China-pub.com
下载

后　记

如果人能从历史中学习，我们将能学到多少东西啊！但是激忿和党派蒙住了我们的双眼，而经验的光亮就像船的艉灯，只能在我们背后的波涛上留下一点余辉。

Samuel Taylor Coleridge，《回忆》

计算的世界每时每刻都在变化，步伐看起来是越来越快。程序员必须不断应付新的语言、新的工具和新的系统，它们总有一些与老东西不兼容的新特性。程序越来越大，界面越来越复杂，而任务的时限也越来越短。

但是，总有某些东西是不变的，总有一些稳定点，在这种地方从过去中学到的东西和洞察力，对于未来必定能有所帮助。本书的背景就是基于这些具有持久性的概念。

简单性和清晰性是第一位的、最重要的，因为几乎所有其他东西都只能跟着它们而来。做那种最简单的能解决问题的东西，选择那些应该是足够快的最简单的算法、能够满足需要的最简单的数据结构；用整齐清楚的代码把它们组合起来。除非性能测试的结果说明需要做更多的事，我们绝不要把事情复杂化。界面应该是整齐和简单的，至少是在发现无可辩驳的证据、说明把它弄得复杂一些有极大优越性之前。

普遍性常与简单性同在，它使我们可能一次就完全解决了问题，而不是对各种情况一个个地重复去做。普遍性常常也是达到可移植的正确途径：找一个一般性的，能够在所有系统上工作的解，而不是去扩大不同系统之间的差异。

进化接踵而来。想第一次就构造出一个绝好的程序通常是不可能的。发现正确解决方法的必要洞察力只能来自思考和经验的结合；纯粹的内省不可能造就出好系统，纯粹靠玩命干也不行。由用户得来的反馈在这个地方非常重要。通过循环的方式：原型、试验、用户反馈和进一步精化，常常是最有效的。我们自己构建的程序常常进化得不够；从别人那里买来的大程序变得太快，根本没经过必要的改进。

界面是程序设计战斗中的一个大战场，界面问题出现在许多不同的地方。程序库是最明显的例子，还有不同程序之间的、程序与用户之间的界面问题。对简单性和普遍性的需求在界面设计方面表现得特别强烈。我们应该使界面具有一致性，容易学习和使用，应该一丝不苟地追求这些东西。抽象是一种有效技术：首先设想一种完美的部件，或者库，或者程序，让界面尽可能地符合这个理想。应该把实现细节都隐蔽在边界的后面，这永远是最安全的方法。

自动化一直没有得到足够重视。让计算机做你需要的事，常常比自己直接用手做的效率高得多。我们已经看到了许多这方面的例子，在程序测试、排错和性能分析方面，特别是写代码方面。在这些地方，对于合适的问题，程序可以写出人很难写出来的程序。

记法也一直被人所忽视，实际上它远不仅仅是程序员告诉计算机去做什么时采用的方式。记法提供了一种组织框架，在实现应用领域非常广泛的各种工具方面，对指导人们去构造那种写程序的程序方面都很有价值。我们对大型的通用程序设计语言都很习惯了，它们为我们

China-pub.com 下载

的大量程序设计工作服务。但有时也会发现面前的工作变得非常集中，已经理解得很清楚，写这种程序几乎完全是机械性的。这可能就预示着一个时机，说明应该建立一种能自然地表达有关工作的记法和一个实现它的语言。正则表达式是我们最喜爱的例子，实际中存在着无数的这种机会，在那里我们可以为对付特殊工作建立一个小语言。当然，这种语言绝不应该复杂到抵消掉其收益的程度。

作为程序员个人，我们很容易感到自己像是位于某种大机器上的一个小齿轮，必须使用那些强加给我们的语言、系统和工具，做那些我们必须完成的工作。但是，从长远的观点看，真正起作用的还是看我们在使用现有东西的情况下工作做得怎么样。通过应用本书里的某些思想，你将能够发现你的代码更容易用了，你的排错过程也不再那么痛苦了，你对自己的程序设计更加有自信心了。我们希望这本书能带给你一些东西，使你在自己的计算生涯中获得更多的成果和回报。