

本PDF仅是样章,书籍版权归著者和出版社所有，未经允许不能在网上传播，  
如有需要，请尽量购买正版实体书，以表示对知识的尊重！实在有必要获  
取完整版本PDF，请联系QQ:2856202282,谢谢！

2856202282



# 聊聊“架构”

ARCHITECTURE TALKS 王概凯 / 著

## ARCHITECTURE TALKS 王概凯 / 著

王概凯 / 著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

架构是如何运作并影响人们的日常生活的，在软件行业中架构是如何运作的？架构又是如何指导代码编写的，如何把架构应用在软件工程实践上？带着这些疑问，本书通过大量的实例一步一步揭示出架构背后的原理，以及架构在软件行业的发展，并通过企业实例来展示软件架构的实际应用。本书没有高深的词汇，不仅适合 IT 从业人员阅读，也适合其他行业的人士阅读。尤其对于想从事架构工作的人而言，是一本不可多得的参考材料。

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

聊聊架构 / 王概凯著. —北京: 电子工业出版社, 2017.4  
ISBN 978-7-121-31122-2

I. ①聊… II. ①王… III. ①软件开发—普及读物 IV. ①TP311.52-49

中国版本图书馆 CIP 数据核字(2017)第 057527 号

策划编辑: 张春雨

责任编辑: 徐津平

印 刷: 中国电影出版社印刷厂

装 订: 中国电影出版社印刷厂

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: 720×1000 1/16 印张: 15.5 字数: 256.2 千字

版 次: 2017 年 4 月第 1 版

印 次: 2017 年 5 月第 2 次印刷

定 价: 69.00 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, 88258888。

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

本书咨询联系方式: 010-51260888-819, [faq@phei.com.cn](mailto:faq@phei.com.cn)。



## 序

在软件行业，写书和教学是两种非常辛苦的职业。一方面对技术知识不断，另一方面业务需求也层出不穷，让人疲于应付。寻找的后果就是常常加班，生活质量低下。只有曾经身在其中的人，才能亲身体会其中的酸甜苦辣。

谨以此书献给我的父亲和母亲！感谢两位老人的照顾和爱护，本书才得以顺利完成。

本书也献给我内人！作为本书的第一位读者，她提了很多建议，并做了很多文字校对工作，本书有她的一份功劳。没有她的体贴和关心，以及支持和帮助，本书也无法顺利完成。

本书也献给我可爱的小朋友们！在爸爸写累了的时候，有你们陪爸爸玩耍放松，你们总是能给家庭带来无尽的快乐。愿小朋友们平安、快乐、健康地长大！

曾多次和我妈商量，希望我能够把果酱阅读扩展成一本书。拒绝了很多次，因为书皮实在是薄，禁不住翻来三番五次的折腾。狠下心来，终于写下了。

把文字写下来传播出去，是要承担很大责任的，一旦说得不对，伤害的就是一大群人。不愿写东西的原因大部分在此。但是想想人非圣贤，总有犯错的时候。把自己的错误暴露给大家，也是帮助大家学习。尽管如此，还是郑重声明，本书的内容都是个人的思想和个人的观点，并非学术。因此，请各位读者不要当作论战来接受。反而应该应该质疑书中的各种观点，尽量自行思考，如此才会有所收获。本书的目的仅仅是为了帮助大家思考。

思及自身水平有限，文字功底也差，难免伤人慧命，深感惭愧和惶恐！望各位读者，宽其愚拙，不吝赐教指正！

## 序

在软件行业，架构师和软件工程师是非常辛苦的职业。一方面新技术层出不穷；另一方面业务需求也层出不穷，让人疲于应付。导致的后果就是常常加班，生活质量低下。只有曾经身在其中的人，才能够体会其中的酸甜苦辣。

在软件行业经历过这么多年，也看到了软件行业普遍存在的一些问题，总觉得自己应该为这个行业贡献一点点力量。不期望能够改变这个行业，能够引起一点点思考也是好的，如果能够帮助一些软件从业者提升工作和生活质量，就超出期望值了。

把自己的想法写出来的过程是痛苦的，从来没有写文字的习惯，也没打算过写书，因此愈发艰难。年初时基于以上同样的想法，在 InfoQ 投稿写了《架构漫谈》专栏，和大家分享一下自己对软件架构的思考，以为算是交差了。不料 InfoQ 的郭蕾多次和我约稿，希望我能够把架构漫谈扩展成一本书。拒绝了很多次，但是脸皮实在是薄，禁不住郭蕾三番五次的游说，狠狠心答应了下来。

把文字写下来传播出去，是要承担很大责任的，一旦说得不对，伤害的就是一大片人。不愿写东西的原因大部分在此。但是想想人非圣贤，总有犯错的时候，把自己的错误暴露给大家，也是帮助大家学习。话虽如此，还是郑重声明，本书的内容都是个人的思考和个人的观点，并非学术的结论，请各位读者不要当作结论全盘接受。反而读者应该质疑书中的各种观点，尽量自行思考，如此才会有所收获。本书的目的也仅仅是为了引发大家的思考。

思及自身水平有限，文字功底也差，难免伤人慧命，深感惭愧和惶恐！望各位读者，鉴其愚诚，不吝慈悲指正！

王概凯 Kevin

## 前言

现代的软件从业者，都受过良好的计算机和软件方面的教育。但是现代的计算机和软件方面的教育，基本上都是从科学研究领域脱胎出来的，教育的目的也理所当然的主要是为科学研究领域服务。而随着社会的发展，软件不断地渗透到不同的业务领域，涉及普通生活的方方面面。以科学研究为目的的软件教育，和日益深入人们生活的软件应用，产生了很大的隔阂。以致很多计算机和软件专业毕业的学生，进入企业工作后，总是感叹学校所学习的知识派不上用场，必须得重新学起，才能够达到企业的要求。

而这些重新学习的内容，又往往是以技术为主的。技术的更新换代太快，往往也导致想跟上新技术的我们力不从心。技术和业务的关系是怎样的？业务又是怎么运作的？很少有人去问这些问题。即使有人问了，也很难有人可以提供建议。

软件技术学习到一定的地步，又会发现软件架构是一个门槛。一直以来，在软件行业，对于什么是架构有很多的争论，每个人都有自己的理解。甚至很多架构师一说架构，就开始谈论应用架构、硬件架构、数据架构等。而事实上，架构在软件发明前就早已存在了。众说纷纭，莫衷一是，这也给大家带来了许多困扰。

业务和架构，是压在软件从业人员身上的两座大山。而软件从业人员手上却只有一个武器：技术。可是这个武器还时灵时不灵，就好像金庸小说《天龙八部》中段誉的六脉神剑，并不总是能够解决问题，有时还会带来麻烦。

软件并不是虚无缥缈的东西，它和现实生活是紧密相关的。业务和架构都是处理人的问题。而技术人员最讨厌处理的就是人的问题，心里面厌恶，却又无法逃避。因为这个排斥的心理，工作中始终想避开和人有关的地方。因此在做技术之前，还需要做一些准备工作，用来连接现实生活，联系上人，让大家知道处理人的问题并不可怕。建立了这个相关性，每个人就都可以自行思考了。

