College and Job 职业教育发展, 2026, 15(1), 322-334
Published Online January 2026 in Hans. https://www.hanspub.org/journal/ve
https://doi.org/10.12677/ve.2026.151044

Hans汉斯

高本贯通人才培养模式下软件工程专业课程体系一体化设计研究

——以上海农林职业技术学院软件工程高本贯通专业为例

李　嘉

上海农林职业技术学院智慧农业工程系，上海

收稿日期：2025年10月14日；录用日期：2026年1月5日；发布日期：2026年1月13日

摘　要

在“高本贯通”政策持续扩容、课程衔接成为瓶颈的背景下，本文基于上海农林职业技术学院－上海建桥学院“3＋2”软件工程高本贯通试点，系统研究了“纵向贯通、横向拓宽”的一体化课程体系构建与量化评估方法。研究首先构建“三阶五层”能力递进模型，将岗位能力－技术能力－工程能力逐层分解为十个学期的阶梯式成长路径；据此设计“三段阶梯 ＋ 两次接口”课程地图，并以“企业级JavaEE开发”等接口课程实现螺旋上升。创新“岗位导向、螺旋上升、学分置换”一体化模型：以Web前端、移动开发、服务端测试三大岗位链为纵轴，设置纵向接口课程与横向拓展包，通过1＋X证书“能力等效”认证抵换学分。研究为软件工程高本贯通提供了可量化、可复制、可推广的课程一体化范式。

关键词

高本贯通，软件工程，课程一体化

Research on the Integrated Design of the Curriculum System for Software Engineering under the Higher Vocational-Undergraduate Integrated Talent Cultivation Model

—A Case Study of the Higher Vocational-Undergraduate Integrated Software Engineering Program in Shanghai Vocational and Technical College of Agriculture and Forestry

Jia Li

文章引用: 李嘉. 高本贯通人才培养模式下软件工程专业课程体系一体化设计研究[J]. 职业教育发展, 2026, 15(1): 322-334. DOI: 10.12677/ve.2026.151044

Department of Smart Agricultural Engineering, Shanghai Vocational College of Agriculture and Forestry, Shanghai

Received: October 14, 2025; accepted: January 5, 2026; published: January 13, 2026

Abstract

Against the backdrop where the Higher Vocational-Undergraduate Integration policy continues to expand, while curriculum articulation has become a bottleneck, this study is based on the "3 + 2" pilot program of Higher Vocational-Undergraduate Integration in Software Engineering jointly carried out by Shanghai Vocational and Technical College of Agriculture and Forestry and Shanghai Jianqiao University. It systematically explores the construction of an integrated curriculum system featuring "vertical integration and horizontal expansion" and its quantitative evaluation methods. First, the study constructs a "three-stage and five-level" competency progression model, decomposing positional competency-technical competency-engineering competency layer by layer into a phased growth path spanning ten semesters. Based on this model, a curriculum map with "three-stage ladder + two interface points" is designed, and interface courses such as "Enterprise-Level JavaEE Development" are used to achieve spiral progression of learning. The study innovatively proposes an integrated model characterized by "position-oriented, spiral progression, and credit transfer": taking three positional chains (Web front-end, mobile development, and server-side testing) as the vertical axis, it sets up vertical interface courses and horizontal expansion modules. Credits are exempted through the "competency equivalence" certification of the 1 + X certificate system. This research provides a quantifiable, replicable, and promotable integrated curriculum paradigm for the Higher Vocational-Undergraduate Integration program in Software Engineering.

Keywords

Higher Vocational-Undergraduate Integration, Software Engineering, Curriculum Integration

1. 引言

在教育部 2023 至 2025 年度的重点工作部署中，“高本贯通”与“职教本科”连续三个年度均被纳入政策范畴。2023 年提出“扩大高职本科贯通培养规模”[1]；2024 年强调“支持在培养周期长、技能要求高的专业领域实施长学制培养”[2]；至 2025 年则进一步明确要求“建立横向融通、纵向贯通的现代职业教育体系，形成中职、高职专科与本科一体化衔接的人才培养方案”[3]。2023 年，全国范围内新增高本贯通点 204 个，课程衔接问题成为关键制约因素。此前，上海市教育委员会已于 2022 年明确要求“高本贯通”专业应实现本科与高职专科专业的逐一对应，并推行五年一体化课程设置；2023 年，上海市新增 8 个此类试点专业，2024 年试点规模扩大到 25 个专业点，覆盖电子信息、装备制造、交通运输等紧缺领域。

现有关于高本贯通人才培养模式的研究多聚焦于制度层面的分析，针对微观课程一体化设计的实证研究较为缺乏。高等职业教育与本科教育衔接过程中存在的课程重复与断层现象亟待系统性解决。如何有效构建“纵向贯通、横向拓宽”的一体化课程体系模型，并对其实际实施效能进行科学评估，构成当前核心的研究议题。上海农林职业技术学院与上海建桥学院联合开展的软件工程专业高本贯通“3 + 2”

