



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

哈尔滨工业计算学部2025年春季学期
软件工程
Software Engineering

刘铭

liuming1981@hit.edu.cn

2025年4月19日

任课教师

■ 刘铭 教授

- 计算学部 社会计算与信息检索研究中心
- 电子邮件: liuming1981@hit.edu.cn
- 联系电话: 86413683
- 办公室: 科创大厦K1226
- 研究方向:
 - 大模型推理
 - 大模型轻量化和加速
 - 多模态信息处理



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY



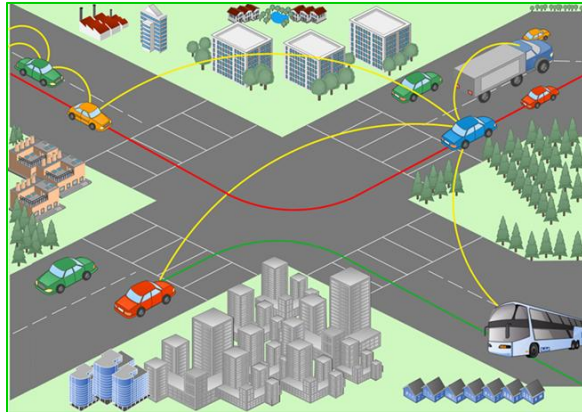
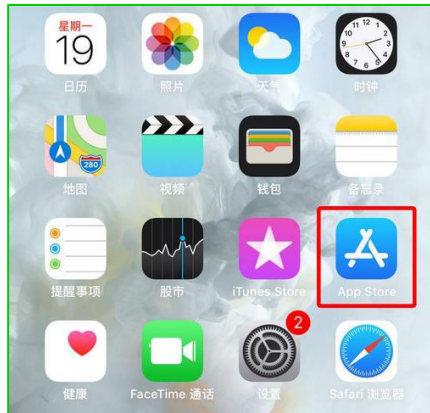
软件不可或缺

■ 信息系统组成

- 微电子芯片、计算机硬件与体系结构、计算机网络与通讯
 - 计算（速度）、存储（容量）、传输（带宽和速率）
- 计算机软件
 - 处理数据、实现功能、提供服务

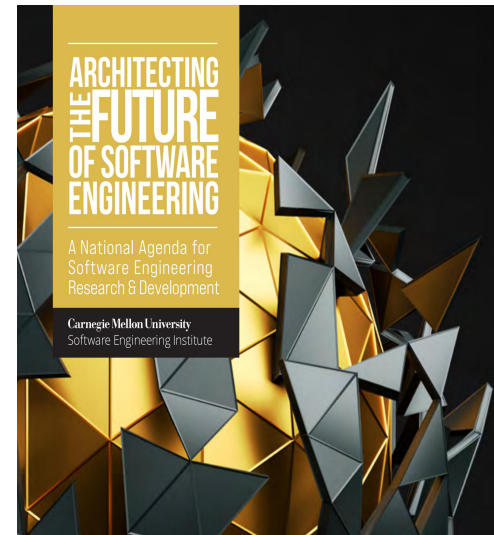
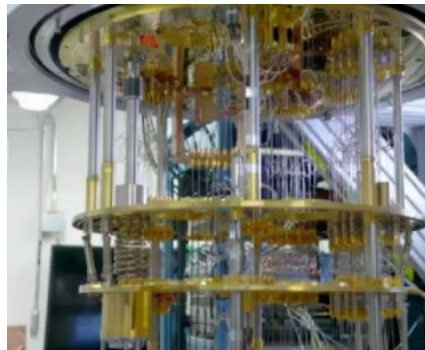
■ 软件所扮演的角色

- 不可或缺、核心和关键的要素
- 没有软件的手机和PC会怎样？



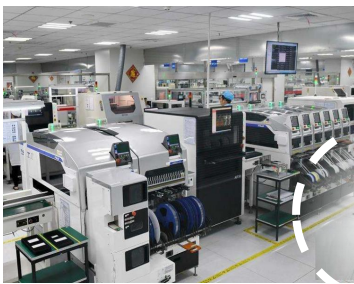
软件的关键性和重要性

- 软件对一个国家的**全球竞争力、创新和国家安全**至关重要
- 世界运行在软件之上，软件是国家的**关键基础设施**
- 软件赋能**能力创新**
- 软件是**安全攸关系统**的脊梁
- 软件是**生态系统**重要组成部分



