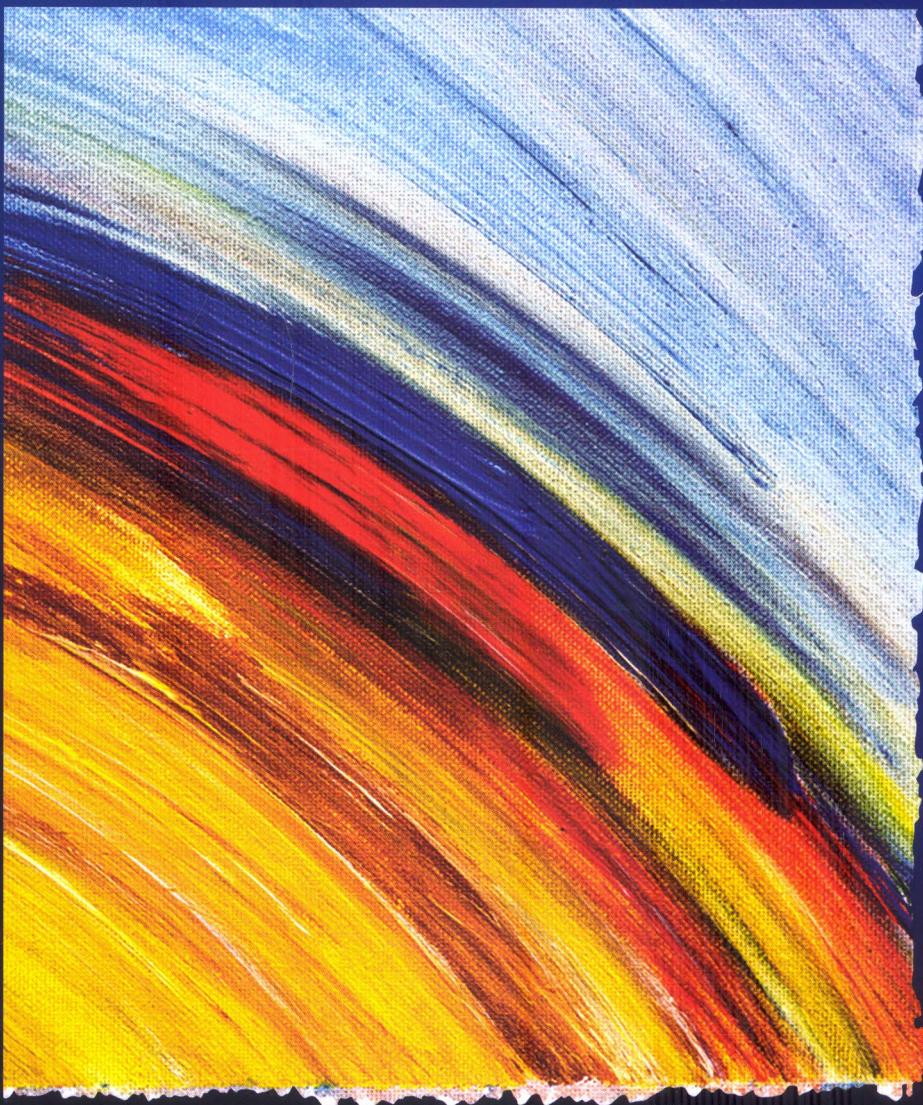


**Roger S. Pressman • Bruce R. Maxim**

# **Engenharia de Software**

**UMA ABORDAGEM PROFISSIONAL**

**8<sup>a</sup> 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

---

1	6.7	Engenharia de software usando a nuvem	97
Projetos	6.8	Ferramentas de colaboração	98
Ler	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	Ánalise 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 raias	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

- CAPÍTULO 15** Etapas no projeto de interfaces 332
- 20.1 15.4.1 Aplicação das etapas para projeto de interfaces 332
  - 20.2 15.4.2 Padrões de projeto de interfaces do usuário 334
  - 20.3 15.4.3 Questões de projeto 335
- 15.5 Projeto de interfaces para WebApps e aplicativos móveis 337
- 20.4 15.5.1 Princípios e diretrizes para projeto de interfaces 337
  - 20.5 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

