

21世纪软件工程专业规划教材

软件系统分析与设计实训教程

张家浩 编著

10010101000101

111010010010

10010101000101

111010010010

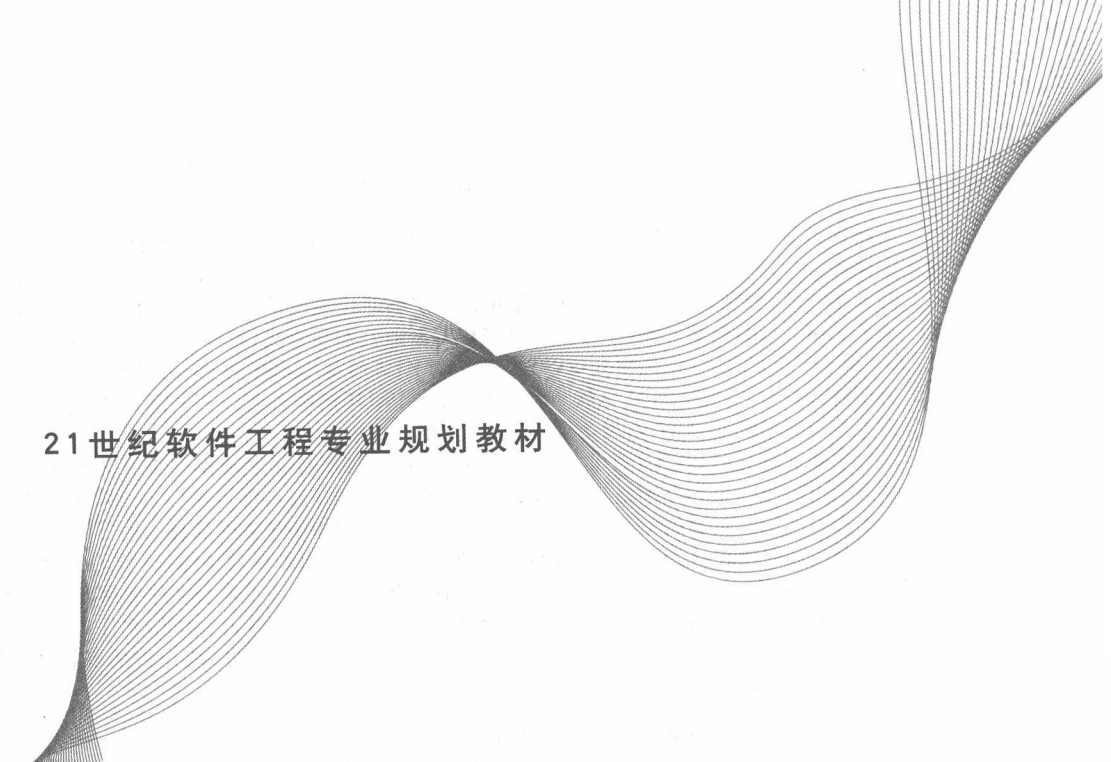
1000101

10010101000101

111010010010

清华大学出版社





21 世纪软件工程专业规划教材

软件系统分析与设计实训教程

张家浩 编著

清华大学出版社
北京

内 容 简 介

本教程是与计算机软件工程专业学生“软件体系结构”课程配套的实训教材,包括三个各为3~5周的实训课程。

全书共9章,分别是:第1章 软件架构设计实训课程导论,第2章 软件的逻辑与思维训练,第3章 基于Arduino的前端开发,第4章 基于树莓派的开发与集成,第5章 物联网服务器的开发与集成,第6章 STKUI的总体架构分析,第7章 STKUI的基本功能分析与二次开发,第8章 STKUI的扩展功能分析与二次开发,第9章 面向服务的STKUI架构再造。其中第2章、第3~5章、第6~9章分别为三个完整的实训课程,因为篇幅关系,分在各章中。

三个实训课程在软件架构知识、学生认知能力、系统与项目开发规模与综合体验的复杂程度上,是三个递进、迭代层次,目标分别是系统逻辑思维训练、小系统开发与集成实训、大型“企业级”应用系统分析与二次开发实训。可根据学生情况和课程需要,分别或持续组织教学实施。

