesi, bilgisayarın donanımından en iyi şekilde yararlanılabilmesi için iyi tasarlanmış ve geliştirilmiş yazılımın çok önemli olduğu artık ortadadır.

Günümüzde yazılımın yalnızca "program kodu" yazmaktan ibaret olmadığı, hatta donanımdan daha ileri bir önem derecesine ulaştığı artık anlaşılmıştır. Bilgisayara hazır paket programlar dışında özel bir iş yaptırmak ya da "bilgisayarlı bir sistem" geliştirmek istendiği takdirde mutlaka bir yazılım geliştirilmesi zorunluluğu vardır. Yazılım bir bilgisayarda koşan, birkaç satırlık bir program olabileceği gibi, birden çok bilgisayar üzerinde koşan milyonlarca satırlık programlar grubu da olabilir. Programların karmaşıklık derecesi arttıkça geliştirilmelerini kontrol altına almak da zorlaşır. Dolayısıyla program geliştirme *yöntembilimleri*ne (methodology) gereksinim duyulmaktadır.

Özellikle 1970'li yıllarda Amerika Birleşik Devletleri'nde ortaya çıkan yazılım krizinden sonra, yazılım mühendisliğinin ve etkin programlama dillerinin önemi daha iyi anlaşılmıştır. Giderek küreselleşen ve gereksinimleri ortak hale gelen dünyada yazılım konusu ortak bir araştırma konusu olmuştur. Bu nedenle de üniversitelerde Bilgisayar Bilimleri ana bilim dalı kapsamında, veri tabanı, yapay zeka, mimari, grafik gibi, "Yazılım Mühendisliği" (Software Engineering) adı altında bir alt bilim dalı oluşmuş, bu alanda uluslararası kuruluşlar tarafından çeşitli disiplinler ve standartlar tanımlanmış, gelişmeler ortak olarak izlenmeye başlamıştır.

## 1.1. Bu Kitap Hakkında

Bu kitap, yazılımın ve yazılım mühendisliğinin önemini vurgulamakta, sistem mühendisliği penceresi ile yazılım mühendisliğine yaklaşmakta, yazılım mühendisliğinin ne olduğu konusunda bilgiler vermekte, en modern yazılım geliştirme tekniklerinden, yöntemlerden ve standartlardan bahsetmektedir. Ancak, kitapta belirli bir yöntem, standart, ticari bir paket yazılım veya bir programlama dili taban olarak alınmamış, yalnızca temel kurallar ve ilkeler üzerinde durularak çeşitli genel bilgiler verilmiştir.

Kitapta, her şeyden önce bilgisayar tabanlı sistemler tanıtılmakta, sistemlerin genel özelliklerine değinilmekte, yazılımın tek başına bir sistem olmadığı ve mutlaka bir sistem düşüncesiyle geliştirme yapılması gerektiği üzerinde durulmaktadır. Bu amaçla sistem mühendisliği, sistem geliştirme sürecinde uygulanan yöntemler ve yazılımın sistem yaşam çevrimindeki yeri üzerinde önemle durulmaktadır. Sistemi oluşturan yazılım ve donanım geliştirme etkinliklerinden yalnızca yazılım geliştirme süreçleri, yöntemleri ve nitelik unsurları anlatılmakta, yazılım yaşam çevrimi ayrıntıyla açıklanmaktadır. Tüm kitap boyunca, geliştirme ve belgelendirme standartlara dayandırılmakta, bazı belgeler için örnek şablonlar verilmektedir. Küçük yazılımlar artık herkes tarafından geliştirilebildiği için, sürekli olarak orta ve büyük ölçekli yazılımların yüksek nitelikle geliştirilmesine yönelik yöntembilim ve etkinliklerden bahsedilmektedir.

Bilgisayar mühendisliği çeşitli alt dalları bulunan çok geniş bir bilim dalıdır. Bu eserde yalnızca yazılım mühendisliği dalı üzerinde durulmaktadır. Eserde yer alan yazılım mühendisliği alanındaki konular birer ön bilgi olarak değerlendirilmeli, isterler belirtimi, yapısal tasarım, nesneye yönelik çözümleme ve tasarım, test mühendisliği gibi belirli bir konuda daha ayrıntılı bilgi almak için yalnızca o konuyu işleyen başka kaynaklara başvurulmalıdır.

