

**Roger S. Pressman • Bruce R. Maxim**

# Engenharia de Software

UMA ABORDAGEM PROFISSIONAL

8ª EDIÇÃO



**Mc  
Graw  
Hill**  
Education



# Sumário

---

## **CAPÍTULO 1 A natureza do software 1**

- 1.1 A natureza do software 3
    - 1.1.1 Definição de software 4
    - 1.1.2 Campos de aplicação de software 6
    - 1.1.3 Software legado 7
  - 1.2 A natureza mutante do software 9
    - 1.2.1 WebApps 9
    - 1.2.2 Aplicativos móveis 9
    - 1.2.3 Computação em nuvem 10
    - 1.2.4 Software para linha de produtos (de software) 11
  - 1.3 Resumo 11
- Problemas e pontos a ponderar 12
- Leituras e fontes de informação complementares 12

## **CAPÍTULO 2 Engenharia de software 14**

- 2.1 Definição da disciplina 15
  - 2.2 O processo de software 16
    - 2.2.1 A metodologia do processo 17
    - 2.2.2 Atividades de apoio 18
    - 2.2.3 Adaptação do processo 18
  - 2.3 A prática da engenharia de software 19
    - 2.3.1 A essência da prática 19
    - 2.3.2 Princípios gerais 21
  - 2.4 Mitos do desenvolvimento de software 23
  - 2.5 Como tudo começa 26
  - 2.6 Resumo 27
- Problemas e pontos a ponderar 27
- Leituras e fontes de informação complementares 27

## **PARTE I O processo de software 29**

---

### **CAPÍTULO 3 Estrutura do processo de software 30**

- 3.1 Um modelo de processo genérico 31
  - 3.2 Definição de uma atividade metodológica 32
  - 3.3 Identificação de um conjunto de tarefas 34
  - 3.4 Padrões de processo 34
  - 3.5 Avaliação e aperfeiçoamento de processos 37
  - 3.6 Resumo 38
- Problemas e pontos a ponderar 38
- Leituras e fontes de informação complementares 39

## **CAPÍTULO 4 Modelos de processo 40**

- 4.1 Modelos de processo prescritivo 41
    - 4.1.1 O modelo cascata 41
    - 4.1.2 Modelos de processo incremental 43
    - 4.1.3 Modelos de processo evolucionário 44
    - 4.1.4 Modelos concorrentes 49
    - 4.1.5 Um comentário final sobre processos evolucionários 51
  - 4.2 Modelos de processo especializado 52
    - 4.2.1 Desenvolvimento baseado em componentes 52
    - 4.2.2 O modelo de métodos formais 53
    - 4.2.3 Desenvolvimento de software orientado a aspectos 54
  - 4.3 O processo unificado 55
    - 4.3.1 Um breve histórico 56
    - 4.3.2 Fases do processo unificado 56
  - 4.4 Modelos de processo pessoal e de equipe 58
    - 4.4.1 Processo de Software Pessoal 59
    - 4.4.2 Processo de Software de Equipe 60
  - 4.5 Tecnologia de processos 61
  - 4.6 Produto e processo 62
  - 4.7 Resumo 63
- Problemas e pontos a ponderar 64
- Leituras e fontes de informação complementares 65

## **CAPÍTULO 5 Desenvolvimento ágil 66**

- 5.1 O que é agilidade? 68
  - 5.2 Agilidade e o custo das mudanças 68
  - 5.3 O que é processo ágil? 69
    - 5.3.1 Princípios da agilidade 70
    - 5.3.2 A política do desenvolvimento ágil 71
  - 5.4 Extreme programming – XP (Programação Extrema) 72
    - 5.4.1 O processo XP 72
    - 5.4.2 Industrial XP 76
  - 5.5 Outros modelos de processos ágeis 77
    - 5.5.1 Scrum 78
    - 5.5.2 Método de Desenvolvimento de Sistemas Dinâmicos (DSDM) 79
    - 5.5.3 Modelagem Ágil (AM) 80
    - 5.5.4 Processo Unificado Ágil 82
  - 5.6 Um conjunto de ferramentas para o processo ágil 83
  - 5.7 Resumo 84
- Problemas e pontos a ponderar 85
- Leituras e fontes de informação complementares 85

