

The Art of Software Engineering — 36 Essential Strategies

软件工程实践导论

有关方法、设计、实现、管理之三十六计

金尊和 著

Jin Zunhe

方裕篇 經濟成本 用戶優先
 管理需求 及早集成 連續驗證
 控制變化 規範流程 快速應付
 借雞下蛋 開放標準
 設計篇 標準規範 因地制宜
 變中求靜 外層于齒 內外有序
 模塊封裝 合縱橫口 面向對象
 巧用標版 鬆伸緊縮 齊用成解
 見好就收
 實現篇 清不重復 邏輯易懂
 精益求精 優化性能 貽貓畫虎
 嚴堵漏洞 雁過留聲 自動流程
 初器常新
 管理篇 分工協作 目標驅動
 常奔常往 系統系統 不斷總結



清华大学出版社

Springer



The Art of Software Engineering — 36 Essential Strategies

软件工程实践导论

有关方法、设计、实现、管理之三十六计

金尊和 著



清华大学出版社
北京

 Springer

内 容 简 介

本书把软件工程放在一个更大的商业视角中进行考察和分析,对最新的软件工程理论从实践的角度重新检验,并提出作者独特的思考方法和新的理论。

本书在形式上参照兵法三十六计,总结了软件工程实践中的一些基本原则、出发点和解决问题的思路和方法,涉及软件工程的方法论、软件设计、编程实现以及管理等四个方面。每个方面包含若干计,每一计为一个重要的方法,单独成为一个章节。

本书面向专业软件开发人员、一般软件从业人员、信息产业管理人士、计算机等专业的教师和在校生以及对软件和信息产业感兴趣的任何读者,也包括希望对软件工程有一定了解或对创造性方法论感兴趣的读者。

版权所有,翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

软件工程实践导论——有关方法、设计、实现、管理之三十六计/金尊和著. —北京:清华大学出版社,2005.4

ISBN 7-302-10576-6

I. 软… II. 金… III. 软件工程—研究 IV. TP311.5

中国版本图书馆 CIP 数据核字(2005)第 015248 号

出 版 者: 清华大学出版社
<http://www.tup.com.cn>
社 总 机: 010-62770175

地 址: 北京清华大学学研大厦
邮 编: 100084
客 户 服 务: 010-62776969

责任编辑: 王一玲

特邀编辑: 陈 力

版式设计: 刘祎森

印 刷 者: 北京市清华园胶印厂

装 订 者: 北京国马印刷厂

发 行 者: 新华书店总店北京发行所

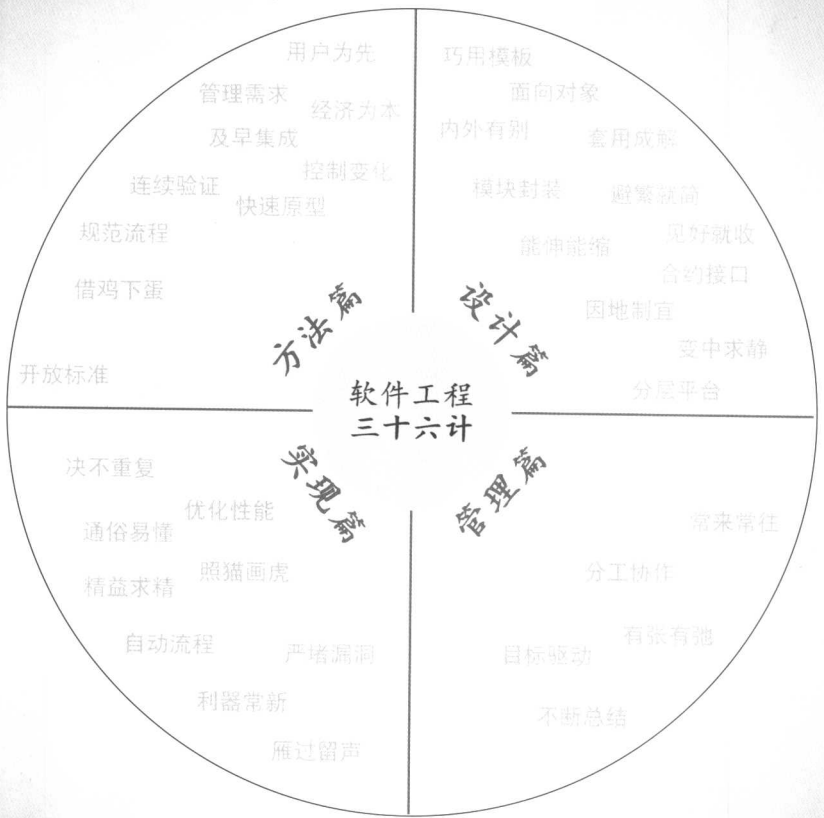
开 本: 153×230 印 张: 26.75 插 页: 1 字 数: 395 千字

