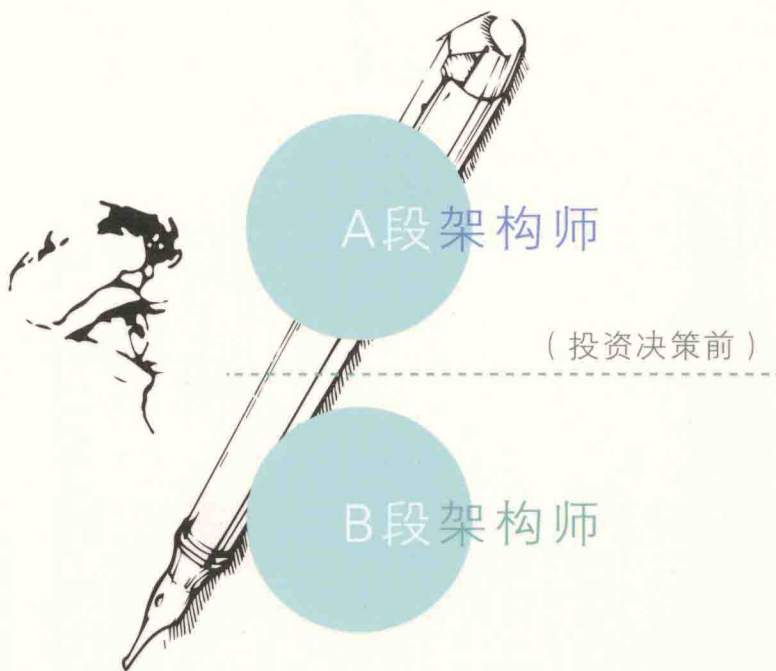


# 思考软件, 创新设计

## A段架构师的思考技术

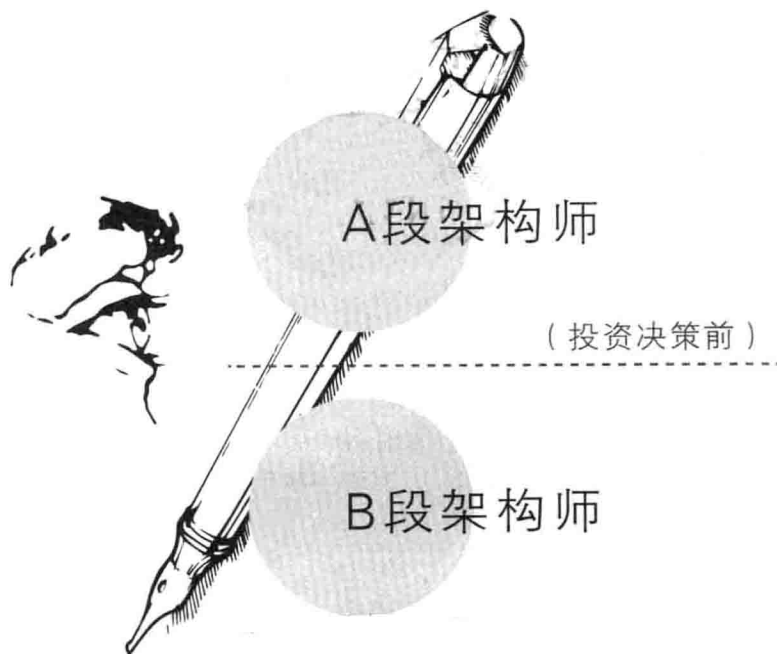
高焕堂 【美】Annping Kao 著



# 思考软件, 创新设计

## A段架构师的思考技术

高焕堂 【美】Annping Kao 著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

架构师是一个技术制高点的岗位，在智能时代（如大数据、云计算和移动互联网潮流）里，它位居于领导位置，与企业高阶经理们是互补的，或者能汇集于一身的。

本书介绍当今全球正蔚为风潮的设计思考（Design Thinking）技术，陪伴你从目前的编程或设计岗位进入架构设计（A段和B段），踏上灿烂的软件（程序）生涯的新旅程。其内容涵盖了A段架构师的主要策略思维，以及许多实践案例，包括软硬整合商业模式范例、产品策略案例演练、架构设计的学习步骤、云平台架构设计法则等；并阐述从策略、设计到代码的落地步骤。