培养模式，于 2020 年启动试点。2020 年至 2025 年间，该专业累计完成五届招生，并已有一届毕业生。在此期间，针对高本贯通人才培养模式下的软件工程专业课程体系一体化设计研究持续深入进行。

2. 研究现状

2.1. 高本贯通课程衔接研究脉络

在国内外学术界，高本贯通课程衔接问题已从政策、理论和实践三个维度进行了深入探讨。研究主要聚焦于课程内容的整合、学分互认以及教学资源的共建共享。

现有研究指出，课程衔接过程中存在的重复与断层问题较为显著[4]，因此构建一个系统化的衔接机制显得尤为必要。上海市的试点项目经验显示，一体化的课程设计能够显著提升人才培养的质量，但课程结构与教学策略的进一步优化仍为必要。衔接机制的核心在于消除学科间的界限，促进知识体系的有机整合。通过模块化的课程设计，可以确保教学内容在各个阶段的递进性和互补性，从而避免内容的重复和断层现象。同时，加强实践教学环节对于提高学生的综合应用能力至关重要，这将为学生的职业发展打下坚实的基础。衔接机制的进一步优化需要借助大数据分析技术，以精确识别课程衔接中的关键问题，并据此动态调整教学内容，确保课程体系的科学性和实用性。

通过跨校合作，共享优质教学资源，可以有效提升整体教育水平，并形成良性循环。实证研究的深入是验证衔接机制有效性的关键，这将为在全国范围内推广提供有力的证据支持。通过案例分析，揭示成功经验与存在的问题，并提出相应的改进策略，可以推动高本贯通教育模式的持续完善。

2.2. 软件工程专业“能力递进”课程地图研究

2021 年《关于推动现代职业教育高质量发展的意见》明确提出“建立各层次专业设置、培养目标、课程体系衔接机制”[5]，强调能力递进应成为软件工程专业课程设计的主线。2023 年上海市教委试点文件进一步细化要求：高本贯通软件工程专业需遵循“前三年重技术、后两年重工程”的培养路径，实现“岗位能力－技术能力－工程能力”的三阶递进目标[6]。

基于上述政策导向，本研究构建高本贯通软件工程专业“三阶五层”能力递进模型框架。三阶能力维度包含：岗位能力、技术能力与工程能力，三者形成逐层递进、有机衔接的人才培养体系，支撑学生从专业入门到企业级人才的全周期发展。五层学习阶段按学期划分为：专业认知(入门)、行业认知(入行)、职业基础(入职)、岗位技能(入岗)、企业实践(入企)，系统提升学生综合实践能力。能力与时间轴精准匹配：各学期聚焦特定能力层级，强化理论与实践融合，通过十个学期的阶梯式培养实现从基础技能到企业级工程师的全面转型。模型核心价值在于：通过系统化培养路径设计，实现个体能力成长与行业人才需求的无缝对接，培养高度适配的软件专业人才。模型框架如图 1 所示。

Figure 1. Framework diagram of the “Three-Stage and Five-Level” competency progression model
图 1. “三阶五层”能力递进模型框架图

基于该模型框架，制定专业能力递进指标体系，如表 1 所示。明确各阶段能力考核标准，确保学生能力逐级递进。依据行业发展趋势动态更新教学内容，实现课程体系与市场需求的同步优化，有效提升学生实践应用能力。通过校企协同机制引入企业真实项目案例，强化实践教学环节，确保学生掌握前沿技术应用能力。建立周期性教学评估机制，持续优化课程结构，形成闭环式动态调整机制，保障人才培养质量。

Table 1. Correspondence between competency stages and training requirements
表 1. 能力阶段与培养要求对应

能力阶段	Bloom 认知层级	证书要求	课程示例	实践学时占比
入门	记忆/理解	信息技术一级	程序设计基础	≥50%
入行	应用	前端中级	HTML + CSS 网页制作	≥55%
入职	分析/评价	二级 Java	JavaEE 架构与程序设计	≥60%
入岗	创造	1 + X 高级	企业级 JavaEE 开发	≥65%
入企	综合创新	毕设 + 实习	毕业设计、顶岗实习	100%