不仅人类受限于自身的生命周期，凡事都有其生命周期。理解了生命周期，

就可以看到很多隐藏在背后的规律,以及这些规律之间的联系。因此,本书试图从生命周期入手,描绘出一张整体的画卷,帮助包括技术人员在内的读者定位自己处于什么地方,自己在起什么作用,别人又在什么地方,他们又在起什么作用。明白了自己的位置和别人的位置,自然也就清楚自己有什么,缺什么,要往哪个地方走,从哪些地方入手了。所谓“知己知彼,百战百胜”,知道这些后,与人打交道时也就有了自己的思考方式,能够进行独立思考,对业务也不再厌恶以致逃避,而是为能帮助业务人员分析及解决问题而自豪。

本书虽然不是技术书籍,不谈技术,却是以帮助技术人员为出发点的。本书的内容可以作为连接技术人员和现实世界的桥梁,使技术人员不再悬在空中使不出力。对于非技术人员,本书可以帮助其理解软件开发的特殊性,拉近与技术人员的距离,能够更有针对性地与技术人员合作。

当然,读完本书不会使读者突然学会神功,打通任督二脉。因为每个人的成长,最终还是要靠自己的思考和实践。本书的思考也不能够代替读者自己的思考,在解决某个业务问题时也无法从书中直接找到答案。本书可以提供给读者的是一个思考的出发点,一个思考的方向,一个思考的角度,使得读者不再惧怕或排斥业务,并可以像业务人员一样思考,和架构师一样思考,不再受困于业务和架构,甚至是技术本身。如果本书能够帮助读者跨过这个门槛,并从这里开始展开思考,那么本书的目的就达到了。

---

轻松注册成为博文视点社区用户 ([www.broadview.com.cn](http://www.broadview.com.cn)),您即可享受以下服务。

- 提交勘误:您对书中内容的修改意见可在【提交勘误】处提交,若被采纳,将获赠博文视点社区积分(在您购买电子书时,积分可用来抵扣相应金额)。
- 与作者交流:在页面下方【读者评论】处留下您的疑问或观点,与我们和其他读者一同学习交流。

页面入口: <http://www.broadview.com.cn/31122>

二维码:



# 目录

第 1 章	软件的生命周期 / 1
1.1	软件的开发生命周期 / 1
1.2	软件开发的成长 / 1
1.3	软件开发的迭代 / 1
1.4	软件的运行生命周期 / 1
第 2 章	什么是软件架构 / 1
2.1	要解决什么问题 / 1
2.2	分别是谁的问题 / 1
第 3 章	生命周期的识别 / 3
3.1	生命周期的识别 / 3
3.2	核心与非核心生命周期 / 3
3.3	生命周期与分工 / 5
第 4 章	时间 / 9
第 5 章	为什么会产生架构 / 11
5.1	分工 / 11
5.2	分工和生命周期 / 12
第 6 章	什么是架构 / 13
6.1	架构产生的条件 / 13
6.2	什么是架构 / 15
6.3	架构的生命周期 / 17
第 7 章	架构和树 / 19
7.1	树与增长 / 19
7.2	架构和树 / 20
第 8 章	概念 / 22
8.1	何为名相 / 22
8.2	究竟什么才是相 / 23
8.3	概念是沟通的基础 / 23
8.4	把握概念的力量 / 24



第7章 什么是抽象 / 26

7.1 个性与共性 / 26

7.2 个性是基础 / 28

第8章 识别问题 / 29

8.1 面对问题有哪些困难 / 29

8.2 如何识别问题 / 30

8.3 寻找问题主体 / 32

第9章 切分的原则 / 34

9.1 切分就是利益的调整 / 34

9.2 为什么需要切分 / 35

9.3 切分的原则 / 35

9.4 树和分层 / 36

9.5 切分与建模 / 37

9.6 切分的输出和组织架构 / 38

第10章 架构与流程 / 40

10.1 什么是流程 / 40

10.2 流程和架构拆分的关系 / 41

第11章 什么是架构师 / 42

11.1 架构师做什么 / 42

11.2 架构师也是人 / 43

11.3 人人都是架构师 / 43

11.4 架构师和权利 / 44

第二部分 软件架构 / 47

第12章 什么是软件 / 48

12.1 以模拟人为目标的冯·诺依曼结构和图灵机 / 48

12.2 成本为王 / 49

12.3 天空才是极限 / 49

12.4 软件的作用 / 50

第 13 章	软件的生命周期	/ 53
13.1	软件的开发生命周期	/ 54
13.2	软件开发的增长	/ 55
13.3	软件开发的迭代	/ 56
13.4	软件的运行生命周期	/ 57
第 14 章	什么是软件架构	/ 58
14.1	要解决什么问题	/ 58
14.2	分别是谁的问题呢	/ 59
14.3	分别有什么问题	/ 59
14.4	分析问题	/ 60
14.5	会生成哪些架构	/ 62
14.6	什么是软件架构	/ 63
第 15 章	什么是软件架构师	/ 65
15.1	软件架构师的区别	/ 65
15.2	软件架构师的困境	/ 66
15.3	生命周期的思考	/ 67
15.4	软件架构师的权力	/ 67
15.5	软件架构师和技术人员对技术的态度区别	/ 68
15.6	架构师是技术的使用者	/ 69
15.7	如何保障架构落地	/ 70
第 16 章	业务、架构和技术三者的关系	/ 73
16.1	什么是技术	/ 74
16.2	业务和架构及技术之间的关系	/ 75
16.3	技术人员和业务人员的关系	/ 76
16.4	重新发明轮子	/ 77
16.5	开源技术	/ 78
第 17 章	软件研发	/ 81
17.1	软件工程师的兴起和使命	/ 81
17.2	分工的困境	/ 83
17.3	软件的迭代	/ 85
17.4	软件开发的分工	/ 87

17.5	软件开发模式和架构	/ 88
17.6	软件工程师的支持者	/ 90
第 18 章	软件的架构拆分	/ 92
18.1	软件拆分的原动力	/ 92
18.2	软件开发团队的拆分	/ 95
18.3	软件的拆分	/ 96
18.4	软件开发的基础技术	/ 98
18.5	软件拆分的第二动力	/ 100
18.6	架构一步到位	/ 100
第 19 章	如何写好代码	/ 102
19.1	什么叫业务逻辑	/ 108
19.2	业务逻辑分散的危害	/ 108
19.3	业务逻辑内聚的好处	/ 110
19.4	代码架构实例	/ 111
19.5	代码误解	/ 113
19.6	软件的拆分	/ 114
第 20 章	单元测试	/ 116
20.1	什么是单元测试	/ 116
20.2	单元测试的困境	/ 116
20.3	单元测试测什么	/ 117
20.4	如何改造代码	/ 118
20.5	为什么要做单元测试	/ 121
20.6	如何做单元测试	/ 123
第 21 章	软件架构和面向对象	/ 125
21.1	什么是面向过程	/ 125
21.2	什么是面向对象	/ 126
21.3	生命周期和面向对象及面向过程	/ 127
21.4	架构和面向对象及面向过程	/ 127
21.5	面向对象的误区	/ 128
21.6	对象和生命	/ 130

