




软件工程（1030064）

第一章 软件工程的基本概念

重庆理工大学
计算机科学与工程学院 李梁
 qq:1255214405
 liliang@cqut.edu.cn
 智慧树课号: K3388095
 智慧树网址: www.zhihuishu.com



16:39:26
重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



课程简介

一、本课程的性质、目的及任务

□ **软件工程**是系统科学（系统工程）与计算机科学相结合的**综合性科学**。软件是在软件工程指导下开发的。


□ **计算机科学与技术专业5门专业核心课之一**，是**软件工程专业4门专业（软件分析与设计、软件测试、软件质量保证、软件项目管理）核心课**的综合核心知识内容

□ **课程目的**：工程项目中用系统工程思想分析设计软件

□ **课程任务**：使学生具有**工程化的软件开发思想**和**掌握软件系统分析与设计方法以及软件开发件管理等方面的知识与能力**，具备**软件系统建设规划、与软件项目管理的初步能力**

□ **计算机类专业**：计科、软工、网络、物联网、智能科学

16:39:26
重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



课程简介

■ **课程基础**

- 学习过至少一种程序设计语言（python、java、C++）
- 有至少小规模软件开发经验（综合课程设计2）


■ **本课将是学习以下课程的基础**

- 综合课程设计3
- 专业实习
- **毕业设计**

■ **实际应用**

- 参与应用单位软件系统建设工作
- 参与商品化软件及应用软件开发

16:39:26
重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



课程简介

二、本课程的基本内容与要求：**思想+方法+工具**

- 了解软件工程的深层含义、软件工程方法从事软件开发的**意义**；
- **掌握常用的软件需求分析的方法（SA和UML）**
- **掌握常用的软件设计的方法（SD和I00P）**
- **掌握软件测试的基本方法（会设计测试用例）**
- 了解软件过程与管理的方法（**质量管理和成本/进度管理**）
- **掌握主要的软件标准及文档（分析、设计、测试文档）**
- **掌握实验教材要求的软件工具应用与案例要求的软件设计过程**


16:39:26
重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



三、课程内容及学时分配

章 次	教学内容	理论学时	实验学时
第1章	软件工程的基本概念	2	
第2章	软件生存周期及开发模型	2	
第3-1章	可行性分析及需求获取	2	
第3-2章	结构化软件需求分析模型	4	4
第4-1章	软件设计-设计思想	2	2
第4-2章	软件设计-设计技术	2	
第4-3章	软件设计-详细设计	2	
第5章	面向对象方法学	2	
第6-7章	面向对象分析与设计	6	6
第9章	软件测试	4	
第10-14章	软件过程及软件工程标准	2	
第11-13章	软件项目管理及总结	6	
合 计：		36H	12H

16:39:26
重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



教学安排

- **理论教学（36学时）**
 - ✓ 本章节内容概述
 - ✓ 重点、难点内容讲述
 - ✓ 补充内容讲解（主要是参考资料上内容）
 - ✓ 下章节内容提示
 - ✓ 布置课后自习内容、实验内容、作业
- **参阅、收集资料及自习（1周1次）**
- **课堂测试、讨论，提问（3次课堂测试）**
- **实验（12学时） 2-3次实验**
 - ✓ 课内实验：检查上次实验情况，讨论问题
 - ✓ 课外实验：完成本次实验内容
- **作业（每6节课1次作业，全部批改，5次作业）**

16:39:26
重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)

成绩组成

□ 平时成绩（共计50%）

- 考勤：病事假必须课前请假，**事假**不准事后补假条。3次迟到算1次缺勤，缺勤超过1/4取消考试资格
- 作业：须以正确的**格式**和**要求时间**交作业（15%，5次，每次作业3分）
- 课堂测试：15%，3次，每次5分
- 实验成绩（实验指导书中实验）（10%，2个实验，每个5分）
 - ✓ 安静、准时、卫生和秩序，不做和实验无关事情，实验时禁止玩游戏、上QQ、看电影、听音乐
- 软件分析设计文档（10%，课程完成后交，实验报告优化）