相信本书会是你的一匹千里马，陪伴你驰骋于广大的IT原野上，并发挥创新潜能，挑战人生颠峰。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

### 图书在版编目（CIP）数据

思考软件，创新设计：A段架构师的思考技术 / 高焕堂，（美）高燕平著. —北京：电子工业出版社，2014.6  
ISBN 978-7-121-22736-3

I. ①思… II. ①高… ②高… III. ①软件设计 IV. ①TP311.5

中国版本图书馆CIP数据核字（2014）第057308号

责任编辑：董 英

印 刷：三河市双峰印刷装订有限公司

装 订：三河市双峰印刷装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编100036

开 本：787×980 1/16 印张：20 字数：424千字

印 次：2014年6月第1次印刷

定 价：59.00元



凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：（010）88258888。

# 序 言

---

2012年6月，我去拜访鲁迅美术学院大连校区时，在其移动新媒体实验室赵奇光责任导师的陪同下，拜会了传媒动画学院院长王亦飞教授。一见面，王院长就问我说：我们未来怎么合作呢？我回答说：IT青蛙想找到一位设计公主，然后希望从此过着幸福快乐的日子。当时我有一个愿景(Vision)：想过着幸福快乐的日子。于是草拟了一个假定(Hypothesis)：只要IT青蛙能找到一位设计公主，就能从此过着幸福快乐的日子。我依循溯因(Abductive)逻辑思维，寻觅众多证据，都未能推翻这项假定，就继续追求下去。于是，这本书诞生了。希望这本书能抛砖引玉，让更多IT青蛙变王子，和公主从此过着幸福快乐的日子。

本书的主要内容，来自于我2013年从与日本IT产业合作项目退休之前，基于日本师徒制的要求而传承给下一代架构师的架构思考技术（俗称设计心法）。25年来我专精于A段（投资决策前）架构设计，退休闲暇将之改写成中文，并加以润饰成为本书。在这个过程中，得到台湾高雄师范大学视觉设计系学生Annping Kao的许多协助，包括提供许多设计思考(Design Thinking)的书籍素材和她的宝贵见解，于此特别致以万分谢意。

最后，由衷感谢传媒动画学院院长王亦飞教授和移动新媒体实验室赵奇光责任导师的支持与鼓励。

高焕堂 谨识于 台北  
台湾 Android 论坛主席  
misoo.tw@qq.com

十载耕耘  
奠定专业地位

## 博文视点诚邀精锐作者加盟

以书为证  
彰显卓越品质

《C++Primer (中文版) (第5版)》、《淘宝技术这十年》、《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目,被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、咎辉Zac、李刚、曹江华……

“明星”级作者济济一堂,他们的名字熠熠生辉,与IT业的蓬勃发展紧密相连。

十年的开拓、探索和励精图治,成就博古通今、文圆质方、视角独特、点石成金之计算机图书的风向标杆:博文视点。

“凤翱翔于千仞兮,非梧不栖”,博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟,与大师并列于IT专业出版之巅。

### 英雄帖

江湖风起,代有才人出。

IT界群雄并起,逐鹿中原。

博文视点诚邀天下技术豪加入,

指点江山,激扬文字

传播信息技术,分享IT心得

### 专业的作者服务

博文视点自成立以来一直专注于IT专业技术图书的出版,拥有丰富的与技术图书作者合作的经验,并参照IT技术图书的特点,打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持:

**善待作者**——我们会把出版流程整理得清晰简明,为作者提供优厚的稿酬服务,解除作者的顾虑,安心写作,展现出最好的作品。

**尊重作者**——我们尊重每一位作者的技术能力和生活习惯,并会参照作者实际的工作、生活节奏,量身制定写作计划,确保合作顺利进行。

**提升作者**——我们打造精品图书,更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力,为作者的事业开拓带来更多的机会。



### 联系我们

博文视点官网: <http://www.broadview.com.cn>

CSDN官方博客: <http://blog.csdn.net/broadview2006/>

投稿电话: 010-51260888 88254368

投稿邮箱: [jsj@phei.com.cn](mailto:jsj@phei.com.cn)



@博文视点Broadview



微信公众平台: 博文视点Broadview



### 关于本书用纸的温馨提示

亲爱的读者朋友: 您所拿到的这本书使用的是**环保轻型纸!**

环保轻型纸在制造过程中添加化学漂白剂较少,颜色更接近于自然状态,具有纸质轻柔、光反射率低、保护读者视力等优点,其成本略高于胶版纸。为给您带来更好的阅读体验并与读者共同支持环保,我们在没有提高图书定价的前提下,使用这种纸张。愿我们共同分享纸质图书的阅读乐趣!

# 博文视点精品图书展台

## 专业典藏



## 移动开发



## 大数据 · 云计算 · 物联网



## 数据库



## Web 开发



## 程序设计



## 软件工程



## 办公精品



## 网络营销



# 目 录

---

第 1 章 商业思维：从商业模式到架构设计	1
1.1 A 段架构师的角色	1
1.2 面对复杂，唯有简单	2
1.3 商业环境与模式是复杂的	3
1.4 软件的复杂是本质性的（并非表象而已）	5
1.5 如何从复杂中设计出简单	7
1.5.1 单纯与美丽的密切联系	7
1.5.2 软件美何在，把酒问<对称>	7
1.5.3 软件架构的第 1 种对称性	8
1.5.4 软件架构的第 2 种对称性	8
1.6 参考文献	9
第 2 章 软硬整合商业模式范例	10
2.1 复习：A 段架构设计	10
2.2 产品设计支持商业模式	11
2.2.1 反思既有的商业模式	12
2.2.2 创立新型商业模式	12
2.3 商业模式范例	13
2.3.1 长期的获利策略	13
2.3.2 短期的获利策略	14
2.3.3 硬硬结合销售	14
2.3.4 软硬整合开发	16
2.4 产品设计范例	17
2.5 延伸到客厅配件市场	18
2.5.1 客厅配件范例	18
2.5.2 客厅配件的商业模式	19

2.5.3 与微信、微博或百度地图等平台对接	20
2.6 产品的架构设计范例	21
2.6.1 基本元素和功能	21
2.6.2 应用情境（举例）	22
2.6.3 架构设计与实践	23
2.7 参考文献	23
<b>第 3 章 获利思维与心境</b>	<b>24</b>
3.1 获利思维 VS.成本思维	24
3.2 让架构师身价百倍的招式	26
3.3 领悟信用卡 VISA 商业模式	27
3.4 以农夫挑水浇花为例	28
3.5 设计客厅配件产品的商业（合作）模式	31
3.5.1 复习：客厅配件的商业模式	31
3.5.2 一样的心境	32
3.5.3 创立 OFA 联盟	33
3.6 参考文献	34
<b>第 4 章 创新思维</b>	<b>35</b>
4.1 溯因推理与创造性	35
4.2 溯因推理基本观念	36
4.3 <假定-否定>= 删除法	39
4.4 孔明的溯因推理范例	41
4.5 麦肯锡的 MECE 原则	45
4.5.1 认识 MECE 原则	45
4.5.2 议题树及其种类	46
4.5.3 持续对议题进行分解和检验	47
4.5.4 符合 MECE 原则	47
4.5.5 持续进行<检验-否定>	48
4.5.6 展开下一阶层的议题	49
4.5.7 MECE 范例	51
4.6 软件敏捷（Agile）开发原则	54
4.6.1 认识敏捷开发原则	54
4.6.2 敏捷幕后的溯因推理	55
4.6.3 A 段架构师的敏捷力	57