## 1.2. Bu Kitabı Kimler Okumalı?

Kitap en basit program geliştiren teknik kişiden üst düzey tasarımcıya ve hatta proje yöneticisine kadar herkese yararlı olabilecek bilgiler içermektedir. Herhangi bir konuda özel bir bilgisayar programı geliştirmek isteyen “amatör programcılar” da bu kitapta kendilerine yol gösterecek birçok bilgi bulacaklardır. Aynı zamanda, profesyonelce yazılım geliştiren kişiler veya daha büyük çaplı bilgisayar sistemi geliştiren yazılım evleri çalışanları da bu kitabı bir yol gösterici olarak kullanabileceklerdir.

Bu kitap belirli bir geliştirme yöntemini ele alıp onu tüm ayrıntılarıyla anlatmak yerine, ortak noktaları, genel tanımlamaları anlatmakta, başlangıç yerine geçebilecek pratik bilgiler vermekte, çeşitli belge şablonları sunmaktadır. Bu bilgiler edinildikten sonra, yazılım mühendisliği görevini yürüten kişiler tarafından, özellikle büyük çaplı proje gruplarında yer alması gereken bir Nitelik Güvence Grubu oluşturulmalı, bu grup tarafından kabul edilip uygulamaya konulan bir geliştirme yöntemin tam bir disiplinle uygulanması gereklidir. Unutulmaması gereken en önemli nokta, standardı veya yöntemi tanımlayıp ortaya koymaktan çok onu uygulamak ve uygulatmaktır.

## 1.3. Kitapta Bulunan Konular

Kitap toplam 14 bölüm ve eklerden oluşmuştur. Herbir bölümde ele alınan konular aşağıda kısaca özetlenmektedir:

- **Bölüm 1: Giriş**
  Bu ilk kısımda kitabın yazılış amacı açıklanmakta ve kimlerin yararlanabileceği belirtilmektedir. Kitapta yer alan bölümlerin kısa açıklamaları da bu bölümde yer almaktadır.
- **Bölüm 2: Bilgisayar Sistemleri**
  Bu bölümde bilgisayar sisteminin ne olduğu anlatılmakta, bilgi sistemlerinin çeşitleri hakkında özet bilgiler verilmektedir. Bir sistemi oluşturan ana bileşenlerin neler olduğu açıklanmakta, bilgi işleme mantığı ve özellikle paralel bilgi işleme üzerinde durulmakta, temel yazılım mimarileri anlatılmaktadır.
- **Bölüm 3: Bilgisayar Sistemi Mühendisliği**
  Bu bölümde bilgisayar sistemi mühendisliğinin neleri kapsadığı bir proje yaşam çevrimi şeklinde açıklanmakta, sistem çözümleme, tasarım, gerçekleştirim, kurulum, test, teslim ve bakım aşamaları açıklanmaktadır.
- **Bölüm 4: Yazılım Mühendisliği**
  Bu bölümde yazılımın ne olduğu, tarihçesi ve temel özellikleriyle beraber anlatılmakta, yazılım mühendisliğinin neleri içerdiği açıklanmaktadır. Yazılım mühendisliğinde en çok kullanılan yöntembilimler tanıtılmakta ve yardımcı araçlar hakkında kısa bilgiler verilmektedir.
- **Bölüm 5: Yazılım İsterleri Çözümlemesi**
  Bu bölüm, yazılım geliştirme sürecinin bir aşaması olan isterler çözümleme aşamasında neler yapılması gerektiğini, ister belirleme ve çözümleme yöntemleri ile belgelendirme konularını kapsamaktadır.
- **Bölüm 6: Yazılım Tasarımı**
  Bu bölümde yazılım tasarım sürecinin neleri kapsadığı hakkında bilgi verilmekte, yaygın tasarım yöntemleri ve tasarım kalıpları anlatılmakta, kullanıcı arayüz tasarımına, gerçek zamanlı sistem tasarımlarına, veri tabanı ve İnternet uygulamaları ile ilgili tasarımlara ayrıntılı bir şekilde değinilmekte, tasarımda uygulanması önerilen kurallar ile belgelendirme konusu ortaya konmaktadır.
- **Bölüm 7: Yazılım Gerçekleştirimi**
  Bu bölüm, programlama dillerinin tarihçesi hakkında bilgi vermekte, kod çevriminin nasıl yapıldığına değinmekte, kodlama biçimleri ve kod etkinliği üzerinde durmakta, temel ilkeler ve belgelendirme konusunu anlatmaktadır.
- **Bölüm 8: Yazılım Testi**
  Yazılım testinin temel özellikleri, test stratejileri, yeterlilik, sistem ve kabul testlerinin neler olduğu, bu testlerin nasıl yapılacağı, belgelendirme ve test yönetimi ile hata ayıklama konusu bu bölümde anlatılmaktadır.

