

CERACU

全国高等学校计算机教育研究会团体标准

CERACU/T XXXXX—XXXX

计算机核心课程规范
软件工程

Specification for Core Courses of Computer Science

Software Engineering

(工作组草案)

Working Draft

2019 - XX - XX 发布

2020 - XX - XX 实施

全国高等学校计算机教育研究会 发布

目 次

目 次	I
前 言	II
计算机核心课程规范 软件工程	3
1 范围	3
2 课程概要和教学目标	3
2.1 课程概要	3
2.2 教学目标	3
3 课程知识体系及教学要求	3
4 课程实践教学及要求	7
4.1 实践教学目标	7
4.2 实践教学内容及要求	7
4.3 实践教学实施及要求	7
5 课程考核要求	7
附 录 A 本科软件工程课程规范案例 A (**学校示例)	错误!未定义书签。

前 言

本标准按照GB/T 1.1—2009给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由全国高等学校计算机教育研究会（CERACU）提出，由全国高等学校计算机教育研究会标准委员会归口。

本标准起草单位：

本标准主要起草人：

计算机核心课程规范

软件工程

1 范围

本标准主要描述了软件工程课程的培养目标、知识点及学习要求、实践教学及要求、课程教学实施及考核要求等，并给出了若干本科软件工程课程规范的示范案例。

本标准适用于高等学校软件工程课程大纲或标准的制定，指导软件工程课程教学的实施和考核。

2 课程概要和教学目标

2.1 课程概要

软件工程是计算机大类专业的一门重要专业课程，计算机科学与技术、软件工程等专业的核心专业课程。课程旨在深入阐述软件工程的思想理念，系统讲授软件开发、管理和维护的方法、技术和工具，帮助学生掌握需求分析、软件设计、编码测试、维护演化、质量保证、团队协作等方面的专业知识，培养学生开发高质量软件产品的能力。

软件工程是一门实践性非常强的课程，需要遵循课堂教学和实践教学相结合、知识传授与案例研讨相结合的教学方式，强调学以致用，突出能力和素质的培养。

本课程的前导课程包括计算机程序设计、数据结构。

2.2 教学目标

本课程的教学目标分为基础目标和增强目标。基础目标是指本课程教学必须达成的合格性目标。增强目标是施教学校根据自身的实际情况，所选定的一组增强性教学目标。

基础目标包括：

- (1) 理解和掌握软件工程的基本原理和思想；
- (2) 应用软件工程方法、技术及工具进行软件系统的开发、管理和维护；
- (3) 培养软件工程的素质和能力。

增强目标包括：

- (1) 大型复杂软件系统的工程开发和项目管理能力；
- (2) 针对特定的技术要求而进行软件开发的能力；
- (3) 结合领域软件的特点和要求进行软件开发的能力；
- (4) 综合考虑隐私、伦理、安全、健康、法律、文化及环境等因素进行软件开发的能力。

3 课程的知识点及教学要求

软件工程课程的知识体系包含一组知识点，每个知识点都有其学习类别、要求和学时安排，不同知识点按照其内在逻辑关系组织成层次化的结构（见表1）。课程知识点的描述包括以下几方面的内容。

- **知识点的标识：**每个知识点都有一个唯一的标识。
- **知识点的类别：**课程的知识点分为核心知识点和扩展知识点。
 - 核心知识点是达成课程基础教学目标所必需的，用符号“C”来表示。
 - 扩展知识点是根据教学实际情况可选择性讲授的知识点，用符号“X”来表示。
- **知识点的学习要求：**每个知识点可从理解、应用、分析三个层次来明确其学习要求。各个学校可以结合各自的具体情况给定知识点的学习要求。
 - **理解：**学生能够解释知识点的内涵并能用自己的语言来表达，用符号“U”来表示。
 - **应用：**学生能够应用知识点来解决问题，用符号“P”来表示。
 - **分析：**学生能够应用知识点来对照区分不同问题解决方法的差异性和优劣性，用符号“A”来表示。
- **知识点的学时要求：**针对该知识点的课堂讲授学时。1个学时通常对应于一节课。表1中各个知识点的学时给出的是最低学时要求。

表1. 软件工程课程的知识及其类别、学习和学时要求。“类别”列中的“C”表示核心知识点，“X”表示扩展可选知识点；“学习要求”列中的“U”表示理解，“P”表示应用，“A”表示分析；“学时要求”栏中的“-”符号表示没有具体的学时要求；一级知识点的学时只累加了下属知识点中核心知识点的学时，不包含扩展知识点的学时。