- 20.6 16.1 Padrões de projeto 348
- 20.7 16.1.1 Tipos de padrões 349
  - 20.8 16.1.2 Frameworks 351
  - 20.9 16.1.3 Descrição de padrões 352
  - 20.10 16.1.4 Linguagens e repositórios de padrões 353
- 16.2 Projeto de software baseado em padrões 354
- 21.1 16.2.1 Contexto do projeto baseado em padrões 354
  - 21.2 16.2.2 Pense em termos de padrões 354
  - 21.3 16.2.3 Tarefas de projeto 356
  - 21.4 16.2.4 Construção de uma tabela para organização de padrões 358
  - 21.5 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
- 21.6 16.6.1 Foco do projeto 365
  - 21.7 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

- 21.8 17.1 Qualidade de projeto em WebApps 372
- 21.9 17.2 Objetivos de projeto 374
- 21.10 17.3 Uma pirâmide de projeto para WebApps 375
- 21.11 17.4 Projeto de interfaces para WebApp 376
- 21.12 17.5 Projeto estético 377
- 21.13 17.5.1 Questões de layout 378
  - 21.14 17.5.2 Questões de design gráfico 378
- 21.15 17.6 Projeto de conteúdo 379
- 21.16 17.6.1 Objetos de conteúdo 379
  - 21.17 17.6.2 Questões de projeto de conteúdo 380
- 21.18 17.7 Projeto de arquitetura 381
- 21.19 17.7.1 Arquitetura de conteúdo 381
  - 21.20 17.7.2 Arquitetura de uma WebApp 384
- 21.21 17.8 Projeto da navegação 385
- 21.22 17.8.1 Semântica de navegação 385
  - 21.23 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

24.3	Estratégias de teste orientado a objetos	528
24.3.1	Teste de unidade em contexto orientado a objetos	528
24.3.2	Teste de integração em contexto orientado a objetos	529
24.3.3	Teste de validação em contexto orientado a objetos	529
24.4	Métodos de teste orientados a objetos	529
24.4.1	As implicações dos conceitos de orientação a objetos no projeto de casos de teste	530
24.4.2	Aplicabilidade dos métodos convencionais de projeto de casos de teste	531
24.4.3	Teste baseado em falhas	531
24.4.4	Projeto de teste baseado em cenários	532
24.5	Métodos de teste aplicáveis no nível de classe	532
24.5.1	Teste aleatório para classes orientadas a objetos	532
24.5.2	Teste de partição em nível de classe	533
24.6	Projeto de caso de teste entre classes	534
24.6.1	Teste de múltiplas classes	534
24.6.2	Testes derivados de modelos comportamentais	536
24.7	Resumo	537
	Problemas e pontos a ponderar	538
	Leituras e fontes de informação complementares	538

## CAPÍTULO 25 Teste de aplicações para Web 540

25.1	Conceitos de teste para WebApps	541
25.1.1	Dimensões da qualidade	541
25.1.2	Erros em um ambiente WebApp	542
25.1.3	Estratégia de teste	543
25.1.4	Planejamento de teste	543
25.2	O processo de teste – uma visão geral	544
25.3	Teste de conteúdo	545
25.3.1	Objetivos do teste de conteúdo	545
25.3.2	Teste de banco de dados	546
25.4	Teste da interface do usuário	549
25.4.1	Estratégia de teste de interface	549
25.4.2	Teste de mecanismos de interface	549
25.4.3	Teste da semântica da interface	551
25.4.4	Testes de usabilidade	552
25.4.5	Testes de compatibilidade	553
25.5	Teste no nível de componente	554
25.6	Testes de navegação	555
25.6.1	Teste da sintaxe de navegação	556
25.6.2	Teste da semântica de navegação	556
25.7	Teste de configuração	558
25.7.1	Tópicos no lado do servidor	558
25.7.2	Tópicos no lado do cliente	559
25.8	Teste de segurança	559
25.9	Teste de desempenho	560
25.9.1	Objetivos do teste de desempenho	561
25.9.2	Teste de carga	562
25.9.3	Teste de esforço	562
25.10	Resumo	563
	Problemas e pontos a ponderar	564
	Leituras e fontes de informação complementares	565