□ 期末理论考试（50%，原则上开卷）

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqupt.edu.cn)

7

参考教材：清华，软件工程导论(第6版)，张海藩机械，软件工程方法与实践（第3版），窦万峰

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqupt.edu.cn)

8

参考书目

- 1. 杜文峰,袁琳等著. 软件工程（微课视频版）. 清华大学出版社. 2023-01.
- 2. 毛新军,董威著. 软件工程：从理论到实践. 高等教育出版社. 2022-06.
- 3. [美]罗杰S. 普莱斯曼 (Roger S. Pressman), [美]布鲁斯 R. 马克西姆 (Bruce R. Maxim) 著, 王林章等译. 软件工程：实践者的研究方法（第9版）. 机械工业出版社. 2021-10.
- 4. [美]戴维·法利 (David Farley) 著, 赵睿, 茹炳晟译. 现代软件工程：如何高效构建软件. 人民邮电出版社. 2023-06.
- 5. [英]伊恩·萨默维尔 (Ian Sommerville) 著, 彭鑫, 赵文耘等译. 软件工程. 机械工业出版社. 2018-02.
- 6. 张海藩, 牟永敏著. 软件工程导论（第6版）. 清华大学出版社. 2013-08.

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqupt.edu.cn)

9

第1-1章 软件工程学概述-软件工程

- 1.1 软件的定义、特点和分类
- 1.2 软件的发展和软件危机
- 1.3 软件工程方法学
- 1.4 开发方法的特点与选择（自学）

本章课程目的

- 了解软件概念、特点及分类方法
- 了解软件发展及软件危机的起因。
- 了解软件工程的定义及其要素。
- 了解软件工程的基本目标和原则。

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqupt.edu.cn)

10

1 软件的定义、特点和分类-软件定义

- **软件**是计算机系统中与硬件相互依存的另一部分，它是包括**程序**，**数据**及其相关文档的完整集合。
- 软件是由程序、数据和文档所组成的一个系统，它可以在计算机系统的支持下实现特定的功能，并提供相关的服务。（101计划中软件概念）
- **程序**是按事先设计的功能和性能要求执行的指令序列
- **数据**是使程序能正常操纵信息的数据结构
- **文档**是与程序开发，维护和使用有关的图文材料
- **软件=程序+文档+数据**
- 软件的特征：**复杂性**、**一致性**（硬件依赖性）、**永不退化**、**易变性**、**不可见性**、**移植性**、**高风险**与**高收益**。
- 软件是高科技的智力产品，需要很高的创造性，但本质上还是不能抹去**工业性**，**做软件就是做工程**。

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqupt.edu.cn)

11

1.1 软件的定义、特点和分类-特点

- 软件是一种**逻辑实体**，不是物理实体。具有**抽象性**，其形态**不可见**，是人力智力密集性产品。必须通过观察、分析、思考、判断来了解其功能、性能和其它特性，没有最终完成的情况下，很难看到它的样子，开发过程对软件的最终结果的**把控性不强**。
- 软件是人脑思维的产物，与**成本**硬件不同，开发中没有明显的制造过程，使用中的**维护工作**相当复杂。在运行和使用期间，没有硬件那样的**机械磨损**，**老化问题**，存在开发过程的质量控制及软件产品保护问题。

年份	硬件成本 (Hardware Cost)	软件成本 (Software Cost)
60	100	20
70	80	30
80	60	40
90	40	50
2000	20	70
2020	20	80

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqupt.edu.cn)

12

1.1 软件的定义、特点和分类-特点

- 软件的开发和运行常受到计算机系统（硬件、系统软件）的限制，对计算机系统有一定的**依赖性**，存在**软件移植问题**。
- 软件的各个关联子系统、模块以及跟其他相关软件之间的接口必须保持**高度一致**，任何一个接口发生变化，必然引起其他关联者无法正常工作。
- 软件的开发至今**尚未完全摆脱手工艺的开发方式**，开发效率低。注意Chat GPT的影响（生成型预训练变换模型：Generative Pre-trained Transformer）
- 软件本身是**复杂的**，实际问题的复杂性和程序逻辑结构的复杂性
- 软件成本相当昂贵（远远超过硬件）
- 相当多的软件工作涉及到**社会因素**：如：机构、体制、管理方式等，包括人的观念及心理，都直接影响软件工作的成败。