版 次: 2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷

书 号: ISBN 7-302-10576-6/TP·7173

印 数: 1~3000

定 价: 48.00 元





当今信息社会,软件和芯片几乎全方位地影响到国家经济、人民生活和国家安全。因此,我国把软件和芯片作为信息产业的重中之重,希望把软件产业做大做强,但发展软件产业的期望与现实之间尚存明显差距。在这样的背景下,本书作者独辟蹊径,从一个全新的角度来审视和分析软件工程实践,借用孙子兵法三十六计的形式,从方法论层次和战略高度概括总结了软件开发的固有规律和原则方法,以推动我国软件开发水平的提高和软件产业的发展,无疑是十分有意义的。

本书写作充满了辩证思维,折射出作者深厚的理论基础、文化素养和丰富的实践经验。书中雪球式开发的借喻,生动贴切地反映了软件本身成长的固有规律;有关控制理论在软件开发系统中的实际应用,别具一格,突出了软件工程作为一个多环系统的特点,从理论上回答了一些软件实践中的问题。在此基础上衍生发展出的软件三十六计,则系统地概括了方法、设计、实现和管理四个方面的原则策略。作者对于软件工程的认识不局限于工程本身,而是从更大的视野来考量,比如对于软件工程中的经济因素的认识(见第一计“经济为本”),揭示了软件工程发展背后的推动力量,更为衡量软件实践和理论确立了一个终极评判和检验的标准。

该书在探讨软件工程理论的同时,非常重视软件实践,强调实用性和辩证性,并结合作者多年的实际项目经验,实例讲解,条理清楚,通俗易懂。而且,注意借鉴国际上工业界和学术界的最新成果和动态,对每个引用的重要观点,作者均有深入的剖析和独立的见解。本书旁征博引,文笔流畅,大量使用了人们熟悉的中国诗文、谚语或俗

语,乃至幽默故事,使原本抽象枯燥的工程知识读来生动有趣。

本书透过现象抓本质,源于实践而又高于实践,反映了作者软件工程方面的深厚底蕴和宽广的知识面。作者在多家国际性大公司中从事商业性大型软件开发和研究工作十年,涉猎广泛,从中积累了非常丰富的实际经验和实践素材,并善于总结思考,提炼出精华,进而上升到理论的高度。从这个意义上来说,今天的这本书是水到渠成之作。

正如作者所言,软件的实践充满了未知和挑战,对它的追求也是永无止境的。广大软件工程从业人员可以本书作为参考与借鉴,拓展思路,不断丰富与完善软件工程的方法、设计、实现和管理的策略。让我们共同为提高软件开发水平,为发展我国软件产业作出应有的贡献。

中国工程院院士

国家 CIMS 工程研究中心主任

清华大学教授 吴澄

2004 年 10 月



清华大学出版社



屈指一算自从第一次接触计算机和软件至今已有近 20 年的时间。想当初,刚进入清华大学自动化系读书不久,便开始学习 FORTRAN 语言,在中央主楼二楼的 PDP 小型机上输入和调试,排队等待针式打印机的结果,感觉计算机和软件是非常神奇和费时的。没想到,从此竟与计算机结下了不解之缘。

大学三年级以后,参加了茅于杭教授的语言研究实验室,在那里的 486 微机上学习 C 语言,修改、调试数万行的英文印刷体字符软件。三年后读研究生时,斗胆走上讲台给 92 级的同学讲授 C 语言编程。因为授课内容强调实用性和简单性,并结合亲身的经验,还颇受好评。