4.7	敏捷设计: 应用于智慧城市顶层设计	58
4.7.1	<顶层设计>含义的迷思	58
4.7.2	创新的起点: 愿景 (Vision)	59
4.7.3	观察现实, 提出假定 (Hypothesis)	60
4.7.4	结论: 提出<敏捷顶层设计方法>	61
4.7.5	应用: 敏捷顶层设计步骤说明	62
4.8	创新设计: <发现 EIT 造形>之旅	62
4.8.1	什么是代码造形?	62
4.8.2	大家熟悉的两种代码造形: 函数和类	63
4.8.3	创新的起点: 愿景 (Vision)	64
4.8.4	提出假定	65
4.8.5	提出新的假定	65
4.8.6	结论: 提出 EIT 代码造形	66
4.8.7	为什么需要有两个配角	68
4.9	参考文献	69
<b>第 5 章</b>	<b>组合思维</b>	<b>70</b>
5.1	从古典抽象思维谈起	70
5.2	创新组合思维	72
5.2.1	<合>的设计法则	72
5.2.2	序中有乱: 以集装箱为例	73
5.2.3	演练: 如何抽象出<集装箱>	74
5.2.4	观摩: 软件集装箱	75
5.3	组合设计的要素: 造形 (Form)	77
5.3.1	从集装箱认识“造形”	77
5.3.2	认识 EIT 软件造形	78
5.3.3	造形的特性	81
5.3.4	造形的内部元素	81
5.3.5	造形的外部组合规律	82
5.4	演练范例: 寻找 (设计) 软件系统的造形	85
5.5	演练范例: 从简单造形组合出复杂	90
5.5.1	范例 (一): 一对 EIT 造形包容善变的通信协议	90
5.5.2	范例 (二): EIT 造形的巢状组合	93
5.5.3	范例 (三): 从 EIT 造形组合出 GoF 设计模式	95
5.6	反思: 儒家文化中的创新组合思维	99

5.6.1	分解的尽理之精神	99
5.6.2	中学为体，西学为用	99
5.7	设计出未来性	103
5.7.1	今日决策的未来性	103
5.7.2	愿景与连线	104
5.7.3	愿景从哪里来	105
5.7.4	组合与创新	105
5.7.5	演练：未来性架构设计	107
5.8	参考文献	113
<b>第 6 章</b>	<b>战术引导战略思维</b>	<b>114</b>
6.1	架构师擅长于战术引导战略	114
6.2	会赢的战术	116
6.3	找会赢的战术，调整战略资源	117
6.4	<战术引导战略>的企业案例	118
6.5	参考文献	121
<b>第 7 章</b>	<b>四项假设性思维</b>	<b>122</b>
7.1	从溯因推理谈起	122
7.2	假设 (Assumption) 是什么	125
7.3	第一项：不自觉的假设，放宽思维局限	125
7.3.1	一般范例	125
7.3.2	孔明空城计的范例	126
7.3.3	其他范例	127
7.4	第二项：有待被检验的假定，激发愿景想象	129
7.5	第三项：完成性的假设，想象最终结果	131
7.5.1	什么是完成性假设	131
7.5.2	Mapping from vision to reality 与溯因推理	133
7.5.3	“Mapping” 如何促进 Know Unknown	135
7.5.4	演练：Mapping from vision to reality	137
7.5.5	演练：全脑观想 (Visualization)	139
7.5.6	演练：完成性假设	140
7.6	第四项：万一性的假设，预留弹性空间	148
7.6.1	Why? 万一性的假设	148
7.6.2	如何预见失败	149

