软件维护
software maintenance

内容提要

- 软件维护的定义
- 软件维护的类型
- 结构化维护VS非结构化维护
- 影响软件维护工作量的因素
- 软件维护的过程
- 可维护性（maintenanceability）
- 软件维护的管理

软件维护的定义

- 软件维护是指软件系统交付使用以后，为了改正错误或满足新的需要而修改软件的过程。

- 一般来说，要求进行维护的原因大致有以下几种：

（1）改正程序中的错误和缺陷。

（2）改进设计以适应新的软、硬件环境。

（3）增加新的应用范围。

软件维护的类型

- 根据软件维护的不同原因，软件维护可以分成4种类型：
  - 改正性维护
  - 适应性维护
  - 完善性维护
  - 预防性维护

改正性维护

- 在软件交付使用后，因开发阶段的问题以及测试得不彻底、不完全，必然会有部分隐藏的错误遗留到运行阶段。
- 为了识别和纠正软件错误、改正软件功能、非功能（性能）上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程就叫做改正性维护。

适应性维护

- 在使用过程中，
  - 外部环境（新的硬、软件配置）
  - 数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）

  可能发生变化。
- 为使软件适应这种变化，而去修改软件的过程就叫做适应性维护。

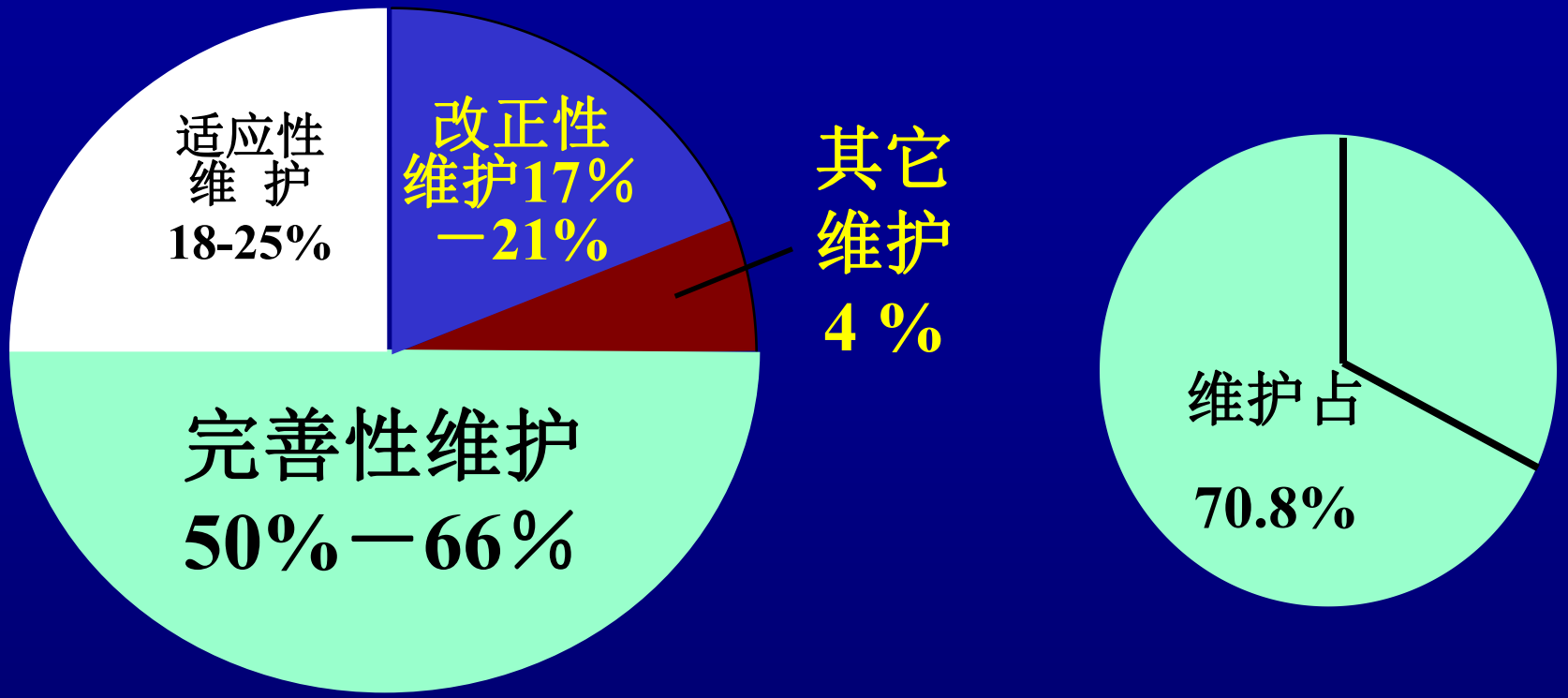
完善性维护

- 在软件的使用过程中，用户往往会对软件提出新的功能与性能要求。
- 为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。
- 这种情况下进行的维护活动叫做完善性维护。