1.1 软件的定义、特点和分类-特点

软件的需求不是很容易一开始就能明确描述，而且会随着客户对其理解和认识的进程随时发生变化，以此来说明软件的**易变性特点**。

软件的复杂性越来越高，对软件人员的要求越来越高；出现了软件复杂性与软件技术发展的不适应现象。

1.1 软件的定义、特点和分类-分类

按功能		按工作方式		按服务对象		按结构和使用方式	
系统软件	支撑软件	应用软件	实时处理软件	分时软件	交互式软件	批处理软件	项目软件
							产品软件
							嵌入式软件
							单机软件
							分布式软件
							移动终端软件
							Web 软件

1.2 软件的发展和软件危机-发展

软件技术的发展：

- 第一代 算法技术
- 第二代 结构化软件技术
- 第三代 面向对象技术
- 第四代 分布式面向对象技术
- 第五代 软件构件技术

软件发展阶段	国外	国内
程序设计阶段	60年代	70年代
系统设计阶段	70年代	80年代
软件工程阶段	80年代	90年代

1.2 软件的发展和软件危机-软件开发风险

软件开发风险：受复杂的多变的多种因素影响。

- 逻辑思维过程的不确定性。举例：疫情、AI影响
- 人的感情、疾病、体力、脑力、智力、欲望等是多变的。

有策略的改变等等？

1.2 软件的发展和软件危机-软件危机

- 例子：
 - 美国波音 737 飞机 2018 年和 2019 年的两次坠机，造成 300 多人遇难！原因就是由于软件缺陷，导致机头抬起不正常！
 - 2008 年中国举办首届奥运会，2007 年 10 月 30 日上午 9 点，奥运官方网站、呼叫中心和 1000 家指定的中国银行网网点同时启动销售。上午 9 点 30 分左右，北京奥运门票的三种购票渠道都陷入“瘫痪状态”。
 - 2021 年 4 月 16 日，“五一”节前，疫情好转背景下的重大节日抢票关键点（提前 15 天购票），当天开抢的第一时间，系统崩溃，页面提示“系统忙，请稍后再试！”
- 软件危机的表现：费用超预算、周期超预期和软件质量问题
- 主要原因-软件开发的特殊性：由于软件是由人开发，不同的程序员，编程习惯、思维模式不同，程序实现的主观性比较强；对于需求的理解，不同的人之间存在差别，且需求容易发生变化-“永远不变的是需求的变化”《人月神话》。布鲁克斯Brooke

1.2 软件的发展和软件危机-软件危机

- 软件危机的概念
软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。即包含两方面的问题：**如何开发软件、如何维护软件**
- 软件危机的主要表现（现象）
 - 软件生产不能满足日益增长的需要，供不应求，供需差增大
 - 软件开发成本难于控制和**进度难于预测或估计不准确，开发速度慢，延期较多**
 - 开发人员对用户的需求缺乏了解。软件产品不符合实际需要，用户对产品功能难以满足
 - 软件产品**质量差难以保证**。软件质量保证技术(审查、复查、测试)没有贯穿于开发的全过程
 - 软件可维护性差，难于维护。错误难以改正，新功能难以增加，“再用性”的软件没能实现，重复开发类似的软件
 - 文档资料不完整，软件交流、管理、维护造成困难
 - 软件成本逐年上升，软件的价格昂贵