7.6.3	认识 Possibility 和备胎思维	151
7.7	参考文献	152
<b>第 8 章</b>	<b>架构师策略思维的案例演练</b>	<b>153</b>
8.1	产品策略思维	153
8.1.1	产品策略目标: 兼顾独特性、普遍性和可靠性	153
8.1.2	实践方案: 软硬整合、跨平台与测试	154
8.1.3	从简单中掌握复杂	155
8.1.4	对假定进行检验	157
8.2	发挥四项假设性思维	158
8.2.1	复习: 溯因 (Abductive) 推理	158
8.2.2	观察现实, 观想愿景	159
8.2.3	推导出假定	160
8.2.4	进行 Mapping from vision to reality	161
8.2.5	重大发现	163
8.2.6	决定跨平台策略的方向	165
8.3	订定跨平台实践策略	165
8.3.1	复习: 探索策略方向	165
8.3.2	拟订实践策略	165
8.3.3	策略-1: 把它“EIT (设计)”了	166
8.3.4	策略-2: 挟天子以令诸侯	167
8.3.5	策略-3: 建立中间件 (middleware)	167
8.4	跨芯片小平台策略	167
8.4.1	什么是<跨平台>	167
8.4.2	终端产品跨 (芯片) 小平台的 3 种情境	169
8.5	跨 Android 大平台策略	174
8.5.1	这个<策略-2>从哪里来	174
8.5.2	策略-2 的实践	176
8.6	跨自己平台的策略	180
8.6.1	这个<策略-3>从哪里来	180
8.6.2	中间件跨平台的效益: 没钱就改版, 改版就有钱	182
8.6.3	中间件跨客户端的效益	183
8.6.4	中间件的设计	183
8.6.5	中间件在整体架构中的角色	185
8.7	参考文献	186

第 9 章 架构设计的十个学习步骤	187
9.1 这<十个步骤>从哪里来	187
9.1.1 从架构思维的 4 个元素出发	187
9.1.2 学好<创意爱上限制>	188
9.1.3 学好<创新组合>	190
9.1.4 学习敏捷测试和设计通用性接口	191
9.2 阐述<十个学习步骤>	192
9.2.1 第 1 步: 学习两种抽象视角 (Abstraction View)	192
9.2.2 第 2 步: 关心下层的变动自由度	193
9.2.3 第 3 步: <系统架构控制力>支撑<商业竞争话语权>	193
9.2.4 第 4 步: <用户体验>是用户从简单中叫出复杂的满足感	194
9.2.5 第 5 步: 创意爱上限制, 即需求检验设计	194
9.2.6 第 6 步: 假设性思维和 Mapping from vision to reality	195
9.2.7 第 7 步: 清晰而明确表述接口 (Interface)	195
9.2.8 第 8 步: 尽快对接口进行检验和测试	195
9.2.9 第 9 步: 设计通用性接口	196
9.2.10 第 10 步: 有效减法设计, 才能开放加法 (设计)	197
9.3 深入领悟 (一): 变动自由度	197
9.4 深入领悟 (二): 创意爱上限制	200
9.5 深入领悟 (三): Mapping from vision to reality	201
9.6 深入领悟 (四): 迅速检测接口	204
9.7 深入领悟 (五): 有效减法设计, 才能开放加法	208
9.7.1 举例: 以 Socket 接口来包容 TCP/IP 通信协议	208
9.7.2 演练步骤	208
9.8 参考文献	218
第 10 章 <Java/HBase + C>云平台架构设计的十项法则	219
10.1 这<十项法则>从哪里来	219
10.1.1 反思假设	219
10.1.2 关注于命令流及其控制点	221
10.2 阐述<十项法则>	222
10.3 思考议题&提示	227
10.4 参考文献	232

