

基于真实项目案例的“软件体系结构”课程建设

孙耀¹, 陈圣国¹, 王智钢¹, 张海涛¹, 张燕²

(1. 金陵科技学院软件工程学院, 江苏 南京 211169; 2. 金陵科技学院, 江苏 南京 211169)

摘要:“软件体系结构”课程存在理论与实践脱节的问题, 据此提出基于真实项目案例的“软件体系结构”课程建设新思路: 基于 OBE 理念, 将真实项目场景融合到理论教学环节, 同时加大实践教学比例, 采取过程监测和多元化评价, 完成课程建设方案, 探索实践驱动理论的教学方式, 激发学生学习热情, 提高学生解决实际问题的能力。

关键词: 软件体系结构; 项目案例; 课程建设; 理论实践; 基于产出的教育模式(OBE)

中图分类号: G42

文献标识码: A

文章编号: 1673-131X(2018)04-0058-05

The Course Construction of “Software System Architecture” Based on Real Project Cases

SUN Yao, CHEN Sheng-guo, WANG Zhi-gang, ZHANG Hai-tao, ZHANG Yan

(Jingling Institute of Technology, Nanjing 211169, China)

Abstract: There is a problem of disconnection between theory and practice in the course of “software system architecture”. Because of this, a new idea for the course construction of “software system architecture” based on real project cases is proposed: According to the basic principle of Outcomes-based Education, the real project scenarios are integrated into the theoretical teaching links, meanwhile increasing the proportion of practical teaching, adopting process monitoring and diversified evaluation to complete the curriculum construction program. And the paper explored the teaching methods of practice-driven theory, stimulating students’ learning enthusiasm, and improving students’ ability to solve practical problems.

Key words: software system architecture; project case; course construction; theory and practice; Outcome-based Education(OBE)

随着信息技术应用领域的不断拓展, 软件产品规模逐渐扩大, 复杂程度也逐步提高, 这使具有搭建系统核心构架、澄清技术细节、扫清主要难点能力的系统架构师成为紧缺高端人才。因此, 培养学生在复杂软件开发过程中具备选择合理的设计模式和软件体系结构以简化系统复杂度、提高软件产品质量的能力, 成为各高校开设“软件体系结构”课程的重要教学目标。传统的“软件体系结构”课程内容侧重对软件体系结构基本概念、理论以及软件体系结构适用性的讲解, 而学生需要记忆大量

枯燥、易遗忘、易混淆的理论概念。同时, 这种灌输式教学方式不但影响教学目标的实现, 而且无法激发学生的好奇心和求知欲, 不利于自主学习和终身学习意识的培养。因此, 为更好地实现“软件体系结构”课程的教学目标, 培养具备利用所学基本理论识别、表达、分析复杂软件工程问题能力的专业人才, 本文以南京市多个重点项目为依托, 提出基于真实项目案例的“软件体系结构”课程建设方案, 以为应用型高端软件人才的培养提供借鉴。

收稿日期: 2018-11-26

基金项目: 江苏高校软件工程品牌专业建设工程项目(PPZY2015B140)

作者简介: 孙耀(1982-), 男, 吉林桦甸人, 讲师, 博士, 主要从事软件工程基础理论与技术应用研究。

一、基于 OBE 理念的课程目标制定

(一)课程目标对毕业要求指标点的支撑

根据工程教育的 OBE(Outcomes-based Education,基于学习产出的教育模式)理念,“软件体系结构”课程目标需对软件工程专业毕业要求提供直接支撑。

1. 支持毕业要求 1(工程知识):能够将数学、自然科学、工程基础和专业用于解决复杂软件工程问题。具体指标点为:具备扎实的软件工程专业知识,具备理解复杂软件工程问题的能力,能够判别软件系统的复杂性,能够运用所学知识和方法进行复杂工程问题求解(1-3)。

2. 支持毕业要求 2(问题分析):能够应用数学、自然科学和工程科学的基本原理识别、表达,并通过文献研究分析复杂软件工程问题,以获得有效结论。具体指标点为:能够应用数学、自然科学和工程科学的基本原理,对软件系统进行抽象分析与识别,并进行推理、求解和验证(2-1);能够有效分析和描述软件系统需求,对于具体的复杂软件工程问题,能够评估各种解决方案的优劣势(2-2)。