软件定义一切时代的特点

- **地位**：无处不在的软件
- **形态**：人机物三元融合的软件
- **复杂**：大规模、持续演化的软件

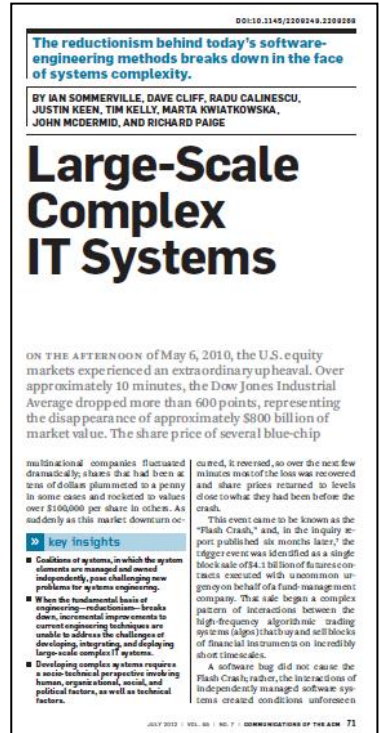


渗透到社会经济文化生活各行各业，从工业生产到国防科技，从衣食住行到娱乐

- 开发数量越来越多
- 交付速度越来越快
- 质量要求越来越高
- 运维能力越来越强

软件形态变化

- 不是**单一孤立**的系统，而是**系统之系统**
 - 诸多系统联盟和组合而成
- 不是纯粹**技术系统**，而是**社会技术系统**
 - 社会、信息、物理等要素共存
- 不是**同构**系统，而是**异构多样系统联盟**
 - 要素异构，客观存在，也是必然
- 不是**封闭**系统，而是**动态适应和持续演化系统**
 - 要素、关系、联盟等持续变化，边界不明确