**CAPÍTULO 26 Teste de aplicativos móveis 567**

- 26.1 Diretrizes de teste 568
  - 26.2 As estratégias de teste 569
    - 26.2.1 As estratégias convencionais são adequadas? 570
    - 26.2.2 A necessidade de automação 571
    - 26.2.3 Construção de uma matriz de teste 572
    - 26.2.4 Teste de esforço (stress) 573
    - 26.2.5 Testes em um ambiente de produção 573
  - 26.3 Considerações sobre o espectro da interação do usuário 574
    - 26.3.1 Teste de gestos 575
    - 26.3.2 Entrada e reconhecimento de voz 576
    - 26.3.3 Entrada por teclado virtual 577
    - 26.3.4 Alertas e condições extraordinárias 577
  - 26.4 Teste além de fronteiras 578
  - 26.5 Problemas do teste em tempo real 578
  - 26.6 Ferramentas e ambientes de teste 579
  - 26.7 Resumo 581
- Problemas e pontos a ponderar 582  
Leituras e fontes de informação complementares 582

**CAPÍTULO 27 Engenharia de segurança 584**

- 27.1 Análise dos requisitos de segurança 585
  - 27.2 Segurança e privacidade em um mundo online 586
    - 27.2.1 Mídia social 587
    - 27.2.2 Aplicativos móveis 587
    - 27.2.3 Computação em nuvem 587
    - 27.2.4 A Internet das coisas 588
  - 27.3 Análise da engenharia de segurança 588
    - 27.3.1 Levantamento de requisitos de segurança 589
    - 27.3.2 Modelagem de segurança 590
    - 27.3.3 Projeto de medidas 591
    - 27.3.4 Verificações de exatidão 591
  - 27.4 Garantia da segurança 592
    - 27.4.1 O processo da garantia da segurança 592
    - 27.4.2 Organização e gerenciamento 593
  - 27.5 Análise de risco de segurança 594
  - 27.6 A função das atividades da engenharia de software convencional 595
  - 27.7 Verificação de sistemas confiáveis 597
  - 27.8 Resumo 599
- Problemas e pontos a ponderar 599  
Leituras e fontes de informação complementares 600

**CAPÍTULO 28 Modelagem formal e verificação 601**

- 28.1 Estratégia sala limpa 602
- 28.2 Especificação funcional 604
  - 28.2.1 Especificação caixa-preta 605
  - 28.2.2 Especificação caixa de estado 606
  - 28.2.3 Especificação caixa-clara 607
- 28.3 Projeto sala limpa 607
  - 28.3.1 Refinamento do projeto 608
  - 28.3.2 Verificação de projeto 608

28.4	Teste sala limpa	610
28.4.1	Teste de uso estatístico	610
28.4.2	Certificação	612
28.5	Reconsideração dos métodos formais	612
28.6	Conceitos de métodos formais	615
28.7	Argumentos alternativos	618
28.8	Resumo	619
	Problemas e pontos a ponderar	620
	Leituras e fontes de informação complementares	621

