

<<大道至易>>

图书基本信息

书名：<<大道至易>>

13位ISBN编号：9787115282170

10位ISBN编号：711528217X

出版时间：2012-6

出版单位：人民邮电出版社

作者：周爱民

页数：386

字数：519000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<大道至易>>

前言

易是变化 台湾的高焕堂先生曾说架构的要旨是“以序容易”，我解释成“用规则来包容变化”，高老师说很合他的本意。

这里的“易”，指的就是变化。

既是变化，那当然是艰难而复杂的了。

然而我们通常说一件事易做或另一件事不易做，这里的“易”却都是指简单的意思。

所以“易”既是无穷的复杂，也是至极的简单，关键在于如何“容”它。

我们得看清什么是变化。

盲人摸象是一个很好的故事，因为每一个盲人的认识都是其固见的、自见的，以及自证的真理。

然而盲人们的观点放在一起的时候，却是一个笑话。

因为看笑话的人看到了“变化”，而这些局外者原本认为那头大象是不变的、确定的、唯一个形象的。

同样，面对任何我们所见的变化，或对变化中的任何一个认识，我们都认为那不过是笑话。

再深刻地透视这一点，其根本在于：我们也有一个对大象的“认识”，只因为这一认识看起来--或我们认为--比那些盲人更为高明、正确，所以我们才别出了盲人，看到了笑话。

正是我们对一个事物固有的认识，制造了盲人与盲人的笑话。

倘若我们的认识是不可易变的，那么我们今天就已经看到了真理，看透了世事万象，我们已成至人，故而不需要存在亦无需进化：一切于我们而言，必须静止；一切于我们的认识而言，不可复加；一切于我们的思维而言，不可偏侧。

而这，看起来不正是我们自盲了双目，自演了笑话吗？

反过来，我们得承认变化的存在。

我们所要做的，只是承认自己是一个盲人，可以从一个方面去触及这一事物，形成一个认识，用一些规则、规格或概念去确指它。

用同样的方法，我们触及这个事物的方方面面，进而得到与这个事物最为接近的一个全像，这就是我们关于这一事物所有的、而又未尽的知识。

知音变而得律，有容则易。

简也不是简单 这本书是《大道至简》（以下称“《简》”）的续作，那我又是怎么认识《简》的呢？

究我们的认识的根底来说，我们向来不是缺乏思考的。

我们有着种种在实践中的选择，这些选择是基于一些认识的，而又产生了一些新的认识。

而我所有的醒觉，并不来源于对后一种认识的反复持续，反而在于对前一种--既有的认识的观察。

《简》一书的源起，正是一次对“我在干什么”的发问。

在这一反思中，我最终找到了我在工程中的种种位置，并将这些位置在我既有的知识中一一标点。

然后，我们就看到了那幅EHM（工程层状模型）图（参见“总论”图0-1）。

追究起来，EHM图以及它所代表的我的认识已经是8年前的事情了；完成《简》一书，也已经是6年前的事情了。

6年前，我将《简》作为电子版发布的时候，也同时得到了一个新的职务：架构师。

然而如果你仔细地观察EHM图，就会发现它根本上是不能容纳“架构”这个角色与职责的。

“我在干什么”，仍然是我这6年来最主要的思考。

事实上，这也促使我抬起头来，观察周围营营碌碌的人们，了解他们在做什么，在围绕什么去做，以及--也是最重要的是：他们得到了什么。

如同《简》中所说的愚公一家，数百年挖山不止，最终却不得不筑关自守，我们是不是也一样背离了初衷，成为一个不知所为的愚者？

《简》事实上是打开了我思维的一扇门，让我看到“我既有的那些东西”是何等粗糙，又曾经是如何粗暴地塞入了我的头脑。

<<大道至易>>

例如我在计算机专业中所学的第一门功课是《数据结构》，这本书的第一页便写着：“用计算机解决具体问题，首先要加以抽象。”但十余年来，我从未思考过“什么是抽象”，又或者我的思维以及由此驱动着的软件开发活动中“何处是抽象”。

