课时7 维护

考点	重要程度	占分	题型
7.1软件维护定义	★★★	4~6	选择、填空、简答
7.2-7.3软件维护特点与过程	★★★	2~6	填空、选择、简答
7.4软件可维护性	★★★	2~6	选择、填空、简答
7.5-7.6预防性维护与再工程过程	★★★	2~8	选择、填空、大题

7.1软件维护定义

软件维护:

是在软件已经交付使用之后，为了改正错误或满足新的需要而修改软件的过程。

分类:

- **改正性维护**：诊断和改正错误的过程(17%~21%)。
- **适应性维护**：为了和变化了的环境适当地配合而进行的修改软件的活动(18%~25%)。
- **完善性维护**：为了满足在用户提出的增加新功能或修改已有功能的要求和一般性的改进要求(50~66%)。
- **预防性维护**：(4%)

7.2软件维护特点

- 结构化维护与非结构化维护差别巨大

  非结构化维护:惟一成分是程序代码，那么维护活动从艰苦地评价程序代码开始。

  结构化维护:有完整的软件配置存在，那么维护工作从评价设计文档开始。

- 维护的代价高昂

  1970年总预算的35%～40%。

  1980年上升为40%～60%。

  1990年上升为70%～80%。

- 维护的问题很多

  理解别人写的程序通常非常困难

  维护的软件往往没有合格的文档，或者文档资料显著不足。

  要求对软件进行维护时，不能指望由开发人员给我们仔细说明软件。

  绝大多数软件在设计时没有考虑将来的修改。

  软件维护不是一项吸引人的工作。

7.3软件维护过程

- 维护组织
- 维护报告
- 维护的事件流
- 保存维护记录
- 评价维护活动

视频讲解更清晰
仅3小时

7.4软件可维护性

维护人员理解、改正、改动或改进这个软件的难易程度。

决定软件可维护性的因素：

- 可理解性:读者理解软件的难易程度
- 可测试性:论证程序正确性的容易程度。
- 可修改性:程序容易修改的程度。
- 可移植性:把程序从一种计算环境（硬件配置和操作系统）转移到另一种计算环境的难易程度。
- 可重用性:同一个软件不做修改或稍加改动，就可以在不同环境中多次重复使用。

7.5预防性维护

为了提高未来的可维护性或可靠性，而主动地修改软件。

维护一行源代码的成本可能是初始开发成本的20-40倍
使用现代设计概念重新设计软件结构，对未来的维护是很有帮助的
软件原型已经存在，软件开发生产率将远远高于平均水平
现在用户已经有丰富的使用软件的经验，很容易确定新的需求和变更方向
利用软件再工程工具可以自动完成部分工作
在完成预防性维护的过程中，可以建立起完整的软件配置（文档、程序、数据）

视频讲解更清晰
仅3小时

7.6软件再工程过程

预防性维护也称为软件再工程。

- **库存目录分析:**分析可能成为预防性维护的对象
  在今后数年内继续使用
  当前正在成功使用的程序
  可能最近的将来要做较大程度的修改或扩充

- **文档重构:**
  如果一个程序稳定，正在走向生命终点，不必为它再建立文档
  只建立系统中当前正在修改的那部分的完整文档
  尽量把文档工作减少到必须的最小量

7.6软件再工程过程

预防性维护也称为软件再工程。

- **逆向工程:**
  分析程序，以便在比源代码更高的抽象层次上创建出程序的某种描述的过程。
- **代码重构:**
  重构一些编码方式难理解、测试和维护的模块代码
- **数据重构:**
  对数据体系结构进行修改维护
- **正向工程:**
  利用现代软件工程概念、原理、技术和方法，重新开发现有的某个应用系统

总结

- 软件维护定义
- 软件维护特点
- 软件维护过程
- 软件的可维护性
- 预防性维护
- 软件再工程过程

视频讲解更清晰
仅3小时