软件的规模和复杂性日趋增长



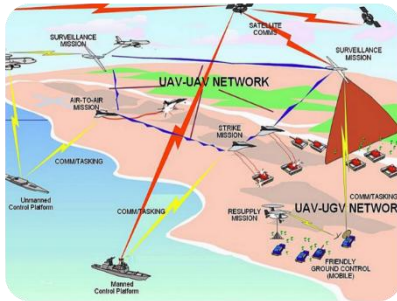
大规模协作系统



金融系统



社会保障信息系统



指挥控制系统



智能交通控制



智能电力系统

- 功能
- 数据
- 实体
- 代码量
- 接入人员
- 连接设备
- 运行进程
-

软件变得越来越复杂

■ 软件环境的复杂性

- 软件表现为人机物三元融合复杂系统
- 软件需要与物理环境、社会环境等进行持续交互
- 软件所在的环境开放、动态、不确定和不可预测

■ 软件规模带来的复杂性

- 大量的软件实体、数据、交互、进程等带来的复杂性

■ 软件需求的复杂性

- 软件需求不可知、不确定和动态变化

■ 软件运维的复杂性

- 软件需要在使用的同时持续演化



12306软件、智能交通控制软件等就是一类复杂软件系统，具有上述复杂性特征

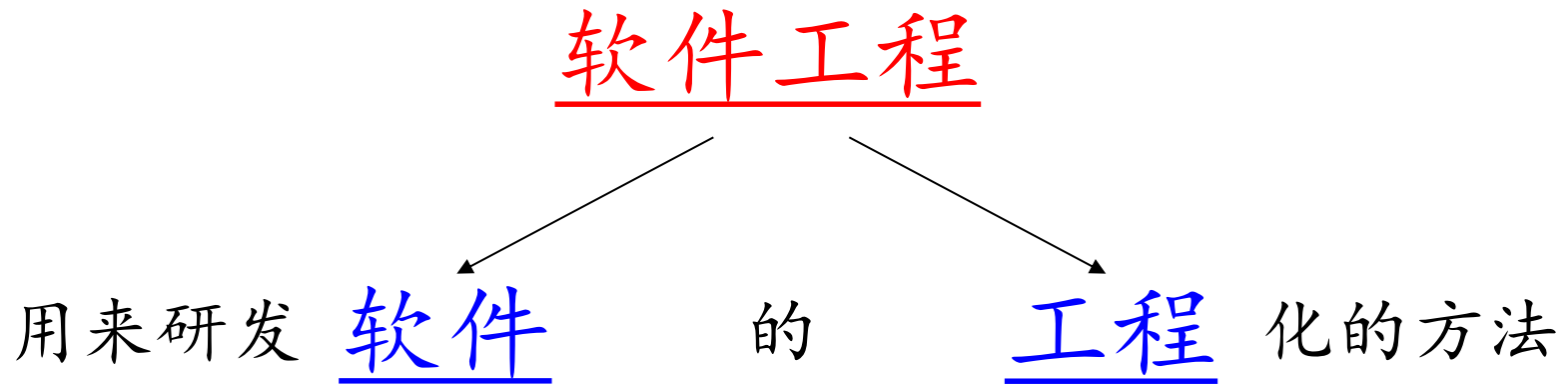
大规模和复杂软件带来的挑战？

- 软件复杂性表现在哪些方面？
- 如何来开发复杂的软件系统？
- 需要哪些方面的知识和技能？



用户方、学术界、产业界共同面临的问题和挑战！

什么是“软件工程”？



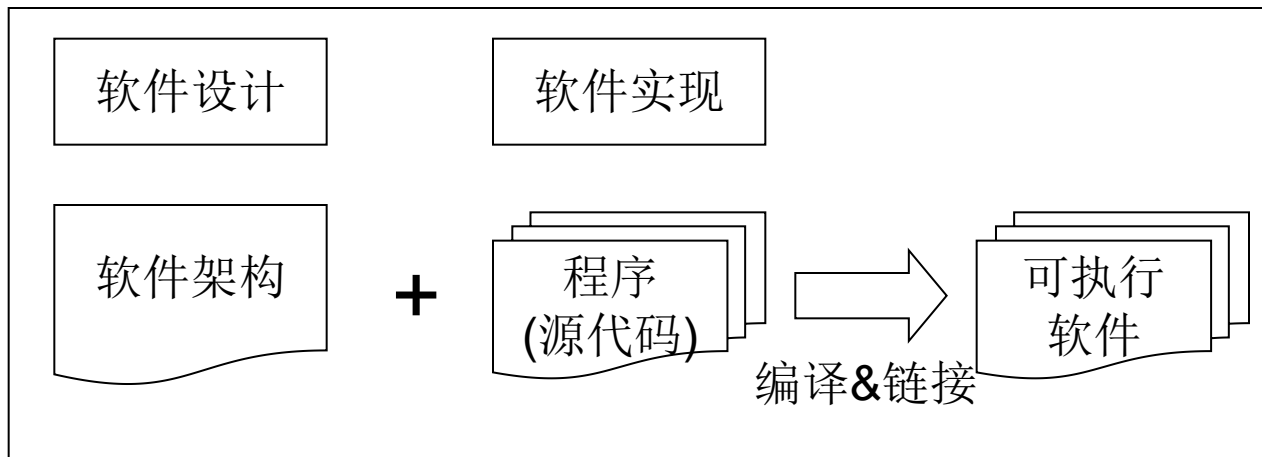
■ 传统的理解：

- 软件：控制计算机硬件功能及其运行的指令、例程序和符号语言
- 工程：将科学及数学原理运用于实际用途的应用手段，即高效率、低成本的设计、制造和运行各类结构、机器、进程和系统

- 对软件工程的直观理解：应用计算机和软件科学中的理论方法来解决软件系统“从无到有”、“从有到好”的过程。

从无到有 (from scratch) : 软件构建

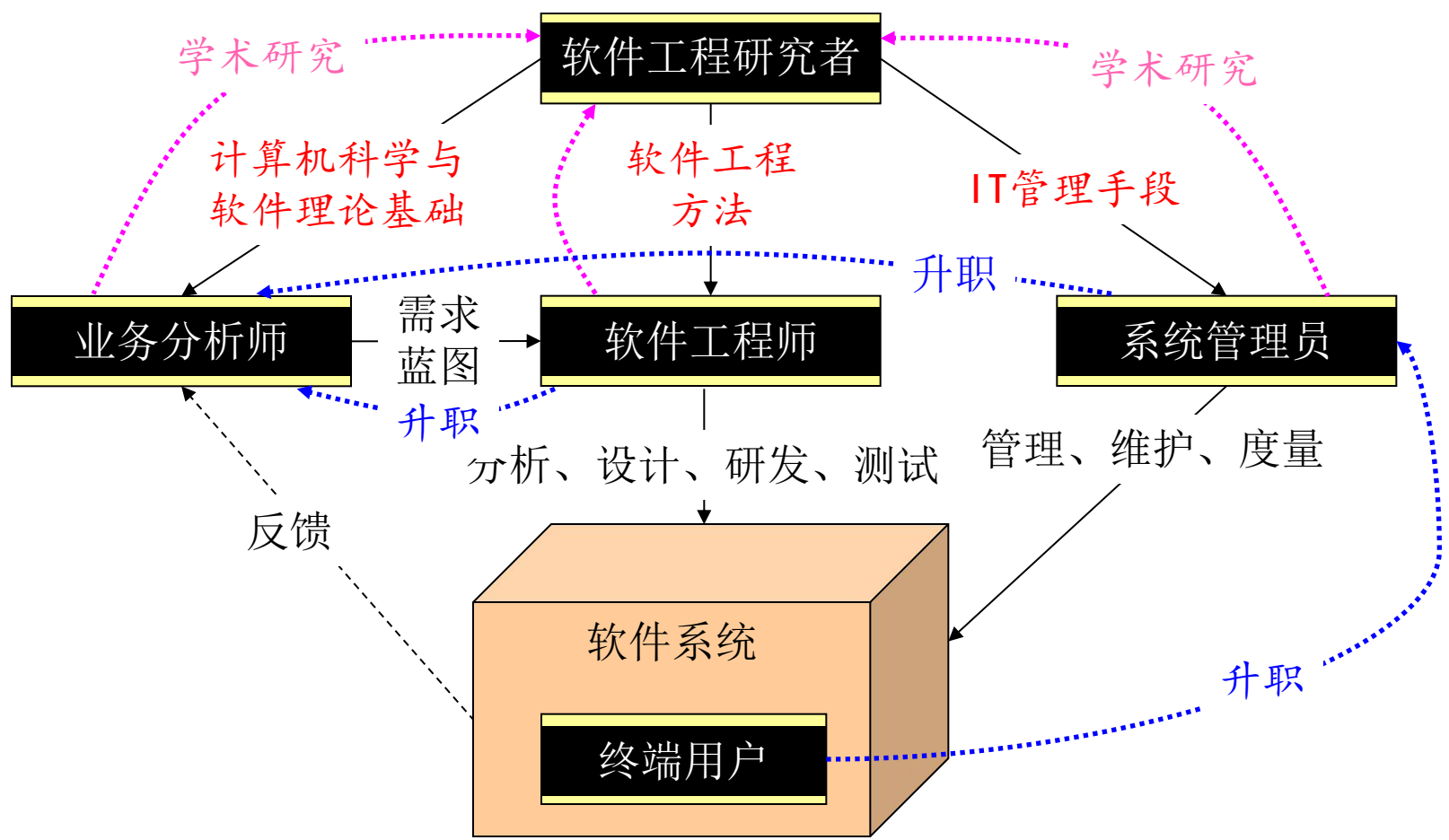
- 程序(program)、源代码(source code): 建立在数据结构(data structure)上的一些算法 (algorithm), 将它们编译成机器能懂的目标代码。
- 对于一个复杂的软件, 要有合理的软件架构 (Software Architecture)、软件设计和实现 (Software Design & Implementation)
- 还要用各种文件来描述各个程序文件之间的依赖关系、编译参数、链接参数等。这些都是软件的构建(Software Construction)。