抽象被作为我们--程序员的一种事实的本能，或被驯育的技能塞入了我们的举手投足，控制着我们而又让我们无知无觉。

《简》在这“无知无觉”的宁静中轻一叩击，“叮”的一声，我的眼前便亮了。

观之简，便在其原始的粗鄙；思之得，便在行道中的一凛。

没有答案 《简》一书只提出了现象，而无有答案。

2009年在北京的QCon大会中，我跟几位朋友谈到过这个问题。

我坦承这一事实：若《简》尚有一点可读处，便在于它粗暴而又挑衅地把种种问题摆在我们的面前。但对许多人来说，这毫无益处。

因为他们或是已经看了这些，或已经为之麻木，或是在贪求某种法子来逃避这些东西带来的影响。

总而言之，《简》是“没有答案”的。

我彼时也是“没有答案”的，我只是觉醒过来，并看到了一些事实上存在已久的东西而已。

。与此相同，《大道至易》这本书也“没有答案”。

尽管我提出了一些貌似答案的东西，但我确实是在保守地、谨慎地谈论它们，并且时时地（譬如现在）警示读者：这是在某些背景下的一时之选。

“讲述某些我知道或想到的答案”是本书中最没有价值的部分，即使它们存在，即使看起来“非常有用”，也仅是一时的表象而已。

与《简》不同的是，《大道至易》提出了种种问题。

就我的认识来说，现象与问题之间是有着一个巨大的跨越的。

例如《简》中提到的“跑不动了”，它可以成为一种催人奋进的现象，也可以是脱水这样的问题。

作为旁观者，若只见现象，便只会责骂鞭策；若能见问题，则知道“一杯冰水”才是解决之道。

在我看来，正确的问题与“解”之间的距离是可显见的，而一个表面的现象或错误的问题，就离真正的解淼远而无际了。

《大道至易》是一本提出问题的书。

不设节名 正因为问题重要，而答案不重要，所以我认为给每段内容加以一个“有诱惑力的标题”是多余的。

因此本书分篇、章、节三级目录，但节下的小节只有“一、二……九”这样的中文标号，没有标题，是谓“不设节名”。

事实上在我撰写的过程中，这些小节多数是有标题的。

但最终我发现，若是有一个标题，则读者便非常容易陷入这个标题的“含义”之中，以之作为小节之纲要、主旨或主张。

而这又正是我所不愿意的，因为我原本只是拿这些标题来搭个架子而已。

也就是说，本书原本是只有骨架，而没有血肉的。

真正有血肉的，是你面对它的种种思考：那些没有名字的小节，是可以由你去补个标题的；那些没有定论的观见，是可以由你去设定的。

我要做的，只是把问题写出来，把我的想法写出来，把我的观见写出来。

这些，既不是真理，也不是我自作了固见的认识。

我之于盲人，相距并不甚远，只是我惊觉我的盲目，故而对周围多做了些探索；又或者我还能觉出蒙眼的黑布才是真实的问题，揭去之后，便一切历历在目、观之若即了。

只不过一刹那，定是相当刺目的。

<<大道至易>>

不为读者 对于许多读者而言，本书的“总论”是一个过高的阅读起点。

因此本书的编审老师曾建议将“总论”移至最末一篇。

为此我犹豫了很久，并也曾数次动摇。

最终，我仍坚持将它作为本书的开篇。

尽管编审在来信中写道：“根据我们的经验，好的开头对销量的影响可能达到三成。

” 这真的是不小的诱惑。

相比来说，本书的第一篇“似乎”更适合作为“好的开头”--这也确实是编审老师们所推荐的。

然而，事实上这一篇我是最晚成稿，写得最累，以及认为它是最不易读的。

尽管表面来看，读者可以从这里轻松入手读上几页，然后觉得这是本“还有点趣”的书。

然而它到底写的是什么呢？

离开总论的框架，所谓第一篇，不过是可以换得你的同感的抱怨而已。

而真相，绝非如此。

就写书出书而言，我当然可以用读者喜欢的方式去写作，并用读者喜欢的方式去包装那些篇章，然而这样一来，读者便会在“一开始”就失去整体的、系统性的认识，并认为这本书不过是又一本讲述软件工程的《大道至简》。

