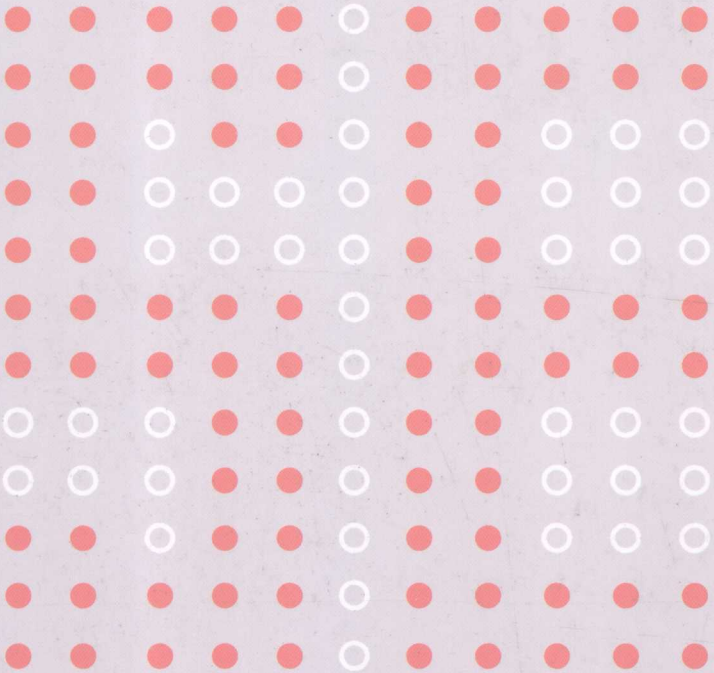
软件工程系列教材

张家浩 编著

软件架构设计实践教程



软件架构设计实践教程

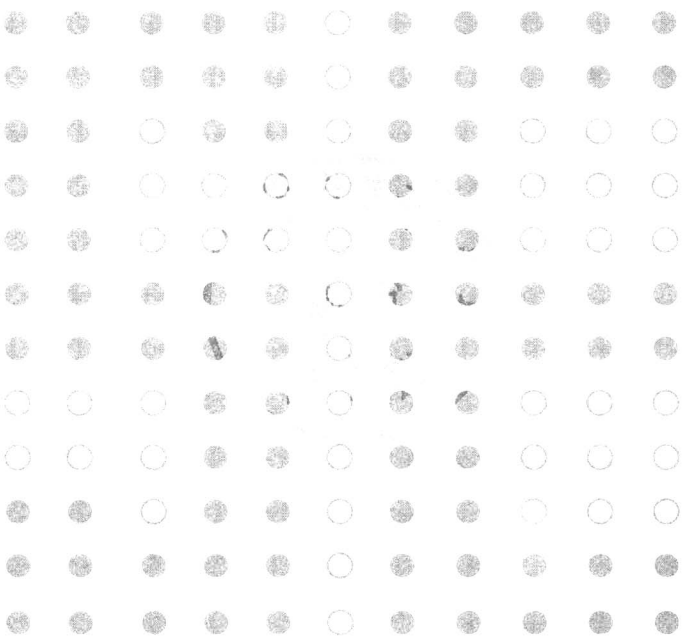
http://www.tup.com.cn

清华大学出版社

软件工程系列教材

软件架构设计实践教程

张家浩 编著



清华大学出版社

内 容 简 介

本教程是为计算机软件工程专业高年级学生或研究生“软件体系结构”课程而编写的教材(含实践内容),包括基本概念和案例分析、实践及思考题等。

全书共 9 章,分别是第 1 章认识软件架构,第 2 章架构与架构师的作用,第 3 章软件架构的描述与可视化,第 4 章从需求到架构,第 5 章软件架构设计的参考模型,第 6 章软件架构的概要设计与实现,第 7 章基于接口、组件和 SOA 的架构设计与实现,第 8 章基于 MVC 设计模式的架构设计与实现,第 9 章基于关键需求的架构设计、验证与评审。