## 第 22 章 软件架构与设计模式 / 131

- 22.1 模式以及模式的意义 / 131
- 22.2 什么是设计模式 / 132
- 22.3 软件设计模式 / 133
- 22.4 设计模式和架构 / 134
- 22.5 设计模式的误区 / 136

## 第 23 章 软件架构和软件框架 / 139

- 23.1 访问类框架 / 139
- 23.2 业务类框架 / 141
- 23.3 什么是框架 / 141
- 23.4 框架的特点 / 142

## 第 24 章 软件运维 / 143

- 24.1 软件运行生命周期 / 143
- 24.2 什么是软件运维 / 144
- 24.3 运维的业务模型 / 146
- 24.4 控制变化 / 147
- 24.5 监控变更 / 150
- 24.6 预警变更 / 152
- 24.7 主导变更 / 154
- 24.8 提升变更质量 / 157
- 24.9 运维的架构拆分 / 158

## 第 25 章 软件访问生命周期 / 161

- 25.1 软件访问的业务模型 / 161
- 25.2 软件访问路径的架构拆分 / 163
- 25.3 大规模软件访问的架构拆分 / 165
- 25.4 集群 / 166
- 25.5 数据中心 / 168

## 第 26 章 软件架构和大数据 / 172

- 26.1 什么是大数据 / 172
- 26.2 如何做好大数据 / 173
- 26.3 软件大数据 / 174

第 27 章 软件架构和建筑架构 / 177

27.1 软件架构和建筑架构的目标之异同 / 177

27.2 软件和建筑的架构扩展之异同 / 180

第三部分 软件架构的应用 / 183

第 28 章 交易 / 184

28.1 什么是交易 / 184

28.2 货币的出现 / 185

28.3 企业的实质 / 186

28.4 软件对交易的影响 / 187

28.5 软件的交易 / 187

28.6 企业的核心 / 188

第 29 章 产品 / 190

29.1 什么是产品 / 190

29.2 什么是商品 / 194

29.3 识别产品 / 195

29.4 产品系统 / 196

29.5 产品列表 / 197

29.6 产品详情 / 197

29.7 商品的规则 / 198

第 30 章 用户 / 199

30.1 什么是用户 / 199

30.2 为什么需要用户 / 200

30.3 客户的出现 / 201

30.4 用户的生命周期 / 201

30.5 用户的识别 / 202

第 31 章 订单 / 203

31.1 什么是订单 / 203

31.2 订单的生命周期架构拆分 / 204

31.3 订单支付 / 206

31.4 订单生命周期 / 207



第 32 章 交易系统 / 208

- 32.1 企业的架构拆分 / 208
- 32.2 软件系统的建模 / 212
- 32.3 访问业务模型 / 216
- 32.4 交易软件系统的架构拆分 / 219
- 32.5 服务的产生和粒度 / 220
- 32.6 用户系统的拆分 / 221

第 33 章 事务 / 226

- 33.1 什么是事务 / 227
- 33.2 软件中的事务 / 228
- 33.3 数据库事务的滥用 / 229
- 33.4 数据库的正确使用方式 / 229
- 33.5 服务调用 / 230

树立对架构的正确理解，建立对架构的基本认识

# 第一部分 认识架构

### 树立对架构的正确理解，建立对架构的基本认识

## 第1章 生命周期

这个世界是如此多彩，每一个个体都是如此美妙，打动人心。这个世界每时每刻都在不断地变化，正因为变，我们每时每刻对这个世界都有新的感受，才觉得她是这么的美丽。

每个个体的变化都是不同的，都有自己的特点。可是所有事物的变化都有一个共同的特点：从出生开始，到消亡结束。用佛家的话说，也就是“生住异灭”。所有的事物一旦出现了，就必然会消亡。而正因为有不断的消亡，才会有不断的出生，所谓“生生不已”。这一切都是在人们掌控之外的，正如人们无法决定自身的生灭。因为人们无法去掌控她，她才会如此美丽，打动人心。

一个事物一旦出生，就必然会长大，变异，一旦长大，就面临着衰老，接下来就是消亡了。这个过程就称为一个事物的生命周期。生命周期（Life Cycle）实际上是近代出现的概念，从西方传入。自古以来，我们称之为生灭。为了叙述上的方便，后续所指的生命周期，实际上指的就是生灭。

人类自古以来就在研究自身的生灭问题，也就是生死问题。这个问题隐藏在每个人的言行举止中，人们经常为之深深地恐惧。这种恐惧的心理驱使着人们做出种种的努力，来延缓甚至避免这个结果，虽然人们明知无法避免。

人类为了保持活着的状态，因而需要满足身体生存的必要条件，那就是要获取能量，并能在复杂的自然条件下保存身体，并生存下来。这个原始的动力，从没有离开过我们。我们必须时刻清楚这一点。不了解这一点，人们很多时候就无法理解自己的行为，因为这个问题深深地隐藏在人们的潜意识当中。看看我们每天的活动，起床，穿衣，吃饭，工作，睡觉，无一不是为了维持身体生存而做的动作。

等一等……人们每一天的活动，是不是也是一个生命周期呢？确实是的，生命周期无处不在。从嚼一口饭，到捡起一支笔，到眨一下眼睛，都是一个生灭，

都是一个生命周期。我们一天里所有的子生命周期组成了这一天，一辈子三万六千天的大生命周期，由三万六千个每天的子生命周期组成。而每一个生命周期的结束，往往又会导致其他事物的生命周期开始。比如人们吃一口饭，饭消失了，产生了能量，人们通过活动，消耗掉能量的同时又产生了新的事物。

所以生命周期并不是孤立的，生命周期里包含生命周期，一个生命周期消亡会产生另一个生命周期。我们生存在一个生灭变化的世界，需要认真认识生命中的这些生命周期，以及生命周期之间的关系，并保持敏锐的观察。这会让我们更加清晰地理解这个世界，并能够更和谐地和这个世界相处。