我可以为了“读者的口味”而改变这本书的著述体系吗？

这是我最终的设问。

而我很快地给出了答案：不！

总论是本书所述思想的核心，而后续几篇则是这一思想下的几个视角。

让读者抛开总论而去窥见某个视角，与盲其目使之摸象何异？

与予其管使之窥豹何异？

我在前言中所反对的思想与方法，何以成了我去争取“读者人数”与迎合“读者口味”的手段？

！

因此写下这段文字，以此自勉：莫因读者“买书如买纸”，便将书作了“字纸”去写，又或将书作了“纸页”去卖。

既如此，便仍让总论做了第一篇罢。

这样一来，即便读者从它处读起，也总还能时时记得：有一篇“总论”在前头，或许蕴含着作者思想的主旨。

致谢 感谢所有的读者、编者以及一直以来支持我的朋友们。

感谢孟岩先生为本书作序。

感谢图灵出版公司。

感谢我的小学数学老师。

感谢Joy。

En, ……我最爱的妻。

<<大道至易>>

内容概要

《大道至易——实践者的思想》可以看成是《大道至简》的姊妹篇，是以软件工程为体系、以组织结构为视角，融合系统架构师、项目管理者 and 软件开发人员三种角色实践的思想总成。本书讨论这些思想，并陈述它们所基于的原则、背景与获得过程。

《大道至易——实践者的思想》适合各类工程管理人员、软件开发人员和架构师阅读。

<<大道至易>>

作者简介

周爱民（Aimingo），国内软件开发界资深软件工程师，架构师。有十余年的软件开发、项目管理、团队建设的经验，曾任盛大网络平台架构师、支付宝业务架构师等职。著有《Delphi源代码分析》、《大道至简》和《JavaScript语言精髓与编程实践》等专著。

<<大道至易>>

书籍目录

目 录

总论	领域角色的关注	1
第一节	什么是领域角色的关注	2
第二节	基于组织视角的观察	11
第三节	工程的本质问题是组织	25
第一篇	具体工程下的组织与行为	
引言	管理中的逻辑	33
第一章	任人治事：组织行为的基本认知	35
第一节	刺秦与灭秦	36
第二节	看到别人能做什么	43
第三节	要做事，不要管理	51
第四节	伯夷与叔齐是怎么死的	59
第二章	谋定后动：项目的存在权	67
第一节	试错通常是无能的托辞	68
第二节	合法的山大王为什么没能成功	78
第三节	自己想办法	86
第三章	具体而微：工程是系统而不是事	93
第一节	做事的选择	94
第二节	你要什么	102
附录一	行在道上，从局部到全局	109
附录二	本来面目——大教堂、集市，与作坊	111
附录三	杀不死的人狼——我读《人月神话》	113
第二篇	程序源流：从计算到系统(上)	
引言	简单的本源	117
第四章	计算系统	119
第一节	数，以及对数据的性质的思考	120
第二节	逻辑	126
第三节	抽象	129
第五章	语言及其面临的系统	133
第一节	语言	134
第二节	从功能到系统	141
第六章	程序设计的核心思想	151
第一节	数据结构：顺序存储	153
第二节	数据结构：散列存储	160
第三节	执行体及其执行过程中的环境	167
第四节	语法树及其执行过程	174
第五节	对象系统：表达、使用与模式	181
第三篇	程序源流：从计算到系统(下)	
第七章	应用开发基础	199
第一节	应用开发的背景与成因	200
第二节	应用开发技术	207
第三节	开发视角下的工程问题	216
第四节	应用程序设计语言的复杂性	223
第八章	系统的基础部件	231
第一节	分布	232

<<大道至易>>

第二节	依赖	241
第三节	消息	245
第四节	系统	250
第九章	系统的基本组织方法与原理	253
第一节	行为的组织及其抽象	254
第二节	领域间的组织	257
附录一	“主要编程范式”及其语言特性关系	273
附录二	继承与混合, 略谈系统的构建方式	275
附录三	像大师们一样思考——从“UML何时死掉”谈起	277
附录四	VCL已死, RAD已死	279
第四篇	架构的思想与指导原则	
引言	架构师的思维	285
第十章	你所关注的系统	287
第一节	了解系统的过程	288
第二节	知识的构建	295
第三节	最初的事实	307
第十一章	架构是过程, 而非结果	315
第一节	架构师的能力结构	316
第二节	系统架构与决策	324
第三节	架构的表达与逻辑	338
第十二章	架构原则, 技艺、艺术与美	355
第一节	架构原则	356
第二节	技艺、艺术与美	368
附录一	做人、做事, 做架构师——架构师能力模型解析	373
附录二	专访: 谈企业软件架构设计(节选)	375
参考资料		377
图表索引		382

