

◆ 一流大学计算机类专业核心课程教材

软件工程实践教程： 基于开源和 群智的方法

Software Engineering Practice:
Restarting from Open Source
and Crowd Wisdom

毛新军 王涛 余跃 编著

高等教育出版社

◆ 一流大学计算机类专业核心课程教材

软件工程实践教程： 基于开源和 群智的方法

Software Engineering Practice:
Restarting from Open Source
and Crowd Wisdom

常州大学图书馆
藏书章

毛新军 玉涛 余跃 编著

高等教育出版社·北京

内容提要

本书是软件工程课程实践教学方面的教材,针对实践教学的特点及要求,以培养学生解决复杂工程问题能力和软件工程素质为目标,围绕实践教学任务的设计、实施和考评三个方面,阐述如何转变思想和观念,基于群智方法、借助开源软件,克服现行实践教学方法的局限性,提高软件工程课程实践教学成效及人才培养水平。

全书设计了两项相对独立、逐级递进的实践任务。一是分析和维护开源软件,二是开发软件系统,并结合具体案例详细介绍了这两项实践任务的实施细节及方法,阐明了如何借助开源软件及群智知识来开展软件开发实践,以及如何采用定性和定量相结合、人工和自动相结合的方式对实践行为和结果进行系统考评。

本书可作为计算机大类专业的软件工程课程教材,也可作为软件工程师的参考用书。为便于教学,提供了丰富的教学资源,包括支撑实践教学的软件开发文档模板,完整的软件项目案例及其软件制品,电子教案 PPT 和教学视频,互联网上的学习社区,支撑软件工具等。

图书在版编目(CIP)数据

软件工程实践教程:基于开源和群智的方法 / 毛新军,王涛,余跃编著. -- 北京:高等教育出版社, 2019.8

ISBN 978-7-04-052423-9

I. ①软… II. ①毛… ②王… ③余… III. ①软件工程-高等学校-教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2019)第 159251 号

策划编辑 倪文慧
插图绘制 于博

责任编辑 倪文慧
责任校对 胡美萍

封面设计 于文燕
责任印制 韩刚

版式设计 马云

出版发行 高等教育出版社
社址 北京市西城区德外大街 4 号
邮政编码 100120
印刷 廊坊市文风档案印务有限公司
开本 787mm×1092mm 1/16
印张 21.75
字数 530 千字
购书热线 010-58581118
咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.hepmall.com.cn>
<http://www.hepmall.com>
<http://www.hepmall.cn>

版 次 2019 年 8 月第 1 版
印 次 2019 年 8 月第 1 次印刷
定 价 42.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物料号 52423-00

软件工程 实践教程：

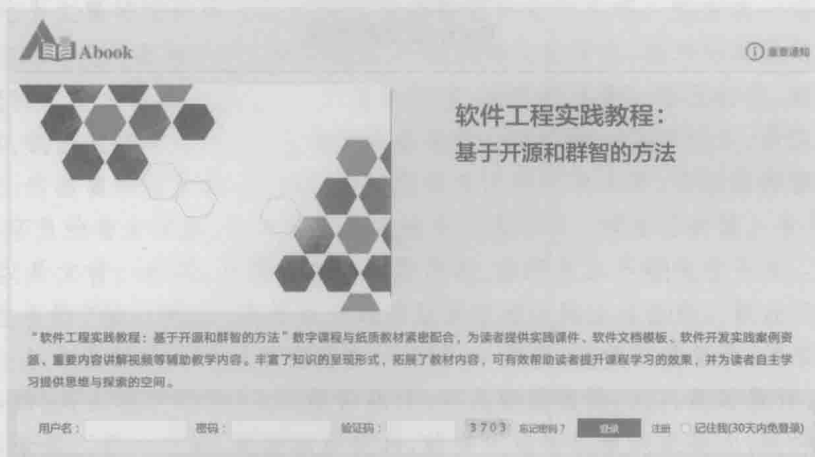
基于开源和
群智的方法

毛新军 王涛 余跃 编著

本书配套的数字资源使用方法如下：

- (1) 电脑访问<http://abook.hep.com.cn/1866338>，或手机扫描二维码、下载并安装 Abook 应用。
- (2) 注册并登录，进入“我的课程”。
- (3) 输入教材封底防伪标签上的数字课程账号（20 位密码，刮开涂层可见），或通过 Abook 应用扫描封底数字课程账号二维码，完成课程绑定。
- (4) 单击“进入课程”按钮，开始本数字课程的学习。

