软件危机：软件开发和维护过程中遇到的严重问题（开发效率低、成本高、质量差、维护难）

软件工程以质量为中心，软件过程（联系桥梁）、方法和工具（为过程和方法提供支持）为三要素，其基础是软件过程（Software Process ）。

软件=程序+数据+文档

软件测试只能表明软件中有缺陷，不能证明软件中没有缺陷

相对硬件，软件的特点是：逻辑的、知识性的智力产品，是对物理世界的一种抽象

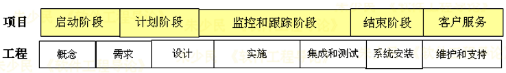
软件工程定义：The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.(2) The study of approaches as in (1)./计算机科学、工程、管理

传统软件工程（问题定义与可行性研究、需求分析、软件设计、编码、测试、运行与维护）/面向对象软件工程/软件过程工程(关键在于软件开发和维护中的管理和支持能力/基于构件的软件开发

软件生命周期(software life cycle)：问题定义与可行性研究/需求分析/软件设计/编码/测试/运行与维护(传统软工)；软件有一个孕育、诞生、成长、成熟、衰亡的生存过程。分析/设计/实现/维护；划分原因：问题分解，分别解决，从时间角度对问题进行分解，为软件人提供交流框架

概要设计：体现对需求的完整实现/保证与需求的一致性/达到向需求的反向可追踪/关注喜用结构设计的逻辑性、合理性和可扩展性；SA(结构分析)/SD(结构设计)

软件过程模型（模型是过程的抽象）

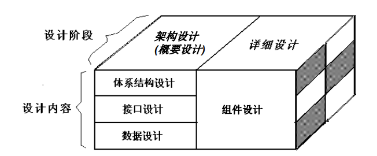
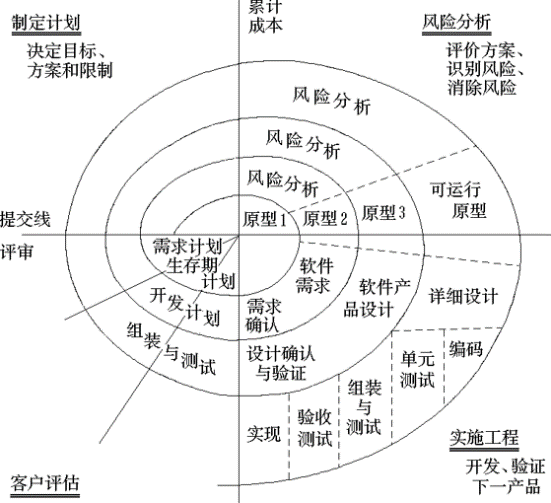
惯例过程模型： **软件过程模型**框架划分视角

瀑布模型：以文档为驱动、线性模型、经典的生命周期模型。顺序性、依赖性/瀑布模型适用于需求很明确并且将来没有太大改变的情况。

原型（不是一个软件产品）模型（Prototype Model）：首先实现软件最核心的 、最重要的功能，原型帮助导引出需求后再实现细节/适用场合：需求不明确或需求经常变更的情况，小型中型系统。/缺点：大型系统难建立稳定系统架构，大型系统结合瀑布和演化

增量（小而可用的软件）模型：进化式、建立一个不完整的系统，如此反复.需要更精心的设计.强调每个增量都是可运行的产品。有计划地管理技术风险/适用：需求经常变化；市场急需而开发人员和资金不能在设定的市场期限之前实现一个完善的产品/第一个增量通常是核心产品

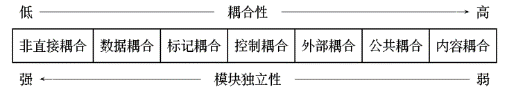
螺旋模型：/优点：强调严格的全过程风险管理/各开发阶段的质量/提供机会检讨项目是否有价值继续下去。 缺点：必须引入非常严格的风险识别，风险分析和风险控制，这对风险管理的技能水平提出了很高的要求。也需要人员，资金和时间的大量投入。



软件设计：可靠性/功能性/易用性；处于概要设计和详细设计时期；技术：抽象/细化/设计模式/模块化/信息隐藏/功能独立/重构(简化组件设计)

软件架构包括构件、连接件和约束、端口和角色

逻辑视图/UML用例图；开发视图/类图、对象图和组件图来表示模块，包表示子系统，连接表示模块或子系统之间的关联；过程视图/状态图、顺序图和活动图来实现/物理视图定义了功能单元的分布状况，描述用于执行用例和保存数据的业务地点/配置图

耦合(coupling)

面向对象设计包括：OOA(Object oriented analysis/分析类图)/OOD(设计类图)/OOP

OOD设计准则：模块化(类、对象、接口、构件/函数过程子程序(传统))/抽象(过程和数据)/信息隐藏(封装)/弱耦合(对象间关联)/强内聚/可重用

单一职责原则：一个类有且仅有一个职责，只有一个引起他变化的原因/开放关闭原则：应该通过扩展实现变化，而不是修改原来的代码/李氏替代原则：所有引用基类的地方必须能透明地使用其子类的对象/接口隔离原则：客户端需要什么接口，就提供给它什么样的接口，其它多余的接口就不要提供，不要让接口变得臃肿/依赖倒置原则：针对接口编程

继承是一种在父类 和子类之间共享属性和行为的方式 ，所以运行时可以用一个子类对象代替其父类对象 。根据为请求提供服务的对象不同可以得到不同的行为， 这种现象称为多态。如果在子类中不覆盖父 类中的任何方法，就不会产生多态行为

SA(结构化分析)SC(结构图)MVC(Model-View-Controller)/ **(whywhatwhenwhowherehowhowmuc)/WBS(work breakdown structure)**

项目管理：通过项目经理和项目组织的努力，运用系统理论的方法对项目及其资源进行计划、组织、协调、控制，旨在实现项目的特定目标的管理方法体系。关注点：人员/产品/过程/项目。管理过程：软件项目启动、建立项目组织、项目计划、跟踪与控制

(确定产品范围)WBS作用：是计划和管理项目进程、成本和变化，风险分析，组织结构、方法的基础

项目估算(成本和工作量)估算方法：LOC/FP(以问题分解为基础)/FP(CT\*0.65+0.01\*F)

进度管理：确保软件项目在规定时间内按期完成。延误原因：不切实际期限，需求变更，低估工作量

Scheduling steps:定义所有的项目任务/确定项目任务间的依赖关系/规划每个任务所需工作量和持续时间，构建项目任务网络/关键路分析/甘特图/跟踪任务进展

风险分类：项目风险(项目预算进度人力资源顾客需求等产生不良影响)，技术风险，商业风险(市场/策略/销售/预算)

风险：人员、经费、进度及需求等方面存在的可能影响项目按计划完成的不确定因素

有效的(风险规划) RMMM：风险避免(mitigation)在风险发生前分析原因采取措施避免发生

风险检测：评估预测风险是否发生，保证正确实施各项缓解步骤，收集信息

风险管理及应急计划(RMP)：风险避免和缓解工作已经失败，风险已经发生

质量控制/质量保证：软件工程师\SQA(software quality assurance)小组

评审：软件工程过程的每个活动(需求分析、设计)的后期进行

软件度量：有效的采用定量的方式进行管理(了解然间工程的执行情况和产品质量)

软件质量：功能性F/usability/可靠性R/performance/serviceability/maintainability