软件组成：数据 程序 文档 知识

软件分类：支撑 可复用 应用 系统

软件危机：软件开发和维护中出现的一系列困难和挑战，导致软件项目无法按时交付，超出预算、质量不可靠或无法满足用户需求的现象

软件危机的主要表现：软件的发展速度远远滞后于硬件的发展速度，不能满足社会日益增长的软件需求，软件开发周期长、成本高、质量差、维护困难

软件工程方法学三要素：方法 过程 工具

三个时期：软件定义（问题定义、可行性研究、需求分析） 软件开发（概要设计、详细设计、编码、测试） 运行维护（改正性维护、适应性维护、完善性维护、预防性维护）

八个阶段：问题定义 可行性研究（技术 操作 经济 社会因素 法律 ） 需求分析 软件设计（概要设计、详细设计） 程序编码 单元测试 集成测试 系统测试 运行维护

软件工具分类：支持软件开发过程的工具 支持软件维护过程的工具 支持软件管理过程和支持过程的工具

软件工程知识体系三类领域知识：软件工程管理类 软件生存期过程类 软件工程基础类

软件生存期模型：

瀑布模型 文档驱动 适用于需求明确

快速原型模型 用户驱动 适用于需求模糊

增量模型 构件驱动

螺旋模型 风险驱动 螺旋模型=瀑布模型+快速原型模型+风险分析

喷泉模型 对象驱动 适用于面向对象

统一过程 用例驱动 （初始阶段、细化阶段、构建阶段、移交阶段）

基于构建的开发模型 采用构件提高效率 复用

敏捷过程 快速响应变化 以人为导向 极限编程（策划、设计、编码、测试）：敏捷高效低风险 结对编程

第三章

获取具体需求：与用户交流 现有产品或竞争产品的描述文档 系统需求规格说明 市场调查和用户调查问卷 观察用户如何工作 当前系统问题报告和改进要求

需求工程：需求开发（需求调查、需求分析、需求定义）、需求管理（需求确认、需求跟踪、需求变更管理）

结构化分析模型：数据模型（实体联系图（E-R图））、功能模型（数据流图（DFD））、行为模型（（状态转换图（STD）））

数据流图：加工、实体、数据存储、数据流 每个处理至少包含一个输入流和一个输出流（黑洞、奇迹、灰洞）

SRS：软件需求规格说明书

DRD:数据规格说明 两者都需要完整性 无歧义性 一致性 可验证性 可修改性 可追踪性

现实性 有效性

软件设计分为概要设计 详细设计

软件设计五项原则：分而治之 问题分解各个击破

模块独立性（高内聚、低耦合）

提高抽象层次

复用性设计

灵活性设计

C/S三层体系结构 表示层 应用逻辑层 数据存储层

详细设计：模块设计 接口设计

概要设计：体系结构设计 数据设计

典型数据流类型：变换型数据流（IPO） 事务性数据流

内聚：功能内聚 信息内聚 通信内聚 过程内聚 时间内聚 逻辑内聚 巧合内聚

耦合：非直接耦合 数据耦合 标记耦合 控制耦合 外部耦合 公共耦合 内容耦合

软件模块结构的改进方法：

模块功能的完善化

消除重复功能，改善软件结构

模块的作用域应该在控制域之内

尽可能减少高扇出结构，随着深度增大扇入

避免和减少使用病态连接

模块的大小要适中

接口设计黄金三原则：用户控制界面 减少用户记忆负担 保持界面一致性

好看：视觉的美观性

好用：交互的易用性

第五章

UML关系：依赖 关联 泛化 实现

第六章

面向对象三种分析模型：功能模型（用例模型） 静态模型（对象模型） 动态模型（交互）

面向对象设计四部分：类设计 数据库设计 接口设计 架构设计

对象模型5各层次：  
 主题层

类-对象层

结构层

属性层

服务层 能干什么事

区分实体类 边界类 控制类

C/S三层体系结构（服务器、客户机、网络）：表示层 应用逻辑层 数据层（通信效率并不高）

