

1. 逆向工程

软件的逆向工程是指通过分析目标系统，标识系统的部件及其交互关系，然后使用其他形式或者更高层的抽象来创建系统表现的过程。凡是在软件生命周期内将软件的某种形式描述转换成更为抽象形式的活动都可称为逆向工程。

① 程序二进制代码->源代码；

② 程序源代码->控制流图、数据流图、类图等。

具体流程包括解密、反汇编等对软件逆向拆解和分析。

2. 重构

不改变代码外在行为来修改代码改进程序内部结构，提高程序可理解性和可维护性。

在程序内部结构中可能出现的各种不合理情形称为代码坏味道，它是设计质量低下的表现。代码坏味道的类型包括①重复代码；②过长方法；③过大的类；④过多方法参数；⑤同样的 switch 语句散布在多处；⑥过多的注释。

- 重构的重点是改进详细设计结构，虽然理论上重构也能改进体系结构设计，但由于体系结构的修改影响范围广泛，因此不要轻易修改。

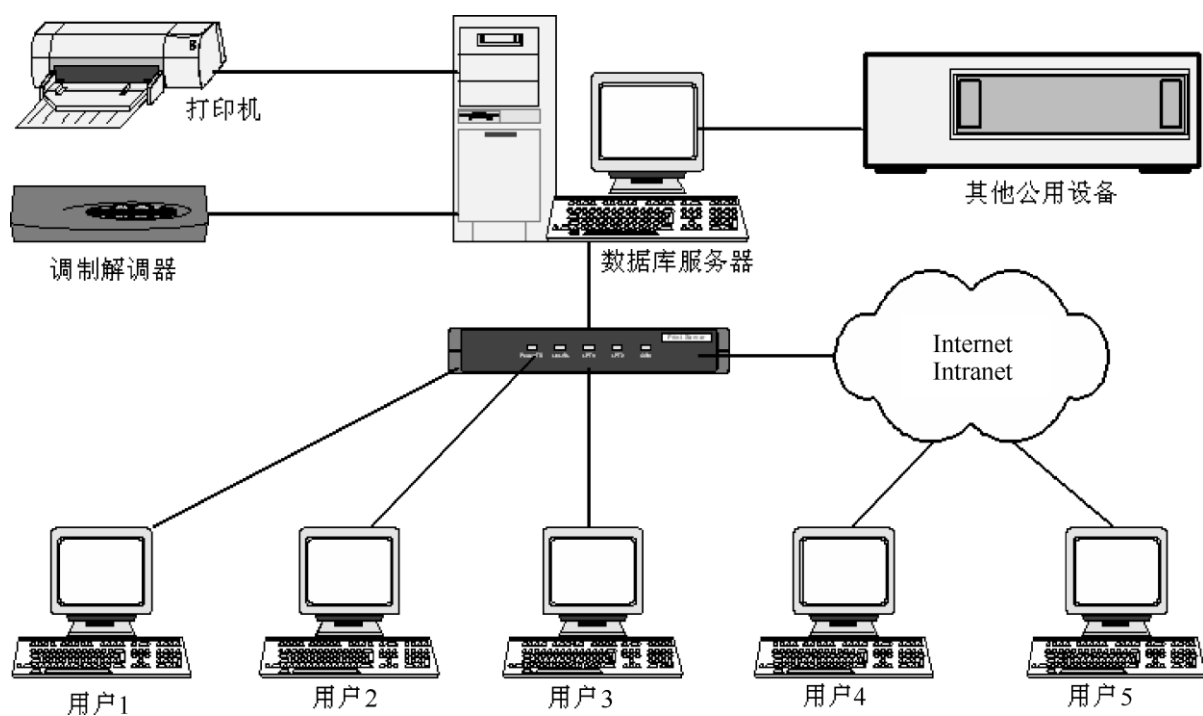
3. 正向工程

比如从 UML 图形生成代码。

4. 仓库风格

数据库系统、超文本系统、黑板系统等。

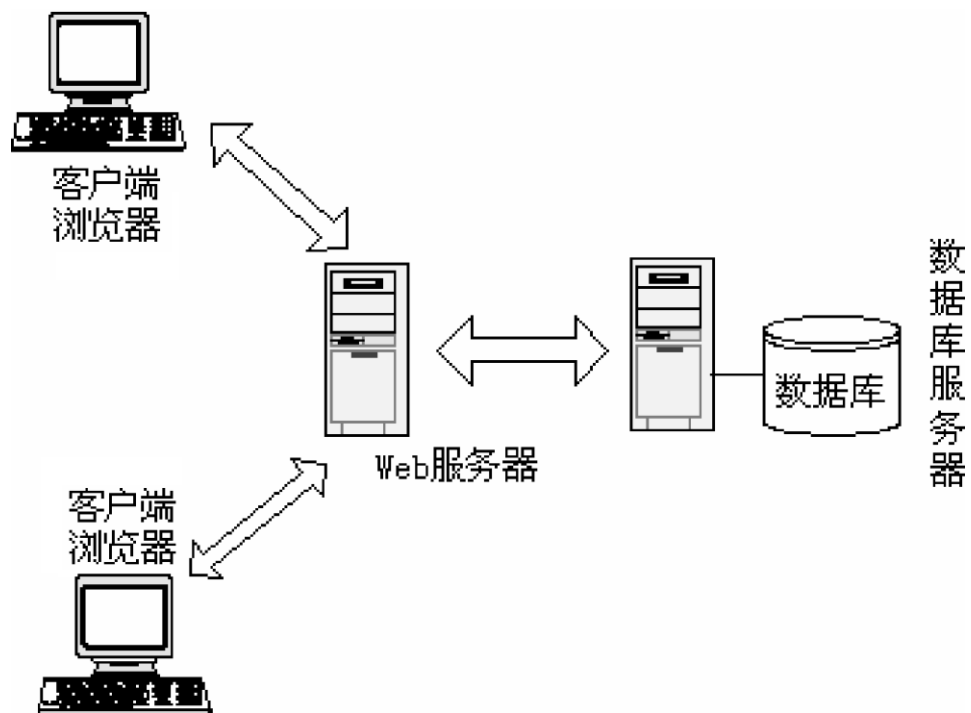
5. C/S 结构



两层 C/S 结构可以有两种形态：① 瘦客户机模型；② 胖客户机模型。

三层的话可以分成表示层、应用逻辑层、数据层。

浏览器/服务器风格（B/S）是三层体系结构的一种实现方法



6. 配置管理

- 软件配置管理的目的是为某个过程或某个项目的软件项建立和保持完整性，以便相关方便于使用。
- 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、交付等。

软件配置管理的对象叫做**软件配置项**。包括文档、数据、代码、相关产品。

7. 版本控制

包括版本管理、发行管理。

(1) 版本管理。

版本是软件系统的一个实例。

是对系统不同版本进行标识和跟踪的过程。

目的是方便区分、检索、跟踪版本，表明各个版本之间的关系。

版本包括**版本标识**，用命名的方式反映了前后版本的传递关系，包括①号码顺序型的版本标识（V1.0、V1.1……）、②符号命名版本标识（V1/VMS/DB Server）、③属性版本标识（客户名、开发语言、开发状态、硬件平台、生成日期）。

(2) 系统发行

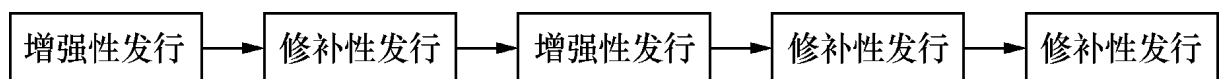
有些版本是会分配给用户的，这类版本就叫系统发行。

除了发行可执行程序外，还可能包括配置文件、数据文件、安装程序、文档等。

➤ 新版本是在修改发现的软件缺陷后，开发出新的程序、形成新的系统；