预防性维护

- 预防性维护即软件再工程，是为了提高软件的可维护性、可靠性等，为以后进一步改进软件打下良好基础。
- 采用先进的软件工程方法对需要维护的软件或软件中的某一部分（重新）进行设计、编制和测试，称为预防性维护。

各种维护类型和维护工作量的比例



结构化维护VS非结构化维护

- 软件的开发过程对软件的维护产生较大的影响。
  - 如果采用软件工程的方法进行软件开发，保证每个阶段都有完整且详细的文档，这样维护会相对容易，被称为结构化的维护。
  - 反之，如果不采用软件工程方法开发软件，软件只有程序而欠缺文档，则维护工作变得十分困难，被成为非结构化的维护。

影响软件维护工作量的因素

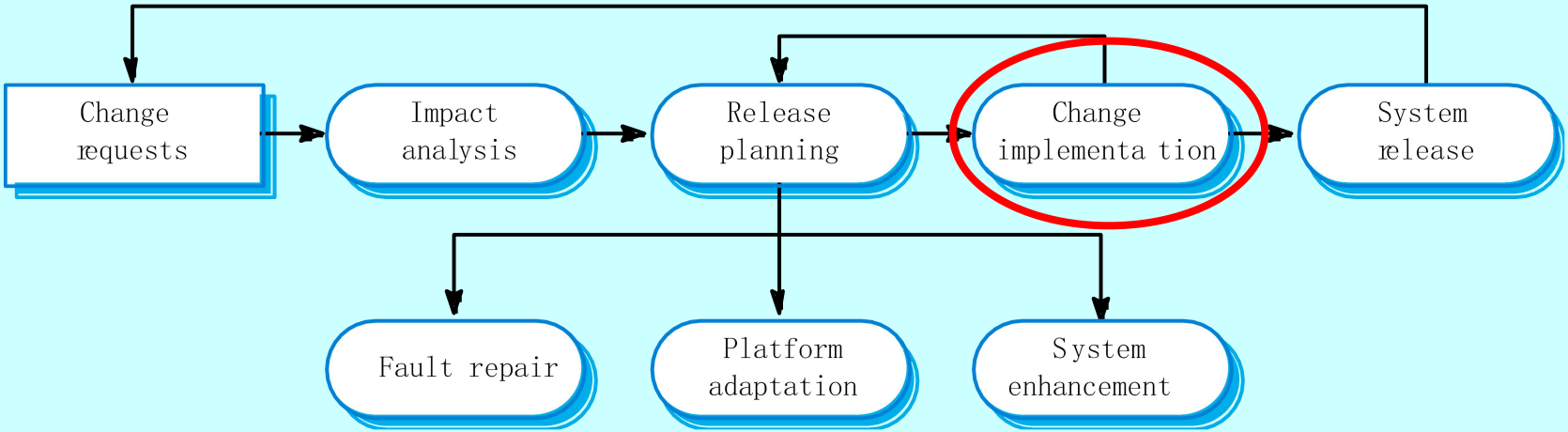
- 在软件维护中，影响维护工作量的因素主要有以下六种：
  - 系统的大小

    系统规模越大，其功能就越复杂，软件维护的工作量也随之增大。
  - 程序设计语言
  - 系统年龄
  - 先进的软件开发技术
  - 其它一些因素

软件维护

- 软件维护的定义
- 软件维护的类型
- 结构化维护VS非结构化维护
- 影响软件维护工作量的因素
- 软件维护的过程
- 可维护性
- 软件维护的管理

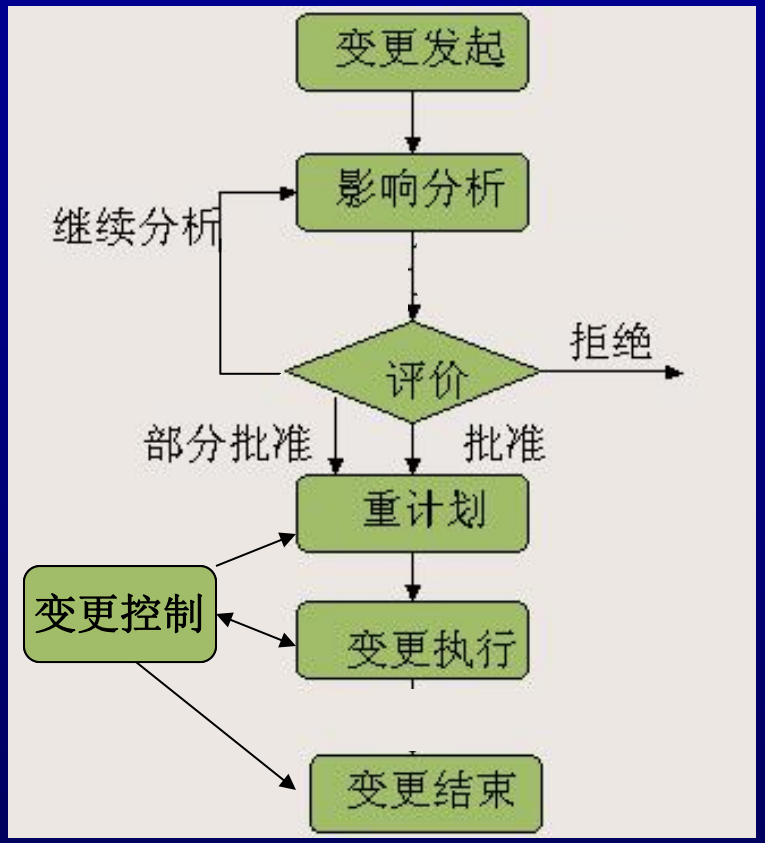
The system evolution process



- Change implementation
  - 修改软件需求说明
  - 修改软件设计
  - 设计评审
  - 对源程序做必要的修改
  - 单元测试
  - 集成测试(回归测试)
  - 确认测试
  - 软件配置评审等。

整体变更控制

- 变更流程



维护的流程

- 维护申请->影响分析与评价->维护修改报告->维护记录保存->维护后评审->维护后测试->维护后验收

软件维护

- 软件维护的定义
- 软件维护的类型