3. 支持毕业要求 3(设计/开发解决方案):能够设计针对复杂工程问题的解决方案,设计满足特定需求的软件系统、模块(组件)或算法流程,并能够在设计环节中体现创新意识,考虑社会、健康、安全、法律、文化以及环境等因素。具体指标点为:能够设计针对复杂软件工程问题的解决方案,设计满足特定需求的软件系统、模块(组件)或算法流程、测试方案及用例(3-3)。

4. 支持毕业要求 5(使用现代工具):能够针对复杂软件工程问题,开发、选择与使用恰当的技术、资源、现代工程工具和信息技术工具,包括对复杂软件工程问题的预测与模拟,并能够理解其局限

性。具体指标点为:能够掌握软件工程领域中所使用的主要方法、平台、工具和技术资源,了解其差异和适用领域(5-1)。

5. 支持毕业要求 10(沟通):能够就复杂软件工程问题与业界同行及社会公众进行有效沟通和交流,包括撰写报告和设计文稿、陈述发言、清晰表达或回应指令,并具备一定的国际视野,能够在跨文化背景下进行沟通和交流。具体指标点为:具有与业界同行及社会公众进行有效沟通和交流的能力,并能在复杂软件工程项目实施过程中运用以上能力(10-1)。

(二)课程目标对毕业要求指标点支撑情况

“软件体系结构”课程目标与毕业要求指标点之间的支撑关系如表 1 所示。课程目标 1 旨在以实际案例促进学生对软件体系结构中基本概念及经典软件体系结构各软件成分之间关系的理解,使学生具备扎实的软件工程专业知识(1-3),具备理解复杂软件工程问题的能力(2-1),进而能够根据问题的特征选择设计模式和软件体系结构并对所选软件体系结构进行评估(2-2);课程目标 2 通过仿真模拟,引导学生参照已有具体案例,逐步设计满足特定复杂软件工程问题的软件系统、模块(组件)或算法流程、测试方案及用例(3-3),体验软件开发体系结构设计过程的各个环节所使用的主要方法、平台、工具和技术资源(5-1)。课程目标 3 通过经典案例,分析不同设计模式的设计思想,总结所解决问题的具体特征,了解经典软件体系结构的优缺点与使用场景,使学生能够针对软件开发中遇到的问题和根据软件系统的特点进行软件体系结构设计。引导学生在分析、设计、开发、测试、验证和文档编撰过程中,选择、使用恰当的技术、资源、现代工程工具和信息技术工具(5-1),分析不同开发周期中各种软件文档的内容、格式、规范,使学生具备撰写软件开发文档的能力(10-1)。

表 1 课程目标对毕业要求指标点支撑情况

课程目标	毕业要求指标点					
	1-3	2-1	2-2	3-3	5-1	10-1
理解软件体系结构的概念及经典软件体系结构各软件成分之间的关系,能够根据问题的特征选择设计模式和软件体系结构并对软件体系结构进行评估;	✓	✓	✓			
具备采用程序设计语言、常见软件体系结构与设计模式,设计具有一定复杂度的软件的能力;				✓	✓	
能够针对软件开发中遇到的问题和根据软件系统的特点进行软件体系结构设计,具备撰写软件开发文档的能力					✓	✓

二、课程教学内容设置

为更好地实现课程目标,本文参考经典软件体系结构^[1-4]和设计模式教程^[5-7],针对毕业要求指标点选择教学内容,实现教学内容对毕业要求指标点的支撑作用(表 2、表 3)。区别于教学内容仅按

照章节和知识点安排的传统模式,基于真实项目案例的软件体系结构课程将理论教学与实践教学相结合,并将真实项目场景融入理论教学环节,同时加大实践教学比例。通过“支持毕业要求指标点的能力要求”和“学生任务”栏目,明确教学内容对应的毕业要求指标点、学生学习任务及学生学习后应获得的能力,将 OBE 理念落实到课堂和实验教学