由于是实训课程教材,因此,教程仅仅在需要的地方,简单回顾软件架构设计有关的相关知识点和内容,全书重点是围绕三个实训的若干个项目,让读者从项目开发实践中体会软件架构设计的深度和广度。

教程配有全部PPT和项目源代码,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”课程的配套实训课程,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。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件系统分析与设计实训教程/张家浩编著. --北京:清华大学出版社,2016

21世纪软件工程专业规划教材

ISBN 978-7-302-43563-1

I. ①软… II. ①张… III. ①软件工程—系统分析—高等学校—教材 ②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(2016)第081946号

责任编辑:张 玥 薛 阳

封面设计:常雪影

责任校对:焦丽丽

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:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:28

字 数:647千字

版 次:2016年8月第1版

印 次:2016年8月第1次印刷

印 数:1~2000

定 价:59.50元

产品编号:069227-01

前言

P R E F A C E

作者自 2011 年离开学校,并未曾离开讲台,甚至还曾又回到原来的学校、站在曾经的讲台上,只是换了一个“身份”,真是颇具中国特色、传奇意味。

在培训机构任职,从 985 学校到高职,有机会接触不同的学校、更多类型的学生,作者发现,现在学生的状况,真的不是用“堪忧”两字可以轻松描述的。在作者讲授“软件架构设计与实践”课程的时候,大三的学生,编程和项目开发的基础之差,连 VS 2010 的 IDE 菜单按钮是干什么的都不知道;而刚学过 C 语言的大一学生,连在 TC 2.0 环境下,需要将 C 语言源程序不单要编译成 .obj 文件,还需要再链接成 .exe 文件后(也可以选择直接生成 .exe),才能运行都不知道。总是问老师,我的代码为什么不能运行(学生只完成了生成 .obj 文件一步)。这说明,这个学生从来没有自己完成过一个完整的源程序编写、编译和运行过程。也有的老师教 Java,不要求学生用 Eclipse/MyEclipse,而是用记事本写源代码。虽然也可以用命令行进行 Java 编译,但是,不使用 Eclipse/MyEclipse 的 IDE,难道代码不用调试吗?难怪学生看到老师用 A 作为变量名,自己用 B 的时候,觉得可能自己用错了。上述这些现象,可能是作者遇到的个别极端情况,但这难道可以简单地批评说,是某个学生的错吗?

本教程是作者近年任职培训机构,担任实训课程教学的总结,也是软件工程、软件架构设计课程实训产品的呈现。有不少老师,用了作者的《软件架构实践教程》后来邮件探讨,如何在大三学生“基础很差”的情况下,上好架构设计这门课。我在该书前言部分的第一句话就讲,上这门课是很吃力、并且是“吃力不讨好”的。因为基础差的根本原因不是这门课难(软件工程专业后端的课程,一定是比编程要难,任何专业的课程都类似),而是学生从一年级开始,就缺乏编程、系统、项目的扎实训练。一本高校情况好一点,二本高校以下难度很大。但是,二本以下也是软件工程专业,也不能放弃。所以,需要从大一开始,用实训的方式,进行补充和强化训练。这就是编写本教程的基本出发点。

本教程原设想是设计为 4 个实训阶段:逻辑思维训练、小系统开发与集成、企业级系统分析与二次开发,以及软件工程过程实训,分别安排在大一至大四或延伸到研究生的 4 个假期实训阶段。由于这样的实训课程的时间安排跨学年、跨课程,因此,不可能跟现阶段教学体系和课程计划中的某一门课程直接挂钩——归于某门课下。所以,目前本课程的名称被套在“软件系统分析与设计实训教程”的课程名称之下。为了减轻学生的负担,本书删去软件工程过程实训内容(可参考作者 2011 年清华大学出版社出版的《软件工程项目实训教程——基于微软 VSTS》一书),并主要围绕软件架构设计的主题进行描述。

