



软件工程一课程介绍

王林章、张天、王豫、李宣东

软件工程组

南京大学计算机学院

2025年08月29日



课前调查





本学期授课团队：软件工程组



■ 课程主讲人：

- 王林章 18951679130, lzwang@nju.edu.cn
 - 张 天 18951991220, ztluck@nju.edu.cn
 - 王 豫 18551415482, yuwang_cs@nju.edu.cn
 - 李宣东 18951679128, lxd@nju.edu.cn
-



南大软件工程组概况

- 团队：学科带头人：李宣东；教师15人，
 - 1位 国家杰出青年基金获得者、享受国务院特殊津贴
 - 2位CCF会士
 - 1位国家万人计划科技创新领军人才、2位万人计划青年拔尖人才、1位海外优青
 - 2位 教育部（跨）新世纪人才
- 人才培养
 - 硕士生：1981年起，近200人取得学位，50+人在读
 - 博士生：1991年起，50+人取得学位，50+人在读
- 平台
 - 计算机软件新技术全国重点实验室
 - 计算机学院、软件学院、智能软件与工程学院
 - 软件新技术与产业化协同创新中心
 - 软件新技术引智基地
- 教学：
 - 程序设计、软件工程、计算机程序设计语言、数据结构、算法、软件体系结构、软件方法学、编译原理、形式语言与自动机、嵌入式系统、软件可靠性方法、程序设计基础实验、软件工程综合实验等

李宣东:

- 软件可信性度量与评估体系
- 软件可信性增强技术



赵建华:

- 模型检验技术及其应用
- 代码验证与分析



王林章:

- 模型驱动软件测试
- 软件测试自动化及工具
- 非功能性建模与测试
- 嵌入式系统建模、仿真与测试

卜磊:

- 嵌入式混成系统模型检验与应用
- 信息物理融合系统在线验证与设计
- 软件代码符号化分析与验证



陈鑫:

- 基于形式推理技术的软件工程方法
- 软件测试技术
- 嵌入式软件开发与测试技术

汤恩义:

- 程序分析技术与应用
- 符号执行方法与动态分析方法的分析与改造
- 软件时间分析与测试
- 软件数值分析与验证



张天:

- 模型驱动工程
- 面向大数据的建模和转换
- 基于Hadoop的转换引擎

王豫:

- 程序静态分析动态测试
- 基于AI的缺陷预测



左志强:

- 程序分析与合成
- 大数据系统
- SAT求解问题

陆一飞:

- 程序测试与分析
- 移动应用漏洞检查



李重:

- 人工智能赋能的软件工程
- 面向人工智能的软件工程



潘敬学:

- 实时与混成系统的分析与验证
- 异步并发系统模型检验技术及其应用
- 嵌入式系统的建模、分析与验证

王轲:

- 程序语言、人工智能
- 程序分析/测试、编译优化



王佳娟:

- 实时系统形式化建模、验证与测试;
- 智能系统运行时监控与智能控制



刘尚清:

- 基于人工智能的软件漏洞检测和修复
- 智能化编程
- 模型的分析与可解释性



南大软件工程组-人才培养



- 重点院校**30+**人，其中获南大博士学位的**20+**人，国外知名大学**8**人
 - 南京大学（**16**人）、上海交通大学（**2**人）、东南大学（**2**人）、华东师范大学(**2**人)、南京航空航天大学、河海大学、苏州大学、北京电子科技学院、东华大学、香港理工大学
 - 美国普度大学、美国匹兹堡州立大学、美国博伊西州立大学、美国**UIUC**、美国马萨诸塞大学罗维尔分校、美国西南医学中心、卢森堡大学
- 产业界**130+**人，获得南大博士学位**10+**人
 - 重点企业：电力、电信、移动、人民银行、上海证券交易所
 - 研究所：**841**、**14**、**28**: **5**人
 - 全球著名企业：**Microsoft** (**7**人)、**Google US** (**2**人)、**IBM**1人、**Intel**(**2**人)、**HP**(**2**人)、**LinkedIn**(**2**人)、**Lucent**(**3**人)、**Ericsson**(**3**人)、**Oracle**1人、**Works Applications**(**2**人)
 - 知名国内企业：**华为**、中兴、趋势、腾讯、百度、阿里、淘宝、网易



南大软件工程组-人才培养



■ 国际知名学者：

- 翟成祥，85届硕士
 - 现为UIUC Prof., 美国总统基金奖获得者
- 徐殿祥，95届博士
 - 现为Boise State University 教授

■ 国内知名学者：

- **2名杰青，2位C9高校计算机系系主任**
 - 过敏意，83届硕士，曾任上海交通大学计算机科学与工程系系主任，国家杰出青年基金获得者
 - 李宣东，94届博士，南京大学智能软件与工程学院院长，曾任南京大学计算机科学与技术系系主任、软件学院院长，国家杰出青年基金获得者



南大软件工程组-人才培养



■ 工业界、创业杰出代表

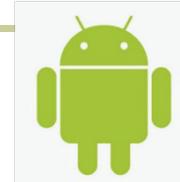
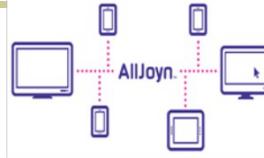
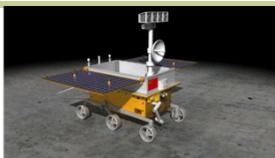
- 张云飞，**92届硕士**
 - 华为副总裁
 - **06年创办艾诺通信（苏州）有限责任公司**
- 袁晓东，**98届博士**
 - 美国甲骨文，趋势科技高级经理，
 - **14年创办南京云白科技，南京创业典范**
- 李勇，**02届博士**
 - 安氏科技副总裁
 - **14年创办多分科技**
-



软件工程研究与合作



- 研究对象：高可信软件与系统
 - 系统软件、移动应用、（航天、列控、车载）嵌入式系统、IOT、CPS、自动驾驶系统、智能系统
- 目标：符合预期、摆脱缺陷
- 途径：工件驱动
 - 规约、模型、代码、用例、系统
- 内容：提高软件开发效率、优化软件性能、保障软件可信
 - 面向学科前沿，研究软件工程理论和方法
 - 面向软件行业，研发实用软件技术与工具
 - 面向具体应用，提供解决方案





软件工程研究与合作



- 研究软件工程领域共性基础问题
 - 建模、设计开发方法、体系结构、程序设计语言
 - 系统建模，软件建模、程序语言扩充，程序逻辑扩展、形式化方法、程序综合与代码生成、编译优化与验证、程序理解与复用
 - 分析、测试、验证、监控、容错
 - 程序分析，模型验证，代码验证
 - 软件测试，错误定位，缺陷修复
 - 监控，容错
 - 性能优化技术：
 - 并行化：多核、分布计算、GPU、FPGA
 - 性能瓶颈/缺陷发现与优化
 - AI 4SE：AI赋能软件工程活动
 - SE4AI：AI模型可信、嵌入AI模型的软件系统可信优化、测试、缺陷预测
- 研发软件工程工具



授课团队-李宣东



- 南京大学：本（1981-1985）硕（1988-1991）博（1991-1994，软件工程）
 - 曾在联合国大学澳门国际软件技术研究所（UNU/IIST）、芬兰Turku Centre for Computer Science（TUCS）和美国CMU进行访问和合作研究。
 - 中国计算机学会软件工程专业委员会主任、国务院软件工程学科评议组成员
 - 国家杰出青年基金获得者（软件工程学）
 - 教学
 - 1994年开始软件工程课程教学
 - 形式语言与自动机
-