1.2 软件的发展和软件危机-软件危机

- 产生软件危机的原因 **发展过快、没有考虑“好”一质量**
 - 客观原因：软件本身的特点(如:软件规模越来越大，软件开发复杂度越来越高)导致开发和维护困难 **波音 737 MAX**
 - 主观原因：
 - 用户需求不明确，需求背景复杂难理解，缺乏正确的理论指导
 - 软件开发方法不正确(个体、作坊式开发、需求分析不深入)
 - 开发人员与管理人员重视开发而轻视问题的定义和软件维护
 - 软件开发技术落后
 - 软件管理技术差
- 目前一个中小规模软件
 - 代码行10~100万行，>100人月
 - 开发费用100~1000万

1.3 软件工程方法学

- 解决软件危机的途径：软件工程方法
 - ✓ 20世纪60年代软件危机爆发，使人们思考、寻求软件开发的“出路”
 - ✓ 人们从工厂的生产组织方法获得启示：能否将软件开发作为工厂的产品，按照工厂产品的生产组织方式组织软件开发工作？

1.3 软件工程方法学

- 解决软件危机的途径：软件工程
 - 组织管理措施
 - ✓ 项目管理方法：采用必要的组织管理措施
 - ✓ 软件工程学：采用工程化方法和工程途径来研制与维护软件
 - ✓ 软件标准化
 - 技术措施：采用先进的技术、方法与工具来开发与设计软件


软件工程正是从**管理**和**技术**两个方面研究如何更好地开发和**维护**计算机软件的学科。

1.3 软件工程方法学

- 工程(Engineering)。将自然科学的理论应用到具体领域中形成的各种学科总称。即将理论和所学的知识应用于实践的**科学**。如**水利工程、化学工程、土木建筑工程、遗传工程**。
- 软件工程——应用计算机科学、数学及管理科学等原理，开发软件的工程。借鉴传统工程的原则、方法，以提高质量，降低成本为目的。其中：
 - ✓ 计算机科学、数学用于构造模型与算法；
 - ✓ 工程科学用于制定规范、设计模型、评估成本；
 - ✓ 管理科学用于计划、资源、质量、成本等的管理。
- 软件工程是一门交叉性学科。

1.3 软件工程方法学-基本概念

- 1983年美国《IEEE软件标准术语》：软件工程是开发、运行、维护和修复软件的**系统方法**。
- 软件工程是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发和维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来——**本书用的定义**。
- 软件工程(Software Engineering, 简称SE)是采用系统工程和管理学相结合的**原理、方法和技术**来指导、管理和实施软件的**各种活动**，主要包括软件开发、运行、维护和服务等内容。 **推荐的定义**
- 软件工程是用系统的、规范的、可度量的方法，开发、运行和维护软件。软件开发活动中**随处弥漫着管理方法、系统科学思想和工程技术——这就是软件工程的难点**




1.3 软件工程方法学-基本原理

软件工程的本质特性

- 1、软件工程关注于**大型软件**的构造
- 2、软件工程的中心课题是**控制复杂性**
- 3、软件**经常变化**
- 4、开发软件的**效率**非常重要
- 5、**和谐地合作**是开发软件的关键
- 6、软件必须有效地支持它的**用户**
- 7、在软件工程领域中通常由具有**一种文化背景的人替具有另一种文化背景的人创造产品**

16:39:26
重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)
25




1.3 软件工程方法学-基本原理

●B.W.Boehm勃姆：软件工程7条基本原理

- 1、用**分阶段**的生命周期计划严格管理---采取项目制
- 2、坚持进行阶段**评审**---并行开发困难
- 3、实行严格的产品**控制**---要有**质量保证**技术方法
- 4、采用现代**设计技术**---先设计后实现（编程）
- 5、**结果**应能清楚地审查---软件的结果用文档来表达
- 6、开发小组的人员应该**少而精**---程序员有**门口**
- 7、承认不断改进软件工程**实践**的必要性---总结持续改进