实整,改变了4个阶段实训、分阶段、综合一体化地培养软件工程专业学生系统、项目、软件工程过程能力的大目标,将其缩小到“软件架构设计”这个软件工程过程的具体目标上,从而降低了本教程原设想的高度和作用域。虽然软件架构在软件工程过程中具有一定的“统领”作用,但毕竟只是角色之一,实在遗憾。

有关课程的目的、内容和教学安排等,在教程的第1章中介绍,不再赘述。

张家浩

2015年12月30日于南京百合果园

目 录

CONTENTS

第 1 章 软件架构设计实训课程导论	1
1.1 实训课程的培养目标	1
1.1.1 架构师是软件开发的“老兵”	2
1.1.2 架构师的知识与能力积累	3
1.1.3 课程目标：“架构意识”的启蒙	4
1.2 实训课程的主要内容	5
1.2.1 实训课程的主要内容和课时安排	5
1.2.2 传统主课程的知识结构和内容安排	6
1.2.3 《软件架构设计实践教程》的知识结构和内容安排	6
1.2.4 与《软件架构设计实践教程》的互补性	7
1.2.5 如何使用本教程	8
1.3 实训课程计划与考核	9
1.3.1 软件逻辑体验实训课程计划	9
1.3.2 树莓派系统开发与集成实训课程计划	10
1.3.3 企业级系统分析与二次开发实训课程	11
1.3.4 考试方法与成绩评定	12
1.3.5 实训课程考核的思路	12
1.4 实训课程的课程资源	14
1.4.1 课程资源	14
1.4.2 参考书目	14
1.5 实训课程的难点与关键点	14
1.5.1 大系统与小系统的区别	15
1.5.2 引进大型系统的难度	15
1.6 构建完整的实训课程体系	17
1.6.1 构建完整实训体系的目标与指导思想	17
1.6.2 主课程改革的三阶段总体设计	18
1.6.3 配套进行的实训课程三阶段设计	20
1.6.4 实训课程的企业本质	20

第2章 软件的逻辑与思维训练	22
2.1 从代码到架构的逻辑思维层次	22
2.1.1 逻辑与程序逻辑	23
2.1.2 程序逻辑与算法逻辑	25
2.1.3 超越程序和算法的系统逻辑	26
2.1.4 逻辑架构与架构逻辑	29
2.2 逻辑思维训练的实训设计	31
2.2.1 本次实训课程的目标	31
2.2.2 本次实训课程的时机安排	32
2.2.3 本次实训的二次开发项目选择	32
2.2.4 本次实训课程的授课方法	33
2.2.5 项目二次开发的意义	33
2.2.6 平台选择	34
2.2.7 实训课程的意义和价值检验	34
2.2.8 课程要求	35
2.2.9 动手能力基础	35
2.2.10 师傅带徒弟式的教学方法	36
2.3 理解程序逻辑	36
2.3.1 读懂别人程序的办法	37
2.3.2 看什么不看什么	38
2.3.3 万年历代码的模块抽取	38
2.3.4 归纳出程序的逻辑流程图	41
2.3.5 增加农历的具体实现步骤分解	42
2.3.6 计算农历的代码	42
2.3.7 显示农历的代码	43
2.3.8 课程小结	44
2.4 从程序到算法	44
2.4.1 修改程序	44
2.4.2 简单改进需求的实现	45
2.4.3 24点计算的第一次优化	47
2.4.4 更复杂的情况	49
2.4.5 课程小结	49
2.5 从算法到业务逻辑	50
2.5.1 五子棋程序的系统构成	50
2.5.2 模块划分与子系统设计	54
2.5.3 添加“人机对弈”功能的系统实现方案	55
2.5.4 智能机器人	58
2.5.5 课程总结	61