从有到好(for better)：软件管理与维护

- **源代码管理 (Source Code Control) 、配置管理 (Configuration Management)**
 - 软件团队的人员每天都在不断修改各种源代码，如何保证软件在不断的修改中能保证质量、不至于崩溃？
 - 要为某个需求写一些特殊功能，如何把这些功能再合并回主要版本？
- **靠一系列的工具和程序来保证软件的正确性，称为质量保证(Quality Assurance)、软件测试 (Testing)。**
- **软件的生命周期 (Software Life Cycle, SLC)**
 - 软件要有人买，就得先找到顾客，实现他们的各种需求；
 - 从需求分析 (Requirement Analysis) 开始，通过设计(软件架构)、实现(数据结构和算法)、测试、到最后发布软件；
 - 软件在运行过程中还会出这样那样的问题，要对软件打补丁，这叫软件的维护 (Software Maintenance)。
- **整个生命周期内软件项目的管理 (Software Project Management)。**

“制造软件”需要社会化分工



知识与能力需求

就业职位	知识与能力需求
软件终端用户	了解典型行业业务及相应软件系统的运作机制，例如金融、保险、通讯、健康医疗保健、物流、旅游、制造等
软件系统管理员	管理系统运行时的资源分配、监控与优化系统性能、响应终端用户的请求、快速解决问题等
业务分析师/ 产品经理	战略眼光：发现IT技术对软件潜在影响的能力；发现行业演化趋势的能力；发现阻碍业务提升的问题所在并进行优化
	发现新价值的能力；业务创新的能力
软件工程师 (需求分析师、系统架构师、程序员、测试人员、项目管理者)	收集软件需求、建立软件模型、设计软件系统的能力
	采用各类IT与软件技术开发测试软件系统的能力
	软件项目管理的能力
软件工程研究者	从各类软件系统进行抽象和数学分析的能力

软件工程知识是IT职业的基础

- IT行业软件工程师
 - 软件程序员
 - 软件设计师
 - 系统架构师
 - 系统分析师
 - 测试工程师
 - 产品质量经理
 - 实施顾问
 - 项目经理
- 企业或事业单位从事IT工作
 - 系统管理员
 - 数据库管理员
 - 首席信息官

课程简介

- **授课对象：** 计算学部 2022级本科
- **课程分类：** 专业必修课
- **学时：** 48(32+16)
- **先修课程：** C++/Java/Python 等至少一门高级程序设计语言；
数据结构与算法；数据库系统。
- **上课时间/地点：**

理论课

- 9-16周 周2 1-2节[2203501, 2203901]/3-4节[2203101, 2203102]正心116,
周5 1-2节[2203101, 2203102]/3-4节[2203501, 2203901] 正心116

实验课

- 9-16周 周三 1-2节[2203501, 2203901]/3-4节[2203101, 2203102]格物楼207

课程网站

- 头歌 <https://www.educoder.net/>
 - 通过手机或邮件注册账号
 - 通过“BDO7K”邀请码加入课堂



The image shows a screenshot of the Educoder website. The top navigation bar is dark with white text links: 首页, 实践课程, 教学课堂, 项目托管, 竞赛组织, 毕业设计, 编程刷题, 交流问答, 精培课程. To the right are icons for search, a plus sign (highlighted with an orange box), a bell, and a user profile. A green arrow points from the plus icon to a modal window titled '加入课堂'. Inside the modal, there is a text input field for '课堂邀请码' (Classroom Invitation Code) containing 'BDO7K'. Below it are radio buttons for '身份' (Identity): 教师, 助教, and 学生/参赛者 (which is selected). At the bottom right are '取消' (Cancel) and '确定' (Confirm) buttons.