## 1.1 生命周期的识别

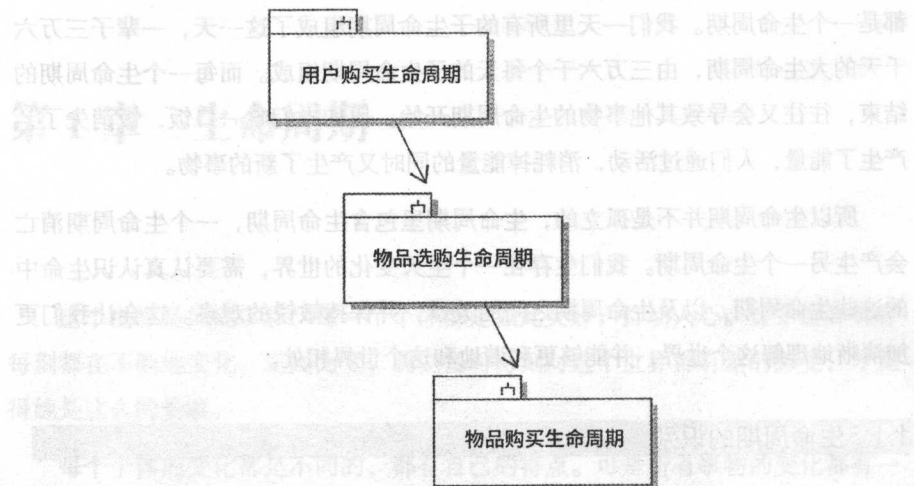
生命周期里面包含有各种活动，这些活动是推进这个生命周期的必要因素，并且这些活动之间有时间上的推进关系。比如，用户购物这个场景，从用户进入到商店，进行浏览、询问、购买等活动，到离开商店，都是按时间顺序一步一步发生的。每个发生的活动，本身也是一个生命周期。

生命周期总是有一个主体，因此更多的是明确指出某某的生命周期。所以识别出生命周期的主体是至关重要的事情。如上文提到的购物场景，从用户进入到商店，进行浏览、询问、购买等活动，到离开商店，这个生命周期的主体是谁？这个生命周期的主体并非用户本身，而是用户的一次购买活动。如果是用户的生命周期，则变成了用户的生成和消亡了。

## 1.2 核心与非核心生命周期

一个生命周期里面的活动可以进行拆分，拆分的原则就是形成若干个新的生命周期，每个新的生命周期都有自己的主体。比如，再承上例，用户的购买活动又可以拆分为<物品的选购>和<物品的购买>两个小的生命周期，分为两个阶段。拆分过程如下图所示，箭头表示执行的顺序。

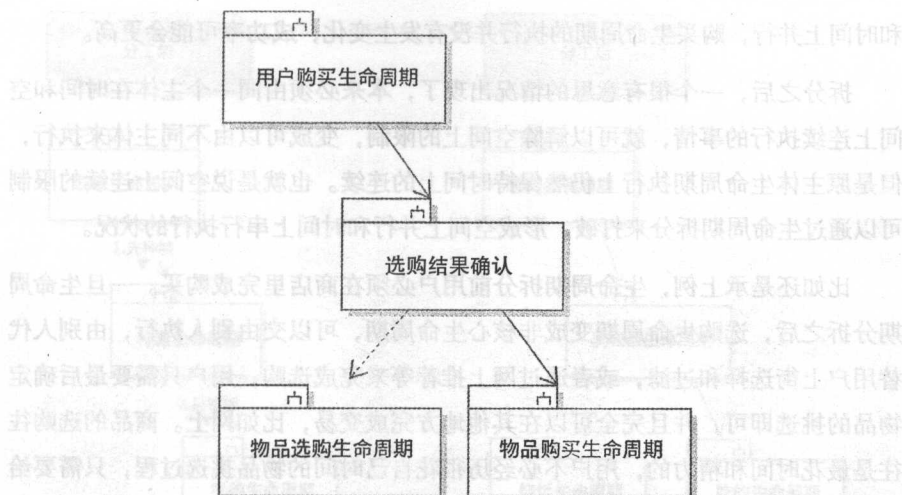
如下例，物品选购结果的确定是核心生命周期需要的，也就是说这个行为是必须在购买前，但是选购结果确定之前的种种行为，如浏览、进店挑选等，也就是选购生命周期的准备，则完全可以交由别人来执行，如导购员、网上购物、智能推荐等都在这个范围发力，这样选购的准备工作就可以和购买行为无缝的执行在空间



一旦拆分为小的生命周期我们就会发现，其中一个生命周期的主体也发生了改变：<物品的选购>这个生命周期的主体为用户的物品意向，以用户进店为开始，以意向确定或放弃为终；<物品的购买>这个生命周期的主体为用户的购买活动，主体并没有发生改变，但是主体的生命周期活动内容减少了，变成以物品确定为开始，以交易成功或失败为终。这两个小的生命周期以物品确定这个行为分别为终始，组成了一个大的生命周期：用户的购买活动。

我们也把拆分之后主体不变的子生命周期，称为核心生命周期。主体改变的子生命周期，称之为非核心生命周期。寻找核心生命周期的过程，实际上也就是一个发现内聚的过程。对于非核心生命周期，本身内部也需要寻找出它自己的核心生命周期，这同样也是发现内聚的过程。如下图所示，虚线表示非核心生命周期，实线表示核心生命周期，箭头表示执行的顺序，树的遍历的次序从左至右。





拆分出来的每个生命周期各自都有自己的边界，不会影响到其他的生命周期，因为各自的变化都在自己的生命周期内确定。每个主体的生命周期活动的变化都累积在该主体本身，这就是所谓的内聚。也就是说，要做到内聚，必然要先确定生命周期的主体和生命周期本身。确定了这两样，内聚就是自然而然的结果。所以，当讨论内聚的时候，实际上就是讨论生命周期主体以及该生命周期内所有的活动，我们一定要意识到这一点。

### 1.3 生命周期与分工

在把一个大的生命周期拆分为多个小的生命周期后，核心生命周期活动的执行都严格地在时间上连续。而非核心生命周期的管理，则围绕着核心生命周期形成了一个树状结构。随着大的生命周期的拆分，树在逐渐地长大。

拆分之前，生命周期内部的活动都是由该生命周期的主体来执行的。在拆分之后，核心生命周期还是由原生命周期的主体来执行，但是非核心生命周期的主体发生了变化。也就是说生命周期的运营和执行产生了分离。

如上例，物品选购结果的确定是核心生命周期需要的，也就是说执行还是必须在购买之前。但是选购结果确定之前的种种行为，如逛街、进店挑选等，也就是选购生命周期的运营，则完全可以交由别人来执行。现今的网上购物、智能推荐等都是在这个地方发力。这样选购的运营就可以和购买生命周期的执行在空间