章节摘录

版权页：插图：（1）记事本作为操作系统的默认组件，其外观表现和交互特性应当是由系统的全局设置来控制的。

对于前者，例如桌面主题导致的记事本前景与背景变化；对于后者，例如系统默认打印设备的设置，或者输入法设置。

（2）如果操作系统的默认设置不能影响到记事本，则系统的其他默认组件也会存在类似的问题或需求，这意味着整体实施的复杂度会增加。

（3）类似于jEdit、EditPlus等产品的用户只是操作系统用户群体的一个较小子集，其需求不具有代表性。

（4）以记事本通常处理的文本长度来说，是不需要用颜色来区分文本的不同部分的。

对于项目管理角色来说，他否决这项需求的理由会更简单直接：（1）在当前的记事本版本中，未定义该项特性；（2）该项特性与原始的设定“文本处理工具”没有必然关系，可以延后决定；（3）该项特性在来自产品、市场等各方的报告中有明显分歧，存在实施风险；（4）在项目实施阶段，增加该特性对项目过程控制存在不确定的影响。

然而与上述两类思考不同，技术实现角色将更多地关注实现的细节问题。

其中一类问题是与项目经理共同关注的，通常与项目背景以及实施环境有关系。

例如：（1）可能的代码量与要求的代码质量；（2）后续维护人员的水平以及因此所需要的注释详细程度；（3）开发环境与测试环境的部署以及性能。

当然，类似于在何种团队中工作，以及开发部门中是否可以玩桌游等，也是技术实现角色经常关注的问题。

不过这类问题的特点是：与具体项目并没有关系，因此大多数情况下会由公司的技术部门或整体组织来平衡。

在具体的项目中，开发人员通常更关注的是另一类更加细节的专业问题，例如：（1）产品性能的具体要求，例如运行在何种设备上，以及定量的稳定性要求如何；（2）采用的语言、框架、程序库，其技术复杂度如何，以及资料是否翔实等；（3）待处理的标准文本格式规范，例如UTF-8编码以及BOM头规格；（4）需测试操作系统初始环境中所有类型的文本，包括.ini、.xml和.reg等；（5）该产品应由多行（ES_MULTILINE）的Edit来实现。

<<大道至易>>

媒体关注与评论

本书作者站在架构师的高度上来看待软件开发，甚至更大的意义上，看待以软件开发能力为核心的企业在产品战略、人力配置和文化方面的组织问题。

作者在过去几年里，先后在两家超大规模互联网企业中做架构师，他在其中的矛盾、迷惘、挣扎和成就就是这本书的思想源泉，而他对于架构师角色的实习和领悟，也是这本书的价值所在。

作者采取的方式，就是老老实实把自己的思维过程和喃喃自语都很原生态地记录下来，从而形成一本“心语”之作。

因此，如果读者也能够“心心相印”，潜下心来仔细品读这本书，那么就相当于跟作者一起进行了一段旅行对话，或者，一起在思维的教武场里杀个几进几出。

或许这种如切如磋、如琢如磨、反复捶打、反复锻造的风格，倒确实能够帮助有心的读者获得一些架构设计的经验。

因此，无论从内容上，还是从行文风格上，这本书都是非常独特的，也有其独特的价值。

作者的作品，一向是长销之作，其价值也往往需要一段时间才能被真正认识，我不敢说自己能够在短暂的阅读当中充分理解这本书，但是我相信其中所蕴藏的价值和诚意。

——孟岩（序作者）

<<大道至易>>

名人推荐

本书作者站在架构师的高度上来看待软件开发，甚至更大的意义上，看待以软件开发能力为核心的企业在产品战略、人力配置和文化方面的组织问题。

作者在过去几年里，先后在两家超大规模互联网企业中做架构师，他在其中的矛盾、迷惘、挣扎和成就就是这本书的思想源泉，而他对于架构师角色的实习和领悟，也是这本书的价值所在。

作者采取的方式，就是老老实实把自己的思维过程和喃喃自语都很原生态地记录下来，从而形成一本“心语”之作。

因此，如果读者也能够“心心相印”，潜下心来仔细品读这本书，那么就相当于跟作者一起进行了一段旅行对话，或者，一起在思维的教武场里杀个几进几出。

或许这种如切如磋、如琢如磨、反复捶打、反复锻造的风格，倒确实能够帮助有心的读者获得一些架构设计的经验。

因此，无论从内容上，还是从行文风格上，这本书都是非常独特的，也有其独特的价值。

作者的作品，一向是长销之作，其价值也往往需要一段时间才能被真正认识，我不敢说自己能够在短暂的阅读当中充分理解这本书，但是我相信其中所蕴藏的价值和诚意。

——孟岩

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>