**一什么是软件危机表现1**20世纪60年代末，软件规模扩大、软件复杂性提高导致软件开发开发周期长、成本高、质量差和维护困难，爆发了软件危机2许多软件项目不能满足客户的要求3许多软件项目超出预算和时间安排 表现1成本和进度估计不准确 2用户不满意 3质量靠不住 4不可维护 5没有适当的文档资料**二管理的目标**1达到功能性能要求2时限要求3开销在预算之内**三软件工程方法学三要素**1**方法**提供如何做的技术2**工具**为方法提供自动的或半自动的软件支撑环境3**过程**是为了获得高质量的软件所需要完成的一系列任务框架，它规定了完成各项任务的工作步骤**四软件工程的目标**1运用先进的软件开发技术和管理方法来提高软件的质量和生产率2短周期低成本高质量工业化生产**五软件规模估算**1功能点方法：外部输入、外部输出、内部逻辑文件、外部接口文件、外部查询2调节因子3交付功能点：DFP = CAF × UFPCAF：复杂度调节因子UFP: 未调节功能点4交付功能点与软件规模**六软件质量**功能、可靠、可使用、效率、可维护、可移植**七CY面向对象设计模型**：问题域、人机交互、任务管理、数据管理**八需求分析概要设计等各阶段在做什么**1需求分析：A准确地回答：“目标系统必须做什么”这个问题B分析内容：对目标系统提出完整、准确、清晰、具体的要求。对用户提出的需求进行分析并给出详细的定义C输出（软件需求说明书、或系统功能说明书、初步的系统用户手册）：提交管理机构评审2可行性：A明确回答：“上一个阶段所确定的要解决的问题是否有行得通的解决办法”B输出（可行性研究报告）：从技术、经济和社会因素等方面研究论证各方案的可行性。 为决策提供依据3概要设计:A概括地回答：“怎样实现目标系统?”A设计内容：设计程序的体系结构，即确定程序的组成模块，确定模块间的关系B输出文档：概要设计说明书4详细设计：A详细回答：“应怎样具体实现这个系统？”B设计内容：详细地设计每个模块，确定实现模块功能所需要的算法和数据结构C输出文档：详细设计说明书5编码和单元测试：A实现：正确的、容易理解的、容易维护的程序模块B输出文档：源程序、程序说明和单元测试报告6集成测试和系统测试：A集成测试：对模块的组装和连接测试，使软件达到预定的要求B系统测试：根据需求规格说明，逐项测试并确认C输出：测试计划、测试方案测试结果7运行维护:A改正性维护B适应性维护C完善性维护D预防性维护**九面向对象设计过程**与准则：1模块化2抽象3信息隐藏4弱耦合5强内聚6可重用**十耦合**：内容耦合、公共耦合、外部耦合、控制耦合、标记耦合、数据耦合、非直接耦合补充：交互耦合：对象之间的耦合通过消息来实现，就是交互耦合**十一内聚**：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

**一什么是软件危机表现1**20世纪60年代末，软件规模扩大、软件复杂性提高导致软件开发开发周期长、成本高、质量差和维护困难，爆发了软件危机2许多软件项目不能满足客户的要求3许多软件项目超出预算和时间安排 表现1成本和进度估计不准确 2用户不满意 3质量靠不住 4不可维护 5没有适当的文档资料**二管理的目标**1达到功能性能要求2时限要求3开销在预算之内**三软件工程方法学三要素**1**方法**提供如何做的技术2**工具**为方法提供自动的或半自动的软件支撑环境3**过程**是为了获得高质量的软件所需要完成的一系列任务框架，它规定了完成各项任务的工作步骤**四软件工程的目标**1运用先进的软件开发技术和管理方法来提高软件的质量和生产率2短周期低成本高质量工业化生产**五软件规模估算**1功能点方法：外部输入、外部输出、内部逻辑文件、外部接口文件、外部查询2调节因子3交付功能点：DFP = CAF × UFPCAF：复杂度调节因子UFP: 未调节功能点4交付功能点与软件规模**六软件质量**功能、可靠、可使用、效率、可维护、可移植**七CY面向对象设计模型**：问题域、人机交互、任务管理、数据管理**八需求分析概要设计等各阶段在做什么**1需求分析：A准确地回答：“目标系统必须做什么”这个问题B分析内容：对目标系统提出完整、准确、清晰、具体的要求。对用户提出的需求进行分析并给出详细的定义C输出（软件需求说明书、或系统功能说明书、初步的系统用户手册）：提交管理机构评审2可行性：A明确回答：“上一个阶段所确定的要解决的问题是否有行得通的解决办法”B输出（可行性研究报告）：从技术、经济和社会因素等方面研究论证各方案的可行性。 