- 新发行是除了写出新的程序，形成新系统之外，还要为用户准备数据、配置文件、编写新文档，准备新包装。
- 新发行要比新版本开销大。

系统发行策略：



8. 里程碑

里程碑就是项目开发过程中一个重要的阶段点，这个阶段点之后将意味着项目的开发进入了另一个层面，比如需求分析结束概要设计开始前就应该有一个里程碑，表示需求阶段结束，将进入另外一个过程，这两个过程之间只有文档的交互就可以了，而不一定非要有人员的交互。

阶段就是项目开发过程中明显可以区分的任务分解，需求收集和需求分析就可以划分为两个阶段，但都数据需求这个大阶段中的，但需求收集和需求分析之间就不能存在一个里程碑，因为你没有完成一个完整的过程。需求收集和需求分析阶段至少有客户是必须始终参与的，而概要设计阶段一般就不需要客户的参与了。

9. 判定表

eg: 假设某航空公司规定, 乘客可以免费托运重量不超过30kg的行李。当行李重量超过30kg时, 对头等舱的国内乘客超重部分每公斤收费4元, 对其他舱的国内乘客超重部分每公斤收费6元, 对外国乘客超重部分每公斤收费比国内乘客多一倍, 对残疾乘客超重部分每公斤收费比正常乘客少一半。用判定表进行表达。

1. 我们先将所有情况进行列出, 就是面对行李是否<30kg, 是否为国内乘客, 是否为头等舱, 是否是残疾乘客我们托运行李的价格不一样
2. T代表满足情况, F代表不满足, X代表我们应该付的价格 (我个人建议你根据这个模板自己写其实更简单, 顺序不同没有关系, 其实就是排列组合所有情况)

表 6.1 用判定表表示计算行李费的算法

	规 则								
	1	2	3	4	5	6	7	8	9
国内乘客		T	T	T	T	F	F	F	F
头等舱		T	F	T	F	T	F	T	F
残疾乘客		F	F	T	T	F	F	T	T
行李重量 $W \leq 30\text{kg}$	T	F	F	F	F	F	F	F	F
免费	×								
$(W-30) \times 2$				×					
$(W-30) \times 3$					×				
$(W-30) \times 4$		×						×	
$(W-30) \times 6$			×						×
$(W-30) \times 8$						×			
$(W-30) \times 12$							×		

https://blog.csdn.net/weixin_45626404

10. IPO 图

IPO 是 Input Process Output 的意思，指结构化设计中变换型结构的输入 (Input)、加工 (Processing)、输出 (Output)。