## **CAPÍTULO 29 Gestão de configuração de software 623**

29.1	Gestão de configuração de software	624
29.1.1	Um cenário SCM	625
29.1.2	Elementos de um sistema de gestão de configuração	626
29.1.3	Referenciais	626
29.1.4	Itens de configuração de software	628
29.1.5	Gestão de dependências e alterações	629
29.2	O repositório de SCM	630
29.2.1	Características gerais e conteúdo	630
29.2.2	Características da SCM	631
29.3	O processo SCM	632
29.3.1	Identificação de objetos na configuração de software	633
29.3.2	Controle de versão	634
29.3.3	Controle de alterações	635
29.3.4	Gestão de impacto	638
29.3.5	Auditoria de configuração	639
29.3.6	Relatório de status	639
29.4	Gestão de configuração para WebApps e aplicativos móveis	640
29.4.1	Problemas predominantes	641
29.4.2	Objetos de configuração	642
29.4.3	Gestão de conteúdo	643
29.4.4	Gestão de alterações	646
29.4.5	Controle de versão	648
29.4.6	Auditoria e relatório	649
29.5	Resumo	650
	Problemas e pontos a ponderar	651
	Leituras e fontes de informação complementares	651

## **CAPÍTULO 30 Métricas de produto 653**

30.1	Framework para métricas de produto	654
30.1.1	Medidas, métricas e indicadores	654
30.1.2	O desafio das métricas de produto	655
30.1.3	Princípios da medição	656
30.1.4	Medição de software orientada a metas	656
30.1.5	Atributos de métricas de software eficazes	657
30.2	Métricas para o modelo de requisitos	659
30.2.1	Métricas baseadas em função	659
30.2.2	Métricas para qualidade de especificação	662
30.3	Métricas para o modelo de projeto	663
30.3.1	Métricas de projeto de arquitetura	663
30.3.2	Métricas para projeto orientado a objetos	666
30.3.3	Métricas orientadas a classes – o conjunto de métricas CK	667

30.3.4	Métricas orientadas a classes – o conjunto de métricas MOOD	670
30.3.5	Métricas orientadas a objeto propostas por Lorenz e Kidd	671
30.3.6	Métricas de projeto em nível de componente	671
30.3.7	Métricas orientadas a operação	671
30.3.8	Métricas de projeto de interface de usuário	672
30.4	Métricas de projeto para WebApps e aplicativos móveis	672
30.5	Métricas para código-fonte	675
30.6	Métricas para teste	676
30.6.1	Métricas de Halstead aplicadas ao teste	676
30.6.2	Métricas para teste orientado a objetos	677
30.7	Métricas para manutenção	678
30.8	Resumo	679
	Problemas e pontos a ponderar	680
	Leituras e fontes de informação complementares	680

## **PARTE IV Gerenciamento de projetos de software 683**

---

### **CAPÍTULO 31 Conceitos de gerenciamento de projeto 684**

31.1	O espectro de gerenciamento	685
31.1.1	As pessoas	685
31.1.2	O produto	686
31.1.3	O processo	686
31.1.4	O projeto	686
31.2	As pessoas	687
31.2.1	Os envolvidos	687
31.2.2	Líderes de equipe	688
31.2.3	A equipe de software	689
31.2.4	Equipes ágeis	691
31.2.5	Questões de comunicação e coordenação	693
31.3	O produto	693
31.3.1	Escopo do software	693
31.3.2	Decomposição do problema	694
31.4	O processo	694
31.4.1	Combinando o produto e o processo	695
31.4.2	Decomposição do processo	695
31.5	O projeto	697
31.6	O princípio W <sup>5</sup> HH	698
31.7	Práticas vitais	699
31.8	Resumo	699
	Problemas e pontos a ponderar	700
	Leituras e fontes de informação complementares	701

### **CAPÍTULO 32 Métricas de processo e de projeto 703**

32.1	Métricas no domínio do processo e do projeto	704
32.1.1	Métricas de processo e aperfeiçoamento do processo de software	704
32.1.2	Métricas de projeto	707
32.2	Medição de software	708
32.2.1	Métricas orientadas a tamanho	709
32.2.2	Métricas orientadas a função	710
32.2.3	Harmonizando métricas LOC e FP	711