课程绑定后一年为数字课程使用有效期。受硬件限制，部分内容无法在手机端显示，请按提示通过计算机访问学习。如有使用问题，请发邮件至 abook@hep.com.cn。



具体的数字资源如下：

(1) 实践课件

提供了一组 PPT 以介绍软件工程课程实践任务的设计、实施和考评。

(2) 软件文档模板

提供了一组模板以描述软件系统构思、分析和设计的成果，形成规范化的软件文档以及相关的自评表格支持实践的考核。具体包括软件需求构思和描述模板、软件需求规格说明书模板、软件设计规格说明书模板、实践团队/结对/个人自评表等。

(3) “分析和维护小米便签开源软件”实践案例资源

提供了往届学生在“分析和维护小米便签开源软件”中所积累的实践资源，具体包



括体系结构UML模型、代码标注、代码质量分析报告、维护后的源程序代码、技术博客、实践成果汇报等。

(4)“空巢老人智能看护系统”软件开发案例资源

针对“空巢老人智能看护系统”，提供了完整的实践成果和软件制品，具体包括软件需求构思和描述文档、软件需求规格说明书、软件需求UML模型、软件设计UML模型、程序代码、技术博客、产品宣传海报、演示视频、实践总结汇报PPT等。

(5)“开发软件系统”实践的其他软件开发案例资源

提供了往届学生在“开发软件系统”实践中所开发的软件系统及相应的实践资源，包括模型、文档、代码、技术博客等，具体的软件系统包括无人值守图书馆、基于AR的3D地图导航、访客自主识别机器人、多无人机联合搜索系统、Baby安全助手等。



序

软件工程教育是专业教育还是通识教育,软件开发能力是“教”出来的还是“练”出来的?我的回答是,软件工程教育既是专业教育也是通识教育,软件开发能力不是“教”出来的而是“练”出来的。这是信息时代发展决定的,也是软件开发活动自身特点决定的。软件工程教育,尤其是软件工程实践教学需要改变。

信息技术的发展过程也是信息技术推动人类社会信息化水平不断升级的过程。我们经历了以数字化为主要特征的第一次信息化浪潮的洗礼,正处于以网络化为主要特征的第二次信息化浪潮之巅,已经听到以智能化为主要特征的第三次信息化浪潮扑面而来的轰鸣。所有这一切都离不开软件,软件已经成为信息时代的基础设施,软件定义一切,软件无处不在,软件的地位越来越重要。语言文字的出现使得人类文明突飞猛进,我们身处其中,也深感其重。信息时代,软件成为承载人类文明的新载体,我们同样身处其中,也同样深感其重。我们每一个现代人,不仅要成为传统文明成果的继承者、传播者和创作者,也将成为现代软件文明的继承者、传播者和创作者。软件开发不再仅仅是程序员的专业技能,还将成为越来越多普通人的一种生活技能。今天,我们可以不是作家,但不可以是文盲。未来,我们可以不是程序员,但不可以不懂软件开发。实际上,今天的软件开发已由过去的“独门绝技”发展成为越来越多普通人的生活技能。软件工程教育必须顺应这一时代要求:从专业教育向通识教育转变。然而,一方面,软件日益普及,越来越多的普通人涉足软件开发活动,人人能用软件,人人能评软件,人人能读软件,人人能写软件,计算思维已成为现代人的基本素养。另一方面,软件日益复杂,软件代码的规模越来越大,软件作用的范围越来越广,软件质量的要求越来越高,即使受过专业训练的程序员也越来越难以驾驭日益复杂的软件。软件工程教育必须因时而变。

早期的软件开发活动是个体性的创作活动,是“匠人”在“私人作坊”中的“手艺”活,人才培养是“师傅领进门,修行在个人”,软件开发效率低,质量难以复制和保证,优秀软件开发人才匮乏。50年前计算机产业发展起来之后,“软件危机”(也可以说是软件开发危机)自然就出现了。当时的解决之道是软件开发“工程化”,就是将工业时代成功的硬件生产和管理方式应用于软件开发活动,软件开发成为大规模工业化生产活动,大学也开始大批培养软件生产线上的工程师,软件开发效率和质量大幅度提升,出现了成功的软件企业。然而,软件工业化生产的努力很快也出现了瓶颈。人们发现,软件开发作为人类的智力活动,不可能也不应该被束缚在严苛的生产线上,软件工程教育不能脱离鲜活的软件创作实践。20年来开源软件的成功实践,使人们看到了大规模群体化软件创作的活力与力量。这其中有何规律可循?对于软件开发方法的研究有何启发?对于软件工程教育的改革有何启发?10年前我们开始开源软件模式的研究,提出了群体软件创作与可信软件生产二者相融合与转化的思想,形成了基于网络的软件开发群体化方法,