加入课堂

课堂邀请码: BDO7K

身份: ☐ 教师 ☐ 助教 ☒ 学生/参赛者

取消 确定

- 课件、作业、实验、项目均在此网站发布，提交物也通过此网站提交
- 在网站论坛上与教师/助教和同学进行讨论交流。

课程网站

- 加入班级
 - 通过“WZHP2G” 邀请码加入课堂

成员管理 | 学生 26 | 教师 10

添加老师 | 添加助教 | 添加学生

🔔 通知公告

📁 实验与项目

📚 教学资料

💬 互动讨论

👤 课堂分班

3

⚙️ 设置

← 刘铭老师班级 邀请码: KS6LO9

删除分班 分班重命名

共 0 个学生

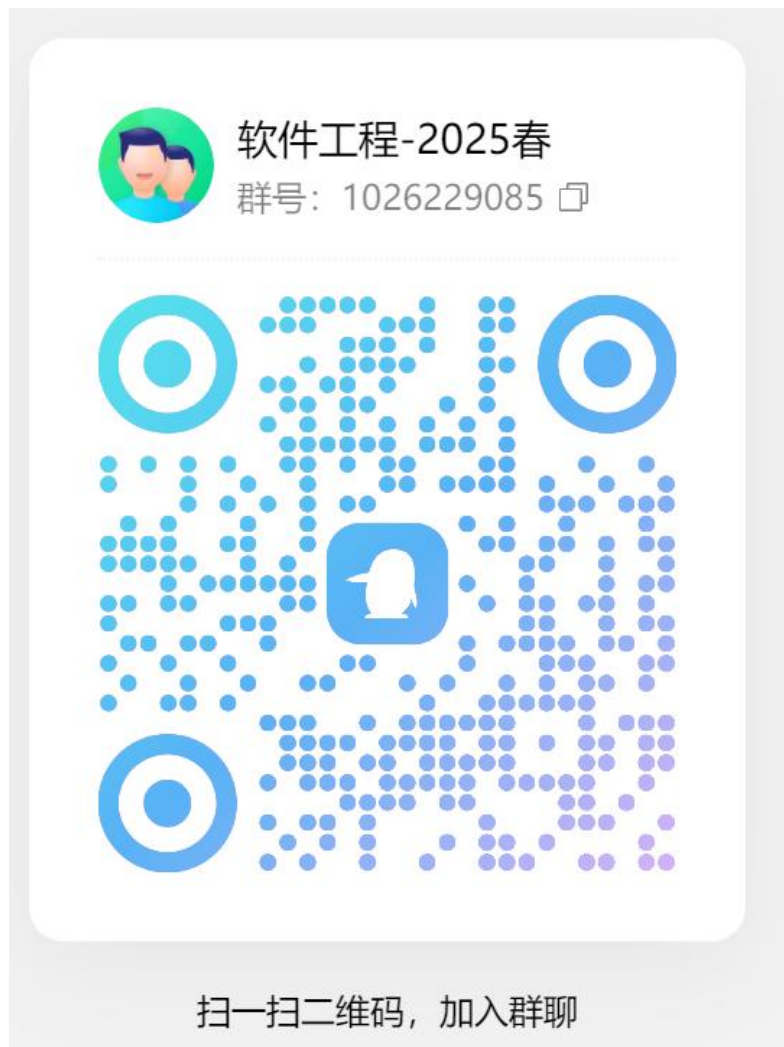
请输入姓名/学号搜索



暂时还没有相关数据哦!

班级QQ群

- 班级QQ群



参考教材

- 毛新军，董威. 《**软件工程-理论与实践**》，高等教育出版社，2024年1月，ISBN: 9787040610109
- Rogers S. Pressman. Software Engineering: A Practitioner's Approach (Ninth Edition). (《**软件工程：实践者的研究方法，第9版**》，机械工业出版社，2021年7月，ISBN: 9787111683940)
- 邹欣. 《**构建之法：现代软件工程(第三版)**》，人民邮电出版社，2014年9月，ISBN 9787115460769



考核方式

- **平时成绩：5%**
 - 课程采用MooC作为教学辅助，需要在MooC中观看视频、完成练习与测试题目；
 - 参与课程网站(头歌平台)上的讨论与交流
- **实验：20%**
 - 共4个，个人完成、三人分组完成；
 - 现场检查、提交实验报告/实验代码至头歌平台；
- **实践项目：25%**
 - 分组完成，分阶段评估。
- **期末考试：50%**
 - 开卷

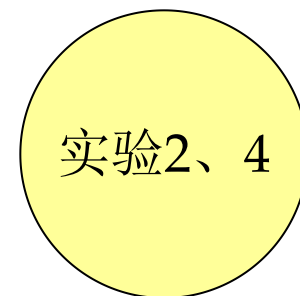
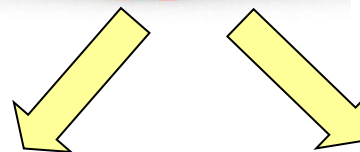
课程过程中的分组



不分组，独立完成实验1、实验3
参与教学网站上的讨论
参加期末考试



3人一组



关于实验

- 共4个实验;
- 16学时实验课, 课上+课后完成;
- 按照提交时间、代码/模型的质量、实验报告的质量、口头问答进行打分;
- 成绩计算:
 - 不管是3人一组、独立完成, 组内成员均获得同样的成绩;
 - 4次成绩加权平均, 得到总成绩。

Lab 1: 基于大模型的编程与Git实战(1人)

Lab 2: 需求获取与原型设计 (3人)

Lab 3: 代码评审与测试 (1人)

Lab 4: OO分析与设计 (3人)