32.2.4	Métricas orientadas a objetos	713
32.2.5	Métricas orientadas a casos de uso	714
32.2.6	Métricas de projeto de WebApp	714
32.3	Métricas para qualidade de software	716
32.3.1	Medição da qualidade	717
32.3.2	Eficiência na remoção de defeitos	718
32.4	Integração de métricas dentro do processo de software	719
32.4.1	Argumentos favoráveis às métricas de software	720
32.4.2	Estabelecimento de um referencial	720
32.4.3	Coleta, cálculo e avaliação de métricas	721
32.5	Métricas para empresas pequenas	721
32.6	Estabelecimento de um programa de métricas de software	722
32.7	Resumo	724
Problemas e pontos a ponderar 724		
Leituras e fontes de informação complementares 725		

## CAPÍTULO 33 Estimativas de projeto de software 727

33.1	Observações sobre as estimativas	728
33.2	O processo de planejamento do projeto	729
33.3	Escopo e viabilidade do software	730
33.4	Recursos	731
33.4.1	Recursos humanos	731
33.4.2	Recursos de software reutilizáveis	732
33.4.3	Recursos ambientais	732
33.5	Estimativa do projeto de software	733
33.6	Técnicas de decomposição	734
33.6.1	Dimensionamento do software	734
33.6.2	Estimativa baseada em problema	735
33.6.3	Um exemplo de estimativa baseada em LOC	736
33.6.4	Um exemplo de estimativa baseada em FP	738
33.6.5	Estimativa baseada em processo	739
33.6.6	Um exemplo de estimativa baseada em processo	740
33.6.7	Estimativa com casos de uso	740
33.6.8	Um exemplo de estimativa usando pontos de caso de uso	742
33.6.9	Harmonizando estimativas	742
33.7	Modelos empíricos de estimativa	743
33.7.1	A estrutura dos modelos de estimativa	744
33.7.2	O modelo COCOMO II	744
33.7.3	A equação do software	744
33.8	Estimativa para projetos orientados a objetos	746
33.9	Técnicas de estimativa especializadas	746
33.9.1	Estimativa para desenvolvimento ágil	746
33.9.2	Estimativa para projetos de WebApps	747
33.10	A decisão fazer/comprar	748
33.10.1	Criação de uma árvore de decisões	749
33.10.2	Terceirização	750
33.11	Resumo	752
Problemas e pontos a ponderar 752		
Leituras e fontes de informação complementares 753		

**CAPÍTULO 34 Cronograma de projeto 754**

- 34.1 Conceitos básicos 755
- 34.2 Cronograma de projeto 757
  - 34.2.1 Princípios básicos 758
  - 34.2.2 Relação entre pessoas e esforço 759
  - 34.2.3 Distribuição de esforço 760
- 34.3 Definição de um conjunto de tarefas para o projeto de software 761
  - 34.3.1 Um exemplo de conjunto de tarefas 762
  - 34.3.2 Refinamento das tarefas principais 763
- 34.4 Definição de uma rede de tarefas 764
- 34.5 Cronograma 765
  - 34.5.1 Gráfico de Gantt 766
  - 34.5.2 Acompanhamento do cronograma 767
  - 34.5.3 Acompanhamento do progresso de um projeto orientado a objetos 768
  - 34.5.4 Cronograma para projetos de WebApps e aplicativos móveis 769
- 34.6 Análise de valor agregado 772
- 34.7 Resumo 774

Problemas e pontos a ponderar 774

Leituras e fontes de informação complementares 776

**CAPÍTULO 35 Gestão de riscos 777**

- 35.1 Estratégias de risco reativas *versus* proativas 778
- 35.2 Riscos de software 778
- 35.3 Identificação do risco 780
  - 35.3.1 Avaliação do risco geral do projeto 781
  - 35.3.2 Componentes e fatores de risco 782
- 35.4 Previsão de risco 782
  - 35.4.1 Desenvolvimento de uma tabela de riscos 783
  - 35.4.2 Avaliação do impacto do risco 785
- 35.5 Refinamento do risco 787
- 35.6 Mitigação, monitoramento e gestão de riscos (RMMM) 788
- 35.7 O plano RMMM 790
- 35.8 Resumo 792

