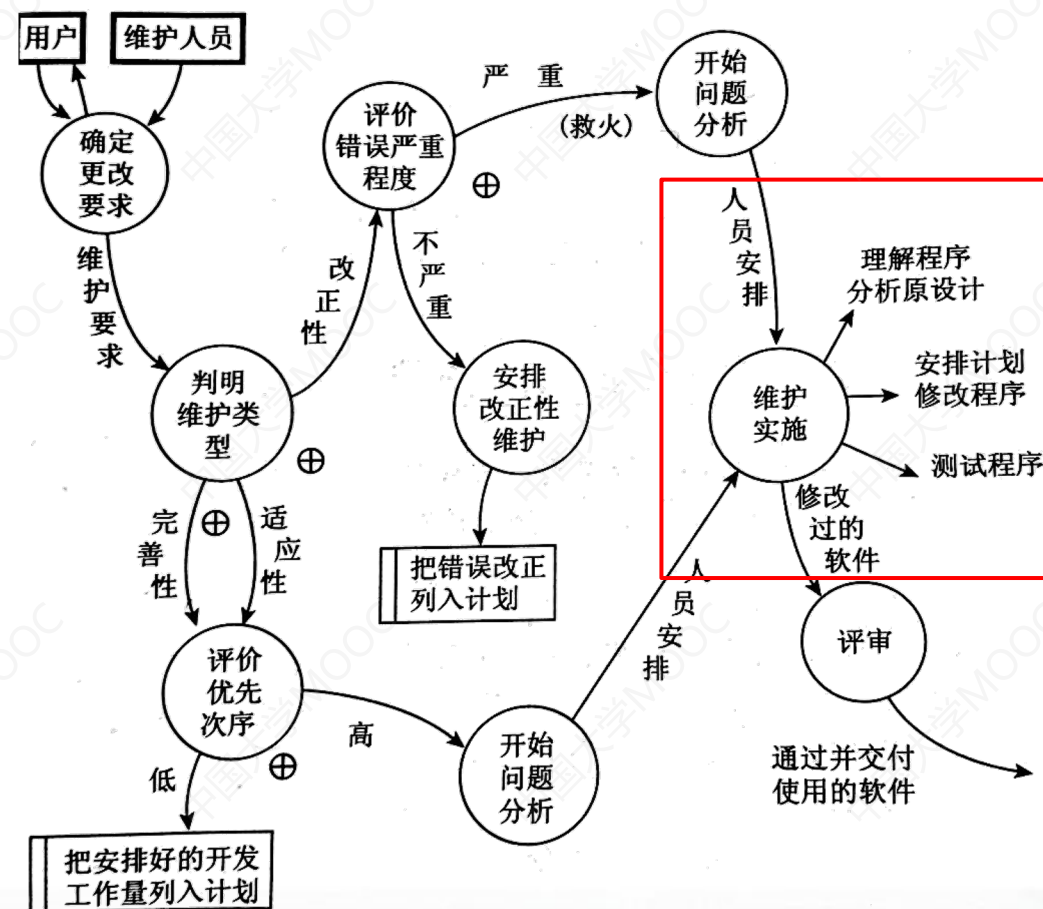


程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

- 对源程序的修改是维护实施中最核心的步骤。
- 修改源程序不能仓促上阵，需要：
 - 分析和理解程序
 - 实施修改
 - 重新测试程序



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

全面、准确、迅速理解程序是决定维护成败和质量好坏的关键

1、分析程序的目标：

- 研究程序使用环境和有关资料，得到尽可能多的背景信息
- 理解程序功能和目标
- 掌握程序结构信息（控制结构、数据结构、输入输出结构等）
- 了解数据流信息（数据来自何处，在哪里被使用）
- 了解控制流信息（执行每条路径的结果）
- 如果设计资料存在，则可利用他们画出结构图和高层流程图
- 理解程序的操作（使用）要求



