Sumário

Introdução **18**

Capítulo 1 ▪ Importar-se com o código **24**

Parte I ▪ você.escreve(código); **27**

Capítulo 2 ▪ Mantendo as aparências **28**

A apresentação é eficaz 29
É uma questão de comunicação 30
Layout 32
Crie uma boa estrutura 32
Consistência 33
Nomes 35
Evite redundância 35
Seja claro 36
Seja idiomático 36
Seja preciso 36
Torne-se apresentável 36
Conclusão 37

Capítulo 3 ▪ Escreva menos código! **40**

Por que devemos nos importar? 41
Lógica oscilante 42
Duplicação 44
Código morto 47
Comentários 48
Verbosidade 49
Design ruim 51
Espaços em branco 51
Então o que devemos fazer? 52
Conclusão 52

Capítulo 4 ■ Melhore o código removendo-o 54
Indulgência no código 55
Não é ruim, é inevitável 56
E daí? 57
Despertando os mortos 58
Extração cirúrgica 58
Conclusão 59

Capítulo 5 ■ O fantasma de um código do passado 62
Apresentação 63
O estado da arte 65
Idioms 66
Decisões de design 67
Bugs 68
Conclusão 68

Capítulo 6 ■ Percorrendo um caminho 71
Uma pequena ajuda de meus amigos 72
Procure pistas 73
Aprenda fazendo 76
Frutos ao alcance das mãos 77
Inspecione o código 77
Estude e depois aja 78
Teste antes 78
Organize a casa 78
Documente o que você encontrar 78
Conclusão 79

Capítulo 7 ■ Chafurdando na lama 82
Fareje os sinais 82
Entrando na fossa 83
A avaliação diz que... 84
Trabalhando em terreno arenoso 85
Limpando a sujeira 86
Fazendo ajustes 86
Código ruim? Programadores ruins? 88

Capítulo 8 ■ Não ignore esse erro! 91
Maneiras de informar erros 92
A loucura 93
As desculpas 94
Conclusão 95

Capítulo 9 ▪ Espere pelo inesperado 97
Erros 97
Threading 98
Encerramento 98
A moral da história 99

Capítulo 10 ▪ Caça aos bugs 101
Uma preocupação econômica 102
Um grama de prevenção 102
Caça aos bugs 103
Coloque armadilhas 105
Aprenda a fazer divisões binárias 105
Empregue a arqueologia de software 106
Teste, teste, teste 106
Invista em ferramentas afiadas 107
Remova códigos para excluí-lo da análise de causa 108
A limpeza evita uma infecção 109
Estratégias oblíquas 109
Não se apresse 110
Bugs que não podem ser reproduzidos 110
Conclusão 112

Capítulo 11 ▪ É hora de testar 115
Por que testar? 115
Reduzindo o ciclo de feedback 116
Código para testar código 117
Quem escreve os testes? 118
Tipos de teste 118
Quando os testes devem ser escritos 120
Quando executar os testes 122
O que deve ser testado 123
Bons testes 123
Como é a aparência de um teste? 125
Nomes de testes 126
A estrutura dos testes 127
Faça a manutenção dos testes 128
Selecionando um framework de teste 128
Nenhum código é uma ilha 129
Conclusão 131

Capítulo 12 ▪ Lidando com a complexidade 134
Círculos 135
Caso de estudo: reduzindo a complexidade dos círculos 137
Linhas 138
E por fim: as pessoas 141
Conclusão 142

Capítulo 13 ▪ Um conto de dois sistemas 145
A Metrópole Confusa 146
Incompreensibilidade 147
Falta de coesão 147
Acoplamento desnecessário 149
Problemas com o código 150
Problemas além do código 150
Um cartão-postal de Metrópole 152
Cidade do Design 153
Localizando as funcionalidades 153
Consistência 154
Expandindo a arquitetura 155
Adiando as decisões de design 155
Mantendo a qualidade 156
Administrando a dívida técnica 156
Os testes moldam o design 158
Prazo para o design 159
Trabalhando com o design 159
E daí? 160