Table 2. Core curriculum map
表 2. 核心课程地图

学期	主要课程(含实践学时)	能力焦点	备注
1	程序设计基础(32) 计算机导论 1 (24)	C 语法/算法/专业认知	与本科阶段“高级编程”接口
2	HTML + CSS 网页制作(32) 面向对象程序设计 1 (32) 数据结构 (32) 计算机导论 2 (16)	Web 前端入门/Java 面向对象/数据结构	前端方向第一条主线
3	数据库原理及应用(32) 面向对象程序设计 2 (32) 计算机网络(24) 软件工程导论(32)	数据库设计/Java 进阶/网络基础/生命周期	与本科“企业级 JavaEE”纵向衔接
4	JSP 程序设计(32) Linux 操作系统(32) 算法设计与分析(16)	动态 Web/系统管理/算法优化	与本科“金融 IT 项目实战”横向拓宽
5	Java 高级应用编程(32) 移动互联网软件开发(32) 数据库设计与开发(32)	移动开发/持久层框架/综合实训	移动方向第二条主线
6	企业级 JavaEE 开发与实践(48) 软件测试技术(32) 智能应用项目实战(44)	微服务/测试工具/AI 场景应用	本科阶段起点；1 + X 中级证书对接
7	金融 IT 项目开发实战(64) 数据分析与应用 Python(48)	行业项目/数据科学	拓展选修包①
8	移动软件开发实战(48) Web 前端开发(32) 人工智能导论(16)	前端框架/移动混合开发/AI 基础	拓展选修包②；1 + X 高级证书对接
9	软工专业综合设计(64) 项目开发实战(32)	综合项目/毕业设计选题	企业真题；可替换毕业设计
10	毕业实习(192) 毕业设计(128)	岗位能力/论文成果	与企业共建基地完成

依据学期递进原则构建高本贯通软件工程专业核心课程地图，详见表 2。总体设计采用“三段阶梯 + 两次接口”的递进模式。第一阶段：技术奠基(第 1~3 学期，48 学分)。通过“程序设计语法→算法基础→面向对象编程→数据库技术→计算机网络”的递进式课程序列，使学生达到《本科高级编程》与《数据结构》课程约 70%的水平，成功将本科阶段难度最高的两门核心课程下沉至专科阶段实施。第二阶段：企业刚需技术栈(第 4~6 学期，60 学分)。课程体系深度聚焦企业级开发核心组件“Java + Web 开发 + Linux 系统 + 数据库技术”。纵向深化方面：将 JavaEE 微服务框架(含 Spring 系列框架)提前至专科第 6 学期

开设，与本科阶段的《企业级开发》课程形成内容重叠，实现“课程层叠”而非简单重复。横向拓展方面：同步引入金融科技(Fintech)、软件测试及人工智能应用场景，构建“业务领域 + 技术栈”双轮驱动模式，为本科阶段的《金融 IT 项目实战》课程提供真实需求储备。第三阶段：领域纵深与毕业出口(第7~10 学期，课程学分 64 分 + 实习学分 192 分)。通过选修模块①“金融数据分析”与选修模块②“移动混合开发”构建可替换路径，实现“同批学生、双轨并行”的行业出口策略。第 9 学期开设的“综合设计”项目直接对标本科毕业设计标准，允许采用企业真实项目替代传统论文，提前完成本科毕业设计约50%的工作量。第 10 学期安排企业驻场实习，该实习包含试用期性质。本科阶段学生可专注于毕业论文的深化研究与技术创新。

2.3. 软件工程专业“能力递进”课程地图研究

目前，关于软件工程专业课程地图的研究表明，课程体系普遍缺乏“任务–能力–课程”三阶映射关系的量化实证依据，具体体现在以下四个方面：

(1) 尚未建立“任务”维度的数据采集与量化标准体系。现有研究未能提供企业任务清单、任务复杂度分级标准、任务完成度评估指标等量化信息；同时，缺乏对“任务”本身的结构化编码体系，例如任务类型(开发/测试/运维)、任务规模(人天/功能点)、技术栈覆盖率、业务领域分布等核心维度，导致难以构建“任务”与“能力”之间的有效映射关系。

(2) 未能量化“能力”指标与任务需求的对应关系。尽管现有框架罗列了“能力阶段”与“Bloom 认知层级”、“证书要求”、“课程示例”的对应关系，但所有指标均局限于教育体系内部维度(如认知层级、证书、课程)，未引入任何外部企业任务维度。尤其缺乏建立“完成特定真实任务所需能力达标阈值”的量化模型，例如：完成“金融 IT 项目开发实战”中的“账户风控模块”需要达到何种具体的 JavaEE 能力分值；完成“移动混合开发”任务需要何种等级的前端框架能力。此外，缺少任务驱动的能力评估实证数据，如学生在企业真实任务中的能力表现指标(代码质量、缺陷率、交付准时率)与其课程成绩之间的关联性分析。

(3) 未能实现“课程”与“任务”映射的实证验证。尽管现有课程地图内容详实，但所有课程设计均以“知识/技术点”为核心逻辑(如“JavaEE 架构”、“Spring 框架”、“数据结构”)，未能明确标识每门课程所对应的企业任务类型及其任务覆盖率。缺乏课程–任务映射矩阵，例如：《企业级 JavaEE 开发与实践》课程覆盖了多少个真实企业任务，这些任务分别属于哪些业务领域(金融/电商/物流)，课程项目与企业实际任务之间的相似性量化指标(如功能点重合率、技术栈重合度)亦未明确。同时，缺少课程任务映射有效性的验证数据，例如学生课程项目表现与其后续在企业任务中表现的相关性分析(如课程成绩与实习任务评分的对比)。