构建了支撑环境 Trustie,开展了群体化方法的大规模验证和应用,并在此基础上开展了软件工程实践教学改革的探索。本书就是这一实践的阶段成果。

本书所谓的“软件工程”是广义的软件工程,是学科意义下的软件工程。本书所谓的“软件工程实践”是广义的软件开发实践,不仅包括经典的“工程化”软件生产实践,还包括网络环境中群体开源软件创作实践。几千年来,人们通过“听、说、读、写”涉足传统语言文字文明的继承、传播和创造活动,当今,人们已经通过“用、评、读、写”涉足现代软件文明的继承、传播和创造活动。移动互联网的普及使得越来越多的普通人成为软件的使用者,普通用户对软件的在线使用反馈(哪怕是一次“点击”、一次“吐槽”)也已成为软件开发需求获取的渠道,成为软件质量提升的途径,普通用户已经不知不觉地参与到软件开发活动中,在“人-机-物”日益融合的三元世界中,普通人对软件开发的参与将更加不知不觉、更加不可或缺。随着开源生态的发展,高水平的软件源代码以及软件开发工具将更加丰富、更加容易获得和使用,更多的非软件职业人士将涉足软件代码的编写,软件开发的群智特征将更加凸显。

本书所讨论的“基于开源和群智方法的软件工程实践”,是国防科技大学教学团队推进软件工程实践教学改革的实录,与其说是面向软件工程专业教学的实践“教程”,还不如说是面向更广泛读者的实践“邀约”,希望与读者共同探索基于群智方法的软件工程实践。

王怀民

2019.5.26 于长沙

前 言

软件工程是一门实践性要求非常高的课程。实践教学是该课程教学的要点、重点和难点,其成效直接决定了该课程教学质量和人才培养水平。本书聚焦于软件工程课程实践教学,以面向对象软件工程技术为基础,利用开源软件和开源社区中的软件开发知识,围绕实践教学的设计、实施和考评三个方面,结合具体的软件开发实践案例,介绍了基于群智的软件工程实践教学的理念、内容、方法和工具。

本书主要特色

① 以能力和素质培养为目标,改革软件工程课程的实践内容、实施方法和考评方式。在分析软件工程课程教学特点及当前实践教学方法不足的基础上,以培养解决复杂工程问题能力和软件工程素质为目标,本书对软件工程课程实践教学的任务和内容、实施方法、考评方式进行了针对性的改革,要求实践内容有创意、上规模、高质量且具综合性,提出依托社区、借助群智、迭代开发的实践实施方法,建立以能力培养为导向、持续点评和改进、定性和定量相结合的考评方式,为软件工程课程实践任务的完成和培养目标的达成提供可行、有效的解决途径。

② 基于循序渐进、逐级递进的实践教学理念,设计了两项既相对独立又相互支撑的实践任务。软件工程的实践需要循序渐进地开展,能力和素质的培养是一个持续性的递进过程。本书基于此理念设计了两项实践任务,一项是分析和维护开源软件,旨在通过对高质量开源软件代码的阅读、分析、标注和维护,理解软件质量的重要性,掌握高质量软件设计和程序设计的方法和技能,培养良好的软件工程素质;另一项是开发有创意、上规模、高质量的软件系统,旨在综合运用软件工程的方法、技术和工具开展实践,开发出具有一定规模和复杂性的软件系统,在此过程中培养解决复杂工程问题的能力。前一项是基础性和先导性的实践任务,后一项是关键性和主体性的实践任务。

③ 采用依托社区、基于群智、迭代开发的方法,为实践实施提供有效技术手段。本书提出基于迭代的方式来逐步、渐进地开展软件工程实践,在此过程中充分利用开源社区中的群智知识来帮助实践人员解决多样化和个性化的开发问题,重用开源社区中的开源软件来促进有创意、上规模 and 高质量软件系统的开发。该方法的本质就是让开源社区中的高水平软件开发者成为指导实践开展的“老师”,海量的软件开发知识成为辅助实践问题解决的“教材”,高质量的开源软件成为推动软件开发的“资源”,开源社区成为实践实施的“教室”。