Parte II ▪ A prática leva à perfeição 164

Capítulo 14 ▪ Desenvolvimento de software é... 165
Esse negócio de software (comida) 165
Desenvolvimento de software é... uma arte 166
Desenvolvimento de software é... uma ciência 168
Desenvolvimento de software é... um esporte 170
Desenvolvimento de software é... uma brincadeira de criança 171
Desenvolvimento de software é... uma obrigação 173
Excesso de metáforas 174

Capítulo 15 ▪ Jogando segundo as regras 176
Precisamos de mais regras! 177
Defina as regras 178

Capítulo 16 ▪ Mantenha a simplicidade 181
Designs simples 182
Simples de usar 182
Evita usos indevidos 182
O tamanho importa 183
Caminhos mais curtos no código 183
Estabilidade 184
Linhas de código simples 184
Mantenha a simplicidade e não a estupidez 185
As suposições podem reduzir a simplicidade 185
Evite uma otimização prematura 186
Suficientemente simples 186
Uma conclusão simples 187

Capítulo 17 ▪ Use o seu cérebro 189
Não seja estúpido 189
Evite o descuido 190
Você tem permissão para pensar! 191

Capítulo 18 ▪ Nada está gravado a ferro e fogo 194
Mudanças sem medo 195
Mude a sua atitude 196
Faça a mudança 198
Crie o design visando a mudanças 198
Ferramentas para fazer alterações 199
Escolha suas batalhas 199
Mais alterações 200

Capítulo 19 ▪ Um estudo sobre reutilização de código 202
Caso de reutilização 1: copiar e colar 202
Caso de reutilização 2: faça o design visando à reutilização 204
Caso de reutilização 3: promover e refatorar 204
Caso de reutilização 4: compre ou reinvente a roda 205

Capítulo 20 ▪ Controle eficiente de versões 208
Use-o ou você se arrependerá 209
Escolha um, qualquer um 210
Armazenando os itens corretos 211
Resposta um: armazene tudo 212
Resposta dois: armazene o mínimo possível 212
Armazenando versões de software 213
Layout do repositório 214

Utilize bem o sistema de controle de versões 214
Faça commits atômicos 215
Enviando as mensagens corretas 215
Componha bons commits 216
Branches: vendo o bosque no lugar das árvores 217
Um lar para o seu código 218
Conclusão 219

Capítulo 21 ■ Marcando um gol 222

Desenvolvimento de software: jogando adubo 224
Uma falsa dicotomia 225
Corrija a equipe para corrigir o código 227
Disponibilizando um build para a equipe de QA 228
Teste o seu trabalho antes 229
Tenha um propósito ao disponibilizar uma versão 230
Mais pressa, menos velocidade 230
Automatize 231
Respeite 231
Ao receber um relatório de falha 231
Nossas diferenças nos tornam mais fortes 233
Peças do quebra-cabeça 234

Capítulo 22 ■ O curioso caso do código congelado 237

Atrás do congelamento de código 237
Uma nova ordem mundial 239
Formas de congelamento 240
Branches são apropriados 241
Mas o código não está realmente congelado! 242
Duração do congelamento 242
Sinta o congelamento 243
O fim se aproxima 244
Anticongelamento 244
Conclusão 245

Capítulo 23 ■ Por favor, libere a versão 247

Parte do processo 249
Uma engrenagem na máquina 249
Passo 1: inicie a disponibilização de versão 250
Passo 2: prepare a versão 250
Passo 3: gere a versão 251
Passo 4: empacote a versão 252
Passo 5: implante a versão 253

Disponibilize versões cedo e com frequência 254
E tem mais... 255

Parte III ▪ Envolvendo-se pessoalmente 257

Capítulo 24 ▪ Viva para amar o aprendizado 258
O que você deve aprender? 259
Aprendendo a aprender 260
Modelos de aprendizado 264
O portfólio de conhecimento 266
Ensine para aprender 267
Faça para aprender 267
O que aprendemos? 268