表 2 理论教学内容对毕业要求及指标点的支撑

章节或知识模块	教学内容	支持毕业要求指标点的能力要求	学时	学生任务
第一章 构件与软件复用	软件复用的作用; 构件与软件复用; 构件模型及实现; 构件获取与管理; 构件组装技术; 软件体系结构的 概念及作用	支持毕业要求指标点: 1-3:具备扎实的软件工程专业知识,具备理解复杂软件 工程问题的能力,能够判别软件系统的复杂性,能够运用 所学知识和方法进行复杂工程问题求解。 能力要求: 理解软件复用技术在软件设计中的作用; 理解构件的概念及分类; 了解构件常见的构件标准,例如 COM+、CORBA、EJB; 了解构件获取与管理方法; 掌握基于功能构件组装技术、基于对象的构件组装技术 (构造法、子类法); 理解软件体系结构的概念	6	作业要求: 理解构件软件重用中的作用及构 件组装技术 自学要求: 要求课后查找资料进一步了 解 COM+、Corba、EJB 讨论: 讨论基于构件的软件开发的优 势是什么;基于构件的软件开 发面临哪些挑战和困难
第二章 软件体系结构	经典软件体系 结构风格; 客户/服务器风格; 三层 C/S 结构 风格; 浏览/服务器风格; 异构结构风格	支持毕业要求指标点: 2-1:能够运用数学、自然科学和工程科学的基本原理,对 软件系统进行抽象分析与识别,并进行推理、求解和验证; 3-3:能够设计针对复杂软件工程问题的解决方案,设计 满足特定需求的软件系统、模块(组件)或算法流程、测试 方案及用例 能力要求: 了解经典软件体系结构风格如管道过滤器、面向对象系 统、基于事件的系统、分层系统、仓库系统及知识库等的优 缺点 理解客户服务器、三层 C/S 结构、浏览/服务器风格的特 点及各部分构件的功能划分原则	8	作业要求: 利用 Visual Studio 或其他编程平台 构建 1 个客户/服务器结构的应用、 1 个三层 C/S 结构的应用、1 个浏 览/服务器结构的应用。 自学要求: 自行查资料,构建浏览/服务器结 构应用开发调试环境; 自行查资料,了解 AJAX 等 Web 开 发相关技术,提交 1 个综述报告。 讨论: 讨论 B/S 结构与 C/S 结构的优缺点
第三章 设计模式	设计模式概述; 设计模式的组成; 设计模式的描述; 面向对象设计 原则; 设计模式的层次 与分类; 设计模式示例; MVC 模式的设计; 与实现	支持毕业要求指标点: 2-1:能够运用数学、自然科学和工程科学的基本原理,对 软件系统进行抽象分析与识别,并进行推理、求解和验证; 3-3:能够设计针对复杂软件工程问题的解决方案,设计 满足特定需求的软件系统、模块(组件)或算法流程、测试 方案及用例; 2-2:能够有效分析和描述软件系统需求,对于具体的复 杂软件工程问题,能够评估各种解决方案的优劣势 能力要求: 掌握模式的概念; 理解模式的作用; 理解面向对象的设计原则:单一职责原则、开放封闭原则、 Liskov 替换原则、依赖倒置原则、接口隔离原则; 了解设计模式的分类和常见的设计模式; 掌握基于 MVC 模式的图形用户界面应用程序结构的开 发,理解其各部分的作用	14	自学要求: 阅读设计模式相关资料,了解工厂 方法、抽象工厂、单例、建造者、适 配器、解释器、迭代器、观察者 模式; 理解设计模式在前修课程中 Java 类库中的应用情况 讨论: 常见模式适用的应用场景和面向 设计原则在常见模式中的应用
第四章 软件体系 结构的评估	了解软件体系结 构评估的主要方 式、ATAM 评估 方法、SAAM 评 估方法; 熟悉软件体系结 构的质量属性; 了解基于体系结 构评估主要方法 的优劣势	支持毕业要求指标点: 2-1:能够应用数学、自然科学和工程科学的基本原理,对 软件系统进行抽象分析与识别,并进行推理、求解和验证; 2-2:能够有效分析和描述软件系统需求,对于具体的复 杂软件工程问题,能够评估各种解决方案优劣势 能力要求: 软件体系结构评估概述 SAAM 体系结构分析方法 ATAM 体系结构权衡分析方法 评估方法比较	4	作业要求: 熟悉软件体系结构的质量属性;了 解评估的主要方式、ATAM 评估 方法、SAAM 评估方法;了解基于 体系结构评估主要方法的优劣势 讨论: 比较分析 ATAM 和 SAAM 两种 评估方法的异同及优缺点

