

基于 GitHub 的高职学生协作编程能力培养模式构建与实证

王尧山

成都工业职业技术学院, 四川 成都 610218

摘要: 高职信息技术专业的课堂上, 学校常用的教学平台往往跟不上企业里真正在用的开发工具。成都工业职业技术学院信息工程学院有个“开源软件融入职业教育项目化教学”教改项目, 这项研究就依托它展开。我们把 GitHub 引进了《嵌入式系统》和《物联网系统集成与应用》两门课, 学生得把 Fork、提 Issue、发 Pull Request、做代码 Review 这一整套流程完整走一遍。一个学期教下来, 86.7% 的学生能独立完成协作开发的全流程; 课程项目的代码规范得分比改革前提高了约 30%; 参加过开源实践的学生, 毕业后的平均起薪比未参与的对照组高出约 8%。当然, 后面要推广的话, 还得结合各校实际, 把师资投入和学生基础之间的匹配问题处理好。

关键词: 开源软件; GitHub; 协作编程; 项目化教学; 职业教育

DOI: 10.64649/yh.shygl.issn3105-0085.202609005

0 引言

现在不管是做系统开发还是做信息技术相关工作, 基本都绕不开开源软件。安永发布的《2025 中国 ICT 人才发展白皮书》里有预测数据, 国内 ICT 领域的人才缺口总量会突破 2000 万^[1]; 工信部印发的《“十四五”软件和信息技术服务业发展规划》给出的数据更直观, 到 2025 年国内开源软件相关岗位的人才需求规模是 85 万, 当前能匹配岗位要求的有效供给量只有 35 万, 缺口占比超过 58%^[2]。教育部在 2025 年 12 月更新了《职业教育专业目录》, 第一次把专科层次的“开源技术应用”、本科层次的“开源技术开发与应用”设为独立的专业方向^[3]。

但实际的人才培养端还存在不少问题。高职物联网专业当前在用的教学平台, 基本都是围绕 Windows 系统和商业闭源软件搭建的, 像 Linux、OpenHarmony 这类开源系统, 很少被纳入日常课堂教学内容^[4]; Git 这类协作工具在课堂上几乎没有适用场景^[5-6]。开展摸底调查能发现, 绝大多数学生从来没提交过 Pull Request; 可只要给学生安排成体系的结构化训练, 他们的协作配合效率、写出的代码质量都会有较为明显的提升。GitHub 平台通用的 Fork → Issue → PR → Review 协作流程, 在企业实际工作中早就成了默认的工作方式。

此前已有两轮小范围的教学尝试, 其中一轮是在《物联网系统集成与应用》课程里试行 GitFlow 工作流教学, 学生的代码规范性评分比试行前提高了约 30%; 另一轮是在《嵌入式系统》课程里, 覆盖 2 个班共 80 名左右学生, 期末项目考核的优秀率比之前提升了约 15 个百分点。但都局限在单门课程、单个学期的零散尝试层面, 没能形成连贯的培养体系。本研究把研究问题梳理为: 怎样把零散的工具使用教学、单个项目训练内容整合起来, 搭建出一套可检验

效果的协作编程能力培养模式。

1 基于 GitHub 的“三阶递进”协作编程能力培养模式

1.1 理论依据

本次开展研究设计教学路径设计时, 参考了较多学习相关的理论, 以及项目化教学的基本思路。建构主义提供的是基本出发点, 该理论提出知识不是靠老师单方面灌输给学生的, 是学习者处在真实的场景里, 彼此互相协作共同建构的; GitHub 就是接近实际工程开发的场景。项目化教学对应的是“学什么”的问题, 即依托一个完整的项目, Git 仓库既是存放最终成果的容器, 也是存储整个学习过程痕迹的资料库。

情境学习理论和认知学徒制对应的是“怎么带学生学”的问题, 前者把学习看成是学习者从圈子外围慢慢融入实际实践群体的过程, 这个路径和高职学生从新手到熟手、再到准职业人的成长轨迹基本吻合; 后者给出了具体的带教步骤顺序, 这种“先示范、再帮扶、最后放手”的节奏其实较难把握, 放手太早的话, 学生很容易卡在最开始的环境配置步骤无法推进, 放手太晚的话, 学生又会养成等着老师下指令的习惯。

研究中将协作编程需要的能力拆分为三层, 第一层是工具操作层面, 需要掌握 Git 的核心操作命令; 第二层是项目协作层面, 需要能在团队项目里做好分支管理、参与代码评审和集成工作; 第三层是社区贡献层面, 需要能为真实的开源项目提交有用的内容, 大致和“会用工具、完成项目、融入社区”的阶段对应。