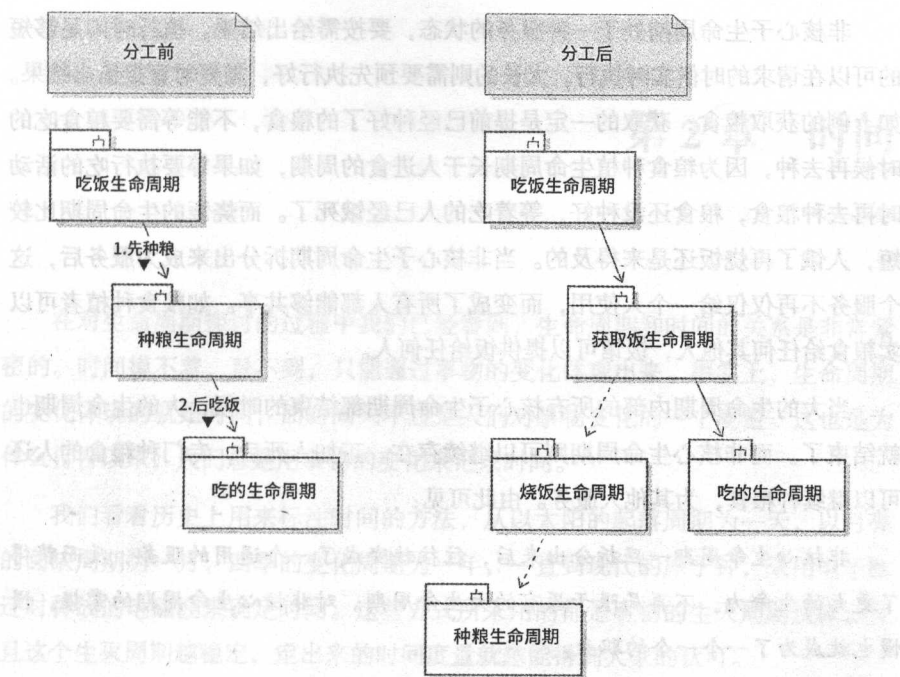
和时间上并行，购买生命周期的执行并没有发生变化，成功率可能会更高。

拆分之后，一个很有意思的情况出现了，本来必须由同一个主体在时间和空间上连续执行的事情，就可以解除空间上的限制，变成可以由不同主体来执行，但是原主体生命周期执行上仍然保持时间上的连续。也就是说空间上连续的限制可以通过生命周期拆分来打破，形成空间上并行和时间上串行执行的状况。

比如还是承上例，生命周期拆分前用户必须在商店里完成购买。一旦生命周期分拆之后，选购生命周期变成非核心生命周期，可以交由别人执行，由别人代替用户上街选择和过滤，或者通过网上推荐等来完成选购，用户只需要最后确定物品的挑选即可，并且完全可以在其他地方完成交易，比如网上。商品的选购往往是最花时间和精力，用户不必经历很花自己时间的物品挑选过程，只需要给出想要的物品型号即可。用户的目的是买到自己需要的东西，而不是选购本身。核心生命周期变得更加简短了。

拆分同时会带动产生很多新的有意思的生命周期，形成新的技术，比如商品的规范化、货物的配送等。有的时候则反过来，新的技术出现导致拆分变成可能。比如照相机的出现，使得人们眼睛看到物体的生命周期发生了根本性的分拆；车的出现，使得人们从一地到另一地的生命周期发生了拆分，不再全靠步行中每一步的生命周期累加；软件的出现，互联网的出现，使得人们生活方方面面的各种生命周期都发生了分拆……

再比如，吃饭是人的核心生命周期，必须要自己执行，别人无法代替。为了完成吃饭这个生命周期活动，在没有产生社会分工之前，每个人自行种粮，用自己种的粮食做饭吃，这个过程按照时间顺序连续发生。我们把这个过程拆分之后，可以发现里面只有“吃”是人类核心的生命周期活动，别人无法代替，必须自己执行，而其他活动都是为吃服务的。随着社会分工的发生，拆分出来种粮的生命周期管理、烧饭的生命周期管理等，可以交给其他人来执行。人们可以通过购买粮食和请人烧饭来直接获取执行的结果，以推进自己吃饭的核心生命周期活动。这就产生了管理和执行的分离。如下图所示，虚线代表非核心生命周期，实线代表核心生命周期，箭头代表执行顺序，树的遍历为从左至右。



非核心生命周期的管理和执行、核心生命周期的管理和执行可以在时间和空间上并行，但执行仍然还是要依照拆分之前的大生命周期的顺序，没有发生改变。比如粮食种植这个非核心生命周期分工给别人了，但是烧饭生命周期执行前先要有粮食，因为“巧妇难为无米之炊”，还是得等别人的粮食种出来之后才能够购买获得，而饭则要等别人烧好了才能吃。

从生命周期内部来看，生命周期的每个阶段是随着时间的推进而发生变化的，因此呈现出来的是一个按时间顺序发展的状况。比如人的吃饭、睡觉等活动，虽然都是独立的活动，但是每天早、中、晚三餐饭都是按顺序发生的。每天晚上也都需要睡觉来补充体力。不可能边吃饭边睡觉，或者三餐一起吃。这些活动需要按照人的生命周期规律逐步地按顺序发生。

反过来，当启动一个核心子生命周期的时候，必须把树上该节点的所有父生命周期启动才可以，也就是所谓的业务流程，因为大生命周期本身内部活动还是按照时间顺序连续发生的。比如，用户下订单之前，必须先要给出选定的商品列表；吃饭之前，先要有饭。

非核心子生命周期处于一种服务的状态，要按需给出结果。执行时间足够短的可以在请求的时候实时执行，太长的则需要预先执行好，需要时直接给出结果。如上例的获取粮食，获取的一定是提前已经种好了的粮食，不能等需要粮食吃的时候再去种，因为粮食种植生命周期长于人进食的周期，如果等要执行吃的活动时再去种粮食，粮食还没种好，等着吃的人已经饿死了。而烧饭的生命周期比较短，人饿了再烧饭还是来得及的。当非核心子生命周期拆分出来成为服务后，这个服务不再仅仅给一个人使用，而变成了所有人都能够共享。如粮食种植者可以卖粮食给任何其他人，饭馆可以提供饭给任何人。

当大的生命周期内部的所有核心子生命周期都结束的时候，大的生命周期也就结束了。而非核心生命周期则可以继续存在。正如人死后，专门种粮食的人还可以继续种粮食，为其他人服务。由此可见：

非核心生命周期一旦拆分出来后，往往就变成了一个通用的服务，反而获得了更大的生命力，不再局限于原有的大生命周期。对非核心生命周期的掌握，慢慢也就成为了一个一个的职业。

而原有的大生命周期则变得更加精简，可以更加专注于自己的核心生命周期活动，以节省更多的时间。

## 第2章 时间

在对生命周期探讨的过程中我们已经看到，生命周期和时间的关系是非常紧密的。时间摸不着，看不到，只能通过事物的变化体现出来。事实上，生命周期的变化体现的就是时间，而时间只不过是人们对事物变化的一个度量。这也是为什么自古以来，人们总是用事物的变化来记录时间。

我们看看历史上用来标注时间的方法，从以太阳的起落周期为一天，以月亮的圆缺周期为一月，四季的变化周期为一年，一直到现代的原子钟，采用电子跃迁时释放的电磁波来确定时间。这些方式所采用的都是事物的生灭周期规律。并且这个生灭周期越稳定，定出来的时间度量就越能得到大家的认可。

所以人们对生死的恐惧，就转而形成了对时间的恐惧。在人的短短一生中，维持生命必需的活动占据了绝大部分的时间，那我们怎样在剩下的时间里做出一些努力呢？无论承认与否，这是每个人内在的焦虑。为减轻这个焦虑而做出的思考和努力，就变成了人类进步的动力。

很多科学研究都在探寻时间，时间本身究竟是怎么回事？我们是否可以回到过去，甚至去到遥远的未来？为了延续人类，很多科学家积极地探索外太空，因为地球终究也会消失的，人类要有其他的地方可以居住。