表 3 实践教学内容对毕业要求及指标点的支撑

项目名称	支持毕业要求的细化指标	学时	类型	每组人数	学生任务
经典软件体系结构风格	支持毕业要求指标点: 3-3、5-1 能力要求: 理解管道-过滤器软件体系结构、面向对象软件体系结构的原理; 掌握管道-过滤器软件体系结构、面向对象软件体系结构特点; 面向对象软件体系结构的编程实现	2	设计型	1	根据软件的功能要求采用管道过滤器体系结构、面向对象软件风格分别实现一个简单的应用程序的设计与实现
C/S 结构应用开发	支持毕业要求指标点: 2-1、2-2、3-3、5-1、10-1 能力要求: 根据软件功能要求,能够分离业务功能与数据库访问功能的能力; 根据软件功能要求,能够分离业务逻辑与视图的能力; 采用编程语言实现三层结构应用的能力	6	设计型	1	根据软件功能要求,分离业务功能与数据库访问功能,在 Windows 环境下采用 Visual Studio 平台实现 1 个客户/服务器结构的应用程序。服务器端数据库进行数据存储管理,客户端采用 asp.net 编程技术进行开发。在此基础上进一步分离业务逻辑与视图,编写调试应用服务器(业务逻辑)、客户端(用户界面视图)程序
B/S 结构应用开发	支持毕业要求指标点: 2-1、2-2、3-3、5-1、10-1 能力要求: 理解 B/S 结构应用中浏览器与 Web 服务器、数据库服务器以及应用服务器之间的关系; 能够采用程序设计语言实现 Web 数据库应用的能力	4	设计型	1	使用 IIS 或其他 Web 服务器软件搭建 Web 服务器; 采用 asp.net 或 jsp(选择一种)为实验 3 的软件增加 Web 访问界面
设计模式应用	支持毕业要求指标点: 3-3、5-1、10-1 能力要求: 根据问题的特点选择设计模式的能力; 根据设计模式完成软件结构设计的能力; 采用程序设计语言实现软件的能力	4	设计型	1	根据软件的功能要求选择设计模式,采用所选模式完成软件结构设计,并选择一种编程语言实现相应功能

单元。在项目案例选择上力求做到以下几点:一是优先选择最新的项目案例,因为软件技术更新迭代迅速,新的项目有利于学生理解最新的软件技术;二是选择质量高的项目案例,所选项目案例应该遵循软件工程规范,具备完备的软件开发文档与良好的代码管理机制;三是类型与难度多样性,针对不同的体系结构,选择具有代表性的项目案例,让学生更好地体会不同体系结构的应用场景。

三、基于案例研究的课程教学方法

相较传统的灌输式教学方式,基于案例研究的教学方法可以提高学生的学习主动性,帮助学生获得更多的专业实践能力。

一是案例真实,理论与实践结合。根据工程项目经验,合理选择最新的、能够与教学内容相结合的工程项目作为教学案例,将枯燥的理论概念与生动的实例项目结合,让学生切身体会每个理论是如何应用到具体的工程项目中去的,从而提高学生工程实践能力。

二是讨论为主,引导为辅。在案例分析中,采用以学生讨论为主、教师引导为辅的方式,让学生最大程度地参与到讨论过程中,激发他们运用所学知识的积极性,鼓励学生大胆尝试。教师通过逐步引导的方式,让学生体会所选案例是如何选择设计模式与软件体系结构的,并了解其优缺点。

三是仿真模拟,深刻体验。学生分组讨论、仿真模拟设计已有的复杂软件系统,并担任不同的角色,体会不同角色在项目管理开发过程中的作用,熟悉工作内容。这能够培养其团队意识和独立完成工作的能力(策划、组织、管理能力),促使学生学会与团队成员交流,提高表达能力,从而更好地协调不同角色之间的工作。

