

第 12章 实现与测试

1

12.1 介绍

1

11.2 准备

2

11.2.1 目的

2

11.2.2 角色与职责

2

11.2.3 启动准则

2

11.2.4 输入

2

11.2.5 主要步骤

2

[Step1] 制定计划

2

[Step2] 审批计划

3

[Step3] 确定相关的规范

3

[Step4] 准备相关的软件工具

3

[Step5] 开发小组技能培训

3

12.2.6 输出

3

12.2.7 结束准则

3

12.2.8 度量

3

12.3 实现与测试

4

12.3.1 目的

4

12.3.2 角色与职责

4

12.3.3 启动准则

4

12.3.4 输入

4

12.3.5 主要步骤

4

[Step1] 编程

4

[Step2] 代码审查

4

[Step3] 单元测试

4

[Step4] 集成测试

5

[Step5] 缺陷管理与改错

5

12.3.6 输出

5

12.3.7 结束准则

5

12.3.8 度量

5

12.4 实施建议

5

# 第 12章 实现与测试

实现与测试（ Implementation and Test, IT）的目的是依据系统设计文档，编写并测试整个系统的代码。在本规范中，实现与测试是“编程、代码审查、单元测试、集成测试、缺陷管理与改错”的综合表述。

实现与测试过程域是 SPP 模型的重要组成部分。本规范阐述了实现与测试过程域的两个主要规程：

- ✧ 准备 [SPP-PROC-IT-PREPARE]
- ✧ 实现与测试 [SPP-PROC-IT-EXECUTE]

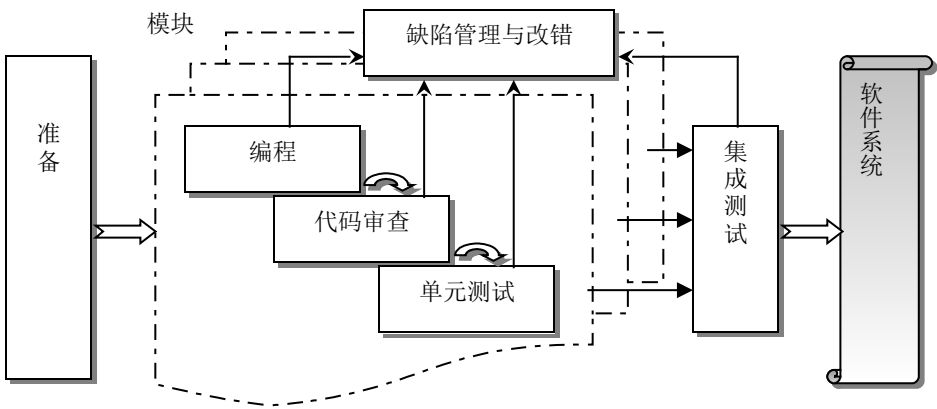
上述每个规程的“目标”、“角色与职责”、“启动准则”、“输入”、“主要步骤”、“输出”、“完成准则”和“度量”均已定义。

本规范适用于国内 IT企业的软件研发项目。建议用户根据自身情况（如商业目标、研发实力等）适当地修改本规范，然后推广使用。

## 12.1 介绍

实现与测试的流程如图 12-1 所示。一般地，编程、代码审查、单元测试、集成测试大致存在先后顺序关系，也可以并行、迭代地开展。上述任何活动中发现的缺陷必须用统一的缺陷管理工具来管理，开发人员应当及时消除缺陷（改错）。

图 12-1 实现与测试流程图



由于实现与测试是工作量最大、时间最长、产生工作成果（代码与文档）最多的一个项目研发过程域，所以需要作充分的准备工作。

实现与测试工作基本上在开发小组内部开展。一个项目可能有一个或者多个开发小组。对于小型项目，项目经理可以兼任开发组长。

特别要注意的是，开发人员应当

对自己的代码进行审查和测试（这是份内的工作），但是不能作为该代码已经通过审查和测试的依据。所以开发人员还要互相审查和测试同伴的代码。

实现与测试过程域产生的主要文档有：

- ✧ 《实现与测试计划》，模板见 [SPP-TEMP-IT-PLAN]。
- ✧ 《编程文档》，模板见 [SPP-TEMP-IT-CODING]。
- ✧ 《代码审查报告》，参考技术评审报告的模板 [SPP-TEMP-TR-REPORT]。
- ✧ 《测试用例》，模板见 [SPP-TEMP-TEST-CASE]。
- ✧ 《测试报告》，模板见 [SPP-TEMP-TEST-REPORT]。
- ✧ 《缺陷管理报告》，由缺陷管理工具自动生成。