一个人静静地坐在池塘一角，看着平静的池塘。随着鱼儿的活动，水面时而浮起涟漪，又转瞬即逝。天空，云影徘徊，变动不居。微风轻吹，直吹到心灵的深处。每个事物，都遵循着自己的规律，遵循自己的生命周期，该来的来，该走的走。此时眼泪不禁从双眼滑落：世界正是因为不断地来来去去，生生灭灭，才是这么多彩，才是这么吸引人。

每个人也都在努力与时间赛跑，尽可能地在事物消逝前多占有一会儿，但是不管人们愿不愿意，事物最终都会离我们而去。过去的已过去，未来的还没来，只有当下的一刻是最真实的，大自然是我们学习的榜样。



人类的生命很短，百年也只有短短的三万六千天。自古以来人们寻求长生不老，希望能够延长自己的生命，得到的结果总是失望。其实反过来思考，如果人类活得足够长，会有这么大的动力去创造出这么多的新技术吗？反而寿命越短，越是焦虑，越希望能够得到更多的东西，物质技术也越发达，分工也越严密细致。得到越多，越希望能够留住，越留不住就越焦虑，形成了一个很糟糕的循环。物质的发达对人来说似乎并不一定是好事。

大部分人都不愿意接受一切都终将消逝的事实，总想活得更久，占有更多，享受更多。在人们短短的一生中，如何延长自身的生命呢？一个办法就是尽可能做出更多的成就，能够让更多的人生活得更好。在同样的时间内创造出更多的产出，相当于把自己的生命延长了。另一个办法就是创造下一代，这在某种程度上也可以视为延长了时间，也是人类延续的动力。这个动力，也推动了人类生命周期的拆分与人类架构的发展。

尽可能做出更多的成就的办法，就是尽量做自己最擅长的事情，以期获得最大的产出，人类的社会化正由此而来。

## 第3章 为什么会产生架构

延长每个人自己生命的时间，就要尽可能做自己擅长的事情，以及产生下一代。

想象一下，人类在最早期，每个人都完全独立生活，衣、食、住、行等等全部都自己搞定，整个人类都是独立的个体，不相往来。每个人都局限在自己所在的地域，独立完成自己的生活必需品的生命周期以支持自身的核心生命周期。每个人为维持生命所做的活动很多，而由于人类结构的原因，一个人同时只能够做一件事情，这会导致很多时间上可以并行的工作变成了串行。

正如我们前面生命周期中所讨论的，除了那些别人无法替代必须要由自己来完成的核心生命周期活动（如吃饭、睡觉等）外，把其他的生命周期活动并行起来，在同样多的时间内做更多的事情，也相当于延长了自己的生命。如何做到这一点呢？

### 3.1 分工

产生下一代的要求就会导致男女群居，这个时候自然就出现了性别分工，男性和女性所做的事情就会有一定的区别。比如男性体力好、强壮，就会更多地野外获取食物。女性细致、耐心，更多地做一些衣物缝制、食物制作、照顾下一代等工作。还会出现地域的分工，比如有些地方适合种粮食，有些地方适合种棉花等。因此，就从原来每个人都要自行获取所有的生活必需品，变成了多个人可以在时间和空间上并行地进行。参与分工的人就可以专心地做好自己擅长的事情，或者所处的不同地域所擅长的事情。

可是人们每天生活的基本需求没有发生变化，还是衣食住行等生活必需品。一旦多人分工配合作为生存的整体，力量就显得强大多了，所以也自然地形成了族群：有些人种田厉害，有些人制作工具厉害，有些地方适合产出粮食，有些地方适合产出棉花等，就自然形成了人的分群，地域的分群。当分工发生后，实际上每个人的生产力都得到了提高，因为做的都是每个人擅长的事情，相应的每个

人的生命都得到了延长。整个人群的生产力和抵抗环境变化冲击的能力都得到了增强,某种程度上人的寿命也同时会得到增长。

既然分工发生了,原来由一个人执行生存所必需的所有的东西,就变成了很多不同分工的角色合作完成这些事情,人的生命周期活动发生了拆分。拆分后每个人只需要执行必须自己做的事情和自己擅长的事情。这些人必须要通过某些机制整合在一起,让每个人都得以完成生存所必需的东西,这实际上也导致了交易的发生。交易这部分暂时就不在这里展开了,后续再讨论。

当每个人都必须自己完成所有生活必须的活动的时候,是没有架构的。一旦产生了分工,就把原来每个人所必须做的人类非核心生命周期工作,切分成由不同角色的人分别来完成,最后再通过交易,使得每个个体都拥有生活必需品,而不需要每个个体都做所有的事情,只需要每个个体做好自己擅长的事情,并具备一定的交易能力即可。这样每个人为自己核心生命周期所花的时间就少多了,有更多的时间用来提升核心生命周期的质量,从而使生命得到了某种程度的延长。这实际上就形成了人类社会的架构。

从以上的例子可以看出来,人有能力按照自己的意愿控制其他事物的生命周期,并重新组合,以在某种程度上延长自己的生命周期,这一点是超越其他生命的,也是人类架构产生的必要条件。

### 3.2 分工和生命周期

再从生命周期上来看,原来需要一个人完成所有的事情,生命周期被细分为很多小的非核心生命周期。这些小的非核心生命周期,就可以由另外一个人来负责,在空间上和时间上都得以并行,原来的那个人只需要获得另一个人的结果就可以了。比如一个人必须要先种田,完成粮食的产生,并消费粮食,结束粮食的生命周期才能完成自己的能量获取,用以维持生命。此时粮食的生命周期和人的生命周期都是由一个人来管理,本来可以并行的事情,由于人本身的限制,也可以说由于空间的限制,变成了串行。当生命周期切分后,粮食生产的生命周期就分解为粮食的产生生命周期和粮食的消费生命周期。人就不必只依赖于自己生产的粮食,大大拓宽了获取的渠道,而人本身的生命周期并没有受到影响,可是却大大地节省了时间,延长了自己的生命。

## 第4章 什么是架构

由上文可见，推动人类架构产生的动力在本质上还是人类对时间的恐惧。而人类为克服这个恐惧，尽可能地去延长自己的时间，去提升自己的生命质量，毫无疑问就变成了人类内在的动力。

### 4.1 架构产生的条件

细分一下架构产生的动力，基本有以下几点：

(1) 必须由人执行的工作。

不需要人介入，就意味着不需要改造；不需要节省时间，也就不需要另外再产生架构了。大自然会自行调整自然界本身的架构。

(2) 每个人的时间有限。

每个人都有自己的强项，个人的产出受限于最短板，并且由于人的结构限制，同时只能专注于做好一件事情。比如，人类虽然有两只眼睛，但是只能同时专注于一件事物；两只手无法同时做不同的事情，虽然有少部分人可以左手画圆右手画框，但这不是普遍现象。人们为了减少自身生命周期活动的时间投入，必然会把自己不擅长的非核心生命周期工作分解出去，给擅长于这些工作的角色来完成，从而缩短自己所花的时间。

