ICS 03.180
A 18

CERACU

全国高等学校计算机教育研究会团体标准

T/CERACU 201—2020

计算机核心课程规范
软件工程

Specification for Core Courses of Computer Science—
Software Engineering

2020-11-27 发布　　2021-05-01 实施

全国高等学校计算机教育研究会 发布

目　　次

前言 2
1 范围 3
2 术语和定义 3
3 课程概述 4
4 课程教学目标 4
5 课程知识点及学习要求 4
5.1 课程知识点 4
5.2 课程知识点学习要求 9
6 课程实践教学及要求 9
6.1 实践教学目标 9
6.2 实践教学内容及要求 9
6.3 实践教学实施及要求 9
7 课程考核要求 10

前　言

本文件按照 GB/T 1.1－2020 给出的规则起草。

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。

本文件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。

本文件起草单位：

国防科技大学、北京大学、清华大学、武汉大学、深圳点猫科技有限公司、北京航空航天大学、湖南师范大学、云南大学、西安电子科技大学、中国地质大学（武汉）、湖北第二师范大学、韩山师范学院、湖南智擎科技有限公司、武汉工商学院。

本文件主要起草人：

毛新军、孙艳春、刘强、应时、李天驰、白晓颖、张莉、孙悦、张锦、王炜、董威、宋胜利、尚建嘎、蔡美玲、胡罗凯、夏立、尹刚、张婷。

计算机核心课程规范
软件工程

1 范围

本文件描述了软件工程课程的教学目标、知识点及学习要求、实践教学及要求、考核要求等。
本文件适用于高等学校软件工程课程、教材、教学过程和教学评价。

2 术语和定义

下列术语和定义适用于本文件。

2.1

知识体系 body of knowledge
知识体系是指构成专业领域的一套完整概念、术语和活动的集合。

2.2

知识点 knowledge point
知识点是构成知识体系的最小知识单位，也是最具体的知识内容。

2.3

核心知识点 core knowledge point
核心知识点是指达成课程基本教学目标所需的知识点，也是构成课程知识体系的基础性知识点。

2.4

扩展知识点 expanded knowledge point
扩展知识点是指核心知识点之外，根据课程教学实际情况，可选择性讲授的知识点。

2.5

理解 understand
理解是指能够解释知识点的内涵和相互间的关系，并能正确地表述、选择和判断知识点。

2.6

应用 apply
应用是指能够综合运用知识点来分析和解决问题。

2.7

分析 analyze
分析是指能够应用知识点来比较和区分不同问题解决方法及结果的差异性和优劣性。

3 课程概述

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。课程旨在阐述软件工程的目标、过程和原则，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，需要遵循理论教学和实践教学相结合、知识传授和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。

4 课程教学目标

软件工程课程的教学目标分为基本目标和增强目标。基本目标是指本课程教学须达成的合格性目标。增强目标是可选项，是指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。

a） 基本目标包括：

1） 理解和掌握软件工程的目标、过程和原则；

2） 掌握和应用软件工程方法、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；

b） 增强目标包括：

1） 培养大型、复杂软件系统的工程开发和项目管理能力；

2） 培养针对特定的技术要求而进行软件开发的能力；

3） 培养结合特定领域软件的特点和要求进行软件开发的能力；

4） 培养综合考虑隐私、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；

各个学校和院系结合具体情况，从本文件中选择知识点及其能力要求、学时要求、实践教学要求，组成一门具体课程的内容和要求。

5 课程知识点及学习要求

5.1 课程知识点

软件工程课程的知识体系按照分层定义，结构层级不超过三级，每个知识点可以容纳下级知识点，也可以是原子知识点。能力要求给出一级、二级和三级三个层次的具体要求，其中一级对应于理解层次，二级对应于应用层次，三级对应于分析层次。学时要求是指理论讲授学时，要求至少达到的学时下限，不设上限。知识点属性描述的定义如表 1 所示。软件工程课程的知识点及其能力要求、学习要求等如表 2 所示。