(4) 尚未构建“任务–能力–课程”三阶映射的闭环评估机制。现行的“周期性教学评估机制”其评估维度过度集中于教育体系内部指标(如课程成绩、证书通过率、实践学时占比)，未纳入任务完成质量、企业导师评价、任务能力成长曲线等外部量化数据。缺乏映射关系有效性的实证证据，例如：修读完“金融数据分析”选修模块的学生，是否在金融企业任务中表现出显著更高的数据处理能力，未选修“移动混合开发”模块的学生，在移动开发任务中的缺陷率是否显著更高。此外，尚未建立基于“任务失败案例”的溯源机制，例如学生在企业实习中遭遇任务失败(如项目延期、代码重构需求)，其根源是否可追溯至特定课程模块的设计不足(如未能覆盖关键任务所需能力)。

3. 现状诊断

3.1. 课程重复率量化

聚焦“专业课程”(含专业基础课、专业核心课、专业拓展选修课、实践教学环节)，剔除公共基础课

程，对高职阶段(第 1~6 学期)、本科阶段(第 7~10 学期)进行专业课程重复率量化分析。高本贯通软件工程专业在高职与本科阶段之间的课程重复率控制在 10%左右，属于合理范围，体现出“分层培养、螺旋上升”的设计思路。重复课程主要为专业核心课和实践环节，内容深度和项目复杂度在本科阶段有明显提升，符合高本贯通人才培养目标。

3.2. 能力断点分析

识别学生在高职→本科过渡阶段可能存在的知识、技能、能力断层，为课程衔接、教学补偿、能力测评提供依据。分析维度与方法如表 3 所示。根据分析结果生成主要能力断点清单，如表 4 所示。

Table 3. Analysis of curriculum evaluation dimensions and methods
表 3. 课程评估维度与方法分析

维度	方法
课程目标	比对课程教学目标是否递进
技能深度	是否从“会用”→“会设计/优化”
项目规模	是否从小型实验→企业级项目
工具链	是否从基础工具→工程化工具链
能力要求	是否从“模仿”→“创新/架构”

Table 4. Comparison of competency requirements between higher vocational and undergraduate stages
表 4. 高职与本科阶段能力要求对比

能力项	高职阶段表现	本科阶段要求	断点等级	典型课程
系统架构能力	掌握单模块开发(如 JSP + Servlet)	要求掌握多层架构、微服务、Spring Cloud	高	企业级 JavaEE 开发
数据库设计能力	掌握基础 SQL、单表操作	要求掌握高并发、索引优化、分布式事务	高	数据库设计与开发
项目工程化能力	无版本控制、手动部署	要求掌握 Git、Maven、Jenkins、Docker	中	软件工程导论、项目开发实战
测试能力	掌握手工测试、黑盒测试	要求掌握自动化测试、性能测试、测试驱动开发	中	软件测试技术
前端工程化能力	掌握 HTML/CSS/JS 基础	要求掌握 Vue/React、Webpack、ES6 模块化	中	Web 前端开发
文档与需求能力	能编写简单功能说明	要求掌握需求建模、UML、接口文档规范	低	软件工程导论
文档与需求能力	能编写简单功能说明	要求掌握需求建模、UML、接口文档规范	低	软件工程导论
团队协作能力	个人开发为主	要求掌握敏捷开发、Scrum、代码评审	低	项目开发实战
性能调优能力	无性能意识	要求掌握 JVM 调优、SQL 优化、缓存机制	高	Java 高级应用编程

对能力断点清单进行梳理，重点阐述以下三个核心能力断点：

(1) 系统架构能力断点

高职阶段能力基线：基于 JSP + JDBC + Servlet 技术栈实现单一功能模块。本科阶段能力要求：掌握 Spring Boot + MyBatis + Redis + Gateway 的微服务架构设计与实现。断点具体表现：对“分层”、“解耦”、“服务拆分”等核心架构理念缺乏理解；无法配置 Spring 容器及实现事务管理；欠缺接口设计、

服务注册与发现等关键技术能力。建议开设桥梁课程《Java 架构基础》(2 学分)，采用案例教学法，引导学生理解从“单体架构”到“分层架构”再到“微服务架构”的演进过程。

(2) 数据库设计能力断点

高职阶段能力基线：掌握基本 CRUD 操作、数据表创建及简单查询。本科阶段能力要求：深入理解数据库范式理论，掌握索引优化策略、事务隔离级别原理及分布式主键生成机制。断点具体表现：缺乏实体关系图(E-R 图)绘制能力；未掌握索引创建与优化方法；对数据库事务、锁机制及并发控制问题理解不足；缺乏 MyCat、ShardingSphere 等主流分库分表中间件的实践经验。建议在项目实训环节嵌入“数据库设计评审”机制；开设实践课程《高并发数据库设计实战》(1 学分)。