四、支持毕业要求达成度分析的多元化课程考核

(一)关注课程目标是否达成的考核方式

本课程考核方式采用试卷考核和实践考核相结合、期末考试和平时考核相结合的多元化考核方

式。对课程目标的评价由3部分构成:期末考试(占比40%)侧重考核课程目标1的教学效果,直接反映学生对软件体系结构中基本概念及经典软件体系结构各软件成分之间关系的理解程度;实践考核(占比40%)侧重考核课程目标2的教学效果,能够反映学生是否具备采用程序设计语言、开发平台常见设计模式和体系结构解决一定复杂度软件问题的能力;报告等其他形式(占比20%)侧重考核课程目标3的教学效果,可以反映学生是否具备撰写软件开发文档的能力。相较于一张试卷定成绩的传统考核方式,多元化的考核方式将学生学习过程评价作为重要依据纳入课程目标评价更具合理性。

1. 期末考试(占总成绩40%)。考试题型为填空题(10分)、选择题(20分)、判断题(10分)、简答题(10分)、体系结构应用(25分)和设计模式应用(25分)。

2. 实践考核(占总成绩40%)。本课程设置4个实验:经典软件体系结构风格(分值占12.5%)、C/S结构应用开发(分值占37.5%)、B/S结构应用开发(分值占25%)、设计模式应用(分值占25%)。每个实验提前1周布置给学生,要求其在课外进行实验预习,对实验内容进行分析和设计。教师在实验课上对学生分析设计的成果进行检查,评分由程序功能完成度(占比40%)、代码对各特定设计模式/体系结构各组成部分分解是否合理(占比30%)、实验报告是否完整详细和描述设计方案文档是否完整(占比30%)3个方面构成。

3. 其他考核(占总成绩20%)。本课程把作业纳入课程考核范围,作业形式由课后习题及报告两种形式构成,其中报告和课后习题(多次)分别为50分。

(二)毕业要求达成度分析

课程考核及成绩评定结果出来后,教师可通过

课程对毕业要求达成度进行分析(图1)。区别于传统的考试成绩的正态分布图,基于OBE的软件体系结构课程考核结果可视化分析可以帮助教师掌握全班在哪些毕业要求指标点上存在短板及每个学生的学习状态,以便持续改进课程教学并对学生实施个性化教学。

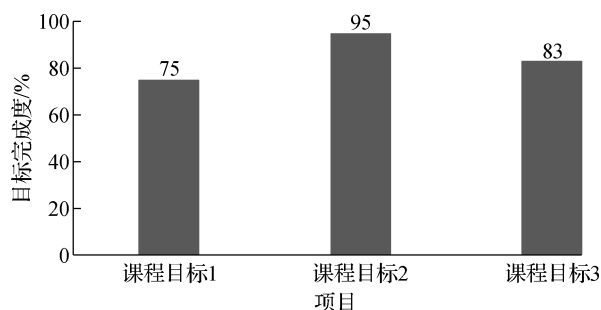


图1 基于OBE的“软件体系结构”课程考核结果可视化分析

参考文献:

- [1] 张友生. 软件体系结构原理、方法与实践[M]. 2版. 北京:清华大学出版社,2014:50-99
- [2] 周华. 软件设计与体系结构[M]. 北京:科学出版社,2012:307-340
- [3] 秦航. 软件设计与体系结构[M]. 北京:清华大学出版社,2014:47-139
- [4] 齐治昌. 软件设计与体系结构[M]. 北京:高等教育出版社,2010:84-213
- [5] Alan Shalloway. 设计模式精解[M]. 熊节,译. 2版. 北京:机械工业出版社,2006:55-278
- [6] Erich Gamma, Richard Helm. 设计模式:可复用面向对象软件的基础[M]. 刘建中,译. 北京:机械工业出版社,2013:54-231
- [7] 程细柱. 软件设计模式(Java版)[M]. 北京:人民邮电出版社,2018:9-25

(责任编辑:刘鑫)