表1 知识点属性

属性项	数据类型	取值范围	可选（O）/必备（M）	说明
标识符	字符串	长度不超过20	M	在本文件中唯一标识知识点
名称	字符串	长度不超过40	M	应采用行业内共识的知识点名称
能力要求	容器类型	包含一级、二级、三级三个子项	M	说明知识点需要达到的能力程度
一级	字符串	长度不超过1000	O	说明某知识点“一级”要求的具体含义
二级	字符串	长度不超过1000	O	说明某知识点“二级”要求的具体含义
三级	字符串	长度不超过1000	O	说明某知识点“三级”要求的具体含义
学时	实数	大于0	M	顶层知识点的学时是下层所有核心知识点学时的总和，不包括下层扩展知识点的学时
类别	字符串	长度为1	M	核心知识点用符号“C”来表示；扩展知识点用符号“X”来表示

表 2 软件工程课程的知识点及学习要求（第 1 页/共 4 页）

标识符	名称	能力要求			类别	学时
		一级	二级	三级		
SE_01	软件工程概述	理解软件工程的概念、目标、原则和职业道德	在软件开发实践中应用软件工程原则，遵循职业道德规范		C	2
SE_01_01	软件概念、特点和分类	理解软件的概念及特点，及有哪些不同类别的软件			C	0.5
SE_01_02	软件工程概念	理解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	0.5
SE_01_03	软件工程目标与原则	理解软件工程的目标以及实现目标的基本原则	在软件开发实践中应用软件工程原则		C	0.5
SE_01_04	软件工程职业道德规范	理解软件工程从业人员需遵守的法律、法规和职业准则	在软件开发实践中遵循职业道德规范		C	0.5
SE_02	软件过程	理解软件过程的概念以及基于软件过程的软件开发	应用软件过程来指导软件开发实践		C	3
SE_02_01	软件生命周期	理解软件生命周期概念、各个阶段的任务和输出			C	1.5
SE_02_02	软件过程模型	理解软件过程模型的概念及典型的软件过程模型及其特点和适用场合，如瀑布模型、原型模型、迭代模型、增量模型、螺旋模型等	应用具体的软件过程模型来指导软件开发实践		C	1.5
SE_02_03	敏捷开发方法	理解敏捷开发方法的概念、思想和过程，及特定的敏捷方法，如Scrum框架	应用敏捷开发方法来指导软件开发实践		X	2
SE_02_04	持续集成与交付DevOps	理解DevOps的概念、方法、思想和过程			X	2

表 2 软件工程课程的知识点及学习要求（第 2 页/共 4 页）

标识	名称	能力要求			类别	学时
		一级	二级	三级		
SE_02_05	群体化软件开发方法	理解群体化软件开发方法的概念、思想和过程	应用群体化软件开发方法来开展软件开发实践		X	2
SE_03	软件开发方法	理解软件开发方法的概念、常见的软件开发方法	应用相应的软件开发方法来开展软件开发实践		C	3
SE_03_01	结构化软件开发方法	理解结构化软件开发方法的相关概念、思想、过程及建模语言	应用结构化软件开发方法来开展软件开发实践		X	3
SE_03_02	面向对象软件开发方法	理解面向对象软件开发方法的相关概念、思想、过程及建模语言UML	应用面向对象软件开发方法来开展软件开发实践		C	3
SE_04	软件需求分析	理解软件需求概念及分析方法、软件需求文档编制和评审	应用软件需求分析方法来开展软件需求分析实践	分析和评判软件需求模型和文档的质量	C	7
SE_04_01	软件需求分析基础	理解软件需求概念、类别及特点，软件需求分析的任务、目标、过程和原则			C	1
SE_04_02	软件需求获取方法	理解软件需求获取的任务、方法和原则	在软件开发实践中应用软件需求获取方法来获取软件需求		C	1
SE_04_03	面向对象需求分析方法	理解面向对象需求分析的概念、过程和策略，及软件需求的面向对象表示及模型	在软件开发实践中应用面向对象需求分析方法来分析软件需求，建立软件需求模型		C	4
SE_04_04	结构化需求分析方法	理解结构化需求分析的概念、过程和策略，以及软件需求描述语言、软件需求的结构化表示及模型	在软件开发实践中应用结构化需求分析方法来分析软件需求，建立软件需求模型		X	4
SE_04_05	软件需求文档	理解软件需求文档的作用、文档规范和编制要求	在软件开发实践中编写软件需求文档		C	0.5
SE_04_06	软件需求评审	理解软件需求评审的目的、要求、原则和方法	在软件开发实践中评审所编写的软件需求文档	分析和评判软件需求模型和文档的质量	C	0.5
SE_05	软件设计	理解软件设计概念及方法、软件设计文档编制及评审	在软件开发实践中应用软件设计方法来设计软件	分析和评判软件设计模型和文档的质量	C	7
SE_05_01	软件设计基础	理解软件设计概念、任务、过程、输出和原则			C	0.5