(3) 项目工程化能力断点

高职阶段能力基线：采用手动打包、FTP 上传部署方式，且未应用版本控制系统。本科阶段能力要求：熟练运用 Git Flow 工作流进行协作开发；掌握 Maven 多模块项目管理；具备持续集成/持续部署(CI/CD)流程实施能力；掌握 Docker 容器化部署技术。断点具体表现：未掌握基于 Git 分支的团队协作开发模式；缺乏 Maven pom.xml 配置文件编写能力及 Dockerfile 构建能力；对 Jenkins、SonarQube 等主流工程化工具缺乏实践经验。建议强制要求所有项目开发采用 Git + Maven + Docker 技术栈；开设集中式工程化实践训练营(为期 1 周，不计入学分但属必修环节)。

4. 一体化课程模型构建

依据人才培养方案，构建高本贯通(3 + 2)软件工程专业学生的能力成长路线图(如表 5 所示)。该路线图以“从技能到工程”为主线，覆盖高职第 1 学期至本科第 10 学期的全过程，划分为 5 个主要阶段，明确了 12 项核心能力及 30 余项关键里程碑，为课程衔接设计提供依据。

Table 5. Staged path of skill development
表 5. 技能发展阶段路径

阶段	时间	核心目标	关键能力	里程碑示例
L1 基础技能层	高职 1~2 学期	会写代码	语法、调试、基础算法	完成 C 语言贪吃蛇
L2 模块开发层	高职 3~4 学期	会做功能	前后端单模块开发	实现 JSP 留言板
L3 项目实战层	高职 5~6 学期	会做项目	完整项目交付	完成 Android 记账本
L4 工程进阶层	本科 7~8 学期	会做工程	工程化、架构、协作	微服务电商系统
L5 综合创新层	本科 9~10 学期	会解决复杂问题	架构设计、性能优化、团队协作	毕业设计 + 企业级项目

4.1. 设计原则

由学生的能力成长路线图，确定一体化课程模型的设计原则为“岗位导向、螺旋上升、学分置换”。

4.1.1. 总体结构：三纵三横一螺旋

(1) 岗位导向主线(三纵)——对准 Web 前端、移动开发、服务端测试三大典型岗位链。

(2) 纵向接口课程(三纵)——高职→本科“螺旋上升”无缝衔接，支持学分置换。

(3) 横向拓展包(三横)——学分模块化选修，打通复合型人才成长通道。

(4) 能力螺旋——每主线“三段六阶”：基础→模块→项目→工程→优化→创新，层层递进。

4.1.2. 岗位导向主线课程地图

(1) Web 前端主线

HTML+CSS(4)→JavaScript*(课外/线上)→Web 前端开发(64, L2)

→ 前端框架(Vue/React)*(课外/竞赛，L3)

→ 企业级前端工程*(本科，L4)→毕业设计(L5)

高职 Web 前端开发 64 学时，本科阶段可学分置换免修对应 3 学分《前端框架基础》；本科第 8 学期插入《微前端与性能优化》，形成 L4 螺旋上升。

(2) 移动开发主线

面向对象 1(4)→面向对象 2(4)→移动互联网软件开发(64, L2)

→ 移动软件开发实战(C, L3)→智能应用项目实战(L3-L4)

→ 移动架构与 Flutter*(本科，L4)→毕业设计(L5)

高职面向对象 2 成绩 ≥80 分，可置换本科《Java 高级应用编程》4 学分(内容递进，不再重复修读)。

(3) 服务端/测试主线

程序设计基础→软件工程导论(3)→软件测试技术(3, L2)

→ 测试自动化与持续质量*(企业选修, L3-L4)

→ 性能测试与质量管理*(本科，L4)→毕业设计(L5)。

4.2. 接口课程

4.2.1. 纵向接口课程：高职→本科螺旋接口

纵向接口课程作为“高本贯通”人才培养体系的纵向衔接机制与能力进阶通道，其核心功能在于确保高职教育阶段的学业终点与本科教育阶段的学业起点实现精准衔接。该机制有效规避课程内容重复、填补知识深度与工程复杂性断层(例如从 JDBC 技术到分布式事务的跨越)，并支撑专业能力的螺旋式递进。具体而言：纵向接口课程打破高职与本科“两段式”课程壁垒，构建连续的能力成长路径；通过“能力等效”认证机制，豁免已掌握内容的重复修读，节省学时用于高阶课程学习；填补知识深度与工程复杂度断层；使学生提前进入本科高阶学习阶段，为企业实践与科研项目预留充足时间。接口课程设置详见表 6。

Table 6. Comparison of information on the series of interface courses
表 6. 课程接口系列信息对比

接口系列	高职课程 (1~6 学期)	本科课程 (7~8 学期)	学分置换规则	螺旋上升要点
Java 系列	面向对象(4 学分)	Java 高级应用编程 (3 学分)	4→3 置换，成绩 ≥80	本科侧重新特性、多线程、JVM 调优
数据库系列	数据库原理(4 学分)	数据库设计与开发 (2 学分)	可置换，Oracle→MySQL 高阶	引入分库分表、分布式事务
测试系列	软件测试技术 (3 学分)	测试自动化 (3 学分)	免修前提：通过自动化脚本测评	企业真实 CI 环境演练