授课团队-李宣东



- 研究工作主要涉及计算机软件工程，近年来研究工作的重点包括软件建模与分析、软件测试与验证
- 科研项目：
 - 李宣东，软件定义的泛在操作系统与环境，国家重点研发计划项目，3000万元，2023.4-2027.4
 - 李宣东，支持软件可信性分析与评估的生产线，863重点项目课题，2007-2010，1009万元
 - 李宣东，基于认识与理解的软件可信性度量与评估技术研究，基金重大研究计划重点课题，2009-2013，260万
 - 李宣东，网构化软件可信评估技术与工具，863主题项目课题，2012-2014，644万元
 - 李宣东，复杂软件测试验证智能化关键技术与支撑环境，国家重点研发计划课题，2016-2019，491万
 - 李宣东，基于符号执行的复杂软件系统分析与验证，基金重点项目，2017-2020，300万



授课团队-王林章



■ 教育

- 2005.2 南京大学计算机科学与技术系获得计算机软件与理论 博士学位

■ 学术工作经历

- 2006.04 - 至今 南京大学计算机科学与技术系讲师、副教授、教授
 - 校学科办副主任、计算机软件新技术国家重点实验室 主任助理、副系主任
- 2005.03 - 2006.03 美国德克萨斯大学达拉斯分校访问学者

在学术共同体、学术会议任职，服务于软件领域学术交流和学科发展

学术任职

- **中国计算机学会系统软件专委会副主任**
- **ACM Sigsoft China Vice Chair (2019-2021)**
- **江苏省计算机学会软件专委会主任**
- **全国信标委软件与系统工程分技术委员会委员**

国际会议

- **程序委员会共同主席**：Internetware (2017、2018)
- **程序委员会委员**：IEEE SEKE、ACM SAC、IEEE QRS

国内会议

- **全国软件与应用大会程序委员会共同主席** (2017、2018、2019)
- **CCF中国软件大会执行委员会委员** (2020-2024)
- **CCF系统软件论坛发起人** (2016年起)
- **江苏省软件新技术学术会议主席** (2019、2020、2021)



授课团队-王林章



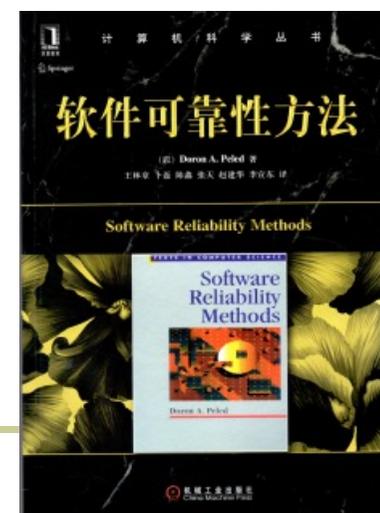
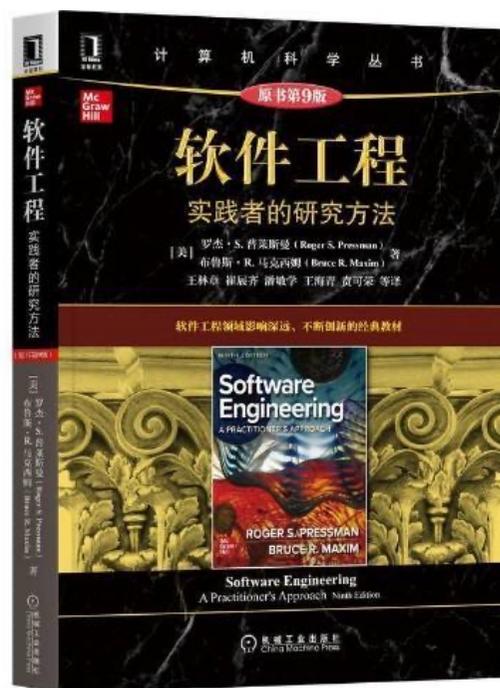
■ 教学

○ 课程

- 《软件测试》（2002-）
- 《软件工程》（2007-）
- 《软件体系结构》（2010-）
- 《软件工程综合实验》（2015-）

○ 教材

- 《软件可靠性方法》（2012年2月）
- 《软件工程导论》（2018）
- 《软件工程：实践者的研究方法（第9版）》（2021）





授课团队-王林章



长期从事复杂软件系统缺陷检测研究，主持多项国家级科研项目、企业核心技术攻关项目，近五年主持14项（经费1800余万元），包括国家自然科学基金重点项目、国家重点研发计划课题。



国家级科研项目

项目类别	名称	时限
国家自然科学基金重点项目	基于模糊测试的物联网设备固件漏洞检测技术研究	2021-2025
国家重点研发计划项目课题	智能制造软件形式化验证和性能优化技术	2018-2023
国家自然科学基金面上项目	模型驱动的移动应用测试方法	2015-2019
国家自然科学基金面上项目	交织场景驱动的信息物理融合系统行为建模与测试技术研究	2012-2015
基金重大研究计划培育项目	基于模型的嵌入式软件测试与验证及国产列控系统应用	2010-2011
国家自然科学基金青年项目	基于Aspect的软件非功能性规约的建模、测试和验证研究	2007-2009



企业核心技术攻关项目

项目来源	名称	时限
华为技术有限公司	静态分析技术	2015-2016
华为技术有限公司	代码分析平台与缺陷自动修复技术	2019-2020
华为技术有限公司	安全与功能类问题检测技术	2019-2020
华为技术有限公司	性能问题检测技术	2019-2020
华为技术有限公司	数通业务新指令自动发掘技术研究	2021-2022
华为技术有限公司	代码分析平台与缺陷自动修复技术(二期)	2021-2022
华为技术有限公司	可扩展的静态分析框架	2021-2022
华为技术有限公司	数据竞争检测工具	2022-2023
华为技术有限公司	漏洞利用情报动态感知预警研究	2022-2023
中国电子科技集团第14研究所	雷达软件测试设计方法与工具	2012-2013
中国电子科技集团第14研究所	雷达软件可复用测试用例库方法与工具	2013-2014
北京控制工程研究所	月球车GNC软件动态时序正确性保证	2012-2013
全球能源互联网研究院	基于模型驱动的软件测试技术研究	2015-2016



科学研究



■ 基于静态分析的方法与技术

- 程序理解/代码评审/代码优化/性能分析/编译优化
- 时间敏感性/循环/递归上界分析
- 基于扰动的数值精度分析
- 高精度静态分析（面向特定任务、特定缺陷）
- 缺陷检测：缓冲区溢出、内存泄漏、数据竞争、数组越界
 - 污点分析
 - 静态分析+ 深度学习
 - 高精度静态分析（面向特定任务、特定缺陷）：基于子路径分段分析
 - 静态分析+符号执行：
 - 静态分析+模糊测试
 - 静态分析+符号执行+仿真
- 基于静态分析的缺陷自动修复
 - 静态分析+ 程序合成
- 并行化分析
 - 基于依赖



科学研究



- 基于静态分析的程序结构可视化和模型重构技术
 - 程序及其依赖关系建模：AST/CFG/DFG/CG/PDG/VFG
 - 行为建模：TA、IA、HA、UML、状态机/设计模式/序列图。。。
 - 性质建模：CTL、LTL
- 基于高精度静态分析的缺陷检测
 - 基于污点分析的数组越界缺陷的静态检测方法
 - 污染数据驱动的Missing/redundant/wrong Check静态分析
 - 基于子路径分段分析的数据竞争检测方法
- 静态分析警报自动确认方法：基于符号执行/模糊测试/AI/混合
 - 数据竞争/内存泄漏/缓冲区溢出。。
- 基于静态分析的缺陷自动修复
- 基于静态分析与深度学习的缺陷预测
- 大规模代码增量分析技术