④ 结合具体的实践案例,翔实阐明实践实施的方法、策略和成果。本书提供了两个具体的实践应用案例,即小米便签开源软件和空巢老人智能看护系统。通过对这两个案例的深入剖析和示例,结合面向对象软件工程技术,详细介绍了开展这两项实践的具体过程、步骤、原则、方法、

策略及输出成果,阐明了如何基于群智的方法来开展实践,提供了相应的模型、文档、代码、标注、测试用例等软件制品和规范,供学习者参考和使用。

⑤ 借助支撑软件工具,利用开源社区,采用群体化软件开发技术来开展实践。本书提供了一组支撑软件工具以辅助实践实施,包括支持分布式协同开发的 Trustie-Forge、支持代码阅读和标注的 Trustie-CodePedia、支持开源软件和软件开发知识搜寻的 Trustie-Ossean、支持技术博客分享和交流讨论的 LearnerHub、支持代码质量分析的 SonarQube 等;推荐了一组开源社区以分享群智知识和重用开源软件,从而促进实践的开展,如 GitHub、Stack Overflow、CSDN、开源中国等;并采用群体化软件开发技术来加强实践任务协同和软件制品管理,包括基于 Issue 的任务分配和问题跟踪、基于 Pull Request 的分布式协同开发、基于 Git 的代码管理等。

⑥ 提出以“评”促“改”、定性和定量相结合、人工和自动相结合的考评方法。本书提供了软件工程课程实践的考评框架,强调要对实践行为和结果两方面同时进行考评,提出“评”为主“考”为辅、持续考评、以评促改等基本原则,以及定性和定量相结合、人工和自动相结合的科学考评方式,突出持续“点评”和“改进”在软件工程实践教学中的重要性,结合实践中常见问题和错误,列举了一系列点评示例供学习者参考。

本书读者对象

本书可作为计算机大类专业本科软件工程课程教学或实践课程教学的教材,也可作为软件工程师的参考用书。

对于教师而言,不仅可以从本书中学习到有关群智软件工程和群体化软件开发方法等方面的知识,掌握基于群智的软件工程实践教学方法,还可以获得有关软件工程课程实践任务设计、实施、点评和考查的具体方法、策略和手段。本书设计了两项软件工程课程实践任务供教师实例化和剪裁,以应用于实践教学;提供了支撑软件工程课程实践教学的配套软件工具、实践案例、软件模型、程序代码、软件文档及模板、考评手段和实践数据等,为其运用本书开展软件工程课程实践教学提供丰富的素材。

对于学生而言,不仅可以从本书中学习到有关面向对象软件工程和 UML、开源软件开发技术、群体化软件开发方法等主流和前沿的软件工程技术,更为重要的是,通过完成本书所设计的软件工程课程实践任务,可帮助他们运用面向对象软件开发技术并借助开源社区中的开源软件和软件开发知识,利用多项 CASE 工具完成具有一定规模和复杂性的软件系统开发,提升其软件开发能力,尤其是解决复杂工程问题的能力,培养良好的软件工程素质,如质量意识、严谨作风等。本书所提供的两个案例,一个来自工业界所开发的软件产品,另一个来自学生在实践教学中的具体成果,它们都为学生开展软件开发实践提供有效的参考和支持。

对于软件工程师而言,不仅可以通過本书掌握基于 Issue 的任务分配和跟踪、基于 Git 的版本管理、基于 Pull Request 的协同开发等现代软件开发技术,还可以通过实践案例讲解来掌握如何运用开源软件、借助软件开发知识问答社区来开发软件系统,解决软件开发中遇到的各种技术问题,以及如何在开源社区中开展软件开发活动、做出贡献。

下表描述了本书各章节对教师、学生和软件工程师而言需要关注的程度。“-”表示一般性关注，“*”表示选择性关注，“+”表示重点关注。

章节	教师	学生	软件工程师
第1章 绪论	+	+	-
第2章 软件工程基础	-	+	-
第3章 群智软件工程及其在实践教学中的应用	+	+	+
第4章 实践任务的设计与要求	+	+	*
第5章 实践支撑软件工具	+	+	*
第6章 实践任务一:分析和维护开源软件	+	+	+
第7章 实践任务二:开发软件系统	+	+	+
第8章 实践考评方法	+	+	*

编写分工与教学资源

全书共8章,毛新军撰写了第1、2、4、6、7、8章,王涛撰写了第5章,余跃撰写了第3章。

为便于高校师生教学,本书提供了丰富的教学资源,包括支撑实践教学的软件开发文档模板;完整的软件项目案例及其软件制品,如UML模型、软件文档、源程序代码、可运行软件系统、技术博客、实践总结材料等;电子教案PPT和教学视频;互联网上的学习社区;支撑软件工具等。