为决策提供依据3概要设计:A概括地回答：“怎样实现目标系统?”A设计内容：设计程序的体系结构，即确定程序的组成模块，确定模块间的关系B输出文档：概要设计说明书4详细设计：A详细回答：“应怎样具体实现这个系统？”B设计内容：详细地设计每个模块，确定实现模块功能所需要的算法和数据结构C输出文档：详细设计说明书5编码和单元测试：A实现：正确的、容易理解的、容易维护的程序模块B输出文档：源程序、程序说明和单元测试报告6集成测试和系统测试：A集成测试：对模块的组装和连接测试，使软件达到预定的要求B系统测试：根据需求规格说明，逐项测试并确认C输出：测试计划、测试方案测试结果7运行维护:A改正性维护B适应性维护C完善性维护D预防性维护**九面向对象设计过程**与准则：1模块化2抽象3信息隐藏4弱耦合5强内聚6可重用**十耦合**：内容耦合、公共耦合、外部耦合、控制耦合、标记耦合、数据耦合、非直接耦合补充：交互耦合：对象之间的耦合通过消息来实现，就是交互耦合**十一内聚**：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

**一什么是软件危机表现1**20世纪60年代末，软件规模扩大、软件复杂性提高导致软件开发开发周期长、成本高、质量差和维护困难，爆发了软件危机2许多软件项目不能满足客户的要求3许多软件项目超出预算和时间安排 表现1成本和进度估计不准确 2用户不满意 3质量靠不住 4不可维护 5没有适当的文档资料**二管理的目标**1达到功能性能要求2时限要求3开销在预算之内**三软件工程方法学三要素**1**方法**提供如何做的技术2**工具**为方法提供自动的或半自动的软件支撑环境3**过程**是为了获得高质量的软件所需要完成的一系列任务框架，它规定了完成各项任务的工作步骤**四软件工程的目标**1运用先进的软件开发技术和管理方法来提高软件的质量和生产率2短周期低成本高质量工业化生产**五软件规模估算**1功能点方法：外部输入、外部输出、内部逻辑文件、外部接口文件、外部查询2调节因子3交付功能点：DFP = CAF × UFPCAF：复杂度调节因子UFP: 未调节功能点4交付功能点与软件规模**六软件质量**功能、可靠、可使用、效率、可维护、可移植**七CY面向对象设计模型**：问题域、人机交互、任务管理、数据管理**八需求分析概要设计等各阶段在做什么**1需求分析：A准确地回答：“目标系统必须做什么”这个问题B分析内容：对目标系统提出完整、准确、清晰、具体的要求。对用户提出的需求进行分析并给出详细的定义C输出（软件需求说明书、或系统功能说明书、初步的系统用户手册）：提交管理机构评审2可行性：A明确回答：“上一个阶段所确定的要解决的问题是否有行得通的解决办法”B输出（可行性研究报告）：从技术、经济和社会因素等方面研究论证各方案的可行性。 为决策提供依据3概要设计:A概括地回答：“怎样实现目标系统?”A设计内容：设计程序的体系结构，即确定程序的组成模块，确定模块间的关系B输出文档：概要设计说明书4详细设计：A详细回答：“应怎样具体实现这个系统？”B设计内容：详细地设计每个模块，确定实现模块功能所需要的算法和数据结构C输出文档：详细设计说明书5编码和单元测试：A实现：正确的、容易理解的、容易维护的程序模块B输出文档：源程序、程序说明和单元测试报告6集成测试和系统测试：A集成测试：对模块的组装和连接测试，使软件达到预定的要求B系统测试：根据需求规格说明，逐项测试并确认C输出：测试计划、测试方案测试结果7运行维护:A改正性维护B适应性维护C完善性维护D预防性维护**九面向对象设计过程**与准则：1模块化2抽象3信息隐藏4弱耦合5强内聚6可重用**十耦合**：内容耦合、公共耦合、外部耦合、控制耦合、标记耦合、数据耦合、非直接耦合补充：交互耦合：对象之间的耦合通过消息来实现，就是交互耦合**十一内聚**：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

**十二软件生存期**：1软件定义：问题定义、可行性分析、需求分析2软件开发：A任务：具体设计和实现软件定义时期定义的软件B执行人：系统设计员，高级程序员，程序员，测试工程师和辅助人员等C阶段划分：分为概要设计、详细设计、编码和单元测试、集成测试和系统测试3运行维护:目的：使软件能持久地满足用户的需要、4类维护**十三C/S**：基于资源不对等，且为实现共享而提出来的，由服务器、客户机和网络三部分组成**十四三层C/S分层**：1表示层：用户界面部分，用户与应用程序之间对话。2应用逻辑层：主体部分，包含具体的业务处理逻辑。