科学研究



■ 静态分析相关工作

○ 性能问题静态分析工具

- 包括内存重复写、冗余判断、循环中的冗余操作、函数调用开销过大等13个性能问题检测

○ 代码分析平台与缺陷自动修复技术

- 包括代码分析基础平台分析算法和技术 and C/C++ 程序特定的缺陷修复算法与技术

○ 针对值流分析的增量分析技术

- 基于LLVM IR，针对值流分析技术，设计并实现增量值流分析

○ 可扩展的静态分析框架

- 包括跨过程的数组下标保护正确性检查、跨过程的循环上界最大值判断、域敏感的污染分析能力、高精度冗余判断检测



科学研究



■ 系统化程序分析技术

■ 目标

- 高可扩展：对超大规模（千万行）软件的高精度静态分析
- 便于使用：前后端分离，提供系统化分析支持

■ 创新思路

- 程序分析理论与大规模数据处理相结合，构建高性能分析系统

■ 工作及系统

- Chianina: 基于演化图处理的流敏感分析系统 [PLDI'21]
- Grapple: 状态相关缺陷静态分析系统 [EuroSys'19]
- GpuSpa: 基于GPU加速的过程间静态分析系统 [TOCS'21]
- BigSpa: 基于分布式加速的过程间静态分析系统 [TPDS'21]
- Graspan: 基于核外计算的过程间静态分析系统 [ASPLOS'17]



科学研究



- 软件测试：工业界普遍采用的有效手段
 - 模型驱动的软件灰盒测试方法
 - 测试、仿真、验证、理解、评审方法
 - 非功能性建模和测试技术
 - 健壮性、实时性、安全性、可用性
 - 访问控制策略测试：XACML
 - 基于代码的自动测试技术
 - 可控符号执行技术、目标制导的混合执行技术
 - 基于子路径的充分度度量准则与工具



科学研究

- 软件测试：工业界普遍采用的有效手段
 - 漏洞检测：内存泄漏、数据竞争、缓冲区溢出、SQL注入、空指针引用、误差
 - 漏洞自动修复
 - 移动应用测试：模型驱动+跨平台+自动化
 - 恶意软件检测
 - 信息物理融合系统测试
 - 嵌入式系统（航天嵌入式系统、工控、列控、IOT、自动驾驶系统）
 - 演化软件的测试技术
 - 测试复用：基于变更的模型维护、测试脚本维护
 - 智能化分析、测试技术
 - 模糊测试技术：结构制导/接口制导/特征+行为制导



难点分析和创新思路

科学问题

难点挑战

创新研究思路





研究内容与目标

科学问题

如何解决大规模程序状态空间爆炸问题?

难点挑战

- 缺陷分析检测**
无法有效处理大规模程序
导致可扩展性差
- 缺陷警报确认**
无法充分探索程序路径,
导致误报多、效率低
- 缺陷定位修复**
关键环节需要人工介入,
导致精度低、难度大

复杂软件系统缺陷检测

创新研究思路

- 大数据处理
分布并行
- 动静态结合
精确制导
- 数据驱动
深度学习

研究内容

- 缺陷分析检测**
代码大规模数据处理
分段增量与并行加速
- 缺陷警报确认**
目标制导符号执行
目标制导模糊测试
- 缺陷定位修复**
深度学习缺陷定位
程序合成补丁模版

研究目标

形成方法技术创新突破, 自主研制工具, 服务国家重大需求



研究成果

技术创新

- ◆ 创新了复杂软件系统缺陷检测的理论与方法，在提高效率、增强可扩展性等方面取得突破；
 - 基于大规模计算的高可扩展性静态分析技术
 - 目标制导、动静态结合的缺陷警报确认方法
 - 基于图区间神经网络的缺陷定位与修复方法
 - 论文发表在CCF A类国际期刊国际会议ACM TOCS、IEEE TPDS、OOPSLA、PLDI、TSE、ICSE、ISSTA、ASE，获OOPSLA'20杰出论文奖、OOPSLA'13最佳论文奖、Internetware'18最佳论文奖

工具与知识产权

- ◆ 提出领域瓶颈问题解决方案，构建复杂软件缺陷检测体系
 - 研制了系列复杂软件自动化缺陷检测技术与工具
 - 获得中国授权发明专利22项、美国授权发明专利4项、中国软件著作权36项
 - 参与制定国家标准3项（GB/T 30847.1-2014、GB/T 30847.2-2014、GB/T 37970-2019）

应用与转化

- ◆ 服务国家重大工程与应用，助力企业关键软件技术攻关与升级
 - 服务于探月工程、国产列控系统/操作系统/编译器/雷达系统等
 - 8项授权发明专利（第一发明人）、3项软件著作权许可华为，许可费用1189万元



研究成果-技术创新突破

◆ 创新了复杂软件系统缺陷检测的理论与方法，在提高效率、增强可扩展性等方面取得突破

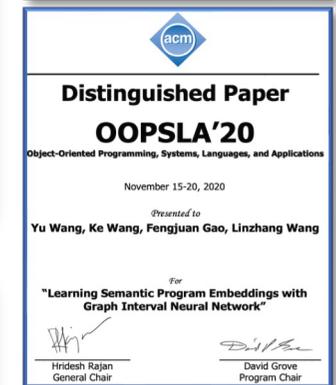
- 基于大规模计算的高可扩展性静态分析方法
- 目标制导、动静态结合的缺陷警报确认方法
- 基于图区间神经网络的缺陷定位与修复方法

◆ 高水平论文发表在国内重要期刊以及CCFA类国际期刊和国际会议上

- 系统软件：ACM TOCS'22、IEEE TPDS'21
- 程序语言：OOPSLA'13、OOPSLA'20、PLDI'21a、PLDI'21b
- 软件工程：TSE'22、ICSE'19、ISSTA'17、ASE'16、ISSTA'13
- 中国科学、计算机学报、软件学报

◆ 多篇论文获奖，被国际权威学者引用，产生了重要的国际影响

- 获OOPSLA'20杰出论文奖、OOPSLA'13最佳论文奖
 - 该CCFA类顶级国际会议自1986年召开36年来国内单位仅有的2次获奖
- Internetware'18最佳论文奖
- 被ACM/IEEE Fellow、重要国际会议主席等权威学者在TOSEM、TSE、OOPSLA、PLDI、ICSE、ISSTA等顶级期刊会议论文中引用。





