

# Obsah

<b>Úvodní slovo</b>	<b>11</b>
<b>Předmluva</b>	<b>13</b>
Kdo by měl tuto knihu číst?	13
Co dělá programátora pragmatikem?	14
Jednotliví pragmatici a velké týmy	15
Nikdy nekončící proces	16
Jak je kniha členěna	16
Jak jsme přišli pojmem na jméno?	16
Zdrojový kód a jiné zdroje	17
Stojíme o zpětnou vazbu	17
Poznámka českého vydavatele	17
Poděkování	18
<b>KAPITOLA 1</b>	
<b>Pragmatická filozofie</b>	<b>19</b>
<b>1 Můj zdrojový kód snědla kočka</b>	<b>20</b>
Přijímání odpovědnosti	20
<b>2 Entropie softwaru</b>	<b>21</b>
Být schopen uhasit požár	22
<b>3 Polévka ze sekyrky a vařené žáby</b>	<b>23</b>
A co na to vesničané	24
<b>4 Dostatečně kvalitní software</b>	<b>25</b>
Zapojte do kompromisů uživatele	26
Vědět, kdy přestat	26
<b>5 Vaše portfolio znalostí</b>	<b>27</b>
Vaše portfolio znalostí	28
Budujte své portfolio	28
Cíle	29
Příležitosti k učení	30
Kritické myšlení	30
<b>6 Komunikujte!</b>	<b>31</b>
Ujasněte si, co chcete říci	32
Poznejte své publikum	32

Zvolte správný okamžik	33
Zvolte vhodný styl	33
Dbejte na formu	33
Zapojte publikum	34
Naslouchejte	34
Odpovídejte	34

**KAPITOLA 2****Pragmatický přístup** **37**

<b>7 Hříchy opakování</b>	<b>38</b>
Z čeho opakování pramení?	39
Vynucené opakování	39
Nechtěné opakování	40
Opakování z netrpělivosti	42
Opakování v týmu	42
<b>8 Ortogonalita</b>	<b>43</b>
Co je ortogonalita?	43
Neortogonální systém	44
Výhody ortogonality	44
Zvýšení produktivity	45
Snížení rizika	45
Projektové týmy	45
Návrh	46
Nástroje a knihovny	47
Tvorba kódu	48
Testování	49
Dokumentace	49
Život s ortogonalitou	49
<b>9 Reverzibilita</b>	<b>51</b>
Reverzibilita	52
Flexibilní architektura	52
<b>10 Trasovací střely</b>	<b>54</b>
Kód zářící ve tmě	55
Trasovací střely nezasáhnou vždy cíl	56
Trasovací kód versus prototypy	57
<b>11 Prototypy a poznámky</b>	<b>58</b>
Co se hodí testovat na prototypch	58
Jak vytvářet prototypy	59
Prototypy architektury	59

Jak prototypy nevytvářet	60
<b>12 Oborové jazyky</b>	<b>61</b>
Implementace minijazyka	63
Datové jazyky a imperativní jazyky	63
Samostatné a integrované jazyky	65
Snadný vývoj, nebo snadná údržba?	65
<b>13 Odhady</b>	<b>66</b>
Jak přesný odhad je dostatečně přesný?	67
Odkud se berou odhady?	68
Pochopte, na co se vás ptají	68
Vytvořte si model systému	68
Rozložte model na komponenty	68
Každému parametru přiřadíte hodnotu	69
Vypočtete odpovědi	69
Sledujte svou odhadovací schopnost	69
Odhady časových plánů projektů	70
Co říci, když jste tázáni na odhad	70

## KAPITOLA 3

**Základní nástroje** **73**

<b>14 Síla prostého textu</b>	<b>74</b>
Co je prostý text?	74
Nevýhody	75
Síla textu	75
Univerzálnost	76
Snadnější testování	77
Nejmenší společný jmenovatel	77
<b>15 Hrátky s příkazovou řádkou</b>	<b>78</b>
Utility pro příkazovou řádku a systémy Windows	80
<b>16 Vyšší škola editování</b>	<b>81</b>
Jeden editor	82
Funkce editoru	82
Produktivita	83
Jakým směrem se dále vydat	84
Jaké editory jsou k dispozici?	84
<b>17 Správa zdrojového kódu</b>	<b>85</b>
Správa zdrojového kódu a sestavení	86
Jenže náš tým správu zdrojového kódu nepoužívá	86
Produkty pro správu zdrojového kódu	87

<b>18 Ladění</b>	<b>87</b>
Psychologie ladění	88
Postoj k ladění	88
Kde začít	89
Strategie ladění	89
Vizualizujte data	90
Trasování	91
Gumová kachnička	91
Proces eliminace	92
Moment překvapení	93
Co zkontrolovat při ladění	93
<b>19 Práce s textem</b>	<b>94</b>
<b>20 Generátory kódu</b>	<b>97</b>
Pasivní generátory kódu	97
Aktivní generátory kódu	98
Generátory kódu nemusí být složité	100
Generátory kódu nemusí generovat kód	100