致谢

本书得益于国家重点研发计划项目“云计算和大数据开源社区生态系统”(编号2018YFB1004202)和国家自然科学基金NSFC重点项目“大规模在线协同学习的机理与方法研究”(编号61532004)的相关研究成果。

感谢国防科技大学王怀民老师和尹刚老师的研究团队开发了Trustie系列软件工具,它们为软件工程课程实践任务的开展提供了一系列支持;感谢我们的研究团队开发了LearnerHub工具(也称知士荟),它可为实践人员分享技术博客、开展交流和讨论、共享实践资源等提供支持。感谢王硕、张汤浩然两位本科生完整地设计和实现了本书的两个软件开发案例。最后还要由衷感谢清华大学刘强老师以及国内多位同行学者为本书提出了许多建设性的意见。

作 者

2019年5月

目 录

第 1 章 绪论	1	2.3 面向对象软件工程	31
1.1 背景	1	2.3.1 核心概念	31
1.1.1 计算机软件的变化	1	2.3.2 基本思想	32
1.1.2 软件工程的发展	6	2.3.3 技术特点	33
1.1.3 对软件工程专业教育和人才 培养提出的要求	9	2.4 统一建模语言 UML	34
1.2 软件工程课程的特点	11	2.4.1 UML 概述	34
1.3 软件工程课程实践教学 目标和要求	11	2.4.2 UML 的图	35
1.4 现行实践教学存在的问题	13	2.5 软件项目的组织方式	46
1.5 互联网技术和开源社区带来的 启发	15	2.5.1 结对模式	46
1.6 软件工程课程实践教学的 指导思想	16	2.5.2 团队模式	47
1.7 本书应用案例说明	18	2.5.3 社区模式	49
1.8 本书的内容组织	18	本章小结	50
本章小结	20	实践作业	51
实践作业	20	第 3 章 群智软件工程及其在实践 教学中的应用	52
第 2 章 软件工程基础	22	3.1 群智软件工程概述	52
2.1 软件工程概述	22	3.1.1 产生背景	53
2.1.1 软件工程的思想	23	3.1.2 核心概念和思想	54
2.1.2 软件工程的目标	24	3.2 群体化软件开发技术及其在 实践教学中的应用	56
2.2 软件开发过程模型与方法	25	3.2.1 基于 Issue 的任务管理	56
2.2.1 瀑布模型	25	3.2.2 基于 Git 的代码管理	59
2.2.2 原型模型	26	3.2.3 基于 Pull Request 的分布式 协同开发	64
2.2.3 增量模型	26	3.2.4 在实践教学中的应用群体化 软件开发技术	66
2.2.4 迭代模型	27	3.3 软件开发知识分享及其在 实践教学中的应用	69
2.2.5 螺旋模型	27	3.3.1 编程知识问答社区及 Stack Overflow	69
2.2.6 敏捷方法	28		
2.2.7 群体化开发方法	30		

3.3.2	技术资讯社区及 CSDN	71	5.2.2	编码实现工具	98
3.3.3	在实践教学中的应用软件开发知识	72	5.2.3	软件测试工具	98
3.4	开源软件及其在实践教学中的应用	75	5.2.4	代码质量分析工具	99
3.4.1	开源软件托管社区	75	5.2.5	软件文档撰写工具	101
3.4.2	开源软件项目资源	79	5.3	实践实施工具	101
3.4.3	基于开源软件的软件开发	79	5.3.1	实践任务管理和协同开发工具 Trustie-Forge	103
3.4.4	在实践教学中的应用开源软件	81	5.3.2	代码阅读和标注工具 Trustie-Codepedia	105
本章小结		81	5.3.3	群智资源检索工具 Trustie-Ossean	106
实践作业		82	5.3.4	群体化学习工具 LeamerHub	107
第 4 章 实践任务的设计与要求		84	5.3.5	软件开发实训工具 Trustie-EduCoder	109
4.1	实践教学的设计理念与指导思想	84	本章小结		111
4.1.1	以能力和素质培养为主要目标	84	实践作业		111
4.1.2	基于群智的实践教学方法	85	第 6 章 实践任务一:分析和维护开源软件		112
4.1.3	循序渐进逐层递进地开展实践	87	6.1	实践实施过程及原则	112
4.2	实践任务的整体设计	89	6.1.1	实施过程和活动	112
4.3	分析和维护开源软件实践任务的设计	90	6.1.2	实施原则和要求	115
4.3.1	实践内容	90	6.1.3	实践输出及成果	116
4.3.2	实践要求	91	6.2	实践实施的准备工作	117
4.4	开发软件系统实践任务的设计	92	6.2.1	选择开源软件	118
4.4.1	实践内容	92	6.2.2	组织实践人员	118
4.4.2	实践要求	92	6.2.3	布置实践任务和创建实践项目	118
本章小结		94	6.2.4	加载开源代码和运行开源软件	122
实践作业		94	6.2.5	访问和加入开源社区	127
第 5 章 实践支撑软件工具		96	6.3	实践案例介绍:小米便签开源软件	128
5.1	实践支撑工具概述	96	6.4	阅读开源代码	129
5.2	软件开发工具	97	6.4.1	泛读开源代码	129
5.2.1	软件建模工具	98	6.4.2	实践成果	134