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

2、分析理解源程序的方法

- 1) 分析程序结构图：收集所有存储程序的文件、阅读文件并记录所有文件名和过程名，分析各个过程的代码，分析各个过程的接口，并估计更改的复杂性
- 2) 数据追踪：建立各层次的程序级上的接口图，展示各模块的调用方式和接口参数；利用数据流分析方法对过程内部的变量进行跟踪
- 3) 控制跟踪：可采用符号执行或实际动态跟踪方法，了解数据如何从输入源到达输出点



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

2、分析理解源程序的方法

- 4) 充分阅读和使用源程序清单与文档，并分析现有文档的合理性
- 5) 充分使用由编译程序或汇编程序提供的交叉引用表、符号表等信息
- 6) 争取参加开发工作



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

对程序的修改也必须事先进行计划，有准备地、周密有效地实施

1、修改程序准备：制定程序修改计划

- 小的修改可以不需要详细计划，但面对历时数月的修改必须进行计划立案
- (1) 程序修改计划的内容
 - 规格说明信息：数据修改、处理修改、作业控制语言修改、系统之间接口的修改等
 - 维护资源：新程序版本、测试数据、所需的软件系统等
 - 人员：程序员、用户相关人员、技术支持人员、厂家联系人、数据录入人员等



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

- (2) 修改前的准备工作过程

- 研究程序各个模块、模块的接口及数据库，从全局观点提出修改计划
- 依次分离需要修改以及受修改影响的模块与数据结构
- 详细分析分离出的受修改影响的模块的内部细节，设计详细修改计划，标明新逻辑及要改动的现有逻辑
- 向用户提供回避措施
 - 在问题原因还未找到时，可先就问题的现象提供回避的操作方法
 - 如：意外停机——消除特定的数据、插入临时代码（打补丁），以人工方式运行系统
 - 弄清问题原因后，可通过临时修改或改变运行控制以回避在系统运行时产生的问题



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

2、修改程序，以适应变化

- 要求：
 - 正确有效地修改代码
 - 保持程序风格及格式，在程序清单上注明改动的指令
 - 不要匆忙删除语句，除非完全肯定其是无用的
 - 插入错误检测语句
 - 保持详细的维护活动和维护结果记录
 - 如果程序结构混乱，修改受到干扰，可考虑重新编写
 -



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

2、修改程序，以适应变化

● 修改程序的副作用

- 1) 修改源代码时，可能引入新的错误
 - 如删除、修改一个标号，改变代码时序关系等，都很容易引入错误
- 2) 修改数据结构时，可能造成软件设计与数据结构不匹配，导致软件出错
 - 如重新定义全局的常量、重新定义数据记录格式等
- 3) 修改数据流、软件结构、模块逻辑等时，必须对相关技术文档进行相应修改（否则造成文档与程序功能不匹配）



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

2、修改程序，以适应变化

- 副作用的控制
 - 1) 按模块把修改分组
 - 2) 自顶向下地安排被修改模块的顺序
 - 3) 每次修改一个模块
 - 4) 在安排下个模块修改前，先确定本模块有无副作用



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

运用静态确认、回归测试、维护后验收的方法，对程序进行充分的确认和测试，保证程序修改的正确性

1、静态确认（代码审查）

- 至少两人参加，对程序进行静态的验证：
 - 修改是否涉及规格说明？有没有歪曲规格说明？
 - 修改是否足以修正软件中的问题？源代码有无逻辑错误？修改时有没有修补失误？
 - 修改部分对其它部分有无不良影响？



北京大学

程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

2、回归测试

- 回归测试的概念：先对修改部分和未修改部分分别进行隔离、单独测试，再将它们集成起来进行集成测试
- 需要为本次修改准备测试用例
- 充分利用软件工具帮助重新验证过程
- 测试种类：对修改程序的测试、操作过程的测试、应用系统运行过程的测试、使用过程的测试、系统各部分之间接口的测试、与系统软件接口的测试、系统之间接口的测试、安全性测试、后备/恢复过程的测试等



程序修改

- 引入
- 分析和理解程序
- 修改程序流程
- 重新验证程序

3、维护后验收

- 维护主管部门在交付前验收：
 - 全部文档是否完备并已更新
 - 所有测试用例和测试结果已经正确记载
 - 记录软件配置所有副本的工作已经完成
 - 维护工序和责任是明确的



北京大学