#### KAPITOLA 4

<b>Pragmatická paranoia</b>	<b>101</b>
<b>21 Programování na základě smlouvy</b>	<b>102</b>
Návrh na základě smlouvy	102
Implementace návrhu na základě smlouvy	105
DBC a havárie v pravý čas	106
Jiná využití invariantů	107
Dynamické smlouvy a agenti	109
<b>22 Mrtvé programy nelžou</b>	<b>111</b>
Havárie je lepší než slon v porcelánu	111
<b>23 Aserktivní programování</b>	<b>112</b>
Aserce nechte zapnuté	113
<b>24 Kdy používat výjimky</b>	<b>115</b>
Co je výjimečné?	116
Jednou z možností jsou rutiny pro obsluhu chyb	117
<b>25 Jak vyvážit prostředky</b>	<b>118</b>
Objekty a výjimky	120
Vyvažování a výjimky	120
Vyvažování prostředků v Javě	122

## KAPITOLA 5

**Ohnout, nebo zlomit 125**

<b>26 Odstraňování závislostí a Demeterův zákon</b>	<b>126</b>
Minimalizace vazeb	126
Demeterův zákon pro funkce	127
Je mezi programy rozdíl?	127
<b>27 Metaprogramování</b>	<b>130</b>
Dynamická konfigurace	130
Aplikace řízené metadaty	131
Program jako dodo mauricijský	133
<b>28 Vazby v čase</b>	<b>134</b>
Pracovní postupy	135
Architektura	137
Návrh pro paralelní programy	138
Čistší rozhraní	139
Nasazení	140
<b>29 Jde jen o pohled</b>	<b>140</b>
Vydavatel/předplatitel	141
Model-pohled-ovladač	142
Co se skrývá za GUI	143
Někde vazby zůstaly (i po těch letech)	145
<b>30 Tabule</b>	<b>146</b>
Některé implementace tabule	147
Příklad aplikace	148

## KAPITOLA 6

**Když píšete program 151**

<b>31 Spoléhání na náhodu</b>	<b>152</b>
Jak spoléháme na náhodu při programování	152
Programovat musíme s rozmyslem	154
<b>32 Rychlost algoritmů</b>	<b>155</b>
Odhady algoritmů	156
Symbol O()	156
Odhad pomocí zdravého rozumu	157
Rychlost algoritmů v praxi	158
<b>33 Refaktorizace</b>	<b>160</b>
Kdy má smysl refaktorizovat	161
Jak refaktorizujeme programy	162

<b>34 Kód, který se snadno testuje</b>	<b>165</b>
Jednotkové testování	165
Testování na základě smlouvy	165
Psaní testů jednotek	167
Standardní testovací postupy	168
Jak vytvoříme testovací okno	170
Kultura testování	170
<b>Zlobiví průvodci</b>	<b>171</b>

**KAPITOLA 7****Předprojektová příprava** **175**

<b>36 Jak vydolovat zadání</b>	<b>176</b>
Zadání musíme vydolovat	176
Dokumentace zadání	177
Diagramy případů užití	179
Nadbytečné specifikace	181
Myslete na budoucnost	181
Ještě jeden malinký penízek ...	182
Veďte si významový slovník	182
Text musí být dostupný	183
<b>37 Jak řešit hlavolamy</b>	<b>184</b>
Stupně volnosti	184
Musí přece existovat snadnější způsob!	185
<b>38 Až budete připraveni, dřív ne</b>	<b>186</b>
Správný úsudek, nebo zbytečné odklady?	186
Náměty k zamyšlení	187
<b>39 Specifikační past</b>	<b>187</b>
<b>40 Kroužky a šipky</b>	<b>189</b>
Vyplatí se formální metody?	190
Měli bychom používat formální metody?	190

**KAPITOLA 8****Pragmatické projekty** **193**

<b>41 Pragmatické týmy</b>	<b>194</b>
Žádná rozbitá okna	194
Vařené žáby	194
Komunikujte	195
Neopakujte se	195

Ortogonalita	195
Automatizace	197
Vědět, kdy přestat	197
<b>42 Všudypřítomná automatizace</b>	<b>198</b>
Vše automaticky	198
Překlad projektu	199
Automatizace sestavování	200
Základy automatické administrativy	201
Ševcova žena	202
<b>43 Nemilosrdné testování</b>	<b>203</b>
Co testovat	204
Jak testovat	206
Ověřování systémů s grafickým rozhraním (GUI)	208
Kdy testovat	209
Utahování sítě	210
<b>44 Vše je o psaní</b>	<b>211</b>
Komentáře v programu	211
Spustitelné dokumenty	213
Technická síla	214
Výstupy můžete tisknout i tkát ...	214
Značkovací jazyky	215
<b>45 Velká očekávání</b>	<b>216</b>
Komunikační předpoklady	217
Čestné kolečko	217
<b>46 Pýcha a předsudek</b>	<b>218</b>

## PŘÍLOHA A

<b>Zdroje</b>	<b>221</b>
Profesní společenstva	221
Budování knihovny	222
Knihy	223
Internet	224
Zdroje na Internetu	224
Bibliografie	232

## PŘÍLOHA B

<b>Odpovědi ke cvičením</b>	<b>237</b>
<b>Rejstřík</b>	<b>263</b>