6.5 分析代码质量	134	7.4.4 软件需求建模与分析	199
6.5.1 人工分析	135	7.4.5 软件需求文档化与评审	214
6.5.2 自动分析	136	7.4.6 迭代开发过程中的软件需求 变更管理	215
6.5.3 实践成果	138	小结	216
6.6 标注开源代码	138	7.5 软件设计与建模	217
6.6.1 理解代码语义	138	7.5.1 任务、过程与输出	217
6.6.2 标注代码	142	7.5.2 软件设计的策略和原则	220
6.6.3 实践成果	148	7.5.3 软件体系结构设计	223
6.7 维护开源软件	148	7.5.4 用户界面设计	238
6.7.1 纠正代码缺陷	149	7.5.5 用例设计	244
6.7.2 完善开源软件的功能	151	7.5.6 子系统 / 构件设计	250
6.7.3 演示维护后的开源软件	158	7.5.7 类设计	255
6.7.4 实践成果	160	7.5.8 数据设计	266
6.8 借助开源社区中的群智资源 开展实践	160	7.5.9 软件设计的整合、文档化及 评审	270
6.9 实践总结	163	小结	271
6.10 实践设计的剪裁	166	7.6 代码编写与测试	272
本章小结	169	7.6.1 任务、过程与输出	272
实践作业	169	7.6.2 编写代码	274
第7章 实践任务二: 开发软件系统	171	7.6.3 软件测试	282
7.1 实践实施过程及原则	171	7.6.4 程序调试和修复	292
7.1.1 实施过程和活动	171	7.6.5 部署和运行	293
7.1.2 实施原则和要求	174	小结	293
7.1.3 实践输出及成果	177	7.7 借助开源社区中的群智资源 开展实践	294
7.2 实践实施的准备工作	178	7.7.1 在软件开发知识分享社区中 寻找问题的解答	294
7.2.1 宣传和动员	178	7.7.2 在开源社区中与软件开发者 群体进行交互	297
7.2.2 布置实践任务	180	7.7.3 搜寻和重用开源软件	298
7.2.3 组织实践人员	181	7.8 实践总结	298
7.2.4 访问和加入开源社区	182	7.9 实践设计的剪裁	299
7.3 实践案例介绍: “空巢老人 智能看护系统”	182	本章小结	303
7.4 需求获取与分析	183	实践作业	304
7.4.1 任务、过程与输出	183		
7.4.2 实践要求与原则	184		
7.4.3 软件需求获取与构思	185		

第 8 章 实践考评方法	306	方法	313
8.1 实践考评的原则	306	8.4.1 考评内容	313
8.2 实践考评的手段	307	8.4.2 考评方法	314
8.3 分析和维护开源软件实践的 考评方法	308	8.4.3 持续点评	320
8.3.1 考评内容	308	8.5 实践实施及成效	324
8.3.2 考评方法	309	本章小结	326
8.3.3 持续点评	312	实践作业	326
8.4 开发软件系统实践的考评 方法	313	后记	328
8.4.1 考评内容	313	参考文献	331
8.4.2 考评方法	314		
8.4.3 持续点评	320		
8.5 实践实施及成效	324		
本章小结	326		
实践作业	326		
后记	328		
参考文献	331		