研究成果-工具与知识产权

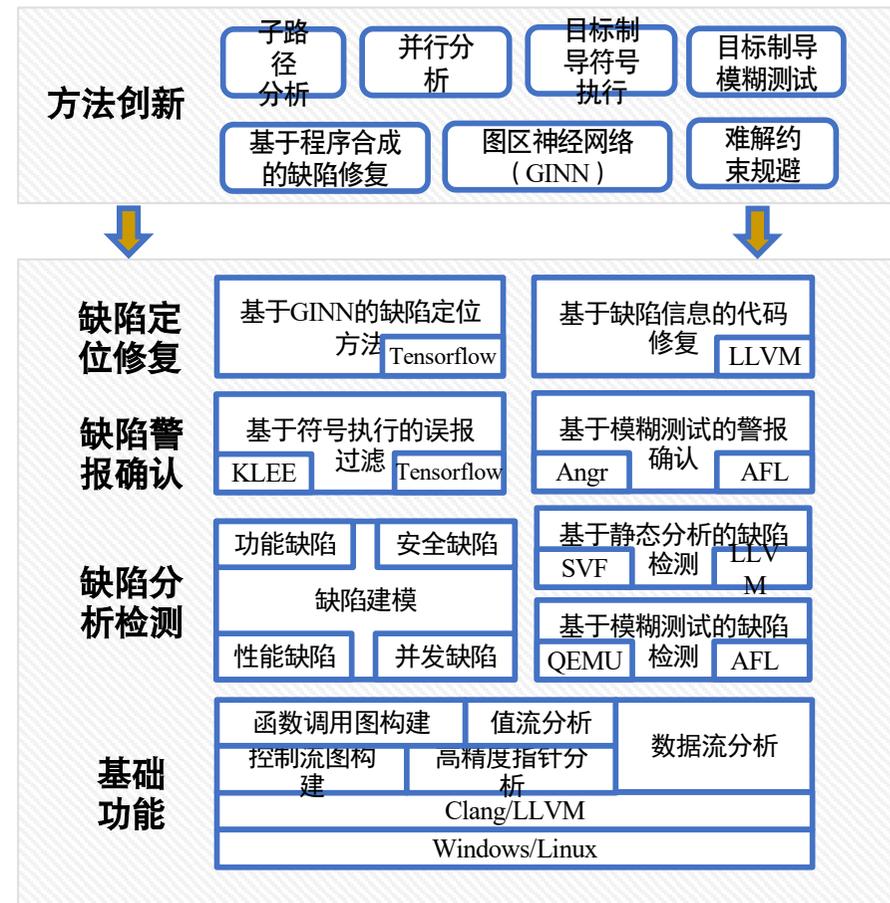
■ 自主研发系列复杂软件系统缺陷检测技术与工具，相关指标领先国际同类工具

- 检测内存类安全缺陷、并发类缺陷、性能类缺陷；
- 自动修复功能类、安全类、并发类、性能类缺陷。
- 分钟级时间内完成对千万行规模软件的精确分析；
- 缺陷警报确认实现完全自动化，确认率80%，准确率达到100%
- 缺陷定位的精准度比基准工作（GGNN、GREAT）提升了23%；

■ 自主知识产权

- 中国授权发明专利22项
- 美国授权发明专利4项
- 中国软件著作权36项
- 参与制定国家标准3项
 - GB/T 30847.1-2014
 - GB/T 30847.2-2014
 - GB/T 37970-2019

复杂软件系统的缺陷检测工具系列





研究成果-工具与知识产权

我们的工具检测了开源社区100余个基础软件项目（包括开源操作系统、分布式系统、编译器、数据库等）、代码总规模**五千余万行**，发现了**500余个真实软件缺陷**，获得7个CVE漏洞编号，9项工具和缺陷代码数据集开源，数据集已有1800余次下载。

- Vanguard: <https://github.com/stuartly/Vanguard>
- RegionFuzz: <https://github.com/stuartly/RegionFuzz>
- BovInspector: <https://bovinspector.github.io/project/>
- DATA: <https://github.com/ITWOI/DATA>
- GpuSpa, <https://github.com/scalablespa/GpuSpa>
- BigSpa, <https://github.com/scalablespa/BigSpa>
- JPortal, <https://github.com/JPortal-system>
- Grapple, <https://github.com/grapple-system>
- Chianina, <https://github.com/GraphFlow-system>
- https://figshare.com/articles/dataset/datasets_tar_gz/8796677

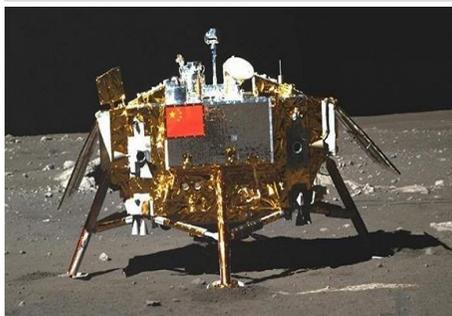
名称	代码行数	缺陷数量	CVE编号
Linux	17.5M	85	
Firefox	7.9M		
PostgreSQL	1.0M		
OpenSSL	519K		
Httpd	196K		
HDFS	546K	49	
Hadoop-MapReduce	568K	54	
ZooKeeper	206K	65	
Hbase	1.3M	191	
Libvips	318K	11	CVE-2021-27847
Libpostal	828K	17	
Tengine	578K	11	
Libsass	36K	18	
leveldb	25K	10	
zfs-fuse	213K	19	
zcash	1.4M	0	
selenium	2.3M	3	
Jasper	7.1M	4	CVE-2021-27845 , CVE-2018-9055 , CVE-2018-9295
Nasm	1.6M	4	CVE-2018-1000667 , CVE-2018-1000886
Libstan	6.2M	1	CVE-2018-1000654
总数	50.4M	542	7



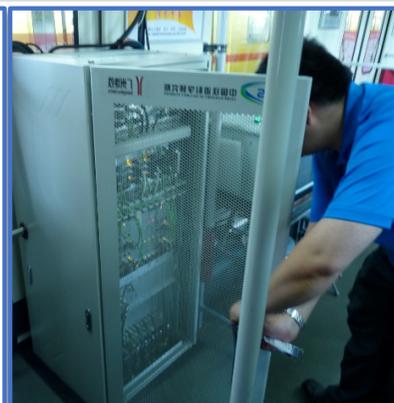
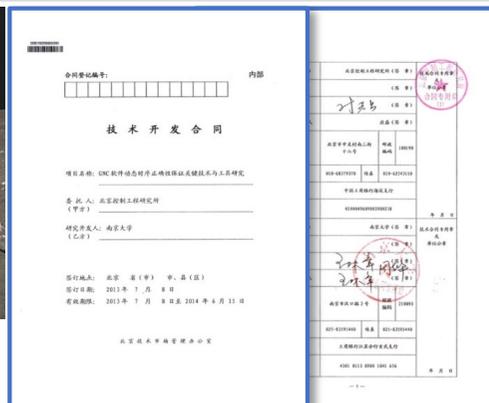
研究成果-应用与转化

服务探月工程、国产列控系统、操作系统、编译器、雷达系统等国家重大工程与应用，支撑中国航天探月工程实现航天嵌入式系统可信保障、国产列控系统通过国际安全标准SIL-4论证。

- 基于静态分析和动态仿真的数据竞争检测、中断驱动的复杂时序建模分析验证应用于探月工程型号系统月球车GNC子系统。
- 对国产CJ-7型CBTC列车控制系统车载、区域控制等6类46个场景进行了建模、测试和验证；工具部署于列车控制国家工程研究中心实物验证平台，成功捕获工程师植入的缺陷，及时预警，安全停车。



探月二期嫦娥三号着陆器



中国航天科技集团 第五〇二研究所
中国航天 有限公司第五研究院

科研合作和成果应用情况说明

中国航天科技集团有限公司第五研究院第五〇二研究所（简称502所）主要从事航天器控制系统和推进系统的研制工作。自2003年以来502所与南京大学在可信软件理论、方法与技术领域长期开展科学研究和成果转化与应用的合作。基于双方合作承担和各自承担的国家自然科学基金重大研究计划项目和重点基金项目、863计划主题项目等重要科研项目，在基于认识与理解途径的软件可信度量与评估体系及支撑技术、航天嵌入式软件可信保障集成环境和示范验证与应用、CPS系统设计与实现技术、复杂软件测试与验证智能化关键与支撑环境等方面开展合作研究和成果应用，相关工作在该所承担的星载软件项目中得到实际验证与应用，达到预期效果。

特此说明。

中国航天科技集团有限公司第五研究院第五〇二研究所
2019年10月30日

CASC



研究成果-应用与转化

通过承担华为10余项核心技术攻关项目，支持华为多部门完成自主可控工具升级，包括**华为数通构建安全和性能缺陷检测工具**、**华为无线构建面向安全缺陷的静态分析框架**、**华为云BU构建静态分析和修复框架**等，许可8项发明专利（第一发明人）和3项软件著作权（许可总金额1189万元），助力华为提升国产化替代以及商用工具不具备的能力、应对美西方封锁。