第 11 章 法则-2 的实践范例: 通用性接口设计 .....	233
11.1 通用性接口 .....	233
11.2 亲自设计通用性接口 .....	235
11.2.1 认识用例 (Use Case) 及其刷景 (Scenario) .....	235
11.2.2 设计通用性接口来支撑用例 (Use Case) .....	238
11.2.3 衔接到<UC-001: 播放音乐>的特殊性接口 .....	239
11.2.4 衔接到<UC-002: 播放某首音乐>的特殊性接口 .....	243
11.3 多个通用性接口的联合应用范例 .....	247
11.3.1 应用范例: 手机与 Android TV 的多机整合 .....	247
11.3.2 iPhone 手机端的规划 .....	248
11.3.3 衔接 iPhone 手机端的 (通用性) 接口: Servlet .....	249
11.3.4 App 的 Context 接口与 myTVServlet 对接 .....	253
11.3.5 Android 本地 App 的设计 .....	259
11.4 参考文献 .....	264
第 12 章 法则-3 的实践范例: 挟天子以令诸侯 .....	265
12.1 一般框架设计: 以<五子棋>为例 .....	265
12.1.1 阶段一: 从传统类 (Class) 造形设计出发 .....	265
12.1.2 阶段二: 继续运用 EIT 造形设计 .....	268
12.2 基于 Android 设计小框架: 以<涂鸦>应用为例 .....	272
12.2.1 介绍多层 EIT 造形 .....	272
12.2.2 简介「涂鸦」(Scribble) .....	274
12.2.3 <涂鸦>范例的功能说明 .....	274
12.3 实践<挟天子以令诸侯> .....	275
12.3.1 阶段一: 从传统类 (Class) 造形设计出发 .....	275
12.3.2 阶段二: 继续运用 EIT 造形设计 .....	276
12.4 参考文献 .....	308

# I

## 第 1 章

# 商业思维：从商业模式到架构设计

### 1.1 A 段架构师的角色

通常，架构设计分为 A 段和 B 段架构设计，如图 1-1 所示。

- A 段：是指“投资决策”或“签订开发合同”之前的产品或项目的规划阶段。
- B 段：则是指上述“投资决策”或“签订开发合同”之后的开发建置阶段。

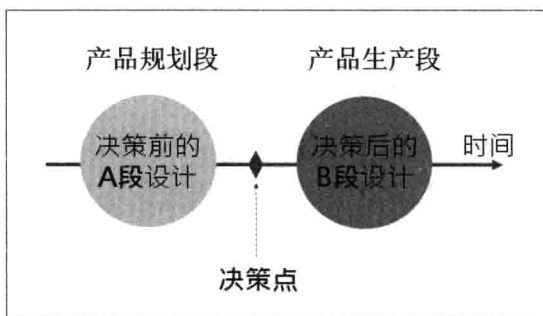


图 1-1 A 段与 B 段架构设计

架构师通常需要涵盖和掌控 A 段（决策前）和 B 段（决策后）的产品相关策略的制定和执行监督。在 A 段，面对环境迅速变迁下的产品策略规划，在 B 段则面对团队的实践策略、执行能力和技术变迁。架构师必须具备鲜活的创新思维、睿智的策略思考、犀利的洞察力和灵活的战术，以便掌握稍纵即逝的商机。笔者与日本 IT 产业合作多年，大多偏于 A 段的架构设计，常被称为 A 段架构师。与其相对应，就是 B 段架构师了。两者比较明显的区别是，在规划段里，A 段架构师大多协助产品（Product）经理；而在生产段里，则协助生产（Production）经理。

位于这两个不同阶段的架构师，在思维或视角上，有许多不同（而且经常是互补的）。

例如, A 段架构师偏于获利思维, 而 B 段架构师则偏于成本思维, 如图 1-2 所示。

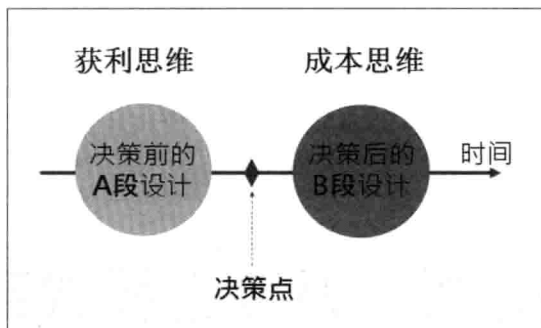


图 1-2 架构师的获利与成本思维

除了金钱方面, A 段和 B 段有获利与成本考虑的差异之外, 还有多项明显的差异。例如:

- A 段获利思维, B 段成本思维。
- A 段算计敌人, B 段设计自己。
- A 段预见失败, B 段势如破竹。

可以看出来, 其实两者之间是非常互补的。对于 A 段架构师而言, “知彼”往往比“知己”还重要; 所以, A 段的架构设计往往不是去设计自己, 而是去“算计”敌人(竞争者)。由于商场如战场, 就像大雾弥漫的丛林一般, 机会与陷阱并存; 知彼是非常重要的。

## 1.2 面对复杂, 唯有简单

——享受从简单中叫出复杂的满足感

面对复杂, 科学家(或架构师)必须设计出简单, 让相关人员皆能享受从简单中叫出复杂的满足感。例如, 17 世纪中期, 牛顿提出了简单公式(即造形)  $F=ma$ , 让人们能轻易理解物体运动的复杂<关系>; 再如, 20 世纪初, 爱因斯坦发表了简单公式  $E=mc^2$ , 让人们能理解复杂的质量、能量与光速之间的复杂关系, 如图 1-3 所示。

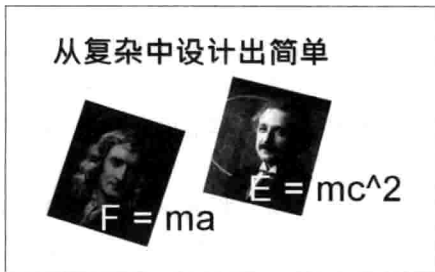


图 1-3 牛顿与爱因斯坦的简单之形(公式)

面对复杂，科学家（或架构师）必须设计出简单，让人人都能从简单中掌握复杂。不仅仅是科学家，艺术家也是如此。例如，画家毕加索曾说过，绘画是减法的过程（Process of Elimination）。画出的作品具有<简单性>，引领人们去领悟具有<复杂性>的真实。他说道：

“艺术并不是真实，它是一种谎言，让我们去领悟真实。”（Art is not truth. Art is a lie that makes us realize truth.）

被尊称为“史上最伟大的平面设计师”的保罗·兰德（P. Rand）也说，我们会从最复杂的部分开始着手，最后的成品却具有<简单性>。这个减法过程本身是复杂的，然而一旦达到简单了，就能发挥杠杆作用，让人们能面对变化、掌握复杂。苹果公司的乔布斯（Jobs）也曾说到：

“简单比复杂更难，因为必须努力让你的想法变得清晰明了，让它变得简单。但终究是值得的，因为只要你获得了其“简单性”，就能搬动大山了。”（Simple can be harder than complex: you have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains.）

### 1.3 商业环境与模式是复杂的

——有效减法设计，才能开放加法

商业（环境与）模式是复杂的，架构师也必须设计出简单，才能进一步支撑复杂的产品创新活动。前面所提到的获利思维，就是商业思维的一部分。商业思维主要涵盖了商业模式和策略，以及产品创新和架构设计等，如图 1-4 所示。