1.2 三阶培养路径设计

第一阶段的目标设置较为务实, 先是帮助

学生打消对命令行的抵触情绪,《嵌入式系统》课程里专门加入了Git基础内容模块,前期先让学生用GitKraken这类带图形界面的客户端上手,等大家熟悉了“提交—推送—拉取”这一套基础流程,再逐步过渡到终端操作。

这一阶段设置了两条硬性要求,一是每个人都都要建立自己的个人代码仓库,整个学期下来有效提交次数不能少于20次;二是每次提交的说明内容,要按照通用的提交规范来写。后面这条要求刚推行的时候招来不少抱怨。课程里还搭配了Markdown写作练习,项目说明文档和代码合并申请的内容,都要求用这种格式完成。

第二阶段的培养方向,是把学生从“单独一个人写代码”的状态,引导到“多人协作共同完成项目”的模式,这部分内容依托《物联网系统集成与应用》课程开展,教师会把学生按4到5人的规模编成小组,组长负责管理代码仓库的访问权限,还有合并代码申请的工作,剩下的组员分别认领不同的功能模块来开发。

整个课程借助GitHub Classroom搭建了一套完整的操作流程:首先教师在平台发布作业仓库,学生先把仓库Fork到自己的账号下,接着新建feature分支编写作业对应的功能,写完提交PR也就是合并请求,先交给同组同伴互相检查,最后由组长把通过审核的代码合并到项目主干中;一个学期里每个小组至少要完成3轮这样的完整迭代,每一轮迭代都要走完提Issue反馈问题、在对应分支写代码实现功能、提交PR后同伴检查评审这三个环节,不能缺漏。

到了第三阶段,学生不能只关注课程内的作业,要走出课程范围,给真实存在的开源项目提交一次代码;对应的开源项目由各小组自行挑选,挑选的标准较为简单,就是有人维护、文档齐全;具体的贡献形式没有严格限制,可以是修复项目里的程序漏洞,唯一的硬性要求是提交的内容要被项目维护者真正接纳才算完成。

课程还设置了开源贡献积分制度,积分数值会折算计入学生的平时成绩,积分的具体核算标准是:成功提交代码算0.5分,提交有效Issue反馈算0.2分;另外课程还和成都国星通信有限公司共同建设了“开源项目实践基地”。

1.3 基于GitHub Classroom的教学管理

这门课的教学管理,主要靠GitHub Classroom来支撑。GitHub Actions那边也做了调整。学生只要往仓库推新代码,流程就会自动跑起来,几十秒内把检测结果返回来。以前是课后收作业、批改完等下次上课再讲,现在这个反馈周期被压到了几分钟。

课程考核规则的调整幅度同样较大,之前的课程成绩基本由期末提交的项目决定,学生平时在课程上投入的精力完全无法体现,调整

后的成绩共分为三个部分:过程性数据占总成绩的40%,这部分分数由平台自动采集统计,采集内容包括学生提交代码的频率,发起、响应合并申请的数量,参与代码评审的投入程度;成果质量占总成绩的40%,这部分由任课教师对照评分标准逐项打分;剩下20%的成绩,由同组学生匿名互评给出。

2 教学实践与效果分析

2.1 实践设计

本次实践开展时间定在2025到2026学年的第一学期,参与实践实验的是该校物联网技术应用专业2405班、2406班的全体学生,共计82人,作为实验组;对照组为同专业2305班、2306班的学生,总人数76人,两组教学采用同一份教学大纲。

本次研究用到的数据一共分三类:第一类是GitHub平台自动记录生成的学生协作行为日志;第二类是两份协作编程能力自评问卷;第三类是课程综合项目的最终成果评分,该部分分数由三位授课教师按照同一套评分标准各自独立打分。

研究过程中还补充了访谈环节;学期末,研究者从实验组的高、中、低三个成绩层次里各选出4名学生,开展半结构化访谈。学生提到的实际问题,比问卷回收得到的量化数据包含更多信息。

2.2 结果分析

在我们此次的实验结果中。有82名学生的实验组里,71个人能独立完成“Fork→Branch→PR→Merge”整套操作,占比86.7%。平均每人提交commit达到了47次,发起PR6.2个,参与PR Review4.8次。作为对比,对照组中此前在课外接触过版本控制工具的学生,只占18.4%。

课程结束后,共有7个小组主动将其项目仓库设置为公开状态,其中2个仓库被其他班级的学生Fork后继续修改完善,虽然涉及的数量不算多,但也能看出有部分学生已经开始将“写出能给别人用的代码”视为很自然的事。