4.2.2. 横向拓展包：6 选 3

横向拓展包作为“高本贯通”课程体系中的“能力拓展模块”与“职业发展助推器”，可在同一学习阶段横向拓宽学生的技能边界，促进跨领域知识融合，助力构建“多领域复合型”职业能力结构，从而增强岗位适应性与职业发展潜力。具体体现为：突破单一技术栈限制，融入前端开发、人工智能、数据科学、云原生等多技术领域能力；实现“移动开发与人工智能”、“前端开发与数据分析”、“软件测试与 DevOps”等跨界能力组合；通过“6 选 3”个性化选修机制，学生可根据个人兴趣与职业规划自主选择组合；紧密对接产业前沿趋势，快速引入新兴技术(如 AIGC、低代码、云原生)。据此设计的横向拓展包括以下内容：

(1) 数据分析包

Python 程序设计(3 学分)→Python 数据挖掘(3 学分)→数据分析与应用(Python) (3 学分)

对接岗位：数据分析师、算法助理

(2) 人工智能包

人工智能导论(2 学分)→智能应用项目实战(2 + 2 学分)

对接岗位：AI 应用工程师、智能产品助理

(3) 前端进阶包

Web 前端开发(4 学分)→移动软件开发实战(3 学分)

跨主线横向连接，培养“大前端”复合能力

(4) 云原生与 DevOps 包

Linux 操作系统(4 学分)→Docker 基础→DevOps 与持续交付(企业选修)

对接岗位：运维开发、云平台助理

(5) 金融产品包

金融 IT 项目开发实战(4 学分)→金融系统架构设计*(本科，3 学分)

对接岗位：金融软件工程师

(6) 创新创业包

创新方法→创业实践→专业竞赛/大创项目(学分可替换对应实践环节，激励竞赛成果)。

4.3. 课程－能力－任务映射

4.3.1. 对应关系

建立“课程－能力－任务”映射关系是确保课程内容精准对接人才培养目标、支撑学生能力螺旋式递进的核心机制。该映射模型旨在清晰界定以下三个关键环节：能力目标的分解、课程内容的承载、以及任务驱动的验证。任务设计需体现从简单到复杂、从模拟到真实的递进性，并与能力断点清单(表 4)中所识别的关键能力提升点紧密关联。“课程－能力－任务”映射模型的核心价值体现在多方面：确保课程教学目标明确指向能力培养，避免盲目性；通过任务完成质量追踪学生能力成长轨迹，为教学调整提供依据；将能力评估转向基于任务成果的多元评价，真实反映工程实践水平；并为学分置换提供核心依据，基于能力等效性而非仅课程相似性。主要课程与能力的关系详见表 7 所示。

Table 7. Correspondence between curriculum and competency
表 7. 课程能力对应关系

典型工作任务	能力指标	支撑课程
Web 前端项目分析与设计	需求分析/界面原型	HTML + CSS、软件工程导论
移动前端编码开发	Java/JSON/网络通信	面向对象、移动互联网软件开发
软件测试与质量保障	测试用例/自动化脚本	软件测试技术、JavaEE
服务器部署与运维	Linux 命令/性能调优	Linux 操作系统、数据库设计与开发

4.3.2. 以 Java 后端开发岗位为例

选取核心岗位 Java 后端开发工程师，聚焦其典型工作任务“企业级电商订单管理模块开发”。该任务覆盖“需求分析－架构设计－编码实现－测试优化－部署运维”全流程，可拆解为高职阶段(L2~L3：模块开发层－项目实战层)的“订单基础功能开发”与本科阶段(L4~L5：工程进阶层－综合创新层)的“订

单高并发优化与微服务集成”，完美契合“三阶五层”能力模型中“岗位能力→技术能力→工程能力”的递进逻辑。基于高本贯通“3 + 2”培养周期，将“企业级电商订单管理模块开发”按能力递进拆解为 2 个阶段、6 个子任务，具体如表 8 所示。

Table 8. Phased decomposition of typical work tasks
表 8. 典型工作任务分阶段分解

培养阶段	任务层级	核心子任务	任务复杂度	对应能力维度
高职 (L2~L3)	基础实现层	1. 订单需求文档(PRD)解读与用例梳理 2. 订单数据模型(MySQL)设计 3. 订单 CRUD 接口编码(Spring Boot) 4. 接口功能测试(Postman)	中小型模块，单库单服务	中小型模块，单库单服务
高职 (L2~L3)	优化集成层	1. 订单高并发场景(秒杀)架构优化(Redis 缓存) 2. 订单服务微服务化改造(Spring Cloud) 3. 订单模块 CI/CD 部署(Jenkins + Docker)	中小型模块，单库单服务	中小型模块，单库单服务