本书与其他同类教科书的不同之处在于,作者根据多年企业工作和学校相关课程教学的经验,结合学生的实际情况和特点,有所选择地强化了相关课程中从关键需求分析到概要设计、接口和组件设计、MVC 模式应用、架构测试和验证等针对性、实用性强,学生看得见、摸得着,能感受、有兴趣学的软件架构知识和动手实践内容,淡化了软件架构的形式化描述等学生难于理解和把握且过于理论化的内容。

本书形式与内容编排与同类教科书有较大变化,在简短的基本概念介绍之后,配备大量的配套案例分析,希望能够帮助学生理解概念,并获得真实的架构体验。同时,在案例介绍中尽可能地采用当前比较流行的平台和工具,使学生在了解和掌握相关知识之后,马上就可以使用,缩短了学校与企业实际运用之间的距离。

教程每章还配有实践题和思考题,方便老师和学生使用。本书主要用作软件工程相关专业的“软件体系结构”课程,也可作为其他相关专业的教学用书,或作为从事软件开发的科技人员的参考书、培训教材等。

图书在版编目(CIP)数据

软件架构设计实践教程/[illegible]编著. —北京:清华大学出版社,2014
软件工程系列教材
ISBN 978-7-302-36637-[illegible]

Ⅰ. ①软… Ⅱ. ①张… Ⅲ. ①软件设计—高等学校—教材 Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字([illegible])

责任编辑:张　玥　薛　阳
封面设计:傅瑞学
责任校对:梁　毅
责任印制:何　芊

出版发行:清华大学出版社
网　　址:http://www.tup.com.cn, http://www.wqbook.com
地　　址:北京清华大学学研大厦 A 座　　邮　　编:100084
社 总 机:010-62770175　　邮　　购:010-62786544
投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载:http://www.tup.com.cn,010-62795954

印 装 者:清华大学印刷厂
经　　销:全国新华书店
开　　本:185mm×260mm　　**印　　张**:33.25　　**字　　数**:826 千字
版　　次:2014 年 8 月第 1 版　　**印　　次**:2014 年 8 月第1次印刷
印　　数:1～2000
定　　价:59.00 元

产品编号:053200-01

PREFACE

软件工程系列教材

前言

教过“软件体系结构”这门课的老师可能都有这样的体会：这是一门很“吃力”、可能也是“不讨好”的课。请企业的老师来讲，他们不了解学生的情况，“天马行空”，学生不能理解。学校老师教，教的人自己多半是“从校门到校门”，几乎没有软件系统的实际开发经验，煞费苦心，教得很辛苦，效果也不好。软件架构师在软件工程过程中，不但需要具有丰富的系统开发经验，并且与一般编码工程师相比，还需要具有更有高度、全局性的视角和软件过程抽象的能力，一般任课教师很难达到；同时，一个无法回避的现实是：学生进行编程和系统开发时间较少，基础能力较差（少数学生除外），与课程的要求严重不符。

软件工程专业的学生，又必须开设这门课，但上述教与学两方面的情况都不尽如人意，也是不争的事实。

1. 教材编写的动机

作者曾在某届软件工程研究生班（来自全国几十个高校、均为计算机相关专业）的入学摸底（C++/Java 编程）中发现，大多数学生的 C++ 或 Java 的编程能力，仅限于会编写一个控制台输入输出的简单“排序算法”，很多人居然从来没有用过 MFC（当然包括 VS2005～VS2010）。据说是老师说的，C++ 编程不用 MFC 的理由是，基础知识要保持平台无关性。似乎如果可能，最好连 Windows 或 UNIX 也不用，可惜做不到。更有很多老师，让学生照着老师编写的可运行代码“输”进去，运行起来，就可以交编程作业了。考试的时候，背背名词解释、填填标准答案，大家“和谐”相处，皆大欢喜。