- 结构化维护VS非结构化维护
- 影响软件维护工作量的因素
- 软件维护的过程
- 可维护性
- 软件维护的管理

软件可维护性（software maintenanceability）

- 软件可维护性是指纠正软件系统出现的错误和缺陷，以及为满足新的要求进行修改、扩充或压缩的容易程度。

- 目前广泛使用的是用如下的七个特性来衡量程序的可维护性。

  可理解性　可使用性
  可测试性　可移植性
  可修改性　效率
  可靠性

- 而且对于不同类型的维护，这七种特性的侧重点也不相同。

1. 可理解性

- 可理解性表明人们通过阅读源代码和相关文档，了解程序功能及其如何运行的容易程度。

2. 可靠性

- 可靠性表明一个程序按照用户的要求和设计目标，在给定的一段时间内正确执行的概率。

3. 可测试性

- 可测试性表明论证程序正确性的容易程度。程序越简单，证明其正确性就越容易。而且设计合用的测试用例，取决于对程序的全面理解。
- 一个可测试的程序应当是可理解的，可靠的，简单的。

4. 可修改性

- 可修改性表明程序容易修改的程度。
- 一个可修改的程序应当是可理解的、通用的、灵活的、简单的。

5. 可移植性

- 可移植性表明程序转移到一个新的计算环境的可能性的大小。或者它表明程序可以容易地、有效地在各种各样的计算环境中运行的容易程度。

6. 效率

- 效率表明一个程序能执行预定功能而又不浪费机器资源的程度。
- 这些机器资源包括内存容量、外存容量、通道容量和执行时间。

7. 可使用性

- 从用户观点出发，可使用性定义为程序方便、实用、及易于使用的程度。

软件维护

- 软件维护的定义
- 软件维护的类型
- 结构化维护VS非结构化维护
- 影响软件维护工作量的因素
- 软件维护的过程
- 可维护性
- 软件维护的管理

维护的文档

- 维护合同
- 维护方案
- 维护手册
- 维护申请表
- 维护记录
- 维护报告