

文章编号: 1672-5913(2018)07-0014-04

中图分类号: G642

基于群体化方法的软件工程课程实践教学

毛新军, 尹良泽, 尹刚, 王涛, 余跃, 卢遥

(国防科技大学 计算机学院, 湖南 长沙 410073)

摘要: 介绍如何借助群体化理念和思想, 改革软件工程课程的实践教学内容、实施方法和考评方式, 以加强工程实践和创新能力以及工程素养培养, 对课程实践内容的规模性、综合性、集成性、创新性、高质量等方面提出明确要求, 指出利用基于互联网的群体智慧和资源以及支撑平台来开展工程实践并解决问题, 科学评定实践效果和成绩并依此驱动实践的深入开展。

关键词: 新工科; 群体化方法; 软件工程实践教学; 开源软件

0 引言

如何培养高素质的软件工程人才、如何根据新工科建设的要求^[1-2]加强软件工程人才的能力培养是当前软件工程专业教育面临的重要挑战。软件工程是一门实践性很强的课程, 实践教学是课程教学的一项必备环节。为了满足产业和经济发展对软件工程专业人才的迫切需求, 我们需要从人才培养角度深层次思考这门课程实践教学的建设目标、方法和手段, 以期取得所需的人才培养效果。

首先, 如何认识软件工程课程实践教学的目标? 不同于其他的专业课程(如操作系统、编译原理等), 软件工程课程将诸多“工程”要素融合到课程实践中以系统性地培养学生工程实践能力、训练其工程素养。因此, 加强这门课程实践教学的建设对于提高人才培养质量、提升工程实践能力具有重要的价值。

第二, 如何达成教学目标并解决面临的问题? 软件工程课程实践教学面临的一个普遍性问题是学生经常抱怨实践没有参考对象, 不知道如何去做以及做的对不对; 一个突出的现象是学生经常遇到多样化问题, 老师难以做到一一解答。这就需要针对软件工程课程实践教学的特点和问题, 寻找有效的解决方法。

第三, 如何提供科学的考评手段并以此促进

人才培养? 考评是一个指挥棒, 引导学生按照要求来开展工作; 也是一个基准线, 告诉学生按照什么样的要求和标准来评定其成效和成绩; 同时也是一个推进剂, 以评促改, 帮助学生不断改进其实践成果, 在此过程中培养能力、提升素养和积累经验。因此, 这就需要从能力培养的角度来正确认识考评的目的、制定科学的考评标准和合理的考评手段。

1 改革目标和要求

软件工程课程实践教学需要为学生的工程实践和创新能力培养创造条件、明确要求和提供指南, 这就需要精心设计课程实践的内容, 分解和细化课程实践的要求, 形成课程实践的具体技术指标。结合软件工程专业和技术的发展, 考虑当前软件系统的形态和特征, 软件工程课程实践内容须满足以下的要求。

(1) 规模性。课程实践待开发的软件系统必须具有一定规模。只有通过对一定规模软件系统的开发实践才能让学生领悟软件工程中的“工程”要素并运用工程化的手段来解决问题。这里的规模可以表现为代码的长度、功能点的数量, 也可是构成系统的要素等。

(2) 交叉性。当前大部分的软件系统不再是一个纯粹的技术系统, 而是融合领域知识(如教

基金项目: 国家自然科学基金项目“大规模在线协同学习的机理与方法研究”(61532004)。

第一作者简介: 毛新军, 男, 教授, 研究方向为群体软件开发、多智能体系统、智能软件技术, xjmiao@nudt.edu.cn。

育、航空等)、交叉相关专业知识(如AI、移动计算等)的一类复杂系统。软件工程课程实践的内容需要体现软件系统开发的知识、技能等方面的综合性和交叉性。

(3)集成性。互联网时代的软件系统通常不再是单一、集中和封闭的系统,而是要与诸多系统和设备进行集成和交互。课程实践中应考虑将软件系统与其他系统和设备相结合来解决问题。