## **CAPÍTULO 6 Aspectos humanos da engenharia de software 87**

- 6.1 Características de um engenheiro de software 88
- 6.2 A psicologia da engenharia de software 89
- 6.3 A equipe de software 90
- 6.4 Estruturas de equipe 92
- 6.5 Equipes ágeis 93
  - 6.5.1 A equipe ágil genérica 93
  - 6.5.2 A equipe XP 94
- 6.6 O impacto da mídia social 95

- 6.7 Engenharia de software usando a nuvem 97
- 6.8 Ferramentas de colaboração 98
- 6.9 Equipes globais 99
- 6.10 Resumo 100
- Problemas e pontos a ponderar 101
- Leituras e fontes de informação complementares 102

## **PARTE II Modelagem 103**

### **CAPÍTULO 7 Princípios que orientam a prática 104**

- 7.1 Conhecimento da engenharia de software 105
- 7.2 Princípios fundamentais 106
  - 7.2.1 Princípios que orientam o processo 106
  - 7.2.2 Princípios que orientam a prática 107
- 7.3 Princípios das atividades metodológicas 109
  - 7.3.1 Princípios da comunicação 110
  - 7.3.2 Princípios do planejamento 112
  - 7.3.3 Princípios da modelagem 114
  - 7.3.4 Princípios da construção 121
  - 7.3.5 Princípios da disponibilização 124
- 7.4 Formas de trabalhar 126
- 7.5 Resumo 127
- Problemas e pontos a ponderar 128
- Leituras e fontes de informação complementares 129

### **CAPÍTULO 8 Entendendo os requisitos 131**

- 8.1 Engenharia de requisitos 132
- 8.2 Estabelecimento da base de trabalho 138
  - 8.2.1 Identificação de envolvidos 139
  - 8.2.2 Reconhecimento de diversos pontos de vista 139
  - 8.2.3 Trabalho em busca da colaboração 140
  - 8.2.4 Questões iniciais 140
  - 8.2.5 Requisitos não funcionais 141
  - 8.2.6 Rastreabilidade 142
- 8.3 Levantamento de requisitos 142
  - 8.3.1 Coleta colaborativa de requisitos 143
  - 8.3.2 Aplicação da qualidade por QFD (Quality Function Deployment) 146
  - 8.3.3 Cenários de uso 146
  - 8.3.4 Artefatos do levantamento de requisitos 147
  - 8.3.5 Levantamento de requisitos ágil 148
  - 8.3.6 Métodos orientados a serviços 148
- 8.4 Desenvolvimento de casos de uso 149
- 8.5 Construção do modelo de análise 154
  - 8.5.1 Elementos do modelo de análise 154
  - 8.5.2 Padrões de análise 157
  - 8.5.3 Engenharia de requisitos ágil 158
  - 8.5.4 Requisitos de sistemas autoadaptativos 158
- 8.6 Negociação de requisitos 159
- 8.7 Monitoramento de requisitos 160
- 8.8 Validação dos requisitos 161

- 8.9 Evite erros comuns 162
- 8.10 Resumo 162
- Problemas e pontos a ponderar 163
- Leituras complementares e outras fontes de informação 164

## **CAPÍTULO 9 Modelagem de requisitos: métodos baseados em cenários 166**

- 9.1 Análise de requisitos 167
  - 9.1.1 Filosofia e objetivos gerais 168
  - 9.1.2 Regras práticas para a análise 168
  - 9.1.3 Análise de domínio 169
  - 9.1.4 Abordagens de modelagem de requisitos 171
- 9.2 Modelagem baseada em cenários 173
  - 9.2.1 Criação de um caso de uso preliminar 173
  - 9.2.2 Refinamento de um caso de uso preliminar 176
  - 9.2.3 Criação de um caso de uso formal 177
- 9.3 Modelos UML que complementam o caso de uso 179
  - 9.3.1 Desenvolvimento de um diagrama de atividades 179
  - 9.3.2 Diagramas de raios 180
- 9.4 Resumo 182
- Problemas e pontos a ponderar 182
- Leituras e fontes de informação complementares 183