知识点标识	知识点名称	知识点的内容描述	类别	学习要求	学时要求
SE_01	软件工程概述	软件概念、软件工程概念、软件工程目标与原则、软件工程职业道德规范	C	U	2
SE_01_01	软件概念	软件概念、特点和分类	C	U	0.5
SE_01_02	软件工程概念	软件危机的表现和根源、软件工程概念、软件工程发展历史	C	U	0.5
SE_01_03	软件工程目标与原则	软件工程目标与原则	C	U	0.5
SE_01_04	软件工程职业道德规范	软件工程从业人员需遵守的法律、法规和职业准则。	C	U	0.5
SE_02	软件过程	软件生命周期、软件过程模型、软件过程管理与改进	C	U	3
SE_02_01	软件生命周期	软件生命周期概念、各个阶段的任务和产品	C	U	1.5
SE_02_02	软件过程模型	软件过程模型概念、典型软件过程模型	C	U	1.5
SE_02_02_01	软件过程模型概念	软件过程模型的概念	C	U	-
SE_02_02_02	瀑布模型	瀑布模型及其特点和适用场合	C	U	-
SE_02_02_03	原型模型	原型模型及其特点和适用场合	C	U	-
SE_02_02_04	迭代模型	迭代模型及其特点和适用场合	C	U	-
SE_02_02_05	增量模型	增量模型及其特点和适用场合	C	U	-
SE_02_02_06	螺旋模型	螺旋模型及其特点和适用场合	X	U	-
SE_02_02_07	UP模型	UP模型及其特点和适用场合	X	U	-
SE_02_03	敏捷方法	敏捷开发方法的相关概念、思想和过程，Scrum框架等特定的敏捷方法	X	P	3
SE_02_04	DevOps方法	持续集成与交付的概念，DevOps方法的相关概	X	P	2

		念、思想和过程			
SE_02_05	群体化开发方法	群体化开发方法的相关概念、思想和过程	X	P	2
SE_02_06	软件过程管理与改进	以 CMM为代表的软件过程改进模型, 过程评价和改进的基本思想	X	U	2
SE_03	软件开发方法	软件开发方法的概念、常见的软件开发方法	C	P	3
SE_03_01	结构化开发方法	结构化开发方法的相关概念、思想和过程	X	P	2
SE_03_02	面向对象开发方法	面向对象开发方法的相关概念、思想和过程	C	P	3
SE_04	软件需求分析	软件需求概念, 软件需求分析基础, 软件需求获取方法, 软件需求分析方法, 软件需求文档编制, 软件需求评审	C	P	8
SE_04_02	软件需求分析基础	软件需求概念、类别及特点, 软件需求分析的任务、目标、过程和原则	C	U	1
SE_04_03	软件需求获取方法	软件需求获取的任务、方法和原则	C	P	1
SE_04_04	软件需求分析方法	常见的软件需求分析方法	C	P	4
SE_04_04_01	面向对象需求分析方法	面向对象需求分析的概念、过程和策略, 软件需求的面向对象表示及模型	C	P	4
SE_04_04_02	结构化需求分析方法	结构化需求分析的概念、过程和策略, 软件需求描述语言, 软件需求的结构化表示及模型	X	P	4
SE_04_05	软件需求文档编制	软件需求文档, 文档规范和编制要求	C	P	0.5
SE_04_06	软件需求评审	软件需求评审的目的、要求、原则和方法	C	A	1.5
SE_05	软件设计	软件设计基础, 软件概要设计, 用户界面设计, 软件详细设计, 软件设计方法, 软件设计模式, 软件设计文档编制, 软件设计评审	C	P	8
SE_05_01	软件设计基础	软件设计的概念、软件概要设计、用户界面设计、软件详细设计	C	U	2
SE_05_01_01	软件设计概念	软件设计的概念、任务、过程、输出和原则	C	U	0.5
SE_05_01_02	软件概要设计	软件概要设计概念、任务、要求和输出, 软件体系结构的概念, 软件体系结构的类型和风格	C	U	0.5
SE_05_01_03	用户界面设计	人机交互设计的理念与原则, 用户界面设计的概念、任务、原则和输出	C	U	0.5
SE_05_01_04	软件详细设计	软件详细设计概念、任务、要求和输出, 数据结构的逻辑设计, 算法与处理逻辑的设计	C	U	0.5
SE_05_02	软件设计方法	面向对象软件设计方法, 结构化软件设计方法, 基于模式的软件设计方法	C	P	4
SE_05_02_01	面向对象的软件设计方法	面向对象软件设计的概念、过程和策略, 面向对象软件设计的模型及语言	C	P	4
SE_05_02_02	结构化软件设计方法	结构化概要设计语言、过程、模型和策略, 结构化详细设计语言、过程、模型和策略	X	P	4
SE_05_02_03	基于模式的软件设计方法	软件设计模式的概念, 软件设计模式的种类, 常见软件设计模式及其应用, 基于模式的软件设计过程和策略	X	A	4
SE_05_03	软件设计文档编制	软件设计文档编制的任务、规范和要求	C	P	0.5
SE_05_04	软件设计评审	软件设计评审的目的、要求、原则和方法	C	A	1.5
SE_06	编码实现	编码实现基础、编码规范与风格、代码重用	C	P	2