这样的学生到了企业，怎么“对付”自己的程序员人生？作者在给企业讲课的时候，看到有些通过了系统分析师（软考）考试的员工，在介绍系统架构时，不论什么类型的应用系统都硬性地归纳为三层结构，感觉很奇怪。把某著名的软考参考书和复习题找来

一看才明白，原来系统架构师也可以这样“练成”的。我也曾经参与翻译日本 UML 等级考试(UMTP)的考题和教材。对于需求分析师、架构师这样的角色，什么是能力？如何考查学生发现问题、分析问题、解决问题的能力？看看人家是怎么做的。首先一点：考试的目的，是考查学生(或企业员工)实际的知识掌握和运用能力，目的是提高企业的相关运用的整体能力和水平。我们是为了什么？仅仅能背一些概念和名词解释有什么用？何况在软件工程中，有多少概念是“定论”，可以“永葆青春”？

为此，作者也翻阅了一些教科书或企业人士编写的技术书籍(见参考书目，罗列的目的并不都是推荐)，作为有二十多年软件企业经历，又有近十年教学体验的作者，感觉“和谐社会”，不应该只是发发牢骚，还是应该自己来写一本既适合学生实际，又符合企业选人、用人、培养人需要的书，这就是本书的成书动机。

2. 教学目的

这是一本与传统《软件体系结构》教科书不同的教材。不同之处首先在于，作者在此明确提出：课程的培养目标，是为软件企业培养未来的架构师，而不是为了迎合少部分学生“考研、读博”，虽然两个“目标”可能可以“和谐”相处。

为此，作者认为：根据企业当前和未来的需要，教学的内容和形式应该改变，不需要林林总总、包罗万象地灌输一大堆概念，美其名曰：基础扎实(其实写教材者，最怕别人说他不全面、不完整，不成体系)。试想，当现在的学生真正成为一个担负架构师责任的开发者时，那也是 5～10 年之后的事了，如果他还恰巧记得 10 年前所学知识和技术的话(甚至包括一些基本概念)，不知道 10 年后，是否还正确、还可用否？仅此一点，这些书本知识最后的结局，仅仅成为应付考试的工具，当属必然(很多学生是在临考试前一周才买书的，因为此时老师开始“圈”考试范围了)。

讲什么和不讲什么的依据，就是：培养未来 10 年后的架构师，现在需要让他知道什么？作者认为，与软件项目管理、软件工程等(甚至包括组织行为学等)课程一样，开设这些课程的根本目的，不在于教会学生多少具体的相关概念和技术方法，而是从现在开始，培养一种面向软件工程高端(与编程相比)角色(项目经理、架构师、需求分析师等)的意识和觉悟，即进行软件架构的“启蒙”教育、“目录”教育。

什么是“启蒙”？皇帝天天把太子带在身边，虽然年少或年轻的太子什么也不懂，带在身边就是让他看——“耳濡目染”。终有一天，轮到太子自己登基也做皇帝了，他自然知道应该怎么做皇帝。教授治国之道，仅靠一些国师，天天在书房里背“四书五经”是不行的。书本知识固然重要，但单靠书本，是不能治理国家的，架构师也是一样。在某次微软培训班上，有来接受培训的老师问：架构师不是项目经理，架构纪律如何贯彻？同样的问题：帝王的威严就因为他是帝王吗？

与太子生而就是太子不同，架构师是要靠“竞争”上位的，不是所有人最终都能坐上架构师的位置。但有没有给他机会看别人是怎么做的，并培养出一种架构师的意识和觉悟，是一个非常关键的因素。人们常说，机会是留给有准备的人，什么是准备？如何准备？道理就在这里。