## **CAPÍTULO 10 Modelagem de requisitos: métodos baseados em classes 184**

- 10.1 Identificação de classes de análise 185
- 10.2 Especificação de atributos 188
- 10.3 Definição das operações 189
- 10.4 Modelagem classe-responsabilidade-colaborador 192
- 10.5 Associações e dependências 198
- 10.6 Pacotes de análise 199
- 10.7 Resumo 200
- Problemas e pontos a ponderar 201
- Leituras e fontes de informação complementares 201

## **CAPÍTULO 11 Modelagem de requisitos: comportamento, padrões e WebApps/aplicativos móveis 202**

- 11.1 Criação de um modelo comportamental 203
- 11.2 Identificação de eventos com o caso de uso 203
- 11.3 Representações de estados 204
- 11.4 Padrões para a modelagem de requisitos 207
  - 11.4.1 Descoberta de padrões de análise 208
  - 11.4.2 Exemplo de padrão de requisitos: Atuador-Sensor 209
- 11.5 Modelagem de requisitos para WebApps e aplicativos móveis 213
  - 11.5.1 Que nível de análise é suficiente? 214
  - 11.5.2 Entrada da modelagem de requisitos 214
  - 11.5.3 Saída da modelagem de requisitos 215
  - 11.5.4 Modelo de conteúdo 216
  - 11.5.5 Modelo de interação para WebApps e aplicativos móveis 217
  - 11.5.6 Modelo funcional 218
  - 11.5.7 Modelo de configuração para WebApps 219
  - 11.5.8 Modelo de navegação 220

11.6 Resumo 221

Problemas e pontos a ponderar 222

Leituras e fontes de informação complementares 222

## **CAPÍTULO 12** Conceitos de projeto 224

12.1 Projeto no contexto da engenharia de software 225

12.2 O processo de projeto 228

12.2.1 Diretrizes e atributos da qualidade de software 228

12.2.2 A evolução de um projeto de software 230

12.3 Conceitos de projeto 231

12.3.1 Abstração 232

12.3.2 Arquitetura 232

12.3.3 Padrões 233

12.3.4 Separação por interesses (por afinidades) 234

12.3.5 Modularidade 234

12.3.6 Encapsulamento de informações 235

12.3.7 Independência funcional 236

12.3.8 Refinamento 237

12.3.9 Aspectos 237

12.3.10 Refatoração 238

12.3.11 Conceitos de projeto orientado a objetos 238

12.3.12 Classes de projeto 239

12.3.13 Inversão da dependência 241

12.3.14 Projeto para teste 242

12.4 O modelo de projeto 243

12.4.1 Elementos de projeto de dados 244

12.4.2 Elementos do projeto de arquitetura 244

12.4.3 Elementos do projeto de interface 245

12.4.4 Elementos do projeto de componentes 247

12.4.5 Elementos do projeto de implantação 247

12.5 Resumo 249

Problemas e pontos a ponderar 250

Leituras e fontes de informação complementares 250

## **CAPÍTULO 13** Projeto de arquitetura 252

13.1 Arquitetura de software 253

13.1.1 O que é arquitetura? 253

13.1.2 Por que a arquitetura é importante? 254

13.1.3 Descrições de arquitetura 255

13.1.4 Decisões de arquitetura 256

13.2 Gêneros de arquitetura 257

13.3 Estilos de arquitetura 258

13.3.1 Uma breve taxonomia dos estilos de arquitetura 258

13.3.2 Padrões de arquitetura 263

13.3.3 Organização e refinamento 263

13.4 Considerações sobre a arquitetura 264

13.5 Decisões sobre a arquitetura 266

13.6 Projeto de arquitetura 267

13.6.1 Representação do sistema no contexto 267

13.6.2 Definição de arquétipos 269

13.6.3 Refinamento da arquitetura em componentes 270

13.6.4 Descrição das instâncias do sistema 272

13.6.5	Projeto de arquitetura para aplicações web (WebApps)	273
13.6.6	Projeto de arquitetura para aplicativos móveis	274
13.7	Avaliação das alternativas de projeto de arquitetura	274
13.7.1	Linguagens de descrição da arquitetura	276
13.7.2	Revisões da arquitetura	277
13.8	Lições aprendidas	278
13.9	Revisão de arquitetura baseada em padrões	278
13.10	Verificação de conformidade da arquitetura	279
13.11	Agilidade e arquitetura	280
13.12	Resumo	282
	Problemas e pontos a ponderar	282
	Leituras e fontes de informação complementares	283