通常在功能层中还包含有确认用户对应用和数据库存取权限的功能以及记录系统处理日志的功能3数据层：主要包括数据的存储及对数据的存取操作，一般选择关系型数据库管理系统**十五三层C/S特点**：克服两层的缺点，增加了应用服务器。可以将整个应用逻辑驻留在应用服务器上，而只有表示层存在于客户机上**十六瘦客户机**：1特点：数据管理部分和应用逻辑都在服务器上执行，客户机只负责表示部分2缺点：A它将繁重的处理负荷都放在了服务器和网络上，服务器负责所有的计算，这将增加客户机和服务器之间的网络流量。A目前PC机所具有的处理能力在瘦客户机模型中用不上。**十七B/S**：1优点：A系统安装、修改和维护全在服务器端解决B提供异种机、异种网、异种应用服务的联机、联网和统一服务的最现实的开放性基2缺点：A缺乏对动态页面的支持能力，没有集成有效的数据库处理功能B在数据查询等响应速度上，要远远地低于C/S体系结构C数据提交一般以页面为单位，数据的动态交互性不强，不利于在线事务处理**十八过程模型**:瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、统一过程、基于构件的开发模型、敏捷过程**十九瀑布模型特点**：1阶段间的顺序性和依赖性2推迟实现的观点3质量保证的观点 V模型描述了测试阶段的活动与开发阶段相关活动之间的关系**二十瀑布模型的优、缺点及使用范围**：1优点：A强制开发人员采用规范化的方法B严格规定每个阶段必须提交的文档C每个阶段交出的所有产品都必须经过验证2缺点：完全依赖书面的规格说明，如需求规格说明与用户需求之间有差异，则可能导致最终开发出的软件产品不能真正满足用户的需要3适用范围：只适用于项目开始时需求已确定的情况。**二十一设计模式**1创建型模式：工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式2结构型模式：适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式3行为型模式：策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式。**二十二单例模式：1**作用：确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例单例模式2场景：只应在有真正的“单一实例”的需求时才可使用。 **二十三简单工厂模式**1作用就是建立一个工厂类，对实现了同一接口的一些类进行实例的创建2场景：1工厂类负责创建的对象比较少2客户端只知道传入工厂类的参数，对于如何创建对象不关心

**十二软件生存期**：1软件定义：问题定义、可行性分析、需求分析2软件开发：A任务：具体设计和实现软件定义时期定义的软件B执行人：系统设计员，高级程序员，程序员，测试工程师和辅助人员等C阶段划分：分为概要设计、详细设计、编码和单元测试、集成测试和系统测试3运行维护:目的：使软件能持久地满足用户的需要、4类维护**十三C/S**：基于资源不对等，且为实现共享而提出来的，由服务器、客户机和网络三部分组成**十四三层C/S分层**：1表示层：用户界面部分，用户与应用程序之间对话。2应用逻辑层：主体部分，包含具体的业务处理逻辑。通常在功能层中还包含有确认用户对应用和数据库存取权限的功能以及记录系统处理日志的功能3数据层：主要包括数据的存储及对数据的存取操作，一般选择关系型数据库管理系统**十五三层C/S特点**：克服两层的缺点，增加了应用服务器。可以将整个应用逻辑驻留在应用服务器上，而只有表示层存在于客户机上**十六瘦客户机**：1特点：数据管理部分和应用逻辑都在服务器上执行，客户机只负责表示部分2缺点：A它将繁重的处理负荷都放在了服务器和网络上，服务器负责所有的计算，这将增加客户机和服务器之间的网络流量。A目前PC机所具有的处理能力在瘦客户机模型中用不上。**十七B/S**：1优点：A系统安装、修改和维护全在服务器端解决B提供异种机、异种网、异种应用服务的联机、联网和统一服务的最现实的开放性基2缺点：A缺乏对动态页面的支持能力，没有集成有效的数据库处理功能B在数据查询等响应速度上，要远远地低于C/S体系结构C数据提交一般以页面为单位，数据的动态交互性不强，不利于在线事务处理**十八过程模型**:瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、统一过程、基于构件的开发模型、敏捷过程**十九瀑布模型特点**：1阶段间的顺序性和依赖性2推迟实现的观点3质量保证的观点 V模型描述了测试阶段的活动与开发阶段相关活动之间的关系**二十瀑布模型的优、缺点及使用范围**：1优点：A强制开发人员采用规范化的方法B严格规定每个阶段必须提交的文档C每个阶段交出的所有产品都必须经过验证2缺点：完全依赖书面的规格说明，如需求规格说明与用户需求之间有差异，则可能导致最终开发出的软件产品不能真正满足用户的需要3适用范围：只适用于项目开始时需求已确定的情况。