实际上，架构师的成长来自两个方面能力的提升：“领导”(指《组织行为学》中动词而不是名词)意识的提升和实际工作经验的积累。两者都不是某一天到了架构师的岗位才

开始“觉醒”和积累的。因此，现在学习的目的就是“开窍”，开窍越早越好，因为“机会是为有准备的人准备的”这句话正好可以用在这里。

“启蒙”的结果应该怎样？课程结束后，一个好的学生，再也不会简单地只看功能、看代码来理解软件系统。就像医生一样，他已经到了离开了CT、甚至不做核磁共振，就不能确诊的程度。他要看架构(当然，他知道怎么看，看什么，怎么评价架构的好坏)。另一方面，经过“启蒙”的学生会觉得，什么“图书管理系统”、“学籍管理系统”，从架构角度看，都是“小儿科”系统。他再也不会为写了这么一个系统，而沾沾自喜。这就是启蒙后的“觉醒”，就是对软件的认知上了一个“层次”(实际上，这也是企业在某种程度上看不上学校做的项目的理由，尽管可能并不完全符合事实)。

这就是本书的教学目的，显然，在众多教材中，本书有点“另类”，甚至会有人说，这是“培训班”的教材，而不是大学教科书。软件工程不是经典物理，除了少数原则和思路，没有多少技术和方法可以留到10年后，继续发挥实际作用。且问：授之10年后可以发挥作用的“渔”，难道不强于已经“臭”掉的鱼，谁更有意义？你有自信你现在教的、所谓扎实的“基础知识”，例如设计模式、形式化方法，10年后还没有“臭掉”？谁也没有这个自信。

3. 教学方法

由于目标不同，教学方法与传统教材自然有所不同。

不少教科书讲授架构设计，一般或较少分析架构是如何从“需求”中来，及如何通过架构设计去满足需求。反之，也看不到代码是如何实现架构设计的。好像软件架构过程是“超然”于其他软件过程而独立存在，架构师按照自己的一套理论(教材内容)，“天马行空”、“上不着天、下不着地”地设计架构。

按照这样的教学思路，某些经典的教科书，开篇就是软件体系结构的“构件重用”、“风格和模式”，似乎选这门课的学生(包括研究生)都是软件架构师中的高手，来上这门课，只是为了来进行“理论深造”。而可悲的是，选用这本教材并讲授这门课的老师，看看他自己“可怜”的从业经历就可以知道，多数并没有参加过什么规范的软件系统开发，甚至没有参加过可称得上是“项目”、“产品”的软件开发，有的只是很多年以前、自己做学生时候的一些“作坊”级别的编程经验，当然谈不上具有架构设计的经验和体验。现在搞课题，“干活”的是学生，自己更不会动手了。所以，照搬一些现成的、空洞的理论，以“空对空”的方式(学生也是空)讲课，可能是比较好的应付学生的方法。

本书作者知道学生的程度和水平，不想让学生到了企业，让我的那些老同事说，“看！这就是老张教出来的学生。”学生只会在“控制台”上“排序”，你要讲23种设计模式，不是“对牛弹琴”吗？只好老老实实从最基本的东西开始进行“启蒙”，宁愿被同行们看得层次“比较低”，知识又“不全面”。但对学生来说，还比较实用。

本书的基本教学方法，是立足可以在“机房”上课。每一章都能够通过一两个可亲自动手“实作”的案例，让学生实际体验和感悟相关的知识内容。当然，这些案例可能比较“基础”，也谈不上有什么“技术水平”，但用于理解相关知识则是足够就好。同时，这些案例的一个最大好处，就是让学生熟悉并了解一些当前流行的平台和工具。虽然5～10年后这些平台或工具早已被淘汰或被更新的版本所取代，但是，它们在学生心中埋下的“种子”，10年后仍然能够起作用(“啊，这个我在学校时见过，大概知道是干什么用的，核心思