## **CAPÍTULO 14 Projeto de componentes 285**

14.1	O que é componente?	286
14.1.1	Uma visão orientada a objetos	286
14.1.2	A visão tradicional	288
14.1.3	Uma visão relacionada a processos	291
14.2	Projeto de componentes baseados em classes	291
14.2.1	Princípios básicos de projeto	292
14.2.2	Diretrizes para o projeto de componentes	295
14.2.3	Coesão	296
14.2.4	Acoplamento	298
14.3	Condução de projetos de componentes	299
14.4	Projeto de componentes para WebApps	305
14.4.1	Projeto de conteúdo para componentes	305
14.4.2	Projeto funcional para componentes	306
14.5	Projeto de componentes para aplicativos móveis	306
14.6	Projeto de componentes tradicionais	307
14.7	Desenvolvimento baseado em componentes	308
14.7.1	Engenharia de domínio	308
14.7.2	Qualificação, adaptação e composição de componentes	309
14.7.3	Divergência arquitetural	311
14.7.4	Análise e projeto para reutilização	312
14.7.5	Classificação e recuperação de componentes	312
14.8	Resumo	313
	Problemas e pontos a ponderar	315
	Leituras e fontes de informação complementares	316

## **CAPÍTULO 15 Projeto de interfaces do usuário 317**

15.1	As regras de ouro	318
15.1.1	Deixar o usuário no comando	318
15.1.2	Reduzir a carga de memória do usuário	319
15.1.3	Tornar a interface consistente	321
15.2	Análise e projeto de interfaces	322
15.2.1	Modelos de análise e projeto de interfaces	322
15.2.2	O processo	323
15.3	Análise de interfaces	325
15.3.1	Análise de usuários	325
15.3.2	Análise e modelagem de tarefas	326
15.3.3	Análise do conteúdo exibido	331
15.3.4	Análise do ambiente de trabalho	331

15.4	Etapas no projeto de interfaces	332
15.4.1	Aplicação das etapas para projeto de interfaces	332
15.4.2	Padrões de projeto de interfaces do usuário	334
15.4.3	Questões de projeto	335
15.5	Projeto de interfaces para WebApps e aplicativos móveis	337
15.5.1	Princípios e diretrizes para projeto de interfaces	337
15.5.2	Fluxo de trabalho de projeto de interfaces para WebApps e aplicativos móveis	341
15.6	Avaliação de projeto	342
15.7	Resumo	344
	Problemas e pontos a ponderar	345
	Leituras e fontes de informação complementares	346

## **CAPÍTULO 16 Projeto baseado em padrões 347**

16.1	Padrões de projeto	348
16.1.1	Tipos de padrões	349
16.1.2	Frameworks	351
16.1.3	Descrição de padrões	352
16.1.4	Linguagens e repositórios de padrões	353
16.2	Projeto de software baseado em padrões	354
16.2.1	Contexto do projeto baseado em padrões	354
16.2.2	Pense em termos de padrões	354
16.2.3	Tarefas de projeto	356
16.2.4	Construção de uma tabela para organização de padrões	358
16.2.5	Erros comuns de projeto	359
16.3	Padrões de arquitetura	359
16.4	Padrões de projeto de componentes	360
16.5	Padrões de projeto para interfaces do usuário	362
16.6	Padrões de projeto para WebApps	364
16.6.1	Foco do projeto	365
16.6.2	Granularidade do projeto	365
16.7	Padrões para aplicativos móveis	366
16.8	Resumo	367
	Problemas e pontos a ponderar	368
	Leituras e fontes de informação complementares	369