关于实践项目

- 给定若干个候选题目，各组从中选择，同一小班的组选择不同的题目；学生也可自拟题目，但是需要同教师讨论是否合适。
- 面向选定的题目，根据自己的直觉和当前能掌握的技术，马上就进入开发(Code-and-Fix)，形成一个版本；
- 在写程序的过程中，不断理解澄清需求；
- 在现有版本的基础上，利用软件工程的方法进行需求分析和设计，进入迭代，不断完善前版本；
- 继续深入理解需求，循环进行迭代设计和开发；
- 设计测试用例，测试。

- 在一学期内，遵循敏捷开发过程，完成两次迭代。

关于实践项目

■ 迭代周期：

- 第9-10周：组队、选题、需求列举与优先级、迭代计划；
- 第11-14周：第一次迭代；
- 第14周：演示，教师评价；
- 第15-17周：第二次迭代；
- 第17周：演示/答辩，教师评价。

■ 成绩：

- 第一次迭代：40%
- 第二次迭代：50%
- 项目管理：10%

■ 成绩评定：

- 两次迭代：完成度、代码质量、演示质量、答辩质量
- 项目管理：用户故事、任务墙、迭代计划、GitHub

关于MooC学习

■ 课程利用在线MooC作为辅助教学手段

— 作用：课前预习、课后复习、课堂授课的补充、练习与测试

■ <http://www.cnmooc.org/> (<http://180.76.151.202:7010/home/index.mooc>) 好大学在线

— 注册时请准确输入：学号、姓名、机构（哈尔滨工业大学）

— 然后搜索或选择课程：软件工程_哈工大 2025年春

■ MooC环节的学习成绩，占总成绩的5%

■ MooC的使用要求和考核依据

— 教学视频：要求认真观看MooC教学视频，以观看时间为考核依据；

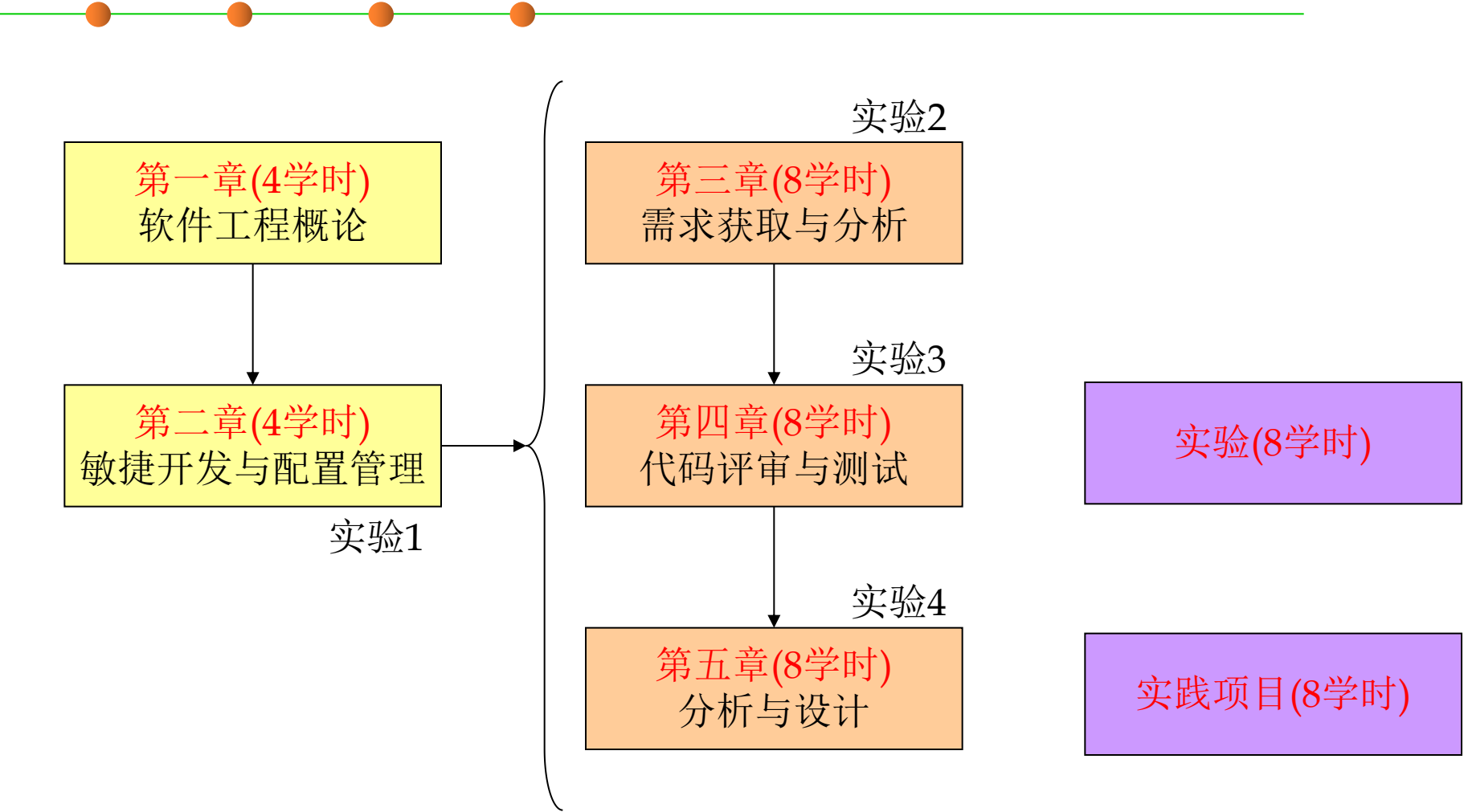
— 练习题：要求完成全部章节的习题测试，以系统给出的测试成绩为考核依据；



小结：累加式成绩

成绩项目	比重	考核方式			提交时间
平时成绩	5%	根据观看视频、练习与测试、讨论交流进行打分			根据期末考试时个人MooC学习状态进行打分
实验	20%	根据实验报告质量和实验课现场验收表现，TA主观打分			当次实验课之后的两周内，提交实验报告
实践项目	25%	第一轮迭代	40%	由教师 和TA主观 打分	第14周
		第二轮迭代	50%		第17周