- **Bölüm 9: Yazılım Bakımı**
  Bu bölümde yazılım bakımının nasıl yapılacağı, aşamaları, bakım kolaylığının nasıl sağlanabileceği, bakımın yan etkilerinin neler olduğu, belgelendirilmemiş yazılımların bakımının nasıl yapılacağı konularına değinilmektedir.
- **Bölüm 10: Yazılım Nitelik Güvencesi**
  Bu bölüm, yazılımın niteliğinin ne olduğu, nitelik güvence gereksinimi, süreç modelleri, en yaygın nitelik sistem standartları hakkında bilgiler vermekte; gözden geçirme etkinliklerinin nasıl yapılacağını, resmi nitelik güvence yöntemlerinin neler olduğunu, yazılım güvenilirliğinin ne olduğunu, nitelikli bir sistem aktarımın nasıl yapılması gerektiğini anlatmakta; nitelik güvence yönetiminin nasıl yürütüleceğini belirtmekte, niteliği ölçme unsurları, metrik toplama ve değerlendirme hakkında bilgiler içermekte, yazılım niteliğini artırmak için pratik öneriler vermektedir.
- **Bölüm 11: Düzenleşim Yönetimi**
  Sistem ve yazılım düzenleşim yönetiminin ne olduğu, ne gibi süreçlerin izleneceği, düzenleşim denetimi ile geliştirmenin ve ürün çoğaltmanın nasıl yapılacağı bu bölümde anlatılmaktadır.
- **Bölüm 12: Yazılım Geliştirme Yöntembilimleri**
  Bu bölümde, yazılım geliştirme standartlarının tarihçesi anlatılmakta, en son standartlar hakkında ayrıntılı ve karşılaştırmalı bilgi verilmekte, yaygın olarak kullanılan standartlara göre biçimlendirilmiş bir yazılım geliştirme sürecinin tüm aşamaları ayrıntıyla açıklanmakta ve hazırlanması gereken belgeler verilmektedir.
- **Bölüm 13: Proje Yönetimi**
  Bu bölümde proje ve program tanımları yapılmakta, proje yönetimi içinde yer alan tümleştirme, kapsam, zaman, maliyet, nitelik, insan kaynakları, iletişim, risk ve edinme yönetimleri açıklanmakta, proje örgütlenmesinin nasıl yapılması gerektiği hakkında bilgi verilmektedir.
- **Bölüm 14: Yazılım Geliştirme İçin Pratik Öneriler**
  Bu bölümde, yönetsel, süreçlere yönelik ve geliştirmeye yönelik uygulama önerileri, kullanışlı bir sistem geliştirme için pratik öneriler verilmekte, toplantılarda uyulmasında yarar görülen kurallar belirtilmektedir.
- **Ekler**
  Kitabın ekleri arasında belge şablonları, iş kırınım yapısı örneği, UML ve bilgi teknolojisinde kullanılan bazı temel tanımların listesi yer almaktadır.

Her bölüm sonunda o konuyla ilgili çeşitli sorular verilmiştir. Basılı kaynaklar kitabın sonuna eklenmiştir. Değişebilir olmaları nedeniyle İnternet siteleri Kaynakça içinde verilmemiştir!