	面向安全缺陷、性能缺陷的检测工具	面向安全缺陷的静态分析框架	静态分析与修复框架	序号	许可华为的专利/著作权
华为数通	Cachelin e缺陷 ... 低效内存布局 ... 内存泄露 ... 重复内存释放 ... 新指令分析 & 检测	悬垂指针 ... 循环上界异常 ... 数组越界 ... 污染的除数	数据竞争 ... 内存泄露 ... 内存溢出 ... 空指针解引用	1	一种基于测试的静态分析误报消除方法 (ZL201310165634.8)
	性能缺陷检测 ... 安全缺陷检测	高精度污点分析 ... 约束求解器 ... 控制流/数据流分析	值流分析 ... 数据流分析 ... 类继承关系分析 ... 基于AST的代码转换	2	基于路径片段频谱的符号执行搜索方法 (ZL201410055286.3)
	Clang/LLVM	Clang/LLVM	CFG构建 ... 精度组合的指针分析 ... 缺陷定位 ... 模板构建	3	基于符号执行路径剪枝的缓冲区溢出漏洞自动检测方法 (ZL201510161507.X)
华为无线			基础分析框架 ... 修复框架	4	一种基于代码变更的软件模型同步方法 (ZL201310135857.X)
			C/C++程序解析工具	5	动静态结合的中断驱动程序数据竞争检测方法 (ZL201410324303.9)
				6	设计模式制导的爪哇代码评审方法 (ZL201310264462.X)
华为云BU				7	基于代码生成和符号执行的访问控制策略测试自动生成方法 (ZL201310711611.2)
				8	一种基于活动图模型的系统行为仿真方法 (ZL201310168258.8)
				9	Grapple-用于大规模软件系统中状态相关缺陷的静态检测工具软件 (2019SR0452378)
			10	GpuSpa-基于GPU的大规模软件高精度静态分析系统 (2019SR0452369)	
			11	Chianina-基于大规模演化图计算的流敏感和上下文敏感静态程序分析系统 (2020SR1850284)	



助力企业实现关键软件技术攻关和升级

在产学研联合技术创新与攻关中积极发挥引导与组织推动作用，汇集了华为操作系统、编程语言与编译器、系统安全等方面60余项技术需求，协调组织了50余位专家学者重点攻关。

■ 华为-南京大学下一代程序语言创新实验室

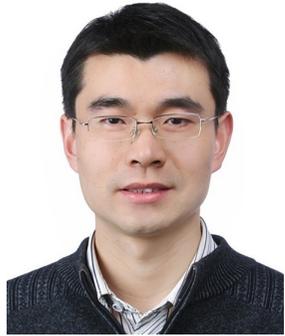
- 华为编程语言与编译器实验室，投入1580万元，连续支持3年（2019-2022）
- 担任**首任执行委员会主任**
- 3年来，支持南京大学29位专家解决编程语言与编译器相关技术问题

■ CCF华为胡杨林基金系统软件专项

- 华为中央软件院，投入1500万元，连续支持3年（2021-2023）
- 担任**首任技术委员会主任**
- 2年来，已经支持了全国从事系统软件相关工作研究的25位专家，解决覆盖操作系统、编译器、系统安全等方面的技术问题。



授课团队-张天



张天 教授 博士生导师

- 院系授课：
 - 本科：软件工程、软件工程综合实验、计算机程序设计语言
 - 研究生：软件开发方法学（计算机学院、软件学院）
- 研究方向：
 - 计算机视觉、自然语言处理、机器学习等多领域交叉的智能化软件工程技术
 - 发表顶会顶刊论文20余篇（获ISSTA20杰出论文奖），受理授权专利30余项，软件著作权30余件
- 企业合作与成果转化：
 - 腾讯（微信）、华为（开源、鸿蒙）、字节跳动、OPPO、小米



授课团队-张天



近期参与和主持的国家课题

1. 国家工信部，2023年软件平台-开源社区软件物料清单（SBOM）平台，300万
2. 国家重点研发计划重点专项“先进计算与新兴软件”课题“软件定义的泛在操作系统与环境”，2023.6-2026.5
3. 国家自然科学基金重点项目，面向移动应用隐私保护的需求规约与合规验证，62232014，2023年01月-2027年12月，284万
4. 国家自然科学基金重大项目，61690204，面向演化的群智能化软件建模与构造方法，2017.01-2021.12，400万
5. 国家重点研发计划重点专项，2017YFB1001801，可成长的智能化网构软件范型理论、方法与技术研究，2017.10-2021.09，750万
6. 国家重点研发计划重点专项，2017YFA0700604，智能制造软件形式化验证和性能优化技术，2018.5-2023.4，700万



授课团队-张天

企业合作与在研项目

- 华为，基于依赖模型分析的C/C++编译优化，50万
- 华为，基于内源代码的软件资产库构造与查询，30万
- 华为，数字架构逆向与模型构造研究，60万
- OPPO，智能化自动探索与脚本修复，两个课题280万
- 华为鸿蒙，界面异常检测，150万
- 字节跳动，测试自动生成，60万
- 腾讯，微信脚本自动修复，20万



授课团队-王豫



- 王豫，南京大学准聘助理教授
- 教学：程序设计基础、高级程序设计实验课、软件工程、软件工程综合实验
- 主要研究方向是程序分析、缺陷预测，智能化软件工程
- 正在承担编译器测试（基金重点项目）、基金面上项目、代码数字化分析（华为项目）
- 研究成果
 - 在PLDI、OOPSLA、TOSEM、TSE、ISSTA、JCST、软件学报国内外高水平期刊会议上发表10篇论文
 - 获得OOPSLA 2020 杰出论文奖，是当年国内唯一，也是该会举办以来国内第二次获奖



本学期授课团队



■ 助教

- 尹诗程, 1115204116@qq.com
- 黄逸飞, 2521065305@qq.com
- 周元昊, 1214549195@qq.com
- 黄宗博, 1398823565@qq.com
- 李铮, 602022320003@smail.nju.edu.cn



提纲



- 背景
- 为什么要学习《软件工程》？
- 课程计划
- 选课要求
- 课程考核



背景



■ 软件很“火”

○ 新一代信息技术：

- 软件：《鼓励软件产业和集成电路产业发展的若干政策》（2000年）
- 云计算
- 大数据
- 互联网、移动互联网、物联网：互联网+行动计划（2015年）
- 人工智能：人工智能+行动计划（2025年）

○ 软件使能一切

○ 软件定义一切

○ 软件成为基础实施

■ 软件从业人员很风光



背景



- 2019年以来“搞软件的”很尴尬
 - 2019年5月16日美国制裁华为
 - “中国制造”面临挑战：“缺芯少魂”、关键技术被“卡脖子”，重要基础软件缺少中国标签
 - 系统软件：OS、Compiler、程序设计语言
 - 基础软件：DB、大数据系统、AI系统...
 - 工具软件：VS, xcode, 开发环境, ...
 - 应用软件：Adobe, office,
 - 工业软件：EDA、CAD....
 - 根本原因
 - 外因：人家做得早、封锁技术
 - 内因：低成本使用，重视不够、投入不够、能力不够



背景



- 2023年以来“搞软件的”很慌：AI何时替代程序员？
 - AI for SE：支持软件工程任务，智能辅助，代码补全、代码搜索
 - ML for SE、DL for SE
 - AIGC：LLM4SE，软件自动化带来新的途径
 - ChatGPT：需求分析、代码生成、测试生成、程序修复、代码优化
 - Deepseeker