表 2 软件工程课程的知识点及学习要求（第 3 页/共 4 页）

标识	名称	能力要求			类别	学时
		一级	二级	三级		
SE_05_02	软件概要设计	理解软件概要设计概念、任务、要求和输出，以及软件体系结构概念、软件体系结构的类型和风格	在软件开发实践中开展软件概要设计		C	0.5
SE_05_03	用户界面设计	理解用户界面设计的概念、任务、原则和输出	在软件开发实践中开展用户界面设计		C	0.5
SE_05_04	软件详细设计	理解软件详细设计概念、任务、要求和输出，以及数据结构的逻辑设计、算法与处理逻辑的设计	在软件开发实践中开展软件详细设计		C	0.5
SE_05_05	面向对象软件设计方法	理解面向对象软件设计概念、过程和策略，以及面向对象软件设计模型及建模语言，如UML	在软件开发实践中应用面向对象软件设计方法来开展软件设计		C	4
SE_05_06	结构化软件设计方法	理解结构化概要设计语言、过程、模型和策略，以及结构化详细设计表示语言、过程、模型和策略	在软件开发实践中应用结构化设计方法来开展软件设计		X	4
SE_05_07	基于模式的软件设计方法	理解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	2
SE_05_08	软件设计文档	理解软件设计文档的作用、文档规范和编制要求	在软件开发实践中编写软件设计文档		C	0.5
SE_05_09	软件设计评审	理解软件设计评审的目的、要求、原则和方法	在软件开发实践中评审软件设计模型和和文档	分析和评价软件设计模型和文档的质量	C	0.5
SE_06	编码实现	理解编码实现基础、编码规范与代码风格、代码重用	在软件开发实践中遵循编码规范编写高质量代码	分析和评价程序代码的质量	C	3
SE_06_01	编码实现基础	理解编码实现的概念、任务、要求和原则			C	1
SE_06_02	编码规范与风格	理解程序代码的质量要求、编码规范、代码风格	在软件开发实践中遵循编码规范编写高质量代码	分析和评价程序代码的质量	C	1
SE_06_03	代码重用	理解程序代码重用的方式和方法、开源代码重用	在软件开发实践中重用代码		C	1
SE_07	软件测试	理解软件测试概念、过程与策略，以及软件测试技术、非功能测试技术	在软件开发实践中开展软件测试		C	6
SE_07_01	软件测试基础	理解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	1