2.6	感受架构逻辑思维	62
2.6.1	推箱子游戏的新需求介绍	62
2.6.2	系统层面的新需求分析	64
2.6.3	推箱子程序的主程序代码分析	66
2.6.4	推箱子的 move 模块代码分析	67
2.6.5	推箱子模块的修改	70
2.6.6	关键质量属性需求	71
2.6.7	推箱子的关键机制分析	73
2.6.8	课程小结	75
第3章	基于 Arduino 的前端开发	77
3.1	从“码工”到架构师	77
3.1.1	“系统级”应用的概念	78
3.1.2	从“码工”到架构师的沉淀	80
3.1.3	“企业级”系统学习与实践的缺失	81
3.2	小型系统开发与集成实训	82
3.2.1	本次实训课程的目标	83
3.2.2	本次实训课程的课时计划	84
3.2.3	实验设备配置	86
3.2.4	本次实训的二次开发项目选择	87
3.2.5	本次实训课程的授课方法	88
3.2.6	实训课程的意义、价值与检验	88
3.3	树莓派应用开发与参赛	89
3.3.1	树莓派与物联网	89
3.3.2	树莓派与云计算	92
3.3.3	树莓派与大数据	93
3.4	应用平台选择与体验	95
3.4.1	物联网应用系统的逻辑与物理架构	95
3.4.2	物联网应用系统的逻辑和运行架构	96
3.4.3	配置树莓派	97
3.4.4	实现树莓派的远程登录	101
3.4.5	让树莓派成为 samba 文件服务器	105
3.4.6	让树莓派成为 Web 服务器	108
3.4.7	让树莓派成为媒体服务器	110
3.4.8	课程小结	112
3.5	用树莓派 GPIO 控制 LED	113
3.5.1	认识树莓派的 GPIO	113
3.5.2	GPIO 接口编程体验	117

3.5.3	使用函数库实现 GPIO 接口控制	121
3.6	了解 Arduino 开发平台	125
3.6.1	Arduino 平台介绍	126
3.6.2	Windows 环境下的 Arduino IDE	127
3.6.3	树莓派环境下的 Arduino IDE	131
3.6.4	树莓派与 Arduino IDE 通信	131
3.7	用 DHT11 获取温度	135
3.7.1	DHT11 温湿度传感器简介	135
3.7.2	DHT11 传感器的技术指标	136
3.7.3	连接电路	136
3.7.4	DHT11 的时序	137
3.7.5	硬件接线	138
3.7.6	DHT11 传感器信息采集软件	138
3.7.7	编译运行	140
3.8	用四位数码管显示温度	141
3.8.1	四位共阴数码管介绍	141
3.8.2	数码管接口说明	141
3.8.3	74HC595 的控制逻辑	142
3.8.4	数码管的片选与刷新	143
3.8.5	代码说明	143
3.9	课程小结	146
第 4 章	基于树莓派的开发与集成	147
4.1	树莓派平台集成的需求	147
4.2	选择树莓派上的 PyQt 界面框架	148
4.3	显示树莓派 CPU 温度的界面	150
4.4	前端 Arduino 与树莓派通信集成	153
4.5	实现温度显示与控制的集成	155
4.6	课程小结	156
第 5 章	物联网服务器的开发与集成	158
5.1	与 Yeelink 服务器集成	159
5.1.1	物联网服务器 Yeelink 的功能与局限	159
5.1.2	Yeelink 的接入设备与存储数据类型	160
5.1.3	在 Yeelink 上创建自己的设备和传感器	161
5.1.4	HTTP 协议简介	162
5.1.5	有关 Requests	164
5.1.6	有关 JSON 库	167