想没有变，只是现在功能更强大了”）。这就是“启蒙”的作用。

4. 章节安排和要点

第 1 章介绍软件架构的定义，并用生活中的例子以及一个最简单的“欢迎”程序创建过程为例，来理解和体验架构定义。架构定义有很多种，熟背 100 种定义，不如学会一种。架构定义是一种“眼光”，可以把它当作“透视”、“剖析”软件架构的“X 光机”。架构定义是贯穿全书、学生认识、分析、理解、设计架构的基本出发点。

第 2 章是软件架构与架构师的作用。本书将软件架构和架构师的作用突出出来，单列一章，并为每一个架构用途安排一个详细的案例分析，讨论架构师在其中的作用。作者如此“煞费苦心”，是因为软件架构师是靠经验的积累、而不是靠“概念的堆砌”成长起来的。如果学生不能切实理解和体会架构的作用，就如同大海航行没有航海图和罗盘。进入企业后，混混沌沌，最终将不可能有机会走到架构师的位置，对现在所学的内容也会毫无兴趣。其实在软件企业中，即使已经担任架构师职责的人，有很多也未必了解架构和架构师的作用，由此，他们给软件开发所带来的危害，无异于在沙滩上盖楼。

第 3 章介绍架构的描述与可视化。本书用了大量的篇幅，介绍如何用架构师的眼光、通过各种不同的工具看架构。如同医生看“X 光片”一样，看（分析）架构是架构师区别于软件过程其他角色的基本能力，更是发挥架构师作用的入口和途径。相反全书几乎并没有涉及“形式化描述”等一般教材必不可少的内容。作者认为：形式化描述方法和内容，如同 Java 语言的中间形式，一定要（实际已经）被“封装”在工具中，如果不是做专门“研究”，如同使用数字电路，已不需要知道二极管、三极管的原理和测试方法一样。

第 4 章介绍从需求模型转化为架构的过程，并讨论架构师在需求阶段的作用。一般教材在此方面也几乎是“一带而过”，而本书则专列一章。因为作者认为，架构师的需求过程，是架构师必须要经历的关键过程，需求转换是架构师必须掌握的基本技能，因为需求是架构设计的源头，而方法和技巧是为需求服务的，否则很容易导致为架构而架构。

第 5 章介绍架构设计的参考模型，从传统软件工程的系统设计思路回顾开始，包括分析网络 7 层协议架构、操作系统架构等经典软件系统的架构，以及给架构师带来的启发和思考。

第 6 章介绍架构设计最重要、也是最核心的部分——子系统划分。除介绍一般子系统的划分原则和方法之外，通过具体案例，让学生理解和体会，划分或分解是为了什么。

第 7 章是架构设计的深入，探讨了接口、面向对象设计模式、基于组件和 SOA 的架构设计等更细致的设计课题，并用一些实际案例，让学生更深入地体会和理解架构中，业务逻辑与业务实体这两个非常重要的架构概念到底是什么。

第 8 章较为系统地介绍了基于 MVC 和 SSH 架构的系统设计，并通过一个医药管理系统权限管理功能二次开发过程的案例剖析，深入到架构内部，看看好的或不好的架构，对二次开发的影响。

第 9 章探讨基于关键质量属性的架构设计和验证、评审，这是对架构设计从来源到结果的贯通和审视。由于有了清晰的源头和目的，比之那些“无病呻吟”、“为架构而架构”的架构设计，到了审视结果的时候，就会对架构的作用，看得更清、把握得更准。

每章最后都安排有一两个实践小项目并配有相应的基础代码(在出版社网站(www.tup.com.cn)上可以下载)和思考题,大多数都是围绕本章主题的体验性程序,且以“二次开发”项目为主,以加深对本章内容的认识和理解。在此也向各位代码的原作者一并致谢。

本教程或许是一个“非主流”教师“自不量力”的抗争,虽然如水面上偶尔冒出的气泡,很弱很无奈。感谢清华大学出版社,给我这么一个“冒泡”的平台和机会。

张家浩

2013年12月30日于南京百合果园

CONTENTS

目 录

软件工程系列教材