大家都知道，商场如战场，商业模式与策略必须持续调整，才能避开陷阱，捕获商机。其中，商业模式、创新产品与架构设计三者之关系可表示如图 1-5 所示。

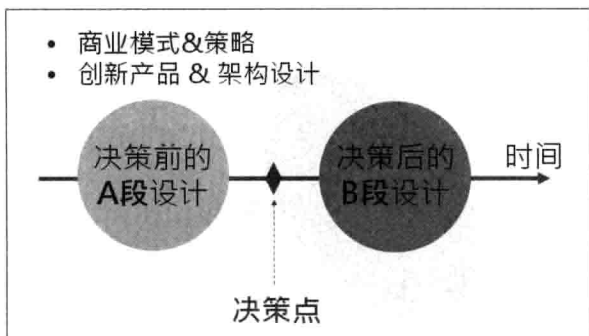


图 1-4 架构师的商业思维要素

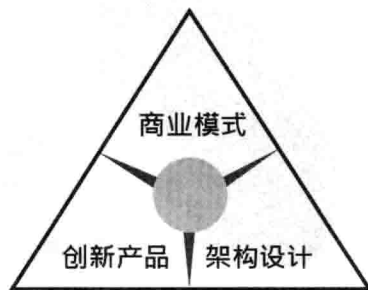


图 1-5 商业思维的三个要素

有效减法设计，才能开放加法

为了支持灵活的商业策略（包括战略和战术），产品必须不断推陈出新，以创新来提



供优越特性来吸引客户。因此，这两项都必须持续做加法，才能推陈出新，如图 1-6 所示。

由于市场环境瞬息万变，所以我们必须持续创新（加法），来面对非常复杂的市场环境。为了大幅支撑创意（加法），规划创新产品，来面对复杂市场。架构设计师必须力求减法（图 1-7），从复杂的市场竞争、用户需求、创新产品和商业模式中设计出简单（架构），让开发团队能从简单中掌握复杂，实践创意，做出创新产品，如图 1-8 所示。

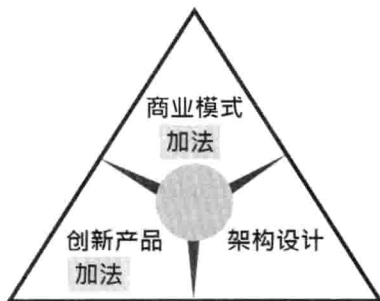


图 1-6 架构师支持商业创新（加法设计）

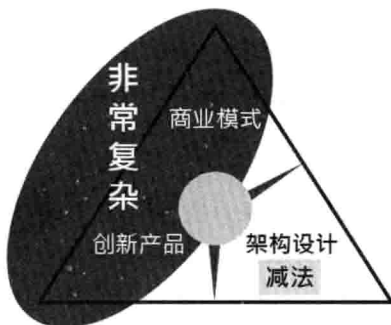


图 1-7 架构（减法）设计支撑商业创新

面对非常复杂，就得熟练商业思维，力求加法，促进创新设计。于是，设计师必须力求减法，从复杂设计出简单（架构），然后从简单来掌握复杂（的加法），捕捉复杂面貌下的无限商机。就像秦朝的书同文、车同轨，力求减法，才能统合天下（加法），随后唐朝的诗同形（例如，七言绝句的造形），激发无限创意，诗风鼎盛。这意味着：熟练商业思维，促进创新设计；有效减法设计，大幅开放加法；书同文诗同形，乃有汉唐盛世。

同样地，商业（环境与）模式是复杂的，架构师也必须设计出简单，才能进一步支撑复杂的产品创新活动，如图 1-9 所示。

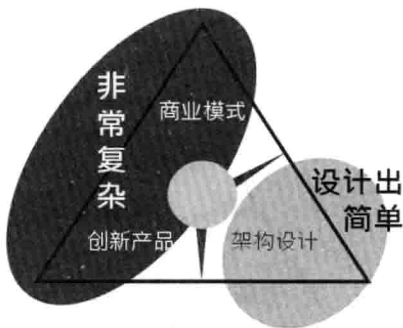


图 1-8 从复杂设计出简单（架构）

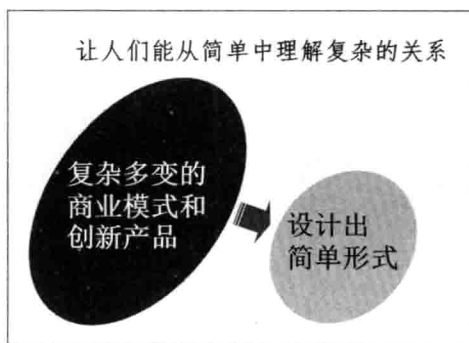


图 1-9 设计出简单是手段，理解复杂是目的

A 段架构师必须具备商业模式与策略思维，发挥洞察力，从复杂中找出简单，设计出有效的可实现计划（即 A 段架构设计），让高阶主管从简单中掌握复杂，做出正确决策。