(3) 对目标系统有更高的要求。

如果满足于现状，没有时间压力，也就不需要进行架构了。

(4) 目标系统的复杂性使得单个人完成这个系统时会受限于时间。

如果自己就可以完成系统的提高，不需要其他人参与，也就不需要架构的设计，这样的人只能算是工匠。因为并不一定会产生分工的思考和尝试，并且一般这个工作对时间的要求也不迫切。当足够熟练之后，这样的人也会有一定的架构

分工思考，但考虑更多的是如何提高质量，提高个人时间的效率。

有人可能会说，一个人建一栋房子，对目标系统进行分解，自己采购材料，自己搭建，难道也不算架构嘛？架构的思考来源于对生命周期的识别，以及对生命周期的拆分。如果没有生命周期拆分的思考，就不能算架构。世界上第一个建房子的人一定是有架构思考的，因为要建房子这件事本身就是一个架构思考，这是对地球上空间的主动切分，对人自身生命周期活动所占空间的切分。但如果只是出于对别人的模仿，则并不会有架构思考。如果对时间不敏感的话，比如已经有住的地方，有足够的时间慢慢造，也并不一定会导致架构思考的产生。如果有足够的自觉，而且足够熟练的话，确实会产生架构的思考，因为这样对于提高生产力是有帮助的，可以缩短建造的时间，并会提高房子的质量。事实上建筑的架构就是在长期进行这些活动后，积累下来的实践。思考的结果，往往会导致新技术的产生，或者新技术的应用。

当上述四个条件同时成立，一定会产生人类的架构。从这个层面上来说，人类的架构是人类发展过程中，由懵懵懂懂、被动地去认识这个世界，变成主动地去认识，并以更高的效率去改造这个世界的方法。以下我们再以建筑为例加强一下理解。

最开始人类是住在山洞里或住在树上的，主要是为了躲避其他猛兽的攻击，以及减少自然环境的变化对人类生存的挑战。山洞和树是大自然所形成的自然结构。为了解决人类生存的问题，人类开始模仿大自然，在平地上用树木和树叶来建立隔离空间的设施，这就是建筑的开始。但是完全隔离也不好，毕竟人类的生命周期完成，离不开自然界。完全隔离会导致进出不方便，也不便于观察周围，慢慢就产生了门窗等设施，形成了新的技术，新的生命周期拆分。

建筑的本质就是从自然环境中划出一块独占的空间，但是仍然能够通过门窗等和自然环境保持沟通。这个时候建筑架构就已经开始了，建筑的技术也跟着发展。

建筑架构按照人类生命周期的需求，对地球上的空间进行切分，并通过门窗、地基等技术，保持和地球以及空间的有机的沟通。

人类的必要生命周期活动，如吃、睡等，是和时间相关的，自然也就会根据人的生命周期规律按照时间来划分内部空间。



比如当人类开始学会用火之后,茅棚里自然而然慢慢就会被进一步切分,一部分用来烧饭,一部分用来生活。当人的排泄慢慢移到室内后,洗手间也就慢慢地出现了。这就是建筑内部的空间切分。这个时候人们对建筑的需求也就逐渐越来越多,空间的切分也会变成很多种,组合的方式也会有很多种,比如个人居住的房子,群居所产生的宗教性质的房子,集体活动的房子等等。这个时候人们开始有意识地去设计房子,建筑架构师就慢慢地出现了。一切都是为了满足人类越来越高的需求,提升质量,减少时间浪费,更有效率地切分空间,并且让空间之间更加有机地进行沟通,让人类可以更好地享受生命,享受时间,完成核心生命周期活动。这就是建筑的架构以及建筑的架构的演变。

## 4.2 什么是架构

总结一下,什么是人类的架构呢?主要包含以下几个要点:

(1) 根据要解决的人类的问题,对目标系统的边界进行界定。问题主体的确定一般就能够确定边界,问题的确定可以明确核心生命周期。

(2) 围绕目标系统核心生命周期进行切分。切分的原则是要让非核心生命周期独立出来,便于不同的角色并行地开展工作。首先要在空间上进行拆分,使得空间上的管理可以并行,而时间执行上串行。因为只有空间上并行才能减少每个人的执行时间。达到空间上的并行后,进一步可以通过预先执行好结果,达到时间上并行的效果,缩短整个生命周期的执行时间。

(3) 对这些切分出来的部分,确立各自的生命周期及其主体,以及负责的角色。不同角色对所负责的生命周期负全责,并具备享受该生命周期推动带来的利益,达到权责对等。每个切分出来的生命周期都应该是完整的,活动的结果都累积在该生命周期的主体上,也就是内聚。

(4) 在这些拆分出来的非核心生命周期和核心生命周期之间设立沟通机制,使得这些非核心生命周期能够围绕核心生命周期,通过树状架构组装起来成为一个整体,共同为核心生命周期做出贡献。为什么是树状架构呢?因为核心生命周期本身是按照时间线性推进的,拆分后非核心生命周期会围绕核心生命周期形成树状的架构。

人类正是因为有了分工,才形成了人类社会。人类社会有自己的核心生命周



期，不同分工的人围绕在这个核心生命周期下形成了一个树状结构。人类社会因此形成了一个新的生命，并且比单个人的生命长得多。因为人类社会的生命并不因为某个人的死亡而消亡。人类生命同样也因此依赖于人类社会而得到了增长，达到了双赢。

是不是只有人类才有架构呢？大家应该也注意到了，前文的架构都加上了“人类”做了限定。自然界也是存在架构的。各种生命都有自己的生命周期，生命周期之间形成一个个的食物链，循环往复。这也是为什么叫作“周期”。这个世界就是通过一个事物的生命周期的推进，不断地向前发展，而不是往后看的。

人不过是自然界中的一种生命而已，在古代被称为“保虫之长”。<sup>1</sup>各种生物组成了整个地球，地球则形成了一个生命体，有自己的生命周期。不同的星球又形成一个个的星系，进而组成整个宇宙。这就是大自然的架构。老子云：“人法地，地法天，天法道，道法自然”，这里的自然指的是自然如是，本来如此的意思。（现代语言中的“自然”，是从日本舶来的，是对英文“nature”的翻译，不是古代的本意。）任何事物都按照自己本身的生命周期规律向前发展。做架构也应该效法地、天、道、自然。大自然的架构是人类学习的榜样，人类设计的架构也必须遵循大自然的规律才能稳定、长久，因为人类是生活在大自然的架构之下的。违背了这一点，人类就要付出巨大的代价。

大自然的架构又是如何推进的呢？靠的就是生存在其中的每种生命，自生自灭，遵从每种生命自身的规律。而每种生命的活动，在得到自身生命周期活动利益的同时也要承担自身生命周期活动的责任，支撑生态链，权责对等。因此大自然才生机勃勃。

大自然的架构经过亿万年的推进，有其自身的拆分，而人类自身的社会架构拆分和分工还在继续。理解架构拆分背后的规律，对人类有意识地去架构是非常有帮助的，为我们指引了方向。为了区别大自然的架构，也为了行文方便，后续我们讨论的架构都是以人类架构为前提，不再特指。