第1章 绪 论

当前信息技术发展迅速,计算机软件在信息系统中的地位和作用变得越来越重要,成为人类社会不可或缺的基础设施。受移动互联网平台、计算技术、计算机应用需求拓展等诸多因素的影响,当前软件系统的基本形态发生了深刻的变化,软件系统的规模和复杂性日益增长并呈现出新的特点,由此驱动软件工程技术和产业的快速发展。过去十年,无论是在学术界还是产业界,软件工程均取得了令人瞩目的进步。软件工程与人工智能、大数据、社会学、复杂性科学、经济学等诸多学科进行交叉,不断衍生出新的研究方向,产生了一系列新颖的软件工程方法、过程和平台,如群智软件工程、数据驱动软件工程、开源软件技术、软件众包、DevOps(英文 Development 和 Operations 的组合)、智能化软件开发、GitHub 等,软件开发的方式和理念也出现了一些新的变化,学术界与工业界间的联系与合作更为紧密。

在此背景下,如何加强教育和教学改革,提升软件工程专业人才的能力和素质,以适应产业和专业的快速发展等,成为高校和教师面临的一项重要挑战。这就需要我们清醒地认识当前计算机软件、软件工程等出现的变化和未来的发展趋势;深入地分析产业界对软件工程专业人才提出的新需求,尤其在能力和素质方面提出的要求;清晰地洞察当前软件工程专业建设、课程教学和人才培养存在的问题和局限,由此针对性地开展教育和教学改革。

本章作为全书的绪论部分,主要介绍相关的背景以及对软件工程课程实践教学的一些基本认识,具体内容包括:

- 近年来计算机软件形态和复杂性的变化和软件工程的发展,以及由此对软件工程专业教育和人才培养提出的要求。
- 软件工程课程教学的特点,其实践教学的目标和要求;当前软件工程课程实践教学的方法及其存在的问题和局限。
- 互联网技术和开源软件技术给软件工程课程实践教学带来的启示。
- 软件工程课程实践教学的方法学。

1.1 背 景

1.1.1 计算机软件的变化

在过去的十余年,受移动互联网、计算基础设施、应用领域、用户需求等多方面因素的影响,

计算机软件系统的运行环境、自身形态以及系统的复杂性等方面发生了深刻的变化,进而对支持软件系统开发的软件工程方法、过程和工具等提出了新的要求,也对从事软件开发的专业人才提出了新的要求。

1. 运行环境的变化

自 21 世纪以来,随着移动互联网技术、人工智能、大数据和云计算发展迅速,计算机软件的应用领域不断拓展,软件系统的驻留和运行环境发生了显著的变化。越来越多的软件系统部署和运行在基于(移动)互联网的计算平台之上。

不同于传统的计算环境和平台(如单机、局域网等),互联网环境具有开放、动态、异构、难控等特点。不仅如此,(移动)互联网还作为计算和网络基础设施将诸多的物理设备(如智能手机)与人类社会等紧密地连接和集成在一起,由此导致部署于其上的软件系统形态及复杂性也随之发生了变化。

自计算机诞生以来,计算机软件的部署和运行环境经历了主机、个人计算机、局域网计算环境、互联网和移动互联网计算环境的发展历程(见图 1.1),体现了计算环境从集中、封闭和单向发散、开放、动态、多样和异构的转变。