**二十一设计模式**1创建型模式：工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式2结构型模式：适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式3行为型模式：策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式。**二十二单例模式：1**作用：确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例单例模式2场景：只应在有真正的“单一实例”的需求时才可使用。 **二十三简单工厂模式**1作用就是建立一个工厂类，对实现了同一接口的一些类进行实例的创建2场景：1工厂类负责创建的对象比较少2客户端只知道传入工厂类的参数，对于如何创建对象不关心

**十二软件生存期**：1软件定义：问题定义、可行性分析、需求分析2软件开发：A任务：具体设计和实现软件定义时期定义的软件B执行人：系统设计员，高级程序员，程序员，测试工程师和辅助人员等C阶段划分：分为概要设计、详细设计、编码和单元测试、集成测试和系统测试3运行维护:目的：使软件能持久地满足用户的需要、4类维护**十三C/S**：基于资源不对等，且为实现共享而提出来的，由服务器、客户机和网络三部分组成**十四三层C/S分层**：1表示层：用户界面部分，用户与应用程序之间对话。2应用逻辑层：主体部分，包含具体的业务处理逻辑。通常在功能层中还包含有确认用户对应用和数据库存取权限的功能以及记录系统处理日志的功能3数据层：主要包括数据的存储及对数据的存取操作，一般选择关系型数据库管理系统**十五三层C/S特点**：克服两层的缺点，增加了应用服务器。可以将整个应用逻辑驻留在应用服务器上，而只有表示层存在于客户机上**十六瘦客户机**：1特点：数据管理部分和应用逻辑都在服务器上执行，客户机只负责表示部分2缺点：A它将繁重的处理负荷都放在了服务器和网络上，服务器负责所有的计算，这将增加客户机和服务器之间的网络流量。A目前PC机所具有的处理能力在瘦客户机模型中用不上。**十七B/S**：1优点：A系统安装、修改和维护全在服务器端解决B提供异种机、异种网、异种应用服务的联机、联网和统一服务的最现实的开放性基2缺点：A缺乏对动态页面的支持能力，没有集成有效的数据库处理功能B在数据查询等响应速度上，要远远地低于C/S体系结构C数据提交一般以页面为单位，数据的动态交互性不强，不利于在线事务处理**十八过程模型**:瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、统一过程、基于构件的开发模型、敏捷过程**十九瀑布模型特点**：1阶段间的顺序性和依赖性2推迟实现的观点3质量保证的观点 V模型描述了测试阶段的活动与开发阶段相关活动之间的关系**二十瀑布模型的优、缺点及使用范围**：1优点：A强制开发人员采用规范化的方法B严格规定每个阶段必须提交的文档C每个阶段交出的所有产品都必须经过验证2缺点：完全依赖书面的规格说明，如需求规格说明与用户需求之间有差异，则可能导致最终开发出的软件产品不能真正满足用户的需要3适用范围：只适用于项目开始时需求已确定的情况。**二十一设计模式**1创建型模式：工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式2结构型模式：适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式3行为型模式：策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式。**二十二单例模式：1**作用：确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例单例模式2场景：只应在有真正的“单一实例”的需求时才可使用。 **二十三简单工厂模式**1作用就是建立一个工厂类，对实现了同一接口的一些类进行实例的创建2场景：1工厂类负责创建的对象比较少2客户端只知道传入工厂类的参数，对于如何创建对象不关心