5.1.7	上传数据到 Yeelink	168
5.1.8	Yeelink 服务器的客户端 APP	170
5.2	开发自己的云服务器应用	170
5.2.1	搭建苏宁云服务器环境	171
5.2.2	把应用搬到云服务器上去	175
5.2.3	开发自己的 Web 应用	178
5.2.4	树莓派与 Web 服务器交互	178
5.2.5	搭建一个 Web 网站	179
5.2.6	在云上直接控制树莓派点亮 LED	184
5.3	将本例改造成电梯群控系统	186
5.3.1	电梯群控的物理架构与系统集成	187
5.3.2	电梯群控的运行架构与功能边界划分	187
5.3.3	电梯群控的逻辑架构与系统信息交互	188
5.3.4	电梯群控的可行性分析	189
5.4	参加物联网创意大赛	190
5.4.1	参赛的项目层面考虑	191
5.4.2	从获奖项目中了解参赛的机会与目标	192
5.4.3	关注数据来源	194
5.4.4	关注数据采集方式	195
5.4.5	关注数据应用	195
5.5	阶段课程小结	195
第 6 章	STKUI 的总体架构分析	197
6.1	软件系统分析的基本概念	197
6.1.1	软件系统分析的目标与过程	198
6.1.2	软件架构的五个基本架构描述方法	201
6.1.3	软件架构分析的 UML 描述方法	202
6.1.4	基于架构的软件系统分析	204
6.1.5	软件系统的逆向分析	205
6.1.6	应用系统的关键需求与架构	206
6.1.7	基于软件工程过程文档的系统分析	206
6.1.8	理解企业级应用系统的基础与应用框架	208
6.2	股票软件的功能与实现要点	209
6.2.1	股票软件功能及其软件实现概述	209
6.2.2	股票行情功能及其软件实现	212
6.2.3	股票指标分析功能及其软件实现	214
6.2.4	股票交易功能及其软件实现	215
6.3	STKUI 系统的物理架构分析	216

6.3.1	STKUI 的物理架构	216
6.3.2	STKUI 物理架构分析的意义	216
6.4	STKUI 系统的开发架构分析	217
6.4.1	STKUI 系统的 9 个开发工程包	217
6.4.2	STKUI 开发架构分析与综述	221
6.4.3	理解软件系统开发架构的意义和作用	222
6.5	STKUI 系统的运行架构分析	222
6.5.1	系统登录与数据接口界面	223
6.5.2	STKUI 的功能层次结构	223
6.5.3	STKUI 菜单、工具条按钮与功能窗口	226
6.5.4	STKUI 运行架构的评价与改进	228
6.6	STKUI 系统的逻辑架构分析	230
6.6.1	理解 MFC 的应用程序框架	231
6.6.2	从菜单追踪到代码	235
6.6.3	逻辑架构分析中的继承关系与调用关系	239
6.6.4	STKUI 的类与继承关系	240
6.6.5	STKUI 的类与调用关系	244
6.6.6	STKUI 的全局变量	245
6.7	STKUI 系统的数据架构分析	245
第 7 章	STKUI 的基本功能分析与二次开发	247
7.1	股票软件系统的分析与二次开发实训	247
7.1.1	本次实训课程的目标	247
7.1.2	本次实训课程的课时计划	249
7.1.3	本次实训的二次开发项目选择	249
7.1.4	实训课题的意义、价值与检验	250
7.2	股票用户图形界面的实现分析与二次开发	250
7.2.1	MFC 文档与视图的简单回顾	250
7.2.2	STKUI 所使用的第三方用户界面框架	252
7.2.3	STKUI 的窗口实现	255
7.2.4	STKUI 的菜单与工具条实现	257
7.2.5	Qt5 及其用户界面开发	260
7.2.6	基于 Qt5+VS 2010 的 STKUI 用户界面二次开发	264
7.2.7	STKUI 移植的可行性实验	266
7.2.8	STKUI 的移植开发	271
7.3	股票分时行情显示的实现分析与二次开发	271
7.3.1	STKUI 的行情功能	271
7.3.2	STKUI 的分时曲线绘制及其软件实现	273