基于“核心课程地图”(表 2)，构建映射矩阵(表 9)，明确每个能力点对应的课程模块与教学重点。

Table 9. Task-competency-curriculum mapping matrix
表 9. “任务 - 能力 - 课程”映射矩阵

培养阶段	典型子任务	核心能力点	对应课程模块	课程教学重点
高职 L2	PRD 解读与用例梳理	需求分析、UML 用例图绘制	《软件工程导论》(第 3 学期)	1. PRD 文档解读方法 2. 用例图绘制工具(StarUML)
高职 L2	订单数据模型设计	数据库表设计、SQL 编写	《数据库原理及应用》(第 3 学期)	1. 订单表结构设计(含索引) 2. 关联查询 SQL 编写
高职 L3	订单 CRUD 接口编码	SpringBoot 开发、异常处理	《JavaEE 开发与实践》(第 6 学期)	1. Spring Boot 接口开发 2. 订单业务异常处理(如库存不足)
高职 L3	接口功能测试	Postman 测试、错误定位	《软件测试技术》(第 6 学期)	1. 接口测试用例设计 2. Postman 断言编写
本科 L4	订单微服务拆分	微服务架构设计、SpringCloud 配置	《微服务架构设计》(第 7 学期)	1. 订单服务拆分原则 2. Eureka 服务注册与 Feign 调用
本科 L4	Redis 缓存实现	高并发缓存设计、Redis 操作	《高并发数据库实战》(第 7 学期)	1. Redis 缓存策略 2. 缓存穿透/击穿解决方案
本科 L5	Docker 打包与 Jenkins 部署	容器化、CI/CD 流程	《DevOps 与持续交付》(企业选修)	1. Dockerfile 编写(订单服务) 2. Jenkins 流水线配置
本科 L5	团队代码协作与文档编写	GitFlow、技术文档撰写	《项目开发实战》(第 9 学期)	1. Git 分支管理(feature/develop 分支) 2. 接口文档(Swagger)编写

映射机制总结：在“订单管理”任务教学中，高职侧重功能实现，本科侧重性能优化与工程化，体现纵向贯通、横向拓宽理念；各能力点通过课程支撑 - 教学活动 - 考核验证三维路径培养。例如，微服务设计能力由课程、项目、评审形成闭环；任务设计对接岗位需求(如高并发处理、微服务架构)，融入 1 + X 证书标准，符合岗位导向原则。

4.4. 学分置换

4.4.1. 与“1 + X”证书对接

深度对接国家“1 + X”证书制度，构建课程学分与职业技能等级证书(X 证书)的等效置换机制[7]。

具体而言，当学生取得与专业核心能力高度关联的 X 证书(如《Web 前端开发》《Java 应用开发》《移动应用开发与测试》《云计算平台运维与开发》等)，经由学院教学指导委员会联合行业企业专家对其能力覆盖范围及达标程度进行综合评估后，可置换人才培养方案中指定的课程学分[8]。该机制以“能力等效”为基本原则，依托权威第三方认证检验学生能力达成度，显著缩短重复学习周期，激励学生积极获取行业认可的职业技能证书，有效提升就业竞争力，并推动课程内容与行业前沿技术标准动态接轨。证书与学分置换规则详见表 10。

Table 10. Curriculum certificates and credit transfer
表 10. 课程证书与学分抵换

课程/环节	证书	学分抵换
HTML + CSS + Web 前端开发	1 + X Web 前端中级	2 学分
Java 高级 + JavaEE	1 + X Web 前端高级	2 学分
数据库原理 + Oracle 开发	计算机等级考试三级数据库	2 学分

4.4.2. 学分置换与螺旋管理流程

学分置换与螺旋管理流程是保障“高本贯通”人才培养体系高效运转、实现学生能力螺旋式进阶的核心管理机制。该流程旨在规范学分置换的操作细则，并建立贯穿高职与本科学习全过程的管理闭环，确保置换的公平性、有效性及其对能力进阶的支撑作用。具体流程设计如下：

(1) 申请与审核：学生在满足前置条件(如高职阶段相关课程成绩达标、获得指定 X 证书等)后，于本科入学后第一学期内提交学分置换申请，并附相关证明材料(成绩单、证书复印件等)。申请材料经所在院系教学委员会联合行业专家进行“能力等效性”评估审核(重点审核申请置换课程与目标课程在能力覆盖范围、深度及实践水平上的匹配度)，审核通过后予以公示。

(2) 备案与执行：审核通过的置换方案报学校教务处备案，正式生效。被置换的本科课程学分予以认定，学生无需重复修读，节省的学时可用于选修高阶课程、参与企业项目实践或科研训练。教务系统自动更新学生培养计划，标注已置换课程及对应学分。

(3) 螺旋管理：该流程并非静态的一次性操作，而是嵌入整个“螺旋上升”能力培养过程。建立“学分银行”制度，对学生通过横向拓展包学习、企业实践项目、竞赛获奖、发表成果等途径获取的非标准化学分进行认定、积累和转换，支持个性化能力进阶路径[9]。同时，对已置换学分的学生在后续高阶课程(尤其是接口课程对应的螺旋进阶课程，如《微前端与性能优化》《移动架构与 Flutter》《性能测试与质量管理》)中的学业表现进行持续追踪评估[10]。

