

软件工程

第二部分 结构化方法学

软件测试

- 6.1 编码
- 6.2 软件测试基础
- 6.3 软件测试的策略(单元/集成/确认)
- 6.4 软件测试技术(白盒/黑盒)
- 6.5 程序调试
- 6.6 软件可靠性

软件测试

1

6.1 编码

- 作为软件工程过程的一个阶段，**实现**是对设计的进一步具体化。通常把**编码**和**测试**统称为**实现**。
- 编码**就是把软件设计结果翻译成用某种程序设计语言书写的程序。
- 程序的质量主要取决于**软件设计**的质量。所选用的**程序设计语言**的特点及**编码风格**也将对程序的可靠性、可读性、可测试性和可维护性产生深远的影响。

软件测试

2

选择程序设计语言

- 程序设计语言是人机通信的最基本的工具，影响解题方式、实现的效率、程序的测试与维护等。
- 选择程序设计语言的主要标准：
 - ✓ 系统用户的要求
 - ✓ 可以使用的编译程序
 - ✓ 可以得到的软件工具
 - ✓ 工程规模
 - ✓ 程序员的知识
 - ✓ 软件可移植性要求
 - ✓ 软件的应用领域

软件测试

3

编码风格

- 源程序代码应该逻辑简明清晰、易读易懂。为此，应该遵循下述规则：
 - ✓ **程序内部的文档**：包括恰当的标识符、适当的注解和程序的视觉组织等等。
 - ✓ **数据说明**：数据说明的次序应该标准化。
 - ✓ **语句构造**：每个语句都应该简单而直接，不能为了提高效率而使程序变得过分复杂。
 - ✓ **输入输出**：输入/输出数据的格式简单、必要的合法性检验等。
 - ✓ **效率**：包括时间与存储两个方面。注意不要牺牲程序的清晰性和可读性来不必要地提高效率。

软件测试

4

个体软件过程(PSP)

- PSP是一种可用于**控制、管理和改进**个人工作方式的自我持续改进过程，是一个包括软件开发表格、指南和规程的结构化框架。
- 由美国CMU/SEI的Watts Humphrey领导开发的。他对104位参与PSP培训的软件人员的统计数据表明(程序规模从50至5000行不等)，在应用PSP后：
 - ◆ 总的差错减少58.0%
 - ◆ 在测试阶段发现的差错减少71.9%
 - ◆ 生产效率提高20.8%。

软件测试

5