这些思考同样可以在其他的行业展开，比如企业架构、国家架构、组织架构、

1 《内经》将虫分为五类，即毛虫、羽虫、倮（luó）虫、介虫、鳞虫，其中倮虫属土，人为倮虫之长。

音乐架构、色彩架构和软件架构等等。套用三国演义的一句话，合久必分，分久必合。人类架构实际上就是指人们根据自己对世界的认识，为解决某个问题，主动地、有目的地去识别问题，并根据核心生命周期进行分解、合并，以解决这个问题的实践活动。人类架构的产出物，自然就是对核心问题的提炼和分析，对核心生命周期的识别，以及解决问题的方案。它包括拆分的原则以及理由，沟通合并的原则和理由，拆分出来的各个部分的生命周期确立，以及所对应的推动角色和角色所需要的核心能力，还有对这些拆分的落地等。

### 4.3 架构的生命周期

从架构本身来看，架构也是人类活动的一种分工，人类架构的体现就是组织架构。因此，架构本身也有其自身的生命周期，有其自身的规律。人类的架构总是在人类的业务遇到瓶颈的过程中产生，在瓶颈的解决中应用，在业务消亡时，它也跟着消亡。架构是对业务生命周期的拆分，自然业务的生命周期就是架构的生命周期。每种业务，它的架构拆分是确定的，区别只在于规模的不同，树的高度不一样。这就是为什么同行业的业务，往往都是一样的架构。有了这一特征，在某个行业开始一个业务，该行业的架构就可以直接拿过来参照实现，这就是架构的模式。

架构的生命周期可以被拆分为两个子生命周期：架构设计生命周期、架构实施生命周期。其中，架构设计生命周期是为架构实施生命周期服务的，因此架构实施生命周期是架构的核心生命周期，架构设计生命周期是非核心生命周期。非核心生命周期的并行管理，导致诞生了很多专门做架构设计的公司，专门给人做设计外包，或者做架构咨询（Consulting）。但架构的核心是实施，很多雇主为了防止自己购买的设计方案无法落实，往往会要求设计提供者同时提供架构的落地服务。吃过苦头才知道，架构只有方案是没有意义的，最重要的是把架构落地执行。

架构的设计生命周期，主要工作就是研究业务本身的生命周期。然后再根据业务面对的问题，发现瓶颈，进行架构的拆分。拆分的原则则是把非核心生命周期拆分出来，由不同的角色来负责，让人们可以并行工作。每个角色达到权责的对等，并形成不同角色各自的激励机制。架构的目的并非产生一个业务之外新的东西出来，只是为了让业务长得更加高大强壮，服务于更多的人。

架构实施生命周期，则是为了把架构的拆分落实到组织架构上，让每一个人能够按照架构的职责拆分，并行工作，执行各自的生命周期。这些角色按照树状架构协调互相之间的沟通，使得每个拆分出来的非核心生命周期的增长，都贡献给核心生命周期，同时自己也能够受益。也只有树状架构能够做到这一点。

在架构落地之后，架构就具备了生命。人们在这个架构下有条不紊地协作，共同完成业务的核心目标。

## 第5章 架构和树

树是自然界中非常常见的一种结构。看看地球上长出来的东西，基本上都是树状的，地球上生长出来的万物基本都是地球的枝叶。人体是由一个细胞分裂出来的，也是一棵树。再看看算法，基本上效率最均衡的也是树。生命基本上都和树有关系。

我们来观察一棵树，树有主干，有枝叶。主干就相当于核心生命周期，决定了树的生死。枝叶是非核心生命周期，负责为主干获取太阳的能量。到了冬天，阳光不够、天气寒冷了，树叶就开始枯黄掉落，为树保存能量，使得树能够更好的过冬；到了春天，树叶生长，通过吸收阳光，自身长大，也为树干供应营养。同样树根也是一个树状的结构，负责从地球获取营养，自身也往下生长。我们做架构，就要从人、地、天、道这里来进行学习。

### 5.1 树与增长

树状结构有什么好处呢？树状结构特别适合于增长。如果是按照直线往上增长，长得越长，越容易失控，沟通成本越高，风一吹就折。如果按照水平增长，则很难获取阳光，容易被周围高的物体挡住太阳。而树状的增长，在成本方面最低，沟通的路径也没有显著的增长，带来的效果最好。树的结构也保证了分支的能量汇集到树干，保障了整棵树的生长。而架构恰恰是为了应对增长的，自然而然架构都是树状的。

从这个角度我们也可以看出，围绕着核心生命周期，把非核心生命周期剥离，也可以使得核心生命周期更加精简、轻量化，更容易应对周围环境的变化，树干也更容易长得壮实。即使某些非核心生命周期发生问题，也不会造成系统性的问题。

再看看建筑，是一个空间的树状架构。看看人类社会，是一个树状架构。看看组织架构，也是一个树状架构。一个村庄、城市、国家，都是空间树状架构，树状架构无处不在，只要有生命的地方就会有生命周期，就会有增长，就会有树。这是所有做架构的人都必须自觉认识到的。

谈到树，就必须提分层，分层实际上就是架构树状拆分的结果。所以当我们采用分层的时候，心里先要有一个树的概念。大多时候，很多人直接采用分层的方式来设计，最后反而违背了树的原则，导致了很多复杂的问题，影响到系统的长大，得不偿失。比如跨层访问，就形成了图，违反了树的原则。

## 5.2 架构和树

当业务增长到一定的程度，新的瓶颈出现了。架构发生新的拆分，新的非核心生命周期从业务生命周期中拆分出来，但架构仍然是树状的，并没有发生变化。树根和树干仍然是核心业务来决定的，并没有变，只是架构树长高了，枝叶增多了。只要树根和树干没有变，树的品种就没有变。不同的业务，树干和树根就变得不一样了。虽然都是树，却是不同类型的树，比如松树和柳树是有区别的。

有人会问，不是树状的结构，算不算架构呢？严格来说，这不算架构，因为并不能够保证权责对等，业务会受其阻碍不能顺利长大，甚至导致业务的失败。

有时候不是树状的，业务也能够长大啊？从现象看确实是的，但实质是业务本身的增长掩盖了这一点。这种方式就像慢性毒药，往往很快就让业务到达瓶颈，进一步还会干扰业务拆分。

即使是上文说的树状架构，树到一定的高度是否也就到了极限？是的，但是这个高度是由业务本身决定的，由业务所生存的周边环境提供的能量多少决定的，而不是架构。架构的目的是要让树能够按照自己的规律长大，而不是在业务中加入架构自己的材料，干扰核心业务，这样反而会阻碍业务的长大。

架构是顺应业务生命周期规律的一种拆分，这个拆分始终是围绕着业务的核心生命周期的，结果也只有一种，就是一棵树。只有树的结构，才能够保证权责对等。只有权责对等才能够保证参与人能从其自身的工作中获利，才能够保障参与人能够持续不断地推进业务的生命周期，比如树叶自身也要长大，才能给树干供应更多的养分。没有这个激励机制的保障，任何拆分都是徒劳的，无法执行的。