16:39:26
重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)
26




1.3 软件工程方法学-基本原理

●David Hooker（戴维.胡克）7条软件工程整体实践的原则：

- 1**存在价值**：一个软件系统因能**为用户**提供价值而存在。
最重要，最基础，有没有必要做，是否有市场前景与市场价值，比如引入流量、占用市场、竞争客户等用途
- 2**保持简洁**：优雅的设计同时也应该是**简洁而不简单**的设计。
减少错误，易于维护，首先需将问题理解透
- 3**保持愿景**：清晰的愿景是软件项目成功的基础。
软件项目成功的基础，团队具有一个一致的目标
- 4**关注使用者**：必须让**别人**理解和更好地使用你的系统。
广义的使用者一包括架构设计、编码人员、维护人员、客户等
- 5**面向未来**：生命周期**持久**的系统更具价值。生命周期持久，价值高，即有前瞻性（比如遗留系统的核心业务，接纳新功能的能力）
- 6**提前计划复用**：复用**既省时又省力**。具有前瞻性设计和计划，积累经验，质量有保障
- 7**认真思考**：**认真仔细思考**可以提高成功的可能性。**权衡与决**

16:39:26
重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)
27




1.3 软件工程方法学-本质特征

●软件工程原理的另外一种表达：5条基本观点

- 1、推迟实现的观点：**尽量完善**软件的逻辑设计，推迟软件**物理实现**
- 2、**质量保证**观点：满足用户对软件功能需求的有计划、有组织的活动，站在用户的角度来保证质量，**开发活动中质量的**可控。
- 3、**逐步求精**的观点：自顶向下，先宏观进行顶层设计（系统框架），然后逐步设计各项功能（模块），进而设计功能细节（算法），最后实现（程序代码）。
- 4、**分解与抽象**的观点
- 5、**信息隐蔽**观点

16:39:26
重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)
28



1.3 软件工程方法学-软件工程三要素


●软件工程是把软件看作是一个**工程产品**，用**工程化管理方法**

●**软件工程方法学三要素：方法、工具和过程**

- ✓**方法**：完成软件开发任务的技术方法，回答“**如何做**”问题。提供一系列软件开发技术。包括完成开发过程中各方面任务的方法并用某种特殊的语言或图形来描述。
- ✓**工具**：为运用方法而提供的自动或半自动软件工程**支撑环境**。将多种工具集成在一起可构成计算机辅助软件工程（CASE）的软件开发支撑系统。
- ✓**过程**：为了获得高质量的软件所需要完成的一系列任务的**框架**，规定了完成各项任务的工作步骤，回答“**何时做**”问题。**将软件工程方法和工具综合起来，进行软件开发。**
 - 方法使用的顺序
 - 要求交付的文档资料
 - 为保证质量和适应变化所需要的管理

●**软件开发各个阶段完成的里程碑**

16:39:26
重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)
29




1.3 软件工程方法学-基本目标

●**软件工程基本目标**：以工程化思想进行软件开发，生产高质量和高效率的软件。

- ✓**高质量**是最大程度地满足客户需求、帮助客户获得成功。
- ✓**高效率**就是以最小的成本获得最好的收益，即在**规定**的时间和**规定（资源）**预算内，完成规定的功能的软件开发、维护和服务等任务**三个规定**

- ① 付出较低的开发成本
- ② 达到要求的软件功能
- ③ 取得较好的软件性能
- ④ 开发的软件易于移植
- ⑤ 需要较低的维护费用
- ⑥ 能按时完成开发工作，及时交付使用