实验组学生在“团队沟通能力”“任务分工与进度管理能力”“代码审查意识与质量敏感度”这三项上的自评提升幅度,分别为28.6%、31.2%和35.7%,有82.3%的实验组学生表示,对理解企业中真实的软件开发流程“有实质性帮助”。

有学生提到,自己写的代码第一次被同学公开点评的时候,“既紧张又期待”;也有学生表示,翻看其他同学的PR讨论内容,比单纯看教材学习的效果要更好。

2.3 教学反思

经过一学期的教学推进,各环节暴露出来的问题还较多,最直观的就是工具使用的门槛问题,约有15%的学生碰到命令行相关的操作就明显畏难,私下交流的时候会说“看到那个黑色的命令窗口就心里发慌”。

遇到这种情况之后调整了要求,允许这部分学生多花一段时间用GitHub Desktop、GitKraken这类带图形界面的工具完成操作,这部分学生的畏难状况才慢慢有所缓解。小组协作里成员贡献不对等的问题也较为明显,部分小组里贡献最少的成员,提交代码的commit数还不到全组平均值的30%。

平时收集的过程数据可以发现这些不对等的异常情况,可如果只按提交数量来算分数,反而会留出刻意“刷提交次数”的空子,所以后续考虑引入按照实际贡献占比加权的个人成绩核算方式。过程性的考核指标不能只盯着提交数量,要从原先的“数量优先”转向兼顾内容质量、提交意图、实际影响的方向;如果班级总人数超过45人,授课教师很难做到对每一个PR都做细致的检查点评,比较可行的应对方法是试点推行“高年级学生协作担任助教”的做法。

3 结论与展望

本次研究以“高职学生如何真正掌握协作编程方法”为核心方向,搭建了“工具操作—项目协作—社区贡献”三层逐步推进的培养模式,并在物联网技术应用专业的核心课程里完成了一整轮教学实践。

参考文献:

- [1] 安永(中国)企业咨询有限公司,华为.中国ICT人才生态白皮书[R].深圳:华为技术有限公司,2025.
- [2] 工业和信息化部.“十四五”软件和信息技术服务业发展规划:工信部规〔2021〕180号[A/OL].(2021-11-30)[2026-09-17].https://www.miit.gov.cn/jgsj/xxjsfzs/zlgh/art/2021/art_90ad7abe94444628b2efd0d2fcee4224.html.
- [3] 中华人民共和国教育部.职业教育专业目录(2025年)[A/OL].(2025-12-15)[2026-09-17].<http://www.moe.gov.cn/>.
- [4] 王莹.基于OpenHarmony的智慧农业混合式教学创新研究[J].南方农机,2025(18):156-159.
- [5] HERLAMBA A D, RACHMADI A, WIJOYO S H. Git and GitHub application training program to support vocational high school students in collaborative computer programming learning[J]. JPPM (Jurnal Pendidikan dan Pemberdayaan Masyarakat), 2023, 10(1): 1-12.
- [6] 李静,张伟.基于GitHub开源平台的软件开发类课程实践教学探索[J].科教导刊,2023(16):112-115.

作者简介:王尧山(1992.11—),男,汉族,四川平昌人,硕士研究生,研究方向:职业教育信息化、嵌入式软件方法与技术。

项目信息:成都工业职业技术学院2026-2028年职业教育人才培养和教育教学改革研究项目(项目编号:JGXM2026-26)。

第一点,仅靠抽象讲解版本控制相关的原理,能取得的效果十分有限,要是先让学生把相关工具用熟、把代码提交练成本能习惯,他们在后续协作环节的表现就会有明显提升。

第二点,学习过程的全程可见,让教师的介入指导方式发生了改变,现在能在学习推进的过程里及时发现异常情况,提前介入疏导问题。

第三点,研究将参与开源贡献的情况折算成课程评价的组成部分,当学生知道自己写的代码有可能被校外的人看到时,原本“只是完成任务”的心态,就会有一部分转成“要做点真正有用的内容”。

后续研究工作打算沿着三个方向继续推进,第一个方向是检验这套模式的适用迁移性,目前模式只在两门硬件与系统类的课程里试过效果,程序设计基础、Web全栈开发这类课程的知识内容构成差别比较大,一模一样的三层推进路径能不能行得通,还需要再做一轮对照实验来验证。

第二个方向是把平台积累的行为数据充分利用起来;后续打算结合学习分析的相关方法,从这些日志数据里梳理出学生协作主动性、代码质量敏感度、社区融入程度等方面的特征,让教学干预能更对准学生的实际问题,不用全靠教师凭过往经验做判断。

第三个方向是把校企合作的内容做扎实,下一步要把企业提出的真实项目需求,更规范地转化成可以用于教学的项目内容,搭起从需求引入教学,到学生成果反哺企业的完整链条。