IPO图就是用来说明每个模块的输入、输出数据和数据加工的重要工具

B、基本元素：可根据系统流程图中基本元素看，包括输入，处理，输出。

C、例图（机房收费系统以操作员为例）：



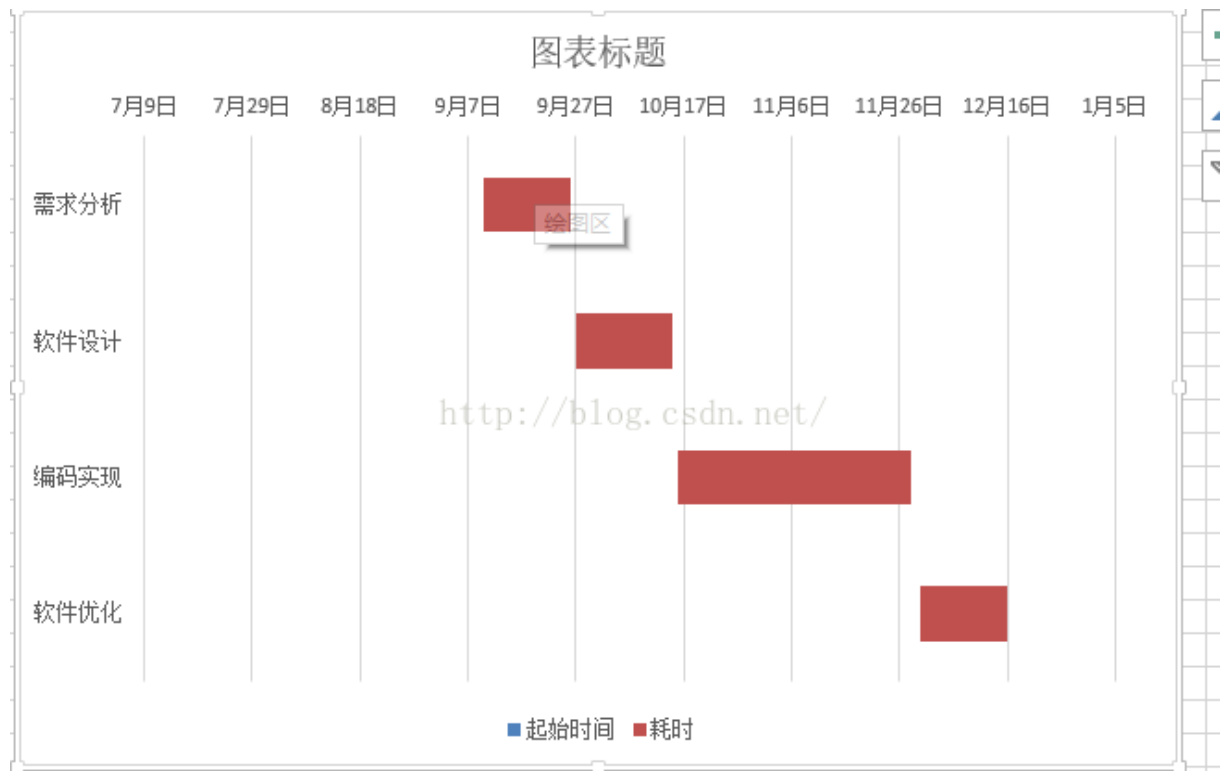
11. PDL

就是伪代码。

12. 甘特图

又称为横道图、条状图(Bar chart)。以图示的方式通过活动列表和时间刻度形象地表示出任何特定项目的活动顺序与持续时间。基本是一条线条图，横轴表示时

间，纵轴表示活动（项目），线条表示在整个期间上计划和实际的活动完成情况。它直观地表明任务计划在什么时候进行，及实际进展与计划要求的对比。



13. 质量管理

1) 质量策划 (Quality Planning — QP)

在确定项目的质量目标基础上，规划需投入的相关资源（包括人力、时间、开发费用等）。

(2) 质量控制 (Quality Control — QC)

明确为达到质量目标应开展的活动，如测试、评审等。

(3) 质量保证（Quality Assurance — QA）

明确为达到质量目标应提供信任的活动，以表明质量控制活动是有效的。

(4) 质量改进（Quality Improvement — QI）

为提高软件产品的有效性和效率而开展的活动。

对比项	质量控制(QC)	质量保证(QA)
目的	尽力使产品达到质量要求	为产品达到质量要求提供信任
对象	产品	生产过程（包括质量控制过程）
做法	通过检测找出产品缺陷，并分析原因加以解决	收集、通报缺陷信息，促进并确保缺陷得到解决
责任人	质检人员、测试人员	质保人员，包括第三方人员
对谁负责	生产部门	企业领导及用户
举例	检验、测试、评审、纠错	评审、审核

常用的质量保障措施：① 计划和标准制定；② 过程控制；③ 文档记录；④ 测试；⑤ 阶段评审；⑥ 客户沟通……