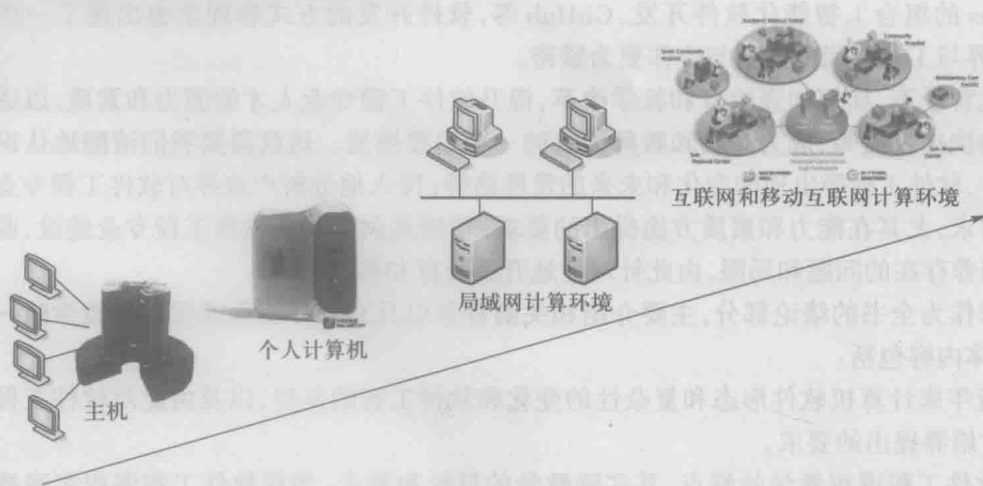


图 1.1 软件系统部署和运行环境的变化

在 20 世纪 90 年代之前,大部分软件系统被部署在同一个计算设施或局域网环境中的若干个计算平台之上(如主机或个人计算机),这些计算设施或平台通常由某些个体或组织来管理和控制。例如,在某个主机上部署软件系统,用户通过连接主机的多个终端来访问和操作软件,或者软件被部署在由局域网连接而成的若干台计算机上,这些软件通过局域网相互交互。它们的部署环境通常是封闭的,不和外界进行交互;环境边界也是明确的,哪些属于环境哪些不属于环境具有清晰的界定。

进入 20 世纪 90 年代,尤其是进入 21 世纪以来,越来越多的软件系统被部署在基于(移动)互联网的多个不同计算设施之上(甚至包括各种嵌入式设备和移动终端),这些计算设施可能属

于不同的组织和机构(如企业、个人、政府等),具有地理和逻辑上分布、高度自治、独立和发散管理、难以全局控制等方面的特点。互联网尤其是移动互联网是一个动态、难控的复杂环境,不断有新的计算节点、服务和计算资源等加入互联网中来,或者已有的计算节点、服务和计算资源等会离开互联网环境,其动态变化呈现出不确定、不可控和不可预测等诸多特点。

2. 软件形态的变化

我们正处在一个“软件定义世界”时代,软件成为社会的一项重要基础设施,渗透到人类社会的方方面面。软件系统使得万物皆可互联(如数据和服务共享),也使得一切均可编程。由于软件的“社会化和普及化”以及软件驻留环境的变化,软件系统自身的基本形态也在发生深刻的变化。

(1) 越来越多的软件系统表现为人-机-物共生系统而非纯粹的技术系统

传统软件系统通常表现为纯粹的技术系统,即软件系统仅仅作作为技术要素而存在,其主要任务是完成计算、提供功能和服务。虽然人与软件系统需要进行交互,如输入软件所需的信息或接受软件输出的信息,但是人和软件系统的边界非常清晰,在业务实现过程中哪些工作由人来完成、哪些工作由计算机软件来完成等有明确的界定。

当前越来越多的软件系统表现为一类由人、社会组织、物理设备、过程等要素共同组成和相互作用的人-机-物共生系统(见图1.2),也称之为信息物理社会系统。软件系统不仅提供各种功能和服务,还连接了大量物理设备(如机器人、手机、传感器等),并通过它将不同的人或机构等组织在一起,以实现信息的交流和共享,典型的例子包括微信、QQ、大众点评等。物理设备不仅包括传统的计算设备,还包含了诸多的物理设施。过程要素定义了软件的用户期望如何来操作软件系统。与此同时,软件系统的行为和服务等还受人-机-物共生系统中的社会法规(rule)、制度(norm)等的影响和限制。

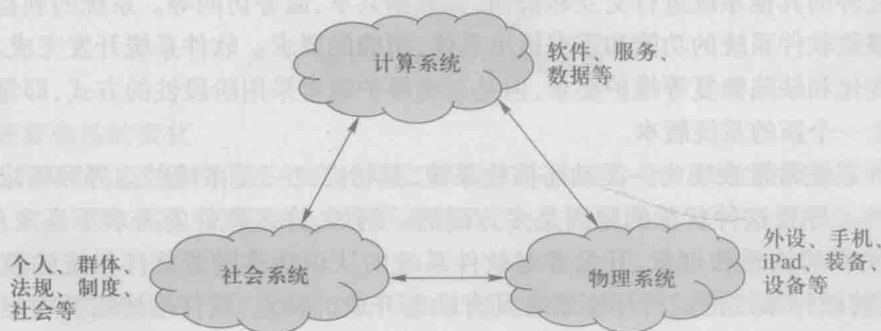


图 1.2 人-机-物共生系统示意图

在人-机-物共生系统中,物理要素、社会要素和技术要素共同存在并且相互作用。系统中的技术要素不能独立于社会要素和物理要素而存在,社会要素和物理要素的变化会引起技术要素的变化。人不仅是系统的使用者,也是系统的组成部分;物理系统和技术系统作为中介实现人与人之间的关联、交互和协同,使得他们构成结构化的组织。典型的例子如社交媒体(如微信)、智能机器人、物流系统等。