		项目管理	10%		日常积累、第17周提交
期末考试	50%	开卷考试			第18周

课程章节安排



课程教学内容

- 软件及其特点
- 软件工程概述
- 软件过程模型
- 敏捷软件开发方法
- 需求分析基础
- 获取软件需求
- 面向对象软件需求分析
- 软件设计基础
- 软件体系结构设计
- 软件详细设计
- 软件测试
- 软件部署
- 软件维护与演化
- 软件项目管理

基本要求

- 大量阅读软件工程方面经典著作
- 多交流、多实践
- 了解国外软件工程发展的技术和趋势
- 理解软件工程的思想和方法
- 掌握常规的软件开发方法和工具
- 初步学会按照敏捷方法开发和维护软件
- 为实践及以后的软件开发打基础



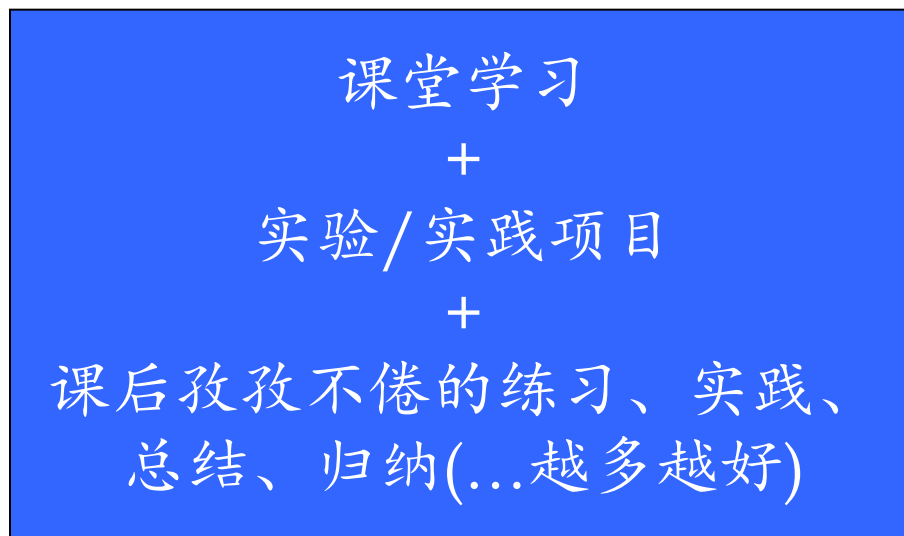
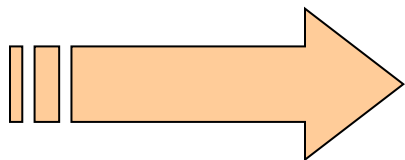
软件工程课程认识误区

- 软件工程课就是一门编程方法课
- 我没有编程基础，肯定学不好这门课
- 我今后不打算开发软件，学这门课没价值
- 软件工程讲很多管理知识，对我们学生没用
- 这门课很重要，学完这门课就成为软件工程师了

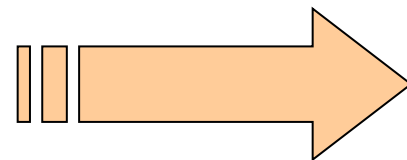
总结属于自己的“最佳实践”

- 多动手、多实践，方可成为合格的软件工程师；
- 实践越多、参与的项目越大，积累经验越多；
- 首先遵循他人提出的“最佳实践”，进而创造自己的“最佳实践”；
- 从“菜鸟程序员”成长为“软件工程师”。

菜鸟程序员



软件工程师



如何学习软件工程

- 在这门课程里，师生之间应该是一种“**健身教练 / 健身学员**”的关系。
 - **Not** Retailer / customer (餐馆/ 食客)
 - **Not** Boss / employee (老板/ 雇员)
 - **Not** Baby-sitter / babies (保姆/ 幼儿)
 - **Not** Buddies / Buddies (哥们/ 哥们)
 - **Not** Stranger / Stranger (路人甲 / 路人乙)
 - **Not** Prison Guard / Prisoner (狱警 / 犯人)



教练，你讲的特别好，我特别想减肥健美，但我太忙了，没时间练，所以我办了卡，就是来听听...