Capítulo 25 ▪ Desenvolvedores orientados a testes 270
Esclarecendo a questão 271
O sucesso gera complacência 272
É hora do exame 272
Desenvolvedores orientados a testes 273
Conclusão 274

Capítulo 26 ▪ Aprecie o desafio 277
É a motivação 277
Qual é o desafio? 278
Não faça isso! 279
Sinta-se desafiado 280
Conclusão 281

Capítulo 27 ▪ Evite a estagnação 283
Suas habilidades são o seu investimento 284
Um exercício para o leitor 284
Segurança no emprego 285

Capítulo 28 ▪ O programador ético 288
Atitude em relação ao código 289
Questões legais 290
Atitude em relação às pessoas 291
Colegas de equipe 292
Gerente 293
Empregador 294
Você 294

O juramento de Hipocódigo 295
Conclusão 295

Capítulo 29 ▪ Amor pelas linguagens 298

Ame todas as linguagens 298
Ame sua linguagem 300
Cultivando o relacionamento com a sua linguagem 302
Amor e respeito 302
Compromisso 303
Comunicação 303
Paciência 304
Valores compartilhados 305
Uma metáfora perfeita? 305
Conclusão 305

Capítulo 30 ▪ Postura dos programadores 308

Postura básica diante do computador 308
A postura de debugging 310
Quando a situação estiver realmente ruim 312
Para aqueles que trabalham a noite toda 312
Intervenção do alto 313
Voltando ao normal 314
É hora do design 315
Vista cansada 315
Conclusão 316

Parte IV ▪ Conseguir que tudo seja feito 318

Capítulo 31 ▪ Mais inteligente, e não mais árduo 319

Escolha suas batalhas 320
Táticas de batalha 320
Reutilize sabiamente 320
Faça com que o problema seja de outra pessoa 321
Faça somente o que for necessário 321
Use uma solução rápida 322
Priorize 322
O que é realmente necessário? 323
Uma tarefa de cada vez 323
Mantenha o seu código pequeno (e simples) 323
Não adie os problemas deixando que eles se acumulem 324
Automatize 324
Evitando erros 325

Comunique-se 325
Evite um esgotamento 326
Ferramentas poderosas 326
Conclusão 326

Capítulo 32 ▪ Estará pronto quando estiver pronto 329
Já terminamos? 329
Desenvolvendo para trás: decomposição 330
Defina "pronto" 331
Simplesmente faça 334

Capítulo 33 ▪ Dessa vez, eu consigo 337
Desenvolvimento em uma ilha deserta 340
Fique na base da montanha 340

Parte V ▪ Uma meta de pessoas 343

Capítulo 34 ▪ O poder das pessoas 344
O que você deve fazer 345
Conheça seus experts 346
Campo de visão 20/20 346

Capítulo 35 ▪ Ser responsável 349
Alongando a metáfora 350
A responsabilidade ajuda 351
Código++ 351
Fazendo com que dê certo 352
Definindo o padrão 353
Os próximos passos 354
Conclusão 354

Capítulo 36 ▪ Fale! 357
Código é comunicação 357
Falando com as máquinas 357
Falando com os animais 358
Falando com ferramentas 360
Comunicação interpessoal 361
Maneiras de conversar 361
Observe o seu linguajar 362
Linguagem corporal 363
Comunicação paralela 364

Conversas nas equipes 364
Falando com o cliente 365
Outras formas de comunicação 366
Conclusão 366

Capítulo 37 ■ Manifestos 369
Um manifesto genérico para o desenvolvimento de software 370
Tudo bem, tudo bem 370
Os manifestos 371
É sério? 371
A conclusão 372

Capítulo 38 ■ Ode ao código 375
Codificação é um problema de pessoas 376

Epílogo 379
Atitude 380
Vá em frente e codifique 381

Sobre o autor 382

Colofão 383