7.3.3	STKUI 行情功能的二次开发	276
7.4	股票指标分析的实现分析与二次开发	277
7.4.1	股票指标分析算法概述	277
7.4.2	股票指标分析的软件实现	281
7.4.3	股票指标分析的二次开发	294
7.5	STKUI 数据接口的实现分析与二次开发	298
7.5.1	有关接口的概念	298
7.5.2	STKUI 的数据接口分析	300
7.5.3	从接口到硬盘	306
7.5.4	通过接口获得实时数据	308
7.5.5	有关数据接口的二次开发	309
7.6	STKUI 数据存储结构的实现分析	311
7.6.1	STKUI 的数据存储	311
7.6.2	STKUI 的数据获取过程	315
7.6.3	数据的使用方法	320
7.7	课程小结	322
第 8 章	STKUI 的扩展功能分析与二次开发	323
8.1	策略模型的建立与模拟交易分析	324
8.1.1	策略及其意义	324
8.1.2	STKUI 策略系统的功能实现分析	328
8.1.3	STKUI 策略的实现代码分析	332
8.1.4	STKUI 策略实现的改进与二次开发	336
8.2	基于策略模拟的程序化交易实现	338
8.2.1	STKUI 的程序交易	338
8.2.2	一个程序交易程序的 DEMO	340
8.2.3	程序交易接口的概念	341
8.2.4	通达信程序化交易接口	342
8.2.5	程序交易的实现方案	349
8.2.6	从策略模拟到量化投资	352
8.3	基于盘口的数据技术分析	354
8.3.1	什么是“盘口”	355
8.3.2	盘口数据与 Level2	355
8.3.3	基于盘口 Level2 的数据分析软件	358
8.3.4	盘口数据程序来源于接口	364
8.3.5	基于盘口数据的二次开发	368
8.4	股票的大数据分析与应用	370
8.4.1	大数据技术的基本概念	370

8.4.2	股市行情的大数据分析	372
8.4.3	热点股票的大数据行情分析	373
8.4.4	“百度股市通”软件	374
8.4.5	三维看股	374
8.5	课程小结	376
第9章	面向服务的 STKUI 架构再造	377
9.1	面向服务的系统新需求	377
9.1.1	机构可提供的股票服务	378
9.1.2	面向股票服务的新需求	379
9.2	面向服务需求与架构关系分析	380
9.2.1	STKUI 功能与架构的关系	380
9.2.2	面向服务的关键质量需求	382
9.2.3	D-Alpha 系统的架构蓝图参考	384
9.3	FB2K 的架构设计新思路与参考	385
9.3.1	FB2K 的用户界面框架样式	386
9.3.2	FB2K 界面框架与插件的载入与使用设置	388
9.3.3	架构师眼中的 FB2K 插件逻辑	390
9.3.4	面向对象设计模式的工厂方法	392
9.3.5	FB2K SDK 工厂方法的运用	396
9.3.6	SDK 部分代码分析	399
9.4	轻量级行业框架的系统架构设计	403
9.4.1	框架与行业应用框架的概念	403
9.4.2	轻量级框架的概念	404
9.4.3	面向股票的行业框架与轻量化方案	405
9.5	轻量级框架下的插件开发	406
9.5.1	业务插件的分类	407
9.5.2	业务插件的加载方式	407
9.5.3	加载业务插件的呈现	408
9.5.4	业务插件的接口标准	409
9.5.5	业务插件的开发实现	409
9.6	面向服务的 SOA 架构分析	411
9.6.1	服务提供的架构和机制	411
9.6.2	面向服务的 SOA 架构	413
9.6.3	SOA 架构的实现机制	415
9.6.4	基于 SOA 的应用案例需求	416
9.6.5	基于 SOA 特色的应用案例实现	418
9.7	基于 SOA 的公式自定义设计与实现	423

9.7.1 股票软件公式的自定义.....	423
9.7.2 华泰专业版 II 的公式自定义功能.....	425
9.7.3 STKUI 的公式自定义功能实现	427
9.7.4 面向服务的公式自定义设计与实现.....	430
9.8 阶段课程小结	432
参考文献.....	433

软件架构设计实训课程导论

本章导读

实训课程与通常课堂教学课程的一个重要区别是：实训课程本身应该被看作是一个教学“项目”，从项目管理要求出发，既然是项目，就必须要有明确的目标；有尽可能细致的过程规划和设计；在项目实施过程中，有可检查的里程碑定义，最终有交付成果的验收方法和评价标准。以项目的方式实施实训课程教学，是本书贯彻始终的基本教学方法，请从第1章开始，就适应这种“教”与“学”的方法。

因此，本章首先介绍并描述“软件系统分析与设计”实训课程项目的项目目标等相关内容。特别需要强调的是：这里所讨论的项目，不是课程中老师分析、学生完成的软件开发项目，虽然这是本课程所必不可少的。本章所讨论的重点，是这次实训课程本身，是本次课程的目标选择、内容安排、过程设计、交付结果验收以及与其他课程的相关关系等内容。