16:39:26
重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn)
30

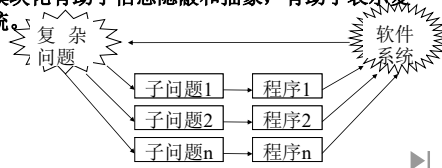


1.3 软件工程方法学-基本原则


① **抽象**：采用分层次抽象，自顶向下、逐层细化的办法控制软件开发过程的复杂性。

② **信息隐蔽**：将模块设计成“黑箱”，实现的细节隐藏在模块内部，不让模块的使用者直接访问。这就是信息封装，使用与实现分离的原则。

③ **模块化**：如C语言程序中的函数过程，C++ 语言程序中的类。模块化有助于信息隐蔽和抽象，有助于表示复杂的系统。



16:39:26
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
31




1.3 软件工程方法学-基本原则

④ **局部化**：要求在一个物理模块内集中逻辑上相互关联的计算机资源，保证模块之间具有松散的耦合，模块内部具有较强的内聚。这有助于控制解的复杂性。

⑤ **确定性**：软件开发过程中所有概念的表达应是确定的、无歧义性的、规范的。

⑥ **一致性**：整个软件系统的各个模块应使用一致的概念、符号和术语。程序内部接口应保持一致。软件和硬件、操作系统的接口应保持一致。系统规格说明与系统行为应保持一致。用于形式化规格说明的公理系统应保持一致。

16:39:26
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
32




1.3 软件工程方法学-基本原则

⑦ **完备性**：软件系统不丢失任何重要成分，可以完全实现系统所要求功能的程度。为了保证系统的完备性，在软件开发和运行过程中需要严格的技术评审。

⑧ **可验证性**：开发大型的软件系统需要对系统自顶向下、逐层分解。系统分解应遵循系统易于检查、测试、评审的原则，以确保系统的正确性。


16:39:26
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
33



1.3 软件工程方法学-过程特征

- ① 可理解性；
- ② 可见性；
- ③ 可支持性；
- ④ 可接受性；
- ⑤ 可靠性；
- ⑥ 健壮性；
- ⑦ 可维护性；
- ⑧ 速度。


16:39:26
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
34



1.3 软件工程方法学-软件过程

- 软件过程是为获得软件产品，在软件工具支持下由软件人员完成的一系列软件工程活动。
- ① 规格说明：规定软件的功能及运行限制
- ② 软件开发：产生满足规格说明的软件
- ③ 软件确认：对软件能否满足用户要求进行确认
- ④ 软件演进：为满足用户的变更要求的演进

16:39:26
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
35



1.3 软件工程方法学-软件过程

- **开发活动**
 - ✓ 软件生命周期模型：沟通、计划、建模、实现、部署、维护等活动
- **项目管理活动**
 - ✓ 项目的范围（要做的内容）
 - ✓ 进度（要花费的时间）
 - ✓ 成本（要耗费的资源）
- **过程改进活动**
 - ✓ 关注质量
 - ✓ 软件开发的最佳实践
 - ✓ 过程定义和改进

16:39:26
重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)
36



1.4 开发方法的特点与选择-开发方法

- 通常把在软件生命周期全工程中使用的一整套技术方法的集合称为**软件开发方法**，也称为**软件工程开发范型**。
- **范型**：模型、模式，就是一套实现具体任务的技术、规范的集合。
- 目前使用得最广泛的软件工程方法分别是**面向过程的方法学**和**面向对象方法学**。
- **结构化开发方法（范型）**：其技术要么面向**行为**，要么面向**数据**
- **面向对象开发方法（范型）**：将对象视作一个融合了数据及在其上操作的行为的、统一的软件**组件**。**数据和行为同等重要**



16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

37



1.4 开发方法的特点与选择-选择

- 开发队伍最熟悉哪种开发方法？
- 提供的软/硬件资源和开发工具？
- 开发方法在计划、组织和管理方面的可行性？
- 对开发项目所涉及领域的知识的掌握情况？



16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

38



1.4 开发方法的特点与选择-结构化方法

- **传统方法（生命周期方法或结构化方法）**
 - (1) 把软件生命周期划分成若干个阶段，然后顺序完成各个阶段的任务。
 - (2) 每个阶段的开始和结束都有严格的标准，对于任何两个相邻的阶段而言，前一阶段的结束标准就是最后一阶段的开始标准。
 - (3) 在每个阶段结束之前都必须正式地进行严格的技术审查和管理复审。
 - (4) 该方法采用结构化技术完成软件开发的各项任务。
 - (5) 构成结构化方法的技术包括：**结构化分析、结构化设计、结构化编程、结构化测试、结构化维护**