SE_06_01	编码实现基础	编码实现的概念、任务、要求和原则	C	U	1
SE_06_02	编码规范与风格	程序代码的质量要求，编码规范，编码风格	C	A	0.5
SE_06_03	代码重用	程序代码重用的方式和方法，开源代码重用	C	P	0.5
SE_07	软件测试	软件测试基础，软件测试过程与策略，软件测试技术，非功能测试技术	C	P	6
SE_07_01	软件测试基础	软件错误和软件测试的概念，软件测试任务，软件测试充分性概念和覆盖准则，软件测试原理	C	U	1
SE_07_02	软件测试过程与策略	软件测试过程，单元测试、集成测试、系统测试、确认测试的实施策略，回归测试	C	U	1
SE_07_03	软件测试技术	白盒测试技术，黑盒测试技术，面向对象软件测试技术，软件测试工具	C	P	4
SE_07_04	非功能测试技术	非功能（如性能、安全性、压力等）的测试技术	X	P	2
SE_08	软件交付与部署	软件交付与部署的概念、任务与方法	X	P	1
SE_09	软件维护与演化	软件维护基础、软件维护过程与策略、软件维护技术、软件自动化维护	C	U	2
SE_09_01	软件维护基础	软件维护和可维护性的概念，软件维护类别，软件维护的副作用，影响软件可维护性的因素	C	U	1
SE_09_02	软件维护过程与策略	软件维护的任务、过程、活动和原则，软件维护的实施策略	C	U	0.5
SE_09_03	软件维护技术	程序理解、软件再工程、逆向工程、软件重构等软件维护技术	C	U	0.5
SE_09_04	软件演化	软件演化的概念、方式及策略选择	X	U	0.5
SE_09_05	软件自动化运维	软件版本更新，软件运行状态监控，软件运行优化，软件运维自动化工具	X	U	1
SE_10	软件项目管理	软件项目管理基础，软件配置管理，软件项目团队管理，软件项目进度管理	C	P	2
SE_10_01	软件项目管理基础	软件项目管理的概念、任务、内容和原则	C	U	2
SE_10_02	软件配置管理	软件配置和配置项概念，软件配置管理的概念、任务、方法和工具	X	P	2
SE_10_03	软件项目团队管理	软件项目团队的组织、交流和合作	X	P	1
SE_10_04	软件项目进度管理	软件项目进度的计划、实施、跟踪和调整	X	P	1

4 课程实践教学及要求

4.1 实践教学目标

实践教学是软件工程课程教学中的一个重要环节，其目标是通过软件开发实践帮助学生加强对软件工程思想和原理的理解，培养学生运用软件工程方法、技术和工具来开发软件系统的能力，帮助学生在实践中体验软件开发的实际场景和问题，体验软件开发的核心环节及面临的各种挑战，培养学生解决复杂工程问题的能力以及良好的软件工程素养。

4.2 实践教学内容及要求

软件工程课程的实践教学必须覆盖基础的实践内容及要求，以确保实践教学的成效以及达成实践教学的基本目标。

● 基础的实践内容及要求

针对特定的应用，运用所学的软件工程过程和方法、技术，使用项目管理、软件建模、软件测试、协同开发等工具，开展软需求分析、软件设计、编写代码、软件测试等软件开发实践，遵循相关的规范和标准，生成多样化、相互一致的软件制品，包括：

- 软件模型
- 软件文档
- 程序代码
- 测试用例
- 可运行软件系统等

● 可选的实践内容及要求

结合不同专业人才培养的要求，考虑具体的施教情况（如施教对象、课程学时、现实条件、校企合作等），选择以下实践内容：

- 围绕软件项目管理，开展项目的计划、跟踪、风险管理等实践；
- 围绕软件质量保证，开展软件制品评审、编码风格与代码质量、软件质量度量与分析、软件验证与确认等实践；
- 针对特定的软件开发技术（如高可信软件技术、安全攸关软件技术），开展软件开发实践；
- 针对特定领域软件（如政务软件、军用软件、工控软件），开展软件开发实践；
- 对实践所开发软件系统在综合性、复杂性、创新性等方面提出要求，开展解决复杂工程问题以及创新能力培养等方面的实践。

4.3 实践教学实施及要求

- 实践教学的组织形式：以项目团队的方式来组织学生开展实践，每个团队的人员规模不少于3人，每个成员在团队中应有明确的角色定位和任务分工。
- 实践教学的课内学时：实践教学应该安排一定的课内学时（应不少于四分之一的知识讲课时），用于对课程实践进行汇报、讲评、点评和指导，以发现和解决问题，交流分享实践经验 and 成果。

5 课程考核要求

- 采用书面考试和实践评价相结合的方式。书面考试可采用开卷或者闭卷的方式。课程实践的评价需根据实践的内容要求，实践成绩占课程总成绩的比例不少于50%。
- 采用定量和定性相结合的考核方式。开卷/闭卷考试侧重定量考核，实践考核侧重定性考核。实践考核可以从个人贡献和团队成果两个角度进行，个人贡献考察个人在项目中的贡献度，团队成果考察团队交付的软件制品（包括文档、模型、代码、数据等），从团队协作、制品质量、项目规模与难度等角度考评学生掌握和运用软件工程知识来开发软件系统的能力。