## **CAPÍTULO 17 Projeto de WebApps 371**

17.1	Qualidade de projeto em WebApps	372
17.2	Objetivos de projeto	374
17.3	Uma pirâmide de projeto para WebApps	375
17.4	Projeto de interfaces para WebApp	376
17.5	Projeto estético	377
17.5.1	Questões de layout	378
17.5.2	Questões de design gráfico	378
17.6	Projeto de conteúdo	379
17.6.1	Objetos de conteúdo	379
17.6.2	Questões de projeto de conteúdo	380
17.7	Projeto de arquitetura	381
17.7.1	Arquitetura de conteúdo	381
17.7.2	Arquitetura de uma WebApp	384
17.8	Projeto da navegação	385
17.8.1	Semântica de navegação	385
17.8.2	Sintaxe de navegação	387



- 17.9 Projeto em nível de componentes 387
- 17.10 Resumo 388
- Problemas e pontos a ponderar 389
- Leituras e fontes de informação complementares 389

## **CAPÍTULO 18 Projeto de aplicativos móveis 391**

- 18.1 Os desafios 392
  - 18.1.1 Considerações sobre o desenvolvimento 392
  - 18.1.2 Considerações técnicas 393
- 18.2 Desenvolvimento de aplicativos móveis 395
  - 18.2.1 Qualidade do aplicativo móvel 397
  - 18.2.2 Projeto de interface de usuário 398
  - 18.2.3 Aplicativos sensíveis ao contexto 398
  - 18.2.4 Lições aprendidas 400
- 18.3 Projeto de aplicativos móveis – boas práticas 401
- 18.4 Ambientes de mobilidade 403
- 18.5 A nuvem 405
- 18.6 A aplicabilidade da engenharia de software convencional 407
- 18.7 Resumo 408
- Problemas e pontos a ponderar 409
- Leituras e fontes de informação complementares 410

## **PARTE III Gestão da qualidade 411**

---

### **CAPÍTULO 19 Conceitos de qualidade 412**

- 19.1 O que é qualidade? 413
- 19.2 Qualidade de software 414
  - 19.2.1 Dimensões de qualidade de Garvin 415
  - 19.2.2 Fatores de qualidade de McCall 416
  - 19.2.3 Fatores de qualidade ISO 9126 418
  - 19.2.4 Fatores de qualidade desejados 418
  - 19.2.5 A transição para uma visão quantitativa 420
- 19.3 O dilema da qualidade do software 420
  - 19.3.1 Software “bom o suficiente” 421
  - 19.3.2 Custo da qualidade 422
  - 19.3.3 Riscos 424
  - 19.3.4 Negligência e responsabilidade civil 425
  - 19.3.5 Qualidade e segurança 425
  - 19.3.6 O impacto das ações administrativas 426
- 19.4 Alcançando a qualidade de software 427
  - 19.4.1 Métodos de engenharia de software 427
  - 19.4.2 Técnicas de gerenciamento de software 427
  - 19.4.3 Controle de qualidade 428
  - 19.4.4 Garantia da qualidade 428
- 19.5 Resumo 428
- Problemas e pontos a ponderar 429
- Leituras e fontes de informação complementares 430

**CAPÍTULO 20 Técnicas de revisão 431**

- 20.1 Impacto de defeitos de software nos custos 432
  - 20.2 Amplificação e eliminação de defeitos 433
  - 20.3 Métricas de revisão e seu emprego 435
    - 20.3.1 Análise de métricas 435
    - 20.3.2 Eficácia dos custos de revisões 436
  - 20.4 Revisões: um espectro de formalidade 438
  - 20.5 Revisões informais 439
  - 20.6 Revisões técnicas formais 441
    - 20.6.1 A reunião de revisão 441
    - 20.6.2 Relatório de revisão e manutenção de registros 442
    - 20.6.3 Diretrizes de revisão 442
    - 20.6.4 Revisões por amostragem 444
  - 20.7 Avaliações post-mortem 445
  - 20.8 Resumo 446
- Problemas e pontos a ponderar 446
- Leituras e fontes de informação complementares 447