后来师从吴澄院士,先后设计并完成了数控机床分布式控制通信软件、CIMS 单元层控制仿真器、基于客户服务器的 CIMS 软件集成平台等项目,并出访中国香港和新加坡的大学机构,进行相关的研究和软件实现工作。

博士毕业后,更是一步踏入 IBM 中国研究中心,专门从事网络多媒体应用系统的研究。后来,来到美国加州从事卫星通信系统和无线网络通信系统的软件开发。2000 年,加入闻名世界的 Rational Software 后,一直从事软件工程的建模工具和流程规范的研究和开发。在业余时间,潜心研读了大量的相关的书籍和文章,参加各种与软件相关的讲座和交流。更重要的是,把别人的经验对比自己的实践,感觉颇多,觉得有写本书的必要。

计算机和软件的发展很快,新的概念名词和技术手段层出不穷,可谓日新月异。停留于表面的现象,可以让人眼花缭乱,应接不暇。只有深入到本质,从哲学方法论的高度上看问题、找答案,才能如庖

了解牛,游刃有余。所以,这本书的着眼点是软件工程的实践,但定位是在方法论层次上的,在强调实用的同时,又超越具体的软件形式和形态。

清华大学老校长蒋南翔对教育有个很好的比方。对一个要到森林中去的人可以给他一大包干粮,或者给他一支猎枪。干粮很快会吃完,但有猎枪等于有了谋生的手段。所以清华的教育是教会使用猎枪,也就是说让学生有正确的自我学习方法,而不仅仅是掌握知识本身。软件的方法论就是一支猎枪。

有了这个基本的想法后,又联想到孙子兵法与三十六计总结千变万化的战争规律的方法,对于复杂千变的软件工程来说,无疑是个切合实际的办法,于是有了软件三十六计的第一稿提纲。从内容上,既从亲身经验出发总结和探讨了软件工程的实践,又借鉴了当今世界最新的研究和工业界的实践成果。

这个提纲立刻得到了正在硅谷短期访问的吴澄院士的赞赏,提出了许多建设性的意见,并推荐给清华大学出版社。在成书的过程中,吴老师仔细审阅了每个部分,对内容和写作各个方面都严加把关。今天此书能面世,首先要感谢我的导师吴澄院士,在我毕业多年后,一如当初在校诲人不倦。

在完成初稿后,作者得到了思科(Cisco)公司高级软件工程师杨云雷先生,甲骨文(Oracle)公司的高级工程师郑佩女士,IBM公司硅谷开发中心高级工程师张建国先生,雅虎(Yahoo)公司高级工程经理吕厚昌先生,高级工程师罗鹏先生,惠普(HP)公司高级工程师陈冬妮女士,中国科学院计算所贺思敏博士等人的大力帮助。他们在百忙中抽出时间审阅了初稿,并提出了很多宝贵的意见和建议。对于他们的帮助和贡献,笔者不胜感激。

在成书的过程中,清华大学出版社王一玲、王仁康和陈力等老师在全书的技术准确性、编辑组织、文字润色等方面均给予了很大的帮助,她们的专业精神和敬业态度令作者非常钦佩,在此一并表达衷心的感谢。

作者

2004年秋于美国硅谷

第一套 胜战计

瞒天过海 围魏救赵 借刀杀人
以逸待劳 趁火打劫 声东击西

第二套 敌战计

无中生有 暗渡陈仓 隔岸观火
笑里藏刀 李代桃僵 顺手牵羊

第三套 攻战计

打草惊蛇 借尸还魂 调虎离山
欲擒故纵 抛砖引玉 擒贼擒王

第四套 混战计

釜底抽薪 混水摸鱼 金蝉脱壳
关门捉贼 远交近攻 假途伐虢

第五套 并战计

偷梁换柱 指桑骂槐 假痴不颠
上屋抽梯 树上开花 反客为主

第六套 败战计

美人计 空城计 反间计
苦肉计 连环计 走为上



卷之二 软件工程导论



绪论 1

- 0.1 历史的回顾 /1
- 0.2 软件的定义 /2
- 0.3 什么是软件工程? /5
- 0.4 软件工程的大环境 /7
- 0.5 软件工程究竟像什么? /8
 - 0.5.1 雪球理论——做软件就像滚雪球 /9
 - 0.5.2 软件工程控制论——软件生产过程是个离散多环控制系统 /10
 - 0.5.3 兵法三十六计的启示 /11
- 0.6 本书的组织 /13