已完成搜索

以下是可以用于代码生成的一些主要的代码大模型：

1. **Code Llama** - Meta公司发布的开源模型，专注于代码生成任务，具有三个不同参数规模的版本（7B, 13B, 34B），并支持多种编程语言。
2. **StarCoder** - HuggingFace BigCode团队推出的开源代码大模型，同样支持多种编程语言。
3. **CodeGeeX** - 一款大规模多语言代码生成预训练模型，拥有130亿参数，能够根据自然语言描述生成代码。
4. **CodeGemma** - 由Google发布的代码生成和理解模型系列，包括2B预训练模型、7B预训练模型及7B指令微调模型。
5. **Salesforce CodeT5** - Salesforce公司发布的模型系列，包括CodeT5-small, CodeT5-base, CodeT5-large, 和 CodeT5-large-ntp-py。
6. **Salesforce CodeGen/CodeGen2** - 另一系列由Salesforce发布的编程模型，具有多个参数规模版本，并且有不同的训练数据集版本。
7. **Amazon CodeWhisperer** - 亚马逊发布的用于代码生成、参考跟踪和安全扫描的内部模型。

这些模型都能够根据给定的自然语言描述或者代码片段生成相应的代码，有的模型还具备代码补全、代码调试等功能。值得注意的是，这些模型的可用性和具体功能可能会随着版本更新而发生变化，因此建议查阅最新的官方文档或公告来获取最新的信息。

李彦宏：以后不会存在“程序员”这种职业了

上观 2024-03-10 13:03



3月9日，百度创始人、董事长兼首席执行官李彦宏在央视《对话》·开年说的访谈中指出，基本上说，以后其实不会存在“程序员”这种职业了，因为只要会说话，人人都会具备程序员的能力。“未来的编程语言只会剩下两种，一种叫做英文，一种叫做中文。”



背景



- 软件不断发展：机遇与挑战并存
 - 1956年“软件”诞生=》1968年“软件工程”元年=》软件定义
 - 影响世界-》改变生活-》定义一切
 - 软件危机-》高质量-》高可信
- 软件系统的规模和复杂性持续上升
 - 规模化：LOC-》KLOC-》MLOC、千万行、亿行
 - Linux 4亿行（内核、驱动、基础包）
 - 淘宝系软件：6亿行
 - 华为系软件：20亿行
 - 持续演化=》开源+生态化



背景



■ 学软件的不服气：我们具备的能力VS做软件需要的能力

○ 软件领域企业与高校的合作历程

○ 来自软件企业的声音，期待“极致的软件能力”：

■ 坚实的理论基础

■ 高超的算法能力

■ 卓越的编程能力

■ 协同的管理能力

■ 可信的质量能力

○ 飞控：适航论证

○ 列控：SEL4论证

○ 无人驾驶汽车：L4论证

○ . . .

○ 我们：？

“我们企业已经领先高校很多！”



作为研究领域的“软件工程”

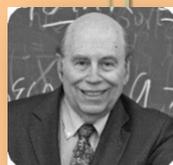
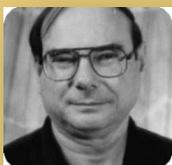


工具与环境

基础理论



形式化方法、结构化方法、面向对象方法、优化方法



程序逻辑

- 程序语言规约与扩充
- 程序逻辑扩展

方法与技术

- 编译构造与优化
- 程序分析
- 模型软件
- 监控与容错
- 程序理解与复用

设计开发方法

程序分析

编译与优化



EDSGER DIJKSTRA, 1972 Turing Award
Illuminating perception of problems at the foundations of program design



ROBERT W FLOYD, 1972 Turing Award
Methodologies for the creation of efficient and reliable software



ALAN KAY, 2003 Turing Award
Object-oriented programming languages, Smalltalk, personal computing



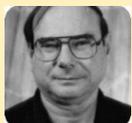
ROBERT W FLOYD, 1972 Turing Award
Automatic program verification and synthesis, and analysis of algorithms



C.A.R. HOARE, 1980 Turing Award
Definition and design of programming languages, Hoare Logic



ARTHUR MILNER, 1991 Turing Award
A practical tool for machine assisted proof construction



AMIR PNUELI, 1996 Turing Award
Temporal logic and contributions to program and system verification



ALAN J PERLIS, 1966 Turing Award
Advanced programming techniques and compiler construction



FRANCES ALLEN, 2006 Turing Award
The theory and practice of optimizing compiler techniques



DONALD KNUTH, 1974 Turing Award
The analysis of algorithms and the design of programming languages



NIKLAUS E. WIRTH, 1984 Turing Award
For developing a sequence of innovative computer languages, PASCAL



PETER NAUR, 2005 Turing Award
Algol 60, compiler design and the art and practice of computing programming



JOHN E HOPCROFT, 1986 Turing Award
Foundations of programming language and system design



JOHN BACKUS, 1977 Turing Award
Contributions to the design of practical high-level programming systems



DENNIS RITCHIE, 1983 Turing Award
Development of generic operating systems theory, Unix



KEN THOMPSON, 1983 Turing Award
Development of generic operating systems theory, Unix



EDMUND CLARKE, 2007 Turing Award
Developing Model-Checking into a highly effective verification technology



ALLEN EMERSON, 2007 Turing Award
Developing Model-Checking into a highly effective verification technology



JOSEPH SIFAKIS, 2007 Turing Award
Developing Model-Checking into a highly effective verification technology

设计开发方法



EDSGER DIJKSTRA, 1972 Turing Award
Illuminating perception of problems at the foundations of **program design**



ROBERT W FLOYD, 1972 Turing Award
Methodologies for the creation of efficient and **reliable software**



ALAN KAY, 2003 Turing Award
Object-oriented **programming languages**, Smalltalk, personal computing

程序分析



ROBERT W FLOYD, 1972 Turing Award
Automatic **program verification** and **synthesis**, and **analysis** of algorithms



C.A.R. HOARE, 1980 Turing Award
Definition and design of **programming languages**, Hoare Logic



ARTHUR MILNER, 1991 Turing Award
A practical tool for machine assisted **proof construction**



AMIR PNUELI, 1996 Turing Award
Temporal logic and contributions to **program and system** verification

编译与优化



ALAN J PERLIS, 1966 Turing Award
Advanced **programming** techniques and **compiler** construction



FRANCES ALLEN, 2006 Turing Award
The theory and practice of optimizing **compiler techniques**



DONALD KNUTH, 1974 Turing Award
The analysis of algorithms and the design of **programming languages**



NIKLAUS E. WIRTH, 1984 Turing Award
For developing a sequence of innovative **computer languages**, PASCAL



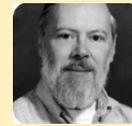
PETER NAUR, 2005 Turing Award
Algol 60, compiler design and the art and practice of **computing programming**



JOHN E HOPCROFT, 1986 Turing Award
Foundations of **programming language and system design**



JOHN BACKUS, 1977 Turing Award
Contributions to the design of practical **high-level programming** systems



DENNIS RITCHIE, 1983 Turing Award
Development of generic **operating systems** theory, Unix



KEN THOMPSON, 1983 Turing Award
Development of generic **operating systems** theory, Unix



EDMUND CLARKE, 2007 Turing Award
Developing Model-Checking into a highly effective **verification technology**



ALLEN EMERSON, 2007 Turing Award
Developing Model-Checking into a highly effective **verification technology**



JOSEPH SIFAKIS, 2007 Turing Award
Developing Model-Checking into a highly effective **verification technology**



作为学科专业的“软件工程”



- 1968年“软件工程”
- 南京大学：
 - 1958年开始软件、计算机的研究与教育
 - 1981年开始在计算机科学与技术一级学科下，计算机软件与理论方向培养硕士、博士，开始软件工程方向研究
 - 2011年“软件工程”成为国家一级学科，开始独立招生培养

	权重	60%	40%
序号	学科名称	计算机科学与技术	软件工程
1	北大	1 A+	1 A+
2	清华	1 A+	1 A+
3	浙大	1 A+	1 A+
4	北航	1 A+	1 A+
5	南大	1 A+	2 A
6	上交	1 A+	
7	西工大	1 A+	



助力软件行业长期生存与发展 “软件工程”