软件工程师的能力评估和职业发展

■ 能力

- 1. **知识**：对具体技术的掌握、动手能力——软件技术纷纭复杂，找准你感兴趣的软件子领域，深挖下去；
- 2. **经验**：对问题领域的知识和经验的积累——多做真实的项目；
- 3. **通用的软件工程思想**：遇到项目可遵循大家公认的方法论，遵循通用哲理，会利用工具，而不总是code-and-fix；
- 4. **职业技能**：自我管理的能力、表达和交流的能力、与人合作的能力、按质按量完成任务的执行力、职业道德、等等。

■ 成长路线

- 课外阅读

<http://www.cnblogs.com/xinz/archive/2011/10/22/2220872.html>

实验指导教师和助教

■ 实验指导教师

— 高峰

[9-16周]

周三 1-2节[2203501, 2203901]

3-4节[2203101, 2203102]

格物楼207

■ TA

— 王泽鑫 zxwang@ir.hit.edu.cn

— 范会明 hmfan@ir.hit.edu.cn

— 唐果 gtang@ir.hit.edu.cn

— 蒋世鑫 sxjiang@ir.hit.edu.cn



結束

2025年4月19日

关于课堂讨论

- 在讨论环节中，教师提出问题，学生发言
 - 为每组分配4张令牌，优先级分别为3、2、1、0。每次提出问题时，希望参与讨论的小组举起手里优先级最高的令牌，教师从所有举起的牌子中选择优先级最高的一组，请其发言。若多个令牌具有同样的优先级，则按举牌先后次序选择。发言之后，相应的令牌被收回。
 - 所有令牌使用完之后，后续讨论中将不能再参与讨论；
 - 提前准备，精炼语言，积极参与。
- 评分标准：根据参与讨论的活跃度(发言次数)、所发表观点的新颖性和合理性(发言质量)综合评分：
 - 分数 = 系数×发言平均成绩；
 - 发言次数为0、1、2、3、4时，系数分别为0、0.6、0.9、1.1、1.15；
 - 平均参与次数为2.5。
 - 例如：参加了1次讨论，得分10，那么最终得分为 $0.6 \times (10/1) = 6$ ；参加了4次讨论，得分为8、10、6、8，最终得分为 $1.15 \times (32/4) = 9.2$ 。

关于课堂讨论

- 分组进行，每组3人，讨论时小组选派代表参与发言，组内成员将获得统一的课堂讨论成绩。
- 课堂讨论分为两类：
 - 集中式讨论：有明确的主题，教师提前给出问题，学生提前准备，组内成员通过讨论达成共识，形成观点，上课时由教师引导进行讨论；
 - 随堂式讨论：讲课过程中，教师根据所讲内容抛出问题，学生阐述观点。
- “翻转课堂”：课堂授课时间会很少，学生需要提前阅读教师指定的教材、讲义、论文等，课堂上就其中某些问题进行研讨；

关于实验

实验	班级	日期	地点	班级	日期	地点
Lab1-1	101	9/25	格物208	201	9/25	格物208
Lab1-2		10/09	格物208		10/09	格物208
Lab2		10/16	格物208		10/16	格物208
Lab3-1		10/23	格物208		10/23	格物208
Lab3-2	102	10/30	格物208	401	10/30	格物208
Lab4		11/06	格物208		11/06	格物208
Lab5		11/13	格物208		11/13	格物208

关于实践项目

- 迭代周期：
 - 第1-2周：组队、选题、需求列举与优先级、迭代计划；
 - 第3-7周：第一轮迭代；
 - 第8-11周：第二轮迭代；
 - 第12-15周：第三轮迭代、验收答辩。
- 上课时间表：

班级	节点	日期	地点	班级	日期	地点
101	第一轮检查	11/06	格物208	201	11/06	格物208
		11/13	格物208		11/13	格物208
102	第二轮检查	11/27	格物208	202	11/28	格物214
	结题检查	12/25	格物208	401	12/26	格物214

如何学习软件工程

- 时刻关注CMS上的课程日历，了解课程的整体进度安排，尤其是各实验的上课时间和提交时间、实践项目各阶段提交物的时间和检查时间；
 - 建议：提前搭建好实验环境，学习实验所用的工具，提前开始实验，实验课上用于与TA的交流，答疑解惑，并接受验收。
 - 单纯使用2学时的实验课，无法完成实验。
- 提前阅读下一次课程的待讲授内容，阅读教材相关章节，进行预习；
- 对下一次课即将进行的课堂讨论问题，提前查阅资料做好准备。
 - “需要我学习的知识，老师一定会在课堂上去讲”——No
 - 不提前预习和准备，难以参与课堂讨论。

关于实践项目

■ 成绩：

- 开题报告：5%
- 前两次迭代：各15%
- 最终验收：35%
- 需求和原型设计成果：5%
- OO分析与设计成果：10%
- 项目管理：10%
- 真实用户数：5%

■ 成绩评定：

- 开题报告：思考的深度与广度
- 前两次迭代：进度、质量
- 最终验收：海报、演示、质量
- 项目管理：团队博客、用户故事、任务墙、结对编程、迭代计划、Github
- 真实用户数量：来源、数据、评价

总结属于自己的“最佳实践”