方法篇

1 经济为本 17

- 1.1 投入产出的概念 /18
- 1.2 经济为本的启示 /20
 - 1.2.1 为技术而技术是条死胡同 /20
 - 1.2.2 以经济原则指导软件项目的决策过程 /21
 - 1.2.3 按照产品规律来营销软件产品 /21
 - 1.2.4 以收益为依据规划设计产品 /23
- 小结 /23

2	用户为先	25
2.1	用户需求分析 /26	
2.2	用户友好的交互界面 /26	
2.2.1	界面设计和风格 /26	
2.2.2	界面的本地化 /27	
2.3	软件安装和部署 /28	
2.4	用户培训和技术认证 /30	
2.5	用户大会和开发人员网络 /30	
2.6	用户文档 /31	
2.7	用户支持和反馈 /31	
2.8	系统升级和版权管理 /32	
2.9	软件咨询服务 /34	
	小结 /35	
3	管理需求	37
3.1	什么是软件需求? /37	
3.2	了解需求 /39	
3.3	记录需求 /41	
3.4	追踪需求 /42	
3.5	在变化中管理需求 /43	
	小结 /44	
4	及早集成	47
4.1	软件工程项目的风险 /49	
4.2	及早集成才能滚雪球 /50	
4.3	少食多餐 /51	
	小结 /52	
5	连续验证	53
5.1	有关测试的理论 /54	
5.2	测试自动化 /55	
5.3	软件缺陷跟踪系统的概念和实际应用 /57	

5.3.1	软件缺陷的生命周期	/58
5.3.2	欲穷千里目,更上一层楼	/58
5.4	软件的“测不准”原理	/59
5.5	另类的验证	/59
	小结	/60
6	控制变化	61
6.1	循序渐进	/62
6.2	同步协调	/64
	小结	/65
7	规范流程	67
7.1	规范要点	/68
7.2	CMM	/69
7.3	Rational Unified Process	/70
7.4	极限编程	/70
7.5	发展方向	/71
	小结	/73
8	快速原型	75
8.1	软件中的“概念车”	/75
8.2	大行不顾细谨	/77
8.3	走向产品	/78
	小结	/79
9	借鸡下蛋	81
9.1	外包	/81
9.2	买进	/83
9.3	开放代码运动	/84
	小结	/86
10	开放标准	89
10.1	开放的意义	/89

10.2	开放标准的种类	/91
10.3	支持开放标准	/92
	小结	/93

设计篇

11 避繁就简 97

11.1	“不简单”的设计	/97
11.1.1	面向履历表的设计	/97
11.1.2	“难得糊涂”的设计	/98
11.1.3	委员会式的设计	/99
11.2	简单就是金钱	/99
11.3	简单才真的不简单	/100
11.4	关键是平衡	/103
11.5	皇帝的新衣	/104
	小结	/105

12 因地制宜 107

12.1	有没有万能钥匙?	/107
12.2	考量的因素	/108
12.3	软件设计中的若干对矛盾	/110
12.4	要马儿跑又要马儿不吃草	/116
	小结	/117

13 变中求静 119

13.1	变在那里?	/119
13.2	针对变化而设计	/122
13.2.1	数据和代码的分离	/122
13.2.2	输入数据和配置参数的分离	/124
13.2.3	不变的和常变的代码分离	/125
13.3	灰色地带	/127
13.4	XML 和数据化	/129