- 百年企业的生存发展秘诀：成熟的产品过程
 - 不断技术创新、满足客户/用户不断变化的需求
- 百年软件企业**VS**软件百年：成熟的软件工程
 - 过程、方法、工具、文档、价值观
 - 作为基础设施的软件
 - 高质量软件、满足客户/用户不断变化的需求



为什么要学习《软件工程》



- 学习《软件工程》，掌握构建软件系统之道，提升软件能力
 - 前人怎么做？ =》 软件工程50+年经验与教训
 - 我们能做什么？ =》 构建符合预期、摆脱缺陷的软件
 - 未来会怎样？ =》 软件百年？

Bill Gates: 课程作业发表为高水平论文
代码引领软件时代，DOS，Windows。。。

Linus Benedict Torvalds: Linux启动了开源生态

○ ○ ○ ○ ○



为什么学习《软件工程》？



- 工程化素养训练：程序员—》工程师
 - 规模化软件-》工程化开发：引导学生规范地开发软件
 - 培养学生工程化软件项目的能力：软实力与硬技能
 - 过程
 - 方法
 - 工具
 - 文档
 - 习惯
 - 责任



为什么学习《软件工程》？



■ 以史为鉴：软件MBA

- 计算机改变人类历史，软件改变计算机的历史，软件工程改变软件的历史
- 学习软件工程，就是了解软件失败的历史，**找到解决问题的途径**，便于预测将来



为什么学习《软件工程》？



终极目标：掌握软件工程知识体系，培养工程化习惯，增强软件开发的可控度和可信度

- 更好地开发软件： **Develop software better**
- 开发更好的软件： **Develop better software**

- 大模型时代。 。 。 。 。



课程计划



■ 理论教学

- 系统地掌握软件工程的基本原理、方法和技术，从而对软件工程有一定层次的理性认识；

■ 实验教学

- 学生能够深入理解软件工程的基本概念和方法，掌握利用具体过程、方法与工具完成软件工程管理和技术活动，培养学生分析和解决问题的能力

■ AI赋能

- 知识、技能
 - 助手：解决问题
-



课程计划



- 教学和实践并重，注重理论和实践结合，以实践驱动理论学习，用理论指导实践。
 - 课程按1.5: 1配置理论课与实验课的课时
 - 课堂教学采用启发式、研究型教学方法，将核心理论知识以粗线条的方式教学，提高课程的宏观性，促进学生课后自学，增加阅读量；
 - 实验采用**阶段性软件开发任务**来驱动，提高学生利用软件工具完成软件工程任务的水平。



课程计划-授课内容



软件工程知识体系

序号	知识域
K1	软件需求
K2	软件设计
K3	软件构造
K4	软件测试
K5	软件维护
K6	软件配置管理
K7	软件工程管理
K8	软件工程过程
K9	软件工程工具与方法
K10	软件质量



课程计划-授课内容



■ 软件工程理论讲授安排与知识域覆盖

序号	软件工程理论	覆盖知识域
1.	软件工程引论：从程序设计到软件工程	K1、K2、K3、K4、K5、K10
2.	软件过程模型和新兴过程方法	K1、K2、K3、K4、K5、K10
3.	需求与需求工程	K1、K2、K3、K6、K7、K8、K9
4.	软件设计与架构建模	K1、K2、K3、K6、K7、K8、K9
5.	软件实现与构建	K3、K4、K5
6.	软件分析、测试与验证技术	K1、K2、K3、K4
7.	软件支持、维护、演化与 DEVOPS	K5
8.	软件工程管理：项目、过程、配置、质量等管理	K6、K7、K8、K9、K10
9.	软件工程展望：研究进展与趋势	K1、K2、K3、K4、K5、K6、K7、K8、K9、K10



课程计划-授课内容



■ 软件工程实验教学内容设置与知识域覆盖

序号	实验	覆盖知识域
1.	软件需求分析实验	K1、K9
2.	软件设计与建模实验	K2、K9
3.	软件实现与构建实验	K2、K9
4.	软件分析实验	K4、K9
5.	软件测试与修复实验	K4、K5、K9、K10



授课计划



课次	周次	日期	软件工程理论（仙 I -319）	软件工程实验 （地址：基础实验楼乙124）	讲授者
1.	1	8.29	(1) 软件工程课程介绍； (2) 软件工程引论：从程序设计到软件工程；		王林章
1.	2	9.5	软件过程模型和新兴过程方法		张天
1.	3	9.12	软件方法与技术发展回顾与展望 需求与需求工程		李宣东张天
1.	4	9.19		实验1: 软件需求分析 内容：需求分析、基于大模型的需求分析、需求管理工具 提交：软件需求实验报告	王豫 助教
1.	5	9.26	软件设计与架构（1）		张天
1.	7	10.10	软件设计与架构（2）		张天
1.	8	10.17		实验2: 软件设计与建模 内容：架构设计、功能设计、UI设计、数据库设计活动，图形化建模工具、UI设计工具、数据库设计工具、PlantUML工具； 提交：软件设计与建模实验报告	王豫 助教
1.	9	10.24	软件实现与构建；		王豫
1.	10	10.31		实验3: 软件实现与构建 内容：编码、编译、调试活动，IDE、构建与调试工具、Git 提交：软件实现实验报告	王豫 助教
1.	11	11.7	软件分析、测试与验证（1）		王豫
1.	12	11.14	软件分析、测试与验证（2）		王豫
1.	13	11.21		实验4: 软件分析 内容：基于静态分析、大模型检测项目缺陷、静态分析工具评估 提交：软件分析实验报告	王豫 助教
1.	14	11.28	软件支持、维护、演化与DEVOPS；		王豫
1.	15	12.5		实验5: 软件测试与修复 内容：白盒测试、黑盒测试、代码验证，测试和验证、基于大模型测试生成、修复 提交：软件测试和修复实验报告	王豫 助教
1.	16	12.11	软件工程管理：项目、过程、配置、质量、集成与构建		王林章
1.	17	12.19	大模型时代软件工程		李宣东
1.	18	12.26	课程总结与答疑、随堂考试		王林章、张天、王豫



(1) 软件工程引论：从程序设计到软件工程

- 软件方法和技术发展回顾与展望
- 工程化方法及其在规模化软件领域的引入：软件工程
 - 软件/程序设计发展历程
 - 软件及其内涵
 - 软件危机
 - 软件工程基本原理和概念
 - 软件工程要素
 - 软件工程目标、原则
 - 软件工程规范



(2) 软件过程模型和新兴过程方法



- 软件过程模型
- 新兴过程方法



(3) 软件需求及需求工程



- 需求工程过程
- 需求获取
- 需求分析：
 - 结构化分析
 - 面向对象分析
- 需求规约与文档化
- 需求验证：需求评审
- 需求管理技术与工具



(4) 软件设计与架构



- 软件设计与建模
 - 传统建模
 - 统一建模语言UML
- 软件功能设计
 - 软件设计的原理和原则、方法和模型
 - 主流的软件设计方法：结构化设计方法和面向对象设计方法；
- 软件体系架构设计
- UI设计
- 设计模式；介绍设计过程中的设计模式及其应用方法。
- 软件设计规约与文档化
- 软件设计规约分析与验证：评审方法与形式验证方法
- 软件设计与建模工具



(5) 软件实现与构建



- 程序设计过程
- 程序设计方法
- 程序设计风格与优化
- 代码规范与评审
- 软件构建与调试
- 集成开发环境



(6) 软件分析、测试和验证



- 代码分析技术
- 软件测试技术
 - 软件测试方法
 - 黑盒方法
 - 白盒方法
 - 灰盒方法：模型驱动软件测试方法
 - 软件测试阶段
 - 软件测试规约与制品文档化
- 软件验证技术
- 分析、测试、验证工具