16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

39



1.4 开发方法的特点与选择-结构化方法

- **结构化方法（面向功能开发方法或面向过程方法）**
 - (1) 把软件开发划分成若干个阶段，然后顺序完成各个阶段的任务。
 - (2) 每个阶段的开始和结束都有严格的标准，对于任何两个相邻的阶段而言，前一阶段的结束标准就是最后一阶段的开始标准。
 - (3) 在每个阶段结束之前都必须正式地进行严格的技术审查和管理复审。
 - (4) 采用结构化技术完成软件开发各阶段的各项任务。
 - (5) 构成结构化方法的技术包括：**结构化分析SA、结构化设计SD、结构化编程SP、结构化测试、结构化维护**



16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

40



1.4 开发方法的特点与选择-结构化方法

- **结构化方法的特点**
 - 按照功能分解的原则；
 - 自顶向下、逐步求精；
 - 直到实现软件功能为止。
 - 简单、实用。
- **结构化方法主要问题**
 - 它以功能分解为基础，而用户的功能是经常改变的，必然导致**系统的框架结构不稳定**。
 - 从数据流程图到软件结构图之间的过渡有明显的断层，导致**设计回溯到需求有困难**。



16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

41



1.4 开发方法的特点与选择-面向对象方法

- **面向对象方法学（面向对象范型）**
- **特征**：将对象视作一个融合了**数据及在其上操作的行为**的、统一的软件**组件**。
 - (1) 把对象作为融合了数据及在数据上的操作的**软件构件**，即用**对象分解**取代了传统方法的分解。
 - (2) 把所有**对象都划分成类**。
 - (3) 按照父类与子类的关系，把若干个**相关类**组织成一个**层次结构**的系统。
 - (4) 对象彼此间仅能通过发送**消息**互相**联系**。



16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililiang@cqu.edu.cn)

42

1.4 开发方法的特点与选择-面向对象方法

① **Booch方法**：该方法的分析能力较弱，是一种偏重设计的方法。

② **OMT方法**：通过建立对象模型、动态模型、功能模型，来实现对整个系统分析和设计工作。

③ **OOSE方法**：最大特点是用用例（Use-Case）与外部角色的交互来表示系统功能，用例贯穿于整个开发过程。

④ **Coad / Yourdon**的面向对象分析和设计方法：即OOA和OOD方法，它是最早的面向对象的分析和设计方法之一。

⑤ **UML**：统一建模语言，国际对象管理组织OMG已批准将UML1.1作为面向对象技术的标准建模语言。

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

43

1.4 开发方法的特点与选择-面向对象方法

● **面向对象范型**

✓ **优势**：

- **对象的概念符合业务或领域的客观实际，将现实世界问题向面向对象解空间直接映射，实现对现实世界的直接模拟。**
- **以数据为中心，而不是基于对功能的分解**
- **使得软件结构相对稳定，软件的重用性、可靠性、可维护等特性都较好。**

✓ **缺点：开发复杂大型系统比较困难**

● **开发出的软件产品是由许多小的，相对独立的单元（对象）组成的。因此，面向对象范型降低了软件产品的复杂度，从而简化了软件开发与维护工作。**

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

44

小结

- **软件工程的宗旨是以工程化思想做指导，以高质量和高效率的开发软件。**
- **软件工程化思想的核心是把软件看作成为一个工程产品。**
- **传统软件工程方法学包含结构化范型和面向对象范型。**
- **软件工程的研究课题**
 - ① 形式语言的研究
 - ② 体系结构与架构技术
 - ③ 质量保证（可靠性研究）
 - ④ 过程控制

目前软件企业中普遍采取的开发与运维方式：DevOps
<https://www.zhihu.com/question/58702398> **敏捷开发+开发+运维**