小结	/130	
14 分层平台	131
14.1 巨人的肩膀	/131	
14.2 为什么软件要分层?	/132	
14.2.1 层次提升	/132	
14.2.2 隐藏细节	/133	
14.2.3 标准互换	/134	
14.3 分层的原则	/135	
14.3.1 实现和接口分离原则	/135	
14.3.2 单向性原则	/135	
14.3.3 服务接口的粒度提升原则	/136	
14.4 典型的分层设计	/136	
14.4.1 OSI 模型	/136	
14.4.2 三层应用结构	/138	
14.5 分层的网络化	/138	
14.5.1 客户/服务器计算模式	/138	
14.5.2 万维网的拓展	/139	
14.6 应用编程接口 API	/139	
14.7 服务编程接口 SPI	/141	
小结	/142	
15 内外有别	143
15.1 MVC 结构	/144	
15.2 万维网应用也要内外有别吗?	/146	
15.3 模型的设计	/147	
15.4 正交化设计	/148	
小结	/149	
16 模块封装	151
16.1 软件芯片	/151	
16.2 静态模块的粒度	/152	
16.2.1 函数	/153	

- 16.2.2 文件 /153
- 16.2.3 类 /154
- 16.2.4 相关类集合 /154
- 16.2.5 子系统 /155
- 16.3 动态模块 /155
 - 16.3.1 构件 /155
 - 16.3.2 插入件 /155
 - 16.3.3 进程 /156
 - 16.3.4 可执行应用 /157
 - 16.3.5 企业运行环境 /157
- 16.4 模块关系的分析 /157
 - 16.4.1 代码类关系 /158
 - 16.4.2 数据类关系 /158
- 16.5 模块划分 /159
 - 16.5.1 分割的原则 /159
 - 16.5.2 模块间的通信 /160
- 16.6 软件构件复用的理想 /160
 - 16.6.1 复用情形下的新问题 /160
 - 16.6.2 没有金刚钻,别揽瓷器活 /162
- 小结 /163

17 合约接口 165

- 17.1 软件世界的“合同” /165
- 17.2 编程接口设计的“五项基本原则” /166
- 17.3 网络条件的新问题 /171
- 17.4 从“貌合”到“神合” /173
 - 17.4.1 隐含规则 /173
 - 17.4.2 言行一致 /174
- 17.5 非程式接口 /175
- 小结 /175

18 面向对象 177

- 18.1 面向对象的三性 /178

18.1.1	继承性和逻辑三段论	/178
18.1.2	封装性	/180
18.1.3	多态性	/181
18.2	继承“税”	/182
18.2.1	多包“含”	/182
18.2.2	单重继承	/183
18.3	面向对象的建模和 UML	/183
18.3.1	软件建模	/183
18.3.2	建模语言	/184
18.4	双向工程	/185
18.5	面向对象的编程语言	/187
	小结	/188
19	巧用模板	189
19.1	变模板为“魔”板	/190
19.1.1	编程	/190
19.1.2	代码生成	/191
19.1.3	数据转换	/193
19.1.4	产生文档	/194
19.2	通用模板语言和引擎	/195
19.2.1	Velocity	/196
19.2.2	XSLT	/200
19.3	模板也要正交吗?	/203
19.3.1	模板流水线	/203
19.3.2	模板的模块化	/203
19.3.3	模板 MVC	/203
	小结	/204
20	能伸能缩	205
20.1	伸缩性要从设计抓起	/205
20.2	结构设计的适应性	/206
20.2.1	并行度规划	/206
20.2.2	逻辑分层和物理分布	/207

20.2.3	同步还是异步	/207
20.2.4	资源的竞争和平衡	/207
20.3	多线程安全性	/208
20.3.1	调用函数重入	/208
20.3.2	变量可见度	/209
20.3.3	死锁和保护	/210
20.4	算法可伸缩	/211
20.4.1	算法并行的可能性	/211
20.4.2	线性化算法的必要性	/211
20.4.3	递归算法	/212
小结		/214

21 套用成解 215

21.1	框架	/215
21.2	面向对象方法中的设计定式	/219
21.2.1	生成类定式	/219
21.2.2	行为类定式	/221
21.2.3	结构类定式	/226
小结		/228

22 见好就收 229

22.1	软件设计的使命	/229
22.2	见好不收的误区	/231
22.2.1	追求完美	/231
22.2.2	画蛇添足	/232
22.2.3	清谈误国	/232
小结		/233

实现篇

23 决不重复 237

23.1	重复是万恶之首	/237
------	---------	------