(7) 软件支持、维护、演化与DEVOPS



- 软件支持
- 软件维护
- 软件演化
- DevOps



(8) 软件工程管理



- 过程与过程管理
- 项目管理
- 配置管理
- 质量与质量管理
- 软件工程管理工具



软件工程实验



■ 实验（LAB）

1. 软件需求分析实验
2. 软件设计与建模实验
3. 软件实现与构建实验
4. 软件分析实验
5. 软件测试与修复实验

■ 实验要求

- 结合一小型软件开发需求，按照实验要求完成实验任务，提交实验报告。



软件工程实验



- L1: 软件需求分析
 - 该实验旨在掌握软件需求分析方法及其实施过程，对不同的需求分析方法进行比较分析，从而能在实际应用中选择合适的分析方法完成系统的分析；给定一个软件开发任务的软件需求片段，选择任意一种软件需求分析方法（如结构化分析方法、面向对象分析方法），对所选择的部分需求进行分析和表达，分析体会不同需求分析方法各自的特点、要求和结果。
 - 任务：需求分析活动、**基于大模型**的需求分析、需求管理工具



软件工程实验



■ L2: 软件设计与建模

- 该实验旨在学习和掌握软件设计方法、软件图形化建模方法及其实施过程，对不同的设计方法进行比较分析，从而能在实际应用中选择合适的设计方法完成系统的设计；给定软件的需求片段，选择任意一种软件设计方法（如结构化设计方法、面向对象设计方法），完成相应的软件设计（包括功能设计、架构设计、界面设计、数据设计等方面），分析体会不同设计方法各自的特点、要求和结果。
- 任务：
 - 架构设计、功能设计、UI设计、数据库设计活动，图形化建模工具、UI设计工具、数据库设计工具
 - 熟悉基于PlantUML工具的UML设计



软件工程实验



■ L3: 软件实现与构建

- 该实验旨在学习和掌握软件实现与构建方法及其实施过程；在相应工具的支撑下，完成各类实现、调试、构建活动
- 任务：
 - 编码、编译、调试活动，**IDE**、构建与调试工具
 - 使用辅助工具进行软件配置管理，实现软件配置项标识、变更控制、版本控制、配置审核和配置管理报告等管理活动
 - 使用**Git**等协同开发环境进行项目开发中的代码管理
 - 使用辅助工具进行软件实现，对辅助工具的能力进行评价



软件工程实验



■ L4: 软件分析

- 该实验旨在学习和掌握软件分析、测试的基本方法和实施过程；
 - 使用静态分析工具对代码进行自动化分析
 - 运用静态分析工具对较大规模的开源项目进行自动化分析，评估静态分析工具的误报率，评估静态分析工具在分析本实验项目（小项目）和开源项目（大项目）时的能力区别。撰写分析报告
 - 使用大模型辅助软件分析
- 内容：基于静态分析、**大模型检测项目缺陷**、静态分析工具评估



软件工程实验



■ L5: 软件测试与修复

- 该实验旨在学习和掌握软件分析、测试的基本方法和实施过程；
 - 运用黑盒测试方法对软件进行测试，选择任意软件系统（包含需求和可运行系统），根据软件需求，使用常规黑盒测试方法设计测试用例，并执行测试，记录测试结果
 - 运用白盒测试方法对软件进行测试，选择任意软件系统（包含可编译源代码），根据源代码进行结构分析，能够根据程序的逻辑结构设计测试用例，并使用白盒测试工具辅助测试，收集相应代码覆盖（如语句覆盖、分支覆盖）
 - 使用大模型辅助测试用例生成
 - 基于软件分析、测试的结果，对代码进行修复
- 内容：白盒测试、黑盒测试、代码验证，测试和验证、**基于大模型测试生成、修复**



软件工程实验



- 详细实验说明会在实验课时发布，并有助教在场答疑



参考资料



■ 参考资料:

- 《软件工程—实践者的研究方法（第9版）》（王林章、崔展齐、潘敏学译）
- 《软件工程导论》（崔展齐、潘敏学、王林章译）
- 《Essentials of Software Engineering(44)》
- SWEBOK3.0 <https://www.computer.org/w>

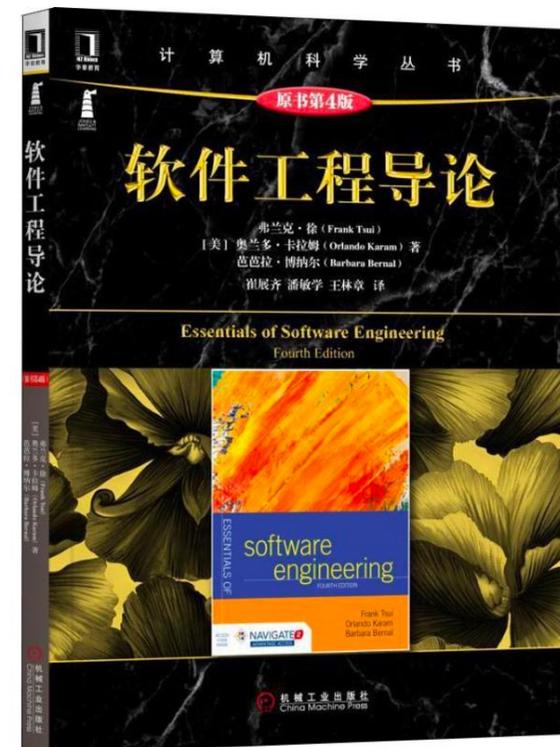
■ 往年课件

■ 网络资源

- OOSE、MDA、UML

■ MOOC-SE

- 国外大学软件工程课程
- 国家精品课程





选课要求



- 能快速学会使用相关软件和工具
- 能阅读中文、英文文献，撰写报告
- 修完《程序设计》、《数据结构》（《数据库》、《算法设计》）等课程
- 掌握至少一种（面向对象）程序设计语言



选课要求



- 学习方法：
 - 书中学——完整看一本书（任意一本软件工程参考书）
 - 听中学——完整听完一位老师讲课（现场、视频）
 - 做中学——完整地走完一个项目，系统地将所学知识用于开发
 - 课堂：到课率、提问，课堂测验
 - 课后：课后阅读、在线学习、作业
-



实验要求



- 形式：
 - 课前任务发布
 - 准备实验和报告
 - 课上完成实验任务、答疑、检查、提交报告
 - **独立**完成实验任务、撰写实验报告



课程考核



考核分类	内容	形式	比例
理论 (60%)	课堂	出勤、课堂提问、测验	10%
	作业	讲授内容 (理论部分的知识点) 相关的开放性和课后作业	20%
	期末考试	开卷考试	30%
实践 (40%)	个人实验	完成实验, 形成实验报告	40%



课程考核



- 理论部分（60%）
 - 考核方式：
 - 课堂（10%）
 - 出勤、课堂提问、课堂测验
 - 课后作业（20%）：
 - 四次：课后阅读、在线学习、读书笔记、调研报告
 - 期末开卷考试（30%）
 - 考核内容：讲授内容（理论部分的知识点），软件工程领域一些开放性问题思考



课程考核



■ 项目实验（40%）

- 考核方式：个人独立完成实验，提交实验报告，根据需要
进行报告或演示
- 考核内容：
 - 完成实验任务
 - 提交实验报告，根据要求进行报告或演示；
- 评分原则
 - 实验完成程度、提交报告内容、演示

■ 申请加分项

- 软件类大赛：鼓励，根据参加、获奖情况额外支持
- 开源：参与维护知名开源项目、自研项目



课程群



2025秋南京大学软件...

群号：910136794



扫一扫二维码，加入群聊

教学课堂：用于 发布实验

链接：<https://cslab-cms.nju.edu.cn/classrooms/gzkij uhs?code=LVSXZ>

教学课堂邀请码：

LVSXZ



■ 谢谢!