承担本课程教学任务的老师，当然应该首先仔细阅读并理解本章的内容，这是“实训课程项目”总体设计和规划的“教学指导书”。而且，也需要让学生知道和了解，因为这是需要老师和学生一起参加、共同完成的“项目”。

1.1 实训课程的培养目标

软件系统的分析与设计技术，在软件企业而言，是在软件产品设计、系统构建、项目开发、过程控制、系统维护等各项工作中，起主导作用的关键技术。微软的比尔·盖茨很多年前有一句著名的话，大意是，自己在微软可以什么位置都不要（例如董事长、CEO等），但唯一要做的，就是做微软产品的架构师。

在软件企业，架构师为什么那么重要，为什么架构师可以拿高薪，Robert Walters（华德士人才咨询有限公司，英国一家著名的猎头公司）2014—2015 全球薪资调查报告（<http://www.robertwalters.co.uk/career-advice/salary-survey.html>）显示，就算在金融工程这样“烧钱”的行业，企业级应用架构师的年薪也是很高的，甚至高于高级金融分析师，并仅次于总监等高管。

在学校，“软件系统分析与设计”是软件工程专业的一门专业主干课程，是每个软件工程专业学生的必修课。但是，由于种种原因，这门课与其他软件工程高端课程一样，成为软件专业的“政治课”。老师难上、学生难学，最后的结局，必然成为课程体系中的“鸡肋”，

在压缩课时的时候,难逃首先被压缩的结局。

既然这么重要的一门课,为什么难上、难学?原因有很多,本书作者在《软件架构设计实践教程》(清华大学出版社2014年8月出版)的第1章中已有分析,不再赘述。

要很好地解决这个问题,作为一名教师,我自己能做的事情,就是在教学过程中,不断探索、不断挖掘,积累更多的教学资源,采用更合适的教学方法,并把这些成果提供出来,供大家分享,大家一起共同提升这门课的教学水平和教学质量,最终真正让学生受益。

本课程以《软件架构设计实践教程》一书作为理论基础和相关知识来源,而本课程则重点突出基于系统分析和设计理论、技术和方法的项目化的实战体验。

不是学过这门课的所有学生,未来都能够成为架构师,除了能力之外,还有环境、条件、机遇等多种“非技术”因素。但是,如果没有成为架构师的潜质,当环境、条件、机遇出现的时候,你一定不可能成为一名架构师,这就是本课程(包括所有软件工程高端课程)的所谓“启蒙”作用——机会只为那些有能力、有潜质的人准备。

1.1.1 架构师是软件开发的“老兵”

与管理岗位不同,在很多软件企业,可能并没有非常严格、规范的软件架构师的岗位设置和岗位描述,可能实质干的是架构师的活、发挥着架构师的作用,却并不曾被公司或项目组正式任命为一名架构师。

在组织行为学中,架构师、项目经理一类“人物”,具有职位和角色双重身份。前者具有明确的职权和位置,可以直接支配资源,并发布命令,是名词意义上的“领导”。而后者主要的作用是影响、带动、感染、示范、沟通、协调,是动词意义上的“领导”。在微软举办的高校教师软件架构师培训班上,有老师提问:架构师不是“领导”,架构纪律如何贯彻执行?之所以会有这样的问题,就是没有搞清楚两种“领导”在软件企业中的角色差异,以及各自是如何发挥各自的作用的。而后的角色,在高校环境中,现在还真的不好找。因为正常高校的院长、系主任其实就应该是后者的角色,但现在已经完全“异化”为前者了。

简单来说,要想成为一名新的架构师,除了首先需要具有架构师的知识与技能之外,在软件企业,往往架构师是先“上路”(先实际上成为负责架构设计的那个人)、后“拿证”(后被指派、任命为某某系统的架构师)的。而且,与项目经理一样,你今天是某某系统的架构师,明天到另一个系统项目上,可能未必就是架构师。因为后一系统可能更大、更复杂,或者可能是一个你并不熟悉的系统。所以,架构师是靠自身的能力活着的。

为什么架构师一定会有一个“无证驾驶”的阶段?