## 11.2 准备

### 11.2.1 目的

- 为编程、代码审查、单元测试、集成测试、缺陷管理与改错等活动作准备，产生《实现与测试计划》。

### 11.2.2 角色与职责

- 一个项目可能有多个开发小组，视项目规模而定。开发组长由项目经理指定。
- 开发组长管理编程、代码审查、单元测试、集成测试、缺陷管理与改错等活动。

### 11.2.3 启动准则

- 系统设计文档已经完成。

### 11.2.4 输入

- 系统设计文档

### 11.2.5 主要步骤

#### [Step1] 制定计划

- 开发小组共同协商实现与测试计划。开发组长按照指定的模板起草《实现与测试计划》。该计划主要包括：
  - ◇ 编程计划
  - ◇ 代码审查计划
  - ◇ 单元测试计划
  - ◇ 集成测试计划
  - ◇ 缺陷管理与改错计划

#### [Step2] 审批计划

- 项目经理审批《实现与测试计划》。如果批准，则转向[Step3]。如果不批准，则退回[Step1]。

### [Step3] 确定相关的规范

- 开发小组确定编程、代码审查、单元测试、集成测试、缺陷管理与改错等规范。如果机构已经存在相应的编程规范，则采用之。如果机构不存在相应的编程规范，则由开发小组共同制定。

### [Step4] 准备相关的软件工具

- 开发小组确定并安装相关的软件开发工具（包括可复用库）、软件测试工具和缺陷管理工具等等。

### [Step5] 开发小组技能培训

- 如果开发组长认为开发小组需要接受编程、单元测试、代码审查、集成测试、缺陷管理与改错等方面的培训，那么由开发组长安排相应的培训。

## 12.2.6 输出

- 本开发阶段所需的规范如编程规范、测试规范等等。
- 《实现与测试计划》

## 12.2.7 结束准则

- 本规程所有的主要步骤都已经完成。

## 12.2.8 度量

- 开发组长统计工作量以及文档的规模，汇报给项目经理。

# 12.3 实现与测试

## 12.3.1 目的

- 依据计划执行编程、单元测试、代码审查、集成测试、缺陷管理与改错等活动。

## 12.3.2 角色与职责

- 开发组长管理编程、代码审查、单元测试、集成测试、缺陷管理与改错等活动。
- 开发人员依据计划编写代码，并对自己的代码进行必要的审查和测试。
- 开发人员依据计划审查、测试同伴的代码。

### 12.3.3 启动准则

- 系统设计文档已经完成。
- 实现与测试计划已经制定。

### 12.3.4 输入

- 系统设计文档
- 实现与测试计划

### 12.3.5 主要步骤

#### [Step1] 编程

- 开发人员根据“编程计划”编写软件的代码，并随时记录编程技术、问题与对策、心得体会等等，产生《编程文档》（类似于编程日记）。
- 开发人员在编写完成每个模块时，必须对自己的代码进行必要的审查和测试。

#### [Step2] 代码审查

- 开发人员根据“代码审查计划”对同伴的代码进行审查，产生《代码审查报告》。代码审查通常采用非正式技术评审规程，请参考 [SPP-PROC-TR-ITR] 。

#### [Step3] 单元测试

- 开发人员首先撰写单元测试用例。
- 开发人员根据“单元测试计划”和相应的“测试用例”来测试同伴的代码，产生“测试报告”。

#### [Step4] 集成测试

- 开发人员首先撰写集成测试用例。
- 开发人员根据“集成测试计划”和相应的“测试用例”来测试同伴的代码，产生“测试报告”。

#### [Step5] 缺陷管理与改错

- 从[Step1]至[Step4]，任何人发现软件中的缺陷时都必须使用开发小组指定的缺陷管理工具。该工具将记录所有缺陷的状态信息，并可以自动产生《缺陷管理报告》。
- 开发人员及时消除已经发现的缺陷。
- 开发人员消除缺陷之后应当马上进行回归测试，以确保不会引入新的缺陷。

### 12.3.6 输出

- 软件的源代码、库文件、可执行程序等等

- 编程文档
- 代码审查报告
- 测试用例
- 测试报告
- 缺陷管理报告

### 12.3.7 结束准则

- 软件的代码已经完成，软件可以运行。
- 本规程所有文档已经完成。

### 12.3.8 度量

- 开发组长统计编程、代码审查、测试、改错等的工作量，以及程序和文档的规模，将此度量数据汇报给项目经理。

## 12.4 实施建议

- 对开发人员进行“高质量程序设计”培训，让他们掌握编写高质量程序的技能。
- 对开发人员进行“代码审查、测试、改错”等方面的培训，提高他们的工作效率。
- 开发小组根据项目的资源、时间等限制因素，可以适当地减少测试的工作量。
- 开发小组根据产品的特征，可以适当地修改本规范的各种文档模板。
- 对实现与测试过程中产生的所有代码和有价值的文档进行配置管理。