(4) 质量保障与反馈：设立定期的流程回溯机制。教学委员会依据学生后续课程成绩、项目表现、毕业设计质量及就业反馈等数据，评估学分置换对学生能力成长的实际效果。若发现置换后学生在相关高阶能力上存在显著不足，则需回溯分析原因，及时调整前置课程的置换标准、接口课程的教学内容或螺旋管理流程本身，形成持续改进闭环。

(5) 信息化支撑：开发或利用现有教务管理系统，实现学分置换申请、审核、备案、执行及效果追踪的全流程信息化管理，确保流程透明、高效、可追溯。

5. 实施与预期成效

在高本贯通人才培养模式深入推进的背景下，本校软件工程专业(高本贯通) 2024 级与 2025 级培养方案已完成一个完整实施周期，2026 级方案修订工作随之启动。本研究将“岗位导向、螺旋上升、学分

置换”一体化课程模型系统融入新版培养方案，取得了以下可量化、可复制的阶段性成效，供同类院校借鉴。

(1) 岗位对口率显著提高

依托“Web 前端–移动开发–测试”三大典型岗位链，校企共同制定课程地图与能力标准。企业工程师全程参与课程设计、教学与评价，实现岗位需求与教学内容“零距离”对接。2024 级学生顶岗实习岗位对口率较 2023 级提升 18.7%，预计 2026 级可达 90%以上。

(2) 重复学时有效压缩

通过纵向接口课程与学分置换机制，学生凭高职阶段达标成绩可直接免修本科同名课程，总计释放约 31 学分、66 学时，折合培养周期压缩 30%。释放学时全部用于“云原生、数据分析、人工智能”前沿模块研修，既避免内容重叠，又提升课程高阶性。

(3) 能力成长路径可视化

每条培养主线均设置“三段六阶”里程碑(基础→模块→项目→工程→优化→创新)，配套能力测评与徽章认证。学生可在学习地图中实时查看个人成长轨迹，教师亦可依据数据及时开展学业预警与个性化指导，教学管理实现精准化。

(4) 毕业学分结构进一步优化

总学分维持 219 不变，但进阶课程与企业实践项目占比由 27%提升至 41%，选修模块(含横向拓展包)学分占比提高 8 个百分点。课程结构呈现“基础够用、核心扎实、拓展灵活、实践强化”的良性态势，为培养复合型、创新型技术技能人才奠定坚实基础。

6. 结论

综上所述，一体化课程模型不仅显著提升了软件工程专业高本贯通培养效率与质量，也为职业教育课程体系重构提供了可复制、可推广的范式。

基金项目

全国高等院校计算机基础教育研究会计算机基础教育教学研究课题：软件工程专业高本贯通课程设计理念和人才培养方案开发研究(编号：2024-AFCEC-517)。

上海农林职业技术学院校内课题：“X”证书多元化在软件技术专业人才培养模式中的应用研究(编号：JY6-0000-24-14)。

参考文献

[1] 中共中央办公厅, 国务院办公厅. 关于深化现代职业教育体系建设改革的意见[Z].

[2] 北京市教育委员会. 关于开展高端技术技能人才贯通培养试验工作的通知[Z]. 京教职成[2024]7 号, 2024-04-17.

[3] 中国教育科学研究院. 2025 职业教育改革与发展报告[EB/OL]. 2025-12-30. http://www.jyb.cn/rmtzgjyb/202512/t20251230_2111431799.html, 2026-01-08.

[4] 赵晓燕, 王屹. 我国高职本科贯通培养的现实困境与路径选择[J]. 教育科学论坛, 2023(9): 45-49.

[5] 中共中央办公厅, 国务院办公厅. 关于推动现代职业教育高质量发展的意见[Z]. 教发[2021]60 号, 2021-10-12.

[6] 上海市教育委员会. 2023 年上海市职业教育工作要点[Z].

[7] 教育部办公厅, 国家发展改革委办公厅, 财政部办公厅, 市场监管总局办公厅. 关于在院校实施“学历证书+若干职业技能等级证书”制度试点方案的通知[Z]. 教职成厅[2019]6 号, 2019.

[8] 耿艳旭, 李敏, 吕磊, 等. 焊接专业 1 + X 证书考核内容对接学分银行的研究[J]. 黑龙江科学, 2022, 13(13): 57-59.

[9] 广东终身教育学分银行管理中心. 广东终身教育学分银行开展“1 + X”证书学分认定和转换的公告[EB/OL]. 2022-04-08. https://www.cbgd.cn/gd_cbank_cms/xwdt/3075.jhtml

[10] 邓燕. 基于学习成果认定与转换的校内学分银行运行机制探索[J]. 现代职业教育, 2025(2): 17-20.