Problemas e pontos a ponderar 792

Leituras e fontes de informação complementares 793

**CAPÍTULO 36 Manutenção e reengenharia 795**

- 36.1 Manutenção de software 796
- 36.2 Suportabilidade do software 798
- 36.3 Reengenharia 798
- 36.4 Reengenharia de processo de negócio 799
  - 36.4.1 Processos de negócio 799
  - 36.4.2 Um modelo de BPR 800
- 36.5 Reengenharia de software 802
  - 36.5.1 Um modelo de processo de reengenharia de software 802
  - 36.5.2 Atividades de reengenharia de software 803
- 36.6 Engenharia reversa 805
  - 36.6.1 Engenharia reversa para entender os dados 807
  - 36.6.2 Engenharia reversa para entender o processamento 807
  - 36.6.3 Engenharia reversa das interfaces de usuário 808

---

36.7	Reestruturação	809
36.7.1	Reestruturação de código	809
36.7.2	Reestruturação de dados	810
36.8	Engenharia direta	811
36.8.1	Engenharia direta para arquiteturas cliente-servidor	812
36.8.2	Engenharia direta para arquiteturas orientadas a objetos	813
36.9	Aspectos econômicos da reengenharia	813
36.10	Resumo	814
	Problemas e pontos a ponderar	815
	Leituras e fontes de informação complementares	816

## **PARTE V Tópicos avançados 817**

---

### **CAPÍTULO 37 Melhoria do processo de software 818**

37.1	O que é SPI?	819
37.1.1	Abordagens para SPI	819
37.1.2	Modelos de maturidade	821
37.1.3	A SPI é para todos?	822
37.2	O processo de SPI	823
37.2.1	Avaliação e análise de lacunas	823
37.2.2	Educação e treinamento	825
37.2.3	Seleção e justificação	825
37.2.4	Instalação/migração	826
37.2.5	Mensuração	827
37.2.6	Gestão de riscos para SPI	827
37.3	O CMMI	828
37.4	People-CMM	832
37.5	Outros frameworks SPI	832
37.6	Retorno sobre investimento em SPI	834
37.7	Tendências da SPI	835
37.8	Resumo	836
	Problemas e pontos a ponderar	837
	Leituras e fontes de informação complementares	837

### **CAPÍTULO 38 Tendências emergentes na engenharia de software 839**

38.1	Evolução da tecnologia	840
38.2	Perspectivas para uma verdadeira disciplina de engenharia	841
38.3	Observação de tendências na engenharia de software	842
38.4	Identificação das “tendências leves”	843
38.4.1	Gestão da complexidade	845
38.4.2	Software aberto	846
38.4.3	Requisitos emergentes	846
38.4.4	O mix de talentos	847
38.4.5	Blocos básicos de software	847
38.4.6	Mudança na percepção de “valor”	848
38.4.7	Código aberto	848
38.5	Rumos da tecnologia	849
38.5.1	Tendências de processo	849
38.5.2	O grande desafio	851
38.5.3	Desenvolvimento colaborativo	852
38.5.4	Engenharia de requisitos	852

38.5.5	Desenvolvimento de software controlado por modelo	853
38.5.6	Projeto pós-moderno	854
38.5.7	Desenvolvimento guiado por teste	854
38.6	Tendências relacionadas a ferramentas	855
38.7	Resumo	857
	Problemas e pontos a ponderar	857
	Leituras e fontes de informação complementares	858

## **CAPÍTULO 39 Comentários finais 860**

39.1	A importância do software – revisitada	861
39.2	Pessoas e a maneira como constroem sistemas	861
39.3	Novos modos de representar a informação	863
39.4	A visão no longo prazo	864
39.5	A responsabilidade do engenheiro de software	865
39.6	Comentário final de RSP	867

## **APÊNDICE 1 Introdução à UML 869**

## **APÊNDICE 2 Conceitos de orientação a objetos 891**

## **APÊNDICE 3 Métodos formais 899**

## **REFERÊNCIAS 909**

## **ÍNDICE 933**