

软件三部分：数据、文档、程序。

软件危机：落后的软件生产方式无法满足迅速增长的计算机需求，从而导致软件开发与维护工程中出现一系列严重问题的现象。

软件工程：指导计算机开发和维护的一门工程学科。

软件工程三要素：方法、工具、过程。

软件开发常见模型

瀑布模型：适用于用户需求明确，完整、无重大变化的软件项目开发，文档驱动。缺点：文档驱动；用户难以清除给出所有需求；用户等待周期长。

V模型：强调测试的重要性，每步都进行测试。

原型模型：在用户不能给出完整明确的需求时，根据用户的基本需求，快速建造原型，然后进行评估，进一步精华、调整原型，使其满足用户的要求，也使开发者对将要做的事情有更好的理解。缺点：没有考虑软件的总体质量和长期的可维护性。不便管理。

增量式模型：迅速向用户提交完成部分工作的产品，分批，逐步的向用户提交产品，需要更精心的设计，但将在维护阶段获得回报。缺点：增量时必须不破坏原来已开发的产品，有无法集成的风险。

喷泉模型：主要用于面向对象软件开发项目。

螺旋模型：基本思想是降低风险，迭代更新产品，有利于软件的重用，减少了过多测试或测试不足，维护和开发没有本质区别。缺点：风险驱动，需要专业知识，否则风险更大。适用于内部开发的大规模软件项目。随着迭代次数增加，工作量加大，软件开发成本增加。

RUP（统一软件开发过程）的四个阶段：初始、细化、构造、移交。

软件生命周期的几个阶段和任务

问题定义：明确要解决的问题是什么。

可行性研究：明确做不做，从经济、技术、社会因素论证解决方案的可行性。

需求分析阶段：明确做什么，提出完整、准确、清晰、具体的要求。

概要设计：明确怎么做，制定最佳方案的详细计划，系统的体系结构设计。

详细设计：明确具体做法，设计出程序的详细规格说明。

编码和单元测试：编写出正确的、易理解的、易维护的程序模块。

综合测试：通过各类的测试使软件达到预定的要求。

软件维护：通过各种必要的维护活动使系统持久的满足用户的要求。

UML（统一建模语言）中包含的五类图：交互图（时序图、协作图）；行为图（状态图、活动图）；用例图；静态图（类图、包图）；实现图（部署图、构件图）。

用例图：从用户的观点对系统行为的一个描述。

获取用户需求的方法：访谈、情景分析、联合分析小组、快速建立软件模型。

类图的关系

依赖：类之间的使用关系。

泛化：类之间的一般和特殊关系。

关联：对象之间的结构关系。

规格说明和其实现之间的关系。

类图的设计过程：研究分析问题领域；发现对象与类并确定属性和含义；发现类之间的关系；设计类与关系；绘制类图并编制相应说明。

时序图：描述对象之间传递消息的时间顺序，用来表示用例的行为顺序。

协作图：显示对象为了一个用例描述的一个系统时间而与另一组对象进行写作的交互图。

协作图与时序图的两点特性：协作图有路径；协作图有顺序号。

活动图：描述了系统从一个活动转换到另一个活动的整个过程。

状态图：包括状态与转换。

状态图和活动图的区别：活动图主要描述动作及其对象状态改变的结果。状态图主要描述的是事件对对象状态的影响。

面向对象建立的三种模型：功能模型；对象模型；动态模型。

OOA：面向对象分析。

OOD：面向对象设计。

四条设计原则中有一条是高内聚，低耦合。

测试：软件测试，为了发现错误而执行程序的过程。

测试的目的：尽可能多的发现软件中潜藏的错误。

白盒测试：知道产品内部工作过程，测试内部动作。

黑盒测试：知道产品应该具有的功能，测试功能。

调试：在测试发现错误之后排除错误的过程。

调试的三种技术手段：蛮干法；回溯法；原因排除法；

软件维护的定义：软件维护就是在软件已经交付使用之后，为保证软件在相当长的时期能够正常运作所运行的软件活动。

软件维护类型：改正性；适应性；扩充与完善性；预防性。