首先,与“学车”相比,架构师岗位是没有“驾校”的,在学校学习的那些内容,最多也只能算是在“教练场”开车。一名刚从学校毕业的学生,来到软件企业,必定是从“码工”做起,那么,一名新的架构师是从何而来的?只能从实战战场“拼杀”而来。所以,架构师是一个要求“实战”能力很强的岗位,是一个要求战场经验、生存能力很强的角色,当然,更不要讲这些实战体验,培养了你的眼光、大局、策划、布局和综合能力。一个资深的“码工”并不等于“架构师”。

所以,架构师的“驾照”实际上是公司、同事对架构师的“追认”,一个好的架构师,当他规划、设计出完美的架构蓝图,并付诸实施之后,不知道过了多少年,后来的人为了升级、

维护旧系统再次回顾和浏览架构设计蓝图和设计文档的时候,可能会禁不住由衷赞叹前任架构师为3~5年,甚至更久远的作者留下一个松耦合的系统模块结构划分、清晰的关联、简洁的接口,预留下充分的扩展空间、灵活的控制手段,省去后来者在系统扩展和维护中可能遇到的分析、定位、修改、重写,甚至大幅调整的麻烦和烦恼;也可能让后来人拍桌子大骂,前面的架构师害人不浅。

在战场上,我可能宁愿被“老兵”骂而活命,也不愿听从一个“书呆子”长官的瞎指挥而送命。在职场也有点类似,架构师就是这样的“老兵”。

1.1.2 架构师的知识与能力积累

老兵是身经百战、在枪林弹雨中摸爬滚打锤炼出来的,软件开发的老兵是不是就是代码写得多,项目做得多?答案显然是否定的。不然,上过战场,没有被打死的战士,最后人人都成为了将军。软件企业也是这样。

从战士成长为将军,还需要“领导”才能。这就是软件架构师所需要、码工则不具备的与架构设计有关的思维和基本技能。

从架构的作用和架构设计的过程,可以归纳出架构师需要具备以下4大关键技能。

(1) 业务背景的认知能力:大型软件系统开发,都是针对领域的。架构师对特定业务领域的业务背景的认知、关键业务需求的理解、核心业务发展的走向,都需要具有深刻的认知和理解能力。这个能力的养成,需要在该领域耕耘多年,才能够达到一定的理解深度。设想,如果没有这方面的理解,他所设计的系统架构,且不要说如何能够充分满足用户的需求,更谈不上经历若干年后,可以完美地应对系统的迭代、扩展开发或必不可少的修改与维护。而现代软件开发更朝向基于专业领域框架的迭代开发方向发展,业务逻辑的抽象层次越来越高,业务实体则已经被“封装”到框架中,理解业务已经大部分变成理解框架。

(2) 架构设计与搭建能力:这个能力是架构师的基本功。从架构蓝图、到设计方案,再到架构实现,甚至架构维护,要求架构师像开发软件代码一样,将架构从设想变为现实。与编写软件代码不同,架构师的工作,很大一部分不是由他自己完成的,而是由他提出方案、指挥别人实现的。从这一点开始,他已经不是一个士兵,而是“领导”了。难就难在,一个没有“头衔”的领导如何“服众”。所以,他必然要经历从起初的“压不住”,到“心服口服”的成长过程。

(3) 架构关键需求的把握能力:关键需求是指与架构设计有关的那些需求,它们通常不会出现在用户的需求文档中,甚至在系统交付之前,用户从来都没有提出过。例如,软件系统的灵活性、可扩展性、实时性、并发性、鲁棒性等。有经验的架构师早早就“预见”到,这些需求的实现,最终是“逃不过”去的“一关”。这一关是什么、在哪里?如何在系统架构设计的蓝图阶段就考虑好解决方案,这个能力即是“远见”,更是“教训和经验”的积累。

(4) 架构设计的组织实施管理能力:架构的规划和设计,不像代码编写,自己调试成功以后,就可以交给别人使用。架构设计需要架构师完成规划和设计后,带领开发团队,共同去实现架构蓝图和设计方案。这包括:架构设计需要让开发团队理解、执行设计方