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

45

小结-软件工程发展过程（自学）

- 1970 年罗伊斯（Royce）提出了著名的瀑布模型（Waterfall Model），直到 80 年代早期一直是唯一被广泛采用的软件开发模型。后来，又有人针对该模型的不足，提出了快速原型法、螺旋模型等。
- 1975 年，布鲁克斯（Brooks）写出了软件工程领域著作《人月神话》（The Mythical Man-Month），作者根据自己在 IBM 公司 System 360 家族和 OS 360 中的项目管理经验，以随笔集的形式对大型软件项目开发提出独到见解。
- 20 世纪 70 年代，结构化软件开发方法学形成。
- 20 世纪 80 年代，面向对象程序设计语言出现。Smalltalk、C++ 软件重用技术被提出，例如面向对象程序语言的可重用类库等；软件工程标准化工作活跃，如 CMU SEI 的 SW-CMM 等。
- 20 世纪 90 年代，随着计算机网络技术的发展，以 CORBA 为代表的分布式面向对象技术发展和流行；以 Java 和 C++ 为代表的面向对象技术成为主流；开源软件和技术出现，如 Linux 出现并产生广泛影响；人机交互技术发展，Windows 操作系统为代表的基于窗口的用户界面技术。
- 1995 年，GoF 写出了设计模式经典著作《Design Patterns: Elements of Reusable Object-Oriented Software》。
- 1997 年 1 月，UML 1.0（Unified Modeling Language 统一建模语言）发布，它解决了开发人员之间互相交流的问题，它包括用例图、类图、活动图、状态图、时序图、协作图、组件图和部署图等。同时，统一过程提出并细化。
- 2000 后，软件开发和运行从单机环境扩展到互联网环境，软件从产品到服务，如微信和 Google Docs 等；敏捷开发方法被应用到软件开发中；DevOps 方法在产业得到广泛智能化软件开发和群体化软件开发不断发展。

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

46

小结-开源软件（自学）

- **开源软件**，也称为**开放源代码软件**，是指其源代码对公众开放的软件。这意味着任何人都可以获取、使用、修改和分发这些软件的源代码。开源软件的特点是允许用户自由地对其进行二次开发和分发，同时通常伴随着一个**开源许可证**，该许可证规定了用户在使用、修改和分发软件时的权利和限制。
- 开源软件与闭源软件相对，后者是指源代码不对外开放，用户只能使用而不能对其进行修改或分发的软件。开源软件的开发和维护通常由全球的开发社区进行，这些社区通过开放协作和透明度来推动软件的发展。
- 开源软件的应用非常广泛，包括操作系统、编程语言、办公软件、图像处理软件等。开源软件的发展不仅促进了技术创新，也促进了全球开发者的交流与合作，同时，由于其源代码的公开性，也使得软件的安全性和可靠性得到了更好的保障。
- **重要的开源社区（微款）**：Github（开源仓库：面向开源及私有软件项目的托管平台）
- Github 于 2008 年 4 月 10 日上线，除了 Git 代码仓库托管及基本的 Web 管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。到 2022-11-17 注册用户已经超过 9400 万，托管版本数量也是非常之多，其中不乏知名开源项目 Ruby on Rails、jQuery、python 等。2018 年 6 月 4 日微软 75 亿收购
- **常用的 18 个开源仓库电子书**：
<https://baijiahao.baidu.com/s?id=1761440118550869203&wfr=spider&for=pc>
- **国内常用开源社区**：<https://www.cnblogs.com/shaoyunpeng/p/6026187.html> 及 <https://m.163.com/dy/article/H4CFRHB705149JG.html>

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

47

小结-常用CASE工具（自学）

- **建模工具**：如 Rational Rose / Microsoft Visio / StarUML 等。
- **原型制作工具**：Balsamiq Mockups / ExtJS / jQuery / EasyUI / 墨刀等。
- **集成开发环境**：Visual C++ / Visual Studio.NET / My Eclipse 等。
- **数据库设计工具**：PowerDesigner / ERWin 等。
- **代码生成工具**：Rational Rose（含 C++/Java 等代码生成模块）等。
- **自动化测试工具**：LambdaTest/ TestComplete 等平台；JUnit / JMeter 等工具。
- **配置管理工具**：Git / GitHub / Gitee 等。
- **项目管理工具**：Microsoft Project 等。

16:39:26

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

48