**CAPÍTULO 21 Garantia da qualidade de software 448**

- 21.1 Plano de fundo 449
  - 21.2 Elementos de garantia da qualidade de software 450
  - 21.3 Processos da SQA e características do produto 452
  - 21.4 Tarefas, metas e métricas da SQA 452
    - 21.4.1 Tarefas da SQA 453
    - 21.4.2 Metas, atributos e métricas 454
  - 21.5 Abordagens formais da SQA 455
  - 21.6 Estatística da garantia da qualidade de software 456
    - 21.6.1 Um exemplo genérico 456
    - 21.6.2 Seis Sigma para engenharia de software 458
  - 21.7 Confiabilidade de software 459
    - 21.7.1 Medidas de confiabilidade e disponibilidade 459
    - 21.7.2 Segurança do software 460
  - 21.8 Os padrões de qualidade ISO 9000 461
  - 21.9 O plano de SQA 463
  - 21.10 Resumo 463
- Problemas e pontos a ponderar 464
- Leituras e fontes de informação complementares 464

**CAPÍTULO 22 Estratégias e teste de software 466**

- 22.1 Uma abordagem estratégica do teste de software 466
  - 22.1.1 Verificação e validação 467
  - 22.1.2 Organizando o teste de software 468
  - 22.1.3 Estratégia de teste de software – visão global 469
  - 22.1.4 Critérios para conclusão do teste 472
- 22.2 Problemas estratégicos 472
- 22.3 Estratégias de teste para software convencional 473
  - 22.3.1 Teste de unidade 473
  - 22.3.2 Teste de integração 475
- 22.4 Estratégias de teste para software orientado a objetos 481
  - 22.4.1 Teste de unidade em contexto orientado a objetos 481
  - 22.4.2 Teste de integração em contexto orientado a objetos 481

- 22.5 Estratégias de teste para WebApps 482
- 22.6 Estratégias de teste para aplicativos móveis 483
- 22.7 Teste de validação 483
  - 22.7.1 Critérios de teste de validação 484
  - 22.7.2 Revisão da configuração 484
  - 22.7.3 Testes alfa e beta 484
- 22.8 Teste de sistema 486
  - 22.8.1 Teste de recuperação 486
  - 22.8.2 Teste de segurança 486
  - 22.8.3 Teste por esforço 487
  - 22.8.4 Teste de desempenho 487
  - 22.8.5 Teste de disponibilização 488
- 22.9 A arte da depuração 488
  - 22.9.1 O processo de depuração 489
  - 22.9.2 Considerações psicológicas 490
  - 22.9.3 Estratégias de depuração 490
  - 22.9.4 Correção do erro 493
- 22.10 Resumo 493
- Problemas e pontos a ponderar 494
- Leituras e fontes de informação complementares 494

## **CAPÍTULO 23 Teste de aplicativos convencionais 496**

- 23.1 Fundamentos do teste de software 497
- 23.2 Visões interna e externa do teste 499
- 23.3 Teste caixa-branca 500
- 23.4 Teste do caminho básico 500
  - 23.4.1 Notação de grafo de fluxo 500
  - 23.4.2 Caminhos de programa independentes 502
  - 23.4.3 Derivação de casos de teste 504
  - 23.4.4 Matrizes de grafos 506
- 23.5 Teste de estrutura de controle 507
- 23.6 Teste caixa-preta 509
  - 23.6.1 Métodos de teste baseados em grafos 509
  - 23.6.2 Particionamento de equivalência 511
  - 23.6.3 Análise de valor limite 512
  - 23.6.4 Teste de matriz ortogonal 513
- 23.7 Teste baseado em modelos 516
- 23.8 Teste da documentação e dos recursos de ajuda 516
- 23.9 Teste para sistemas em tempo real 517
- 23.10 Padrões para teste de software 519
- 23.11 Resumo 520
- Problemas e pontos a ponderar 521
- Leituras e fontes de informação complementares 521

## **CAPÍTULO 24 Teste de aplicações orientadas a objeto 523**

- 24.1 Ampliando a visão do teste 524
- 24.2 Teste de modelos de análise e de projeto orientados a objetos 525
  - 24.2.1 Exatidão dos modelos de OOA e OOD 525
  - 24.2.2 Consistência dos modelos orientados a objetos 526