表 2 软件工程课程的知识点及学习要求（第 4 页/共 4 页）

标识	名称	能力要求			类别	学时
		一级	二级	三级		
SE_07_02	软件测试过程与策略	理解软件测试过程、及单元测试、集成测试、系统测试、确认测试的实施策略，理解回归测试	在软件开发实践中应用软件测试策略		C	1
SE_07_03	软件测试技术	理解白盒测试技术、黑盒测试技术、面向对象软件测试技术、软件测试工具	在软件开发实践中应用软件测试技术和工具开展软件测试		C	4
SE_07_04	非功能测试技术	理解非功能（如性能、安全性、压力等）软件测试技术	在软件开发实践中开展非功能测试		X	2
SE_08	软件交付与部署	理解软件交付与部署的概念、任务与方法			X	2
SE_09	软件维护与演化	理解软件维护概念、软件维护过程与策略、软件维护技术、软件自动化维护			C	2
SE_09_01	软件维护基础	理解软件维护和和可维护性的概念、软件维护类别、软件维护副作用、影响软件可维护性因素			C	1
SE_09_02	软件维护过程与策略	理解软件维护的任务、过程、活动和原则，及软件维护的实施策略			C	0.5
SE_09_03	软件维护技术	理解程序理解、软件再工程、逆向工程、软件重构等软件维护技术			C	0.5
SE_09_04	软件演化	理解软件演化的概念、方式及策略选择			X	1
SE_09_05	软件自动化运维	理解软件版本更新、软件运行状态监控、软件运行优化、软件运维自动化工具			X	1
SE_10	软件项目管理	理解软件项目管理的概念、任务和方法	在软件开发实践中开展软件项目管理		C	3
SE_10_01	软件项目管理基础	理解软件项目管理的概念、任务、内容和原则			C	1
SE_10_02	软件配置管理	理解软件配置和配置项的概念，及软件配置管理的概念、任务、方法和工具	在软件开发实践中对软件制品等进行配置管理		C	2
SE_10_03	软件质量保证	理解软件质量保证的计划和方法	在软件开发实践中开展质量保证		X	2
SE_10_04	软件项目进度管理	理解软件项目进度的计划、实施、跟踪和调整	在软件开发实践中开展进度管理		X	2
SE_10_05	软件项目团队管理	理解软件项目团队的组织、交流和合作	在软件开发实践中开展团队管理		X	2
SE_10_06	软件过程管理与改进	理解软件过程管理、改进和评价的概念，及以CMM为代表的软件过程改进模型及思想			X	2

5.2 课程知识点学习要求

1）为达成课程教学的基本目标，应包括本文件第 5 章规定的 80%以上核心知识点，所选择的具有二、三级能力要求的核心知识点需占到所有具有二、三级能力要求的核心知识点的 60%以上，并依据所选择的知识点，实施相应的基础实践内容。

2）为达成课程教学的增强目标，在基本目标的基础上，应包括本文件第 5 章规定的 20%以上扩展知识点，所选择的具有二、三级能力要求的扩展知识点需占到所有具有二、三级能力要求的扩展知识点的 50%以上，并依据所选择的知识点，实施相应的基础实践内容，完成可选实践内容。

6 课程实践教学及要求

6.1 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知识点的理解，掌握并运用软件工程方法、技术和工具来开发软件系统，在实践中体验软件开发的实际场景、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等方面的能力以及养成良好的软件工程素养。

6.2 实践教学内容及要求

软件工程课程的实践教学须覆盖基础实践内容，以确保实践教学的成效以及达成课程教学的基本目标。

a） 基础实践内容

针对特定的应用，运用所学的软件工程过程、方法和技术，借助项目管理、软件建模、软件测试、协同开发等工具，开展软件需求分析、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，产生和输出多样化、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。

b） 可选实践内容

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、编码规范与代码质量、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、规模性等方面提出要求，培养解决复杂工程问题能力和创新能力等。

6.3 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通常不少于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学时），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，交流分享实践经验和成果。

7 课程考核要求

——课程考核要求应服从各个学校、院系的培养方案和教学实际需求。考核知识点应覆盖所有被选择作为该课程内容的知识点，考题难度应与该课程所选的各知识点能力要求、学时要求相符。

——采用卷面考试和实践考评相结合的方式，并考虑平时的作业和讨论。卷面考试可采用开卷或者闭卷的方式。课程实践的考评需根据实践的内容要求，实践成绩占课程总成绩的比例建议控制在30%-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），从软件制品质量、软件项目规模和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。