(4)高质量。质量是工程的基本要素。因此,课程实践必须重视质量,培养学生的质量意识(如编程规范、模型一致性等),掌握高质量软件产品的开发能力。

(5)创新性。有创意的软件才有价值,才能激发学生的实践热情和激情,帮助学生获得成就感。因此,课程实践要创造机会、提供空间来让学生进行各种创意和构思,从而培养学生的创新意识和能力。

(6)先进性。软件工程相关技术发展很快,业界所采用的技术不断变化,除了要训练学生的基本软件开发技能之外,课程实践要尽可能地采用业界的主流技术,如开发方法、实现语言、支撑平台和工具等,确保课程实践技术手段的先进性。

2 基于群体化的课程实践教学理念和思想

要达成上述目标是一项极具挑战性的工作。一方面,对老师的知识、经验和教学投入提出很高的要求。另一方面,对学生开展课程实践所需的知识结构、开发技能、工具使用、实践结果等提出了更高的要求。当前基于互联网的群体化方法及其在软件工程等诸多领域的成功实践为上述问题的解决提供新颖的思路和独特的方法,可以借助于互联网上的群体(如软件开发人员),利用他们的智慧(如开发经验)和产品(如开源软件)来开展课程实践,以克服老师、助教和学生知识、经验、阅历和认识等方面的局限性和有限性,达成工程实践和能力培养的目标。

(1)借助互联网群体的智慧。互联网连接了大量的人群,其中不乏高手和能人。它们拥有多样的软件工程知识,具有丰富的软件开发经验和技能。开源软件社区提供了灵活的机制来支持他们之间的交流、沟通和讨论,从而来分享他们的智慧和经验^[3-4]。课程实践可以借助于互联网群

体的力量和智慧,运用互联网社区的运行机制和手段,让互联网群体成为课程实践的指导者和贡献者,帮助和指导学生解答问题、提供资源、给出建议、反馈经验等。

(2)利用互联网群体的作品。互联网上汇聚了大量的开源软件,他们是互联网群体实践活动的结晶,其中不乏高水平、高质量^[5]的作品。重用开源软件成为目前重要的软件开发方式和手段。软件工程课程实践的规模性、高质量、交叉性、集成性等要求意味着让学生从头开始、全新开发这样的软件系统是不现实的。基于开源的软件开发方法将开发软件与重用开源软件二者相结合不仅可以满足课程实践教学的目标和要求,而且可以帮助学生掌握主流的软件开发技术。

概括起来,软件工程课程实践教学可借助互联网群体的智慧和力量,依托他们的软件开发知识、经验和作品,指导学生的课程实践,解决软件开发中遇到的种种问题,重用开源软件来实现特定功能,从而来达成和实现实践教学的目标和要求。

3 实践教学改革的关注点及解决方法

针对上述目标和要求,我们开展了基于群体化的软件课程实践教学探索。整个实践采用分组方式,每组由3~4位同学组成,他们共同完成实践任务,包括需求创意、软件设计、编码实现、软件测试等。课程实践要求学生提交分析和设计模型、软件文档、程序代码等软件产品。为了落实人才培养目标、加强工程实践和创新能力培养,践行群体化的实践教学方法,课程实践特别关注以下几个方面的工程实践问题,并采取了一组行之有效的实施方法。

1) 软件需求创意。

要求学生借助群体的智慧和力量,做有创意的课程实践,具体表现为软件系统欲解决的问题、软件需求以及实现技术手段要有新意。鼓励学生参考往届学生的作品,依托基于互联网的实践教学平台 Trustie,与往届学生进行交流和沟通,分享他们的经验和感悟,从而获得创意的灵感和想法。要求学生加入到开源社区,通过对已有开源软件的分析以及与社区中群体的交流来获得有关创意的意见、建议和想法。这两种方法的

本质都是希望扩大学生的视野、获取他人的智慧、听取他人的建议、获得更多的知识,并依此来构想软件需求。在此过程中,老师主要起到引导、评价、建议、把关、促进再思考等作用。