MVC:M(模型model) V（视图view）C（控制器contrl）

第八章

面向对象设计准则：模块化

抽象

信息隐藏

弱耦合

强内聚

可重用

系统分解p197：对于系统分解要求高内聚低耦合，

人机交互部分设计：p201 用户界面设计步骤：

任务管理部分设计：主要包括任务的选择和调整

常见任务：事件驱动型 时钟驱动型 优先任务 关键任务 协调任务

编程语言选择考虑因素：应用领域 系统用户的要求 编程语言自身功能 编码和维护成本和开发环境 编程人员技能 软件可移植性

程序设计风格：源程序文档化（标识符的命名、程序的注释、空格、空行和移行）、数据说明标准化、语句结构简单化、输入/输出规范化

第十章：软件测试方法

什么是软件测试：在软件投入生产使用前，对软件需求分析，设计规格说明和编码的最终审查。

软件测试的目的和原则：

目的1，测试是软件执行过程，目的在于发现错误2.一个好的测试用例能够检测出未出现过的错误3.一个成功的测试时发现了从未发生过的错误

原则：1.把“尽早的不断地测试”作为软件开发者的座右明2。测试用例用该有测试输入数据和预期输出结果两部分3、程序员避免检查自己的程序4、设计测试用例应当包括合理输入条件和不合理输入条件5、注意测试集群现象，坚持测试吗，不要过早认为没有错误6、严格执行测试计划7、对每个测试结果全面检查8、妥善保管测试计划测试用例出错统计最终分析报告

软件测试对象：并不等于测试程序。软件测试应贯穿于软件定义与开发的整个期间

黑盒测试（等价类划分、边界值分析、错误推测、判定表法）和白盒测试（逻辑覆盖语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖）

程序环路复杂性V(G)=P+1

软件测试策略：单元测试 组装测试 确认测试 系统测试

单元测试步骤：驱动模块 桩模块

确认测试：（站在用户角度）1、进行有效性测试（黑盒）2、软件配置复查3、α测试和β测试，α为内测（受控测试），β公测

1. 软件维护

软件维护定义：在软件运行/维护阶段对软件产品进行的修改就是维护

软件维护四种类型（软件维护策略）：1、改正性（交付之后发现的错误进行改正）（修复错误和缺陷）2、适应性（外部环境变化，进行扩展功能增强性能，更新）（适应变化需求）3、完善性（用户提出新的功能，对软件进行的改善加工）（改建性能和效率）4、预防性（提高可维护性，为下一步改进打下基础）（更新和升级）

影响维护因素：可理解性 可修改性 可测试性 可移植性 可复用性

逆向工程：逆向

软件维护性定义：可理解性、可修改性、可测试性、可移植性、可复用性

CMM：capability maturity model能力成熟度模型的缩写

CMMI：初始级-已管理级-已定义级-已量化管理级-持续优化级

GB国标T国家推荐

GB国标Z国家指导

国际标准-国家-行业-地区-企业-项目

第十三章

五大过程组：启动、规划、执行、控制、收尾

4P：人（Person）、产品（Product）、过程（Process）、项目（Project）

十大知识领域：时间、质量、成本、范围、人力资源、沟通、风险、项目干系人、整体、采购

软件开发成本估算：专家判定（Delphi法）：E=（最大值+最小值+4\*最可能值）/6

软件配置管理要开展的活动包括：配置标识、配置控制、配置状态报告、配置评价以及发布管理和交付

软件配置项：1、与合同、过程、计划和产品有关的文档及数据

1. 源代码、目标代码和可执行代码
2. 相关的产品，包括软件工具、库内的可复用软件、外购软件及顾客提供的 软件

基线：通过正式复审的软件配置项叫做基线，基线是可以修改的