2) 分布式协同开发和版本管理。

要求学生基于 Git 开发、管理软件。每个项目团队创建各自的软件版本库,负责人基于 Issue 管理工具来安排和发布任务,成员间基于 Git 来开展交流和解决冲突,每个成员利用 Git 提交各自的作品(如程序代码),并利用 Pull-request 机制将其合并到整个课程实践项目中。它不仅可以帮助学生掌握基于互联网的软件开发模式(如任务分配、作品发布、作品的本地开发和分布式提交等),而且有助于获取学生个体的课程实践情况,搜集第一手课程实践数据,从而对学生的实践情况(如参与程度、所作出的贡献、产品质量等)进行科学和客观的评价。

3) 迭代开发。

由于课程实践对软件的创意、规模和知识交叉性等提出了更高要求,对于学生而言要一次性地开发出如此复杂的软件系统是非常困难的,他们常常面临需求创意难以想清楚、一些复杂功能难以实现等问题。为此我们采用迭代的方法来开展课程实践,在整个课程实践过程中要求学生分若干次迭代来完成整个课程实践任务,每次迭代关注的角度和评价的要求不同。

4) 以评促改。

课程实践要求学生周期性地汇报课程实践的成果(平均每周一次),老师根据学生的汇报对学生的课程实践进行“点评”,帮助学生发现问题并提供建议,目的是引导学生对课程实践成果持续改进和提高,从而起到以“评”促“改”的功效。针对学生的调查问卷结果表明,课堂点评最受学生欢迎,并对学生改进和提高课程实践水平起到关键性的作用。

5) 重用开源软件。

课程实践鼓励学生根据构想的软件需求,到开源软件托管网站(如 SourceForge、Github、Stack Overflow、开源中国、CodeProject 等)去寻找支撑其关键需求的开源软件,并将其集成到整个课程实践的软件项目之中。为了鼓励学生重用开源,我们明确要求学生在课程实践中要重用至

少 5 000 行开源软件。

6) 借助开源社区。

课程实践要求学生加入到相关的开源社区中,学生通过检索获取相关讨论或者在社区中提出问题获得所需的解答。此外,我们还通过实践教学平台收集了每一届学生开展课程实践所遇到的问题、解决问题的方法、软件开发经验等,学生也可以通过分享这些智慧来促进相关问题的解决。

7) 持续检查。

课程实践要求学生采用持续检查(continuous inspection)来确保所开发软件系统的质量,借助持续集成系统 Jenkins 将 SonarQube 质量分析工具集成到 Trustie 平台,实现了对学生提交到版本库代码的自动化质量检查,并要求学生对问题代码进行修复后,才可合并到项目仓库中。

8) 考评方法。

采用定性和定量、主观和客观、过程和结果相结合的考评方法,针对多个方面(如参与度、质量、贡献等)给出一个详尽的实践考核评价要素,借助于课程实践教学平台 Trustie 以及诸多软件开发工具(如 Git 和 SonarQube),搜集学生在课程实践过程中的参与和贡献信息(如提出问题、提供建议、参与讨论、提交作品等的数量),对课程实践结果(如代码质量)进行定量的分析,尽可能做到基于数据、综合实践过程和结果、统筹实践的工作量和质量、团队汇报和个体问答相结合的方式进行考评,从而通过考评方法的改变来驱动学生更为独立、自主、积极开展实践。

9) 工具和平台支持。

为了帮助老师和学生高效地开展课程实践,跟踪实践情况和效果,科学客观地评价实践成绩,课程实践教学借助一系列的工具和平台,包括分布式协同开发平台 Trustie-Forge、开源软件检索和分析平台 Trustie-OSSEAN、开源代码阅读和标注工具 Trustie-Codepedia 等。

4 教学成效分析

基于上述实践教学方法,针对软件工程专业学生开展了 2 年的探索,每年大约有 24 名学生参与,采用小班教学模式,配备 2~3 名助教。课程实践延续一年时间,覆盖了 4 门课程的实践教学任务,包括软件工程、软件体系结构与设

计、软件项目管理以及软件课程综合实践。整体上看软件工程实践教学改革取得了以下成果:

(1) 学生通过课程实践所开发的软件系统代码规模平均超出 10 000 行。2 个年级课程实践提交的代码统计结果表明, 课程实践的各项目开发程序代码量平均超出 5 000 行, 重用开源软件超出 5 000 行(见表 1)。

项目名称	开发代码	重用开源软件代码
基于 AR 的 3D 导航	5 000	17 000
图书管理机器人	8 500	3 000
家庭服务机器人	7 500	3 000
ROBOT 机器人战斗模拟	7 000	80 000
校园信息集成 APP	5 000	4 000

(2) 软件系统的集成性和综合性。所开发的软件系统大部分是基于互联网的分布式系统, 许多软件系统集成成了机器人、无人系统、智能手机、Web 服务等系统和设备, 交叉了模式识别、移动计算、数据分析等相关知识。

(3) 软件质量意识和水平得到提升。学生软件开发的质量意识得到加强, 运用各种技术(如持续检查、软件测试等)和工具(如 Junit、SonarQube 等)来开发高质量软件系统的能力得到提升, 分析结果表明学生开发的软件产品的质量处于较高水平。

(4) 软件工程综合素养获得提升。课程实践过程中学生持续进行总结、报告和讨论, 制作宣传彩页、视频和汇报 PPT, 撰写产品手册、凝练创新点, 因而学生的总结、分析、表达、汇报、交流、讨论等方面的综合素养得到了显著提升。

(5) 学生掌握了基于互联网的现代软件工程技术和工具, 学会了借助于开源软件的软件开发技能(如查找、阅读、分析和集成开源软件等)、依托开源社区获得群体智慧的方法(如提出和发布问题、查询解答等), 运用基于互联网的 CASE 工具来进行协同开发。

(6) 课程实践成果的创新性得到认可。两年来学生构思了许多有创意的软件作品, 包括无人值守的图书馆、家庭接待机器人、多无人机联合搜寻系统、Baby 安保助手等, 这些软件作品获得了计算机作品比赛、机器人比赛等多项全国性奖项, 包括金奖 1 项、银奖 2 项、省赛区特等奖 3 项和一等奖 1 项。

5 结 语

实践教学是软件工程系列课程教学的重要环节, 如何改革软件工程课程实践教学, 克服当前实践教学理念、实施方法、评价手段等方面的局限性是一项极为紧迫的任务。基于互联网的连接、共享和开放思想, 借助以开源软件和开源社区为代表的群体化思想, 结合主流的软件工程技术和工具, 针对软件工程实践教学中的突出矛盾和瓶颈问题, 开展了以提升工程实践和创新能力为目标的实践教学改革, 取得了积极的成效, 为加强软件工程专业人才的培养做出了积极的探索, 极大提高了学生开发大规模、高质量软件系统的自信心。两年的教学实践表明, 在互联网时代我们应该采用更为开放的思维来寻找解决实践教学问题的方法, 应该充分发挥互联网的优势和力量来弥补当前实践教学存在的问题和不足, 应该通过教学和科研相结合、满足产业发展需要以及贴近主流的技术来落实实践教学改革。

参考文献:

- [1] 林健. 面向未来的中国新工科建设[J]. 清华大学教育研究, 2017, 38(2): 26-35.
- [2] 张大良. 因时而动返本开新建设发展新工科: 在工科优势高校新工科建设研讨会上的讲话[J]. 中国大学教学, 2017(4): 1-3.
- [3] LaToza T D, van der Hoek A. Crowdsourcing in software engineering: Models, motivations, and challenges[J]. IEEE software, 2016, 33(1): 74-80.
- [4] Papadopoulos P M, Stamelos I G, Meiszner A. Enhancing software engineering education through open source projects: Four years of students' perspectives[J]. Education and Information Technologies, 2013, 18(2): 381-397.
- [5] Neill C J, DeFranco J F, Sangwan R S. Improving collaborative learning in online software engineering education[J]. European Journal of Engineering Education, 